UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCELO AKIRA INUZUKA

Decomposição de Tarefas para Problemas de Linguagem Natural

Segmentação de *Hashtags* e Anotação de Texto Argumentativo

Goiânia-GO
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[ ] Dissertação [X] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Marcelo Akira Inuzuka

3. Título do trabalho

Decomposição de Tarefas para Problemas de Linguagem Natural – Segmentação de *Hashtags* e Anotação de Texto Argumentativo

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Hugo Alexandre Dantas Do Nascimento**, **Professor do Magistério Superior**, em 02/06/2025, às 15:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Marcelo Akira Inuzuka**, **Discente**, em 06/06/2025, às 10:55, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5403749** e o código CRC **98A2B2A6**.

Referência: Processo nº 23070.016461/2025-32 SEI nº 5403749

MARCELO AKIRA INUZUKA

Decomposição de Tarefas para Problemas de Linguagem Natural

Segmentação de *Hashtags* e Anotação de Texto Argumentativo

Tese apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Hugo Alexandre Dantas do Nascimento

Co-Orientadora: Profa. Nádia Félix Felipe da Silva

Goiânia-GO
2025

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Inuzuka, Marcelo Akira
Decomposição de Tarefas para Problemas de Linguagem Natural [manuscrito] : Segmentação de Hashtags e Anotação de Texto Argumentativo / Marcelo Akira Inuzuka. - 2025.
CCXCI, 291 f.: il.

Orientador: Prof. Hugo Alexandre Dantas do Nascimento; co orientador Nádia Félix Felipe da Silva.
Tese (Doutorado) - Universidade Federal de Goiás, Instituto de Informática (INF), , Cidade de Goiás, 2025.
Bibliografia. Anexos. Apêndice.
Inclui siglas, abreviaturas, símbolos, gráfico, tabelas, algoritmos.

1. Anotação de corpus. 2. Processamento de Linguagem Natural. 3. Decomposição de Tarefas. 4. Qualidade de dados. 5. Padrões reutilizáveis. I. do Nascimento, Hugo Alexandre Dantas, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE TESE

Ata nº **09/2025** da sessão de Defesa de Tese de **Marcelo Akira Inuzuka**, que confere o título de Doutor em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos vinte e quatro dias do mês de abril de dois mil e vinte e cinco, a partir das oito horas e trinta minutos da manhã, na sala 250 do INF, realizou-se a sessão pública de Defesa de Tese intitulada **"Decomposição de Tarefas no Processamento de Linguagem Natural"** . Os trabalhos foram instalados pelo Orientador, Professor Doutor Hugo Alexandre Dantas do Nascimento (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professora Doutora Nádia Félix Felipe da Silva, Coorientadora; Professor Doutor Márcio de Souza Dias (DC/UFCAT), membro titular externo; Professor Doutor Wanderley de Souza Alencar (INF/UFG), membro titular externo; Professor Doutor Thierson Couto Rosa (INF/UFG), membro titular interno; e Professor Doutor Wellington Santos Martins (INF/UFG), membro titular interno. A participação dos professores Márcio de Souza Dias, Wellington Santos Martins, Nádia Félix Felipe da Silva e Hugo Alexandre Dantas do Nascimento ocorreu por meio de videoconferência. Durante a arguição os membros da banca fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Tese, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Hugo Alexandre Dantas do Nascimento, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos vinte e quatro dias do mês de abril de dois mil e vinte e cinco.

TÍTULO SUGERIDO PELA BANCA

Decomposição de Tarefas para Problemas de Linguagem Natural – Segmentação de *Hashtags* e Anotação de Texto Argumentativo

Documento assinado eletronicamente por **Hugo Alexandre Dantas Do Nascimento**, **Professor do Magistério Superior**, em 24/04/2025, às 18:19, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Wellington Santos Martins**, **Professor do Magistério Superior**, em 24/04/2025, às 18:29, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Wanderley De Souza Alencar**, **Professor do Magistério Superior**, em 24/04/2025, às 18:32, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Nadia Felix Felipe Da Silva**, **Professora do Magistério Superior**, em 24/04/2025, às 18:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Márcio de Souza Dias**, **Usuário Externo**, em 24/04/2025, às 21:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Thierson Couto Rosa**, **Professor do Magistério Superior**, em 24/04/2025, às 22:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Marcelo Akira Inuzuka**, **Discente**, em 24/04/2025, às 22:13, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5331039** e o código CRC **E5D45D83**.

Referência: Processo nº 23070.016461/2025-32 SEI nº 5331039

MARCELO AKIRA INUZUKA

Decomposição de Tarefas para Problemas de Linguagem Natural

Segmentação de *Hashtags* e Anotação de Texto Argumentativo

Tese defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Doutor em Ciência da Computação, aprovada em 24 de Abril de 2025, pela Banca Examinadora constituída pelos professores:

Prof. Hugo Alexandre Dantas do Nascimento
Instituto de Informática – UFG
Presidente da Banca

Profa. Nádia Félix Felipe da Silva
Instituto de Informática – UFG

Prof. Dr. Thierson Couto Rosa
Instituto de Informática – UFG

Prof. Dr. Wellington Santos Martins
Instituto de Informática – UFG

Prof. Dr. Wanderley de Souza Alencar
Instituto de Informática – UFG

Prof. Dr. Márcio de Souza Dias
Departamento de Ciências da Computação – UFCat

Marcelo Akira Inuzuka

Graduado em Engenharia Elétrica (1995) e Ciência da Computação (1997) ambos pela Universidade Federal de Goiás (UFG), mestre em Tecnologia da Informação e Comunicação na Formação em Ensino a Distância pela Universidade Norte do Paraná(UNOPAR)/Universidade Federal do Ceará (UFC) (2008). É Professor do Instituto de Informática da UFG desde março de 2010. Realizou seu doutorado sanduíche na Faculdade de Ciências de Lisboa em 2023, pesquisando na área de Mapeamento de Argumentos utilizando técnicas de Aprendizado Profundo.

Agradecimentos

A jornada que culmina nesta tese não teria sido possível sem o apoio, a orientação e a parceria de muitas pessoas que, de diferentes formas, contribuíram para o meu crescimento acadêmico e profissional.

Agradeço, em primeiro lugar, aos meus pais, que me criaram e educaram com dedicação, proporcionando a base para meus estudos e formação profissional. Seu incentivo incondicional foi essencial em cada etapa do meu caminho.

À minha esposa, Phlaras, e às minhas filhas, Sophia e Laura, minha gratidão profunda pelo amor, paciência e apoio inestimáveis. Nos momentos mais desafiadores, foram vocês que me deram força e equilíbrio para seguir adiante.

Ao meu orientador, Hugo Alexandre Dantas do Nascimento, por ampliar minha visão e conhecimento nas diversas áreas da computação, sempre com ensinamentos valiosos e encorajamento constante.

À minha coorientadora, Nádia Félix, por me apresentar à área de Processamento de Linguagem Natural e por sua orientação precisa e inspiradora ao longo dessa trajetória.

Ao professor António Branco, meu orientador durante o doutorado sanduíche, por me acolher generosamente no grupo de pesquisa da Faculdade de Ciências da Universidade de Lisboa (FCUL) e proporcionar um ambiente acadêmico rico e desafiador.

Aos colegas do grupo de pesquisa, João Silva e João Rodrigues, por seus conselhos valiosos e pelo compartilhamento de conhecimento, que ajudaram a moldar minha compreensão da área de Processamento de Linguagem Natural (PLN).

Ao colega Walid, pela ajuda na revisão da tese e pelas dicas preciosas que contribuíram para aprimorar o trabalho.

Aos meus alunos, bolsistas e orientandos que participaram das pesquisas e contribuíram com seu empenho e dedicação, meu reconhecimento e sincero agradecimento.

Aos colegas da equipe de especialistas — Alexandre Oliveira, Fernando Almeida, Romário Magalhães e Victor Brandão —, muito obrigado pelas muitas horas de voluntariado em reuniões de desenvolvimento de ferramentas e na anotação do *dataset* Argmap.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio concedido por meio da bolsa de doutorado sanduíche (PDSE), que

viabilizou o desenvolvimento de parte desta pesquisa no exterior. Também agradeço ao LaMCAD — Laboratório Multiusuário de Computação de Alto Desempenho da UFG — pelo suporte computacional essencial à condução dos experimentos desta tese.

A todos que, de alguma forma, estiveram ao meu lado nessa caminhada, muito obrigado!

Resumo

Inuzuka, Marcelo Akira. **Decomposição de Tarefas para Problemas de Linguagem Natural**. Goiânia-GO, 2025. 293p. Tese de Doutorado. Programa de Pós-graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

A anotação de corpora é essencial para o treinamento de modelos de Processamento de Linguagem Natural (PLN), mas enfrenta desafios como alta complexidade cognitiva, inconsistência entre anotadores e custos elevados. Esta tese propõe a decomposição de tarefas como uma estratégia metodológica para modularizar processos complexos em PLN, promovendo maior clareza conceitual, escalabilidade e reprodutibilidade. Inicialmente centrada no Mapeamento de Argumentos, a pesquisa redirecionou seu escopo devido à inviabilidade da tarefa original, concentrando-se na identificação de padrões reutilizáveis aplicáveis a etapas de anotação e automação.

Foram desenvolvidas diretrizes, um algoritmo de decomposição hierárquica e artefatos como conjuntos de dados anotados e a plataforma Argmap, que oferece suporte à anotação colaborativa com controle de qualidade. A abordagem foi validada por meio de três estudos de caso: segmentação de hashtags, curadoria de frases-chave e anotação de estruturas argumentativas. Os resultados demonstram que a decomposição melhora a consistência entre agentes (humanos ou automáticos), a clareza das diretrizes e a viabilidade de automação.

A tese também propõe o padrão arquitetural Recrutador–Selecionador, que estrutura tarefas em dois módulos independentes — geração de candidatos e seleção final —, aplicável tanto a fluxos de anotação quanto a algoritmos baseados em *Large Language Model (LLM)s*. Conclui-se que a decomposição orientada por padrões reutilizáveis aprimora a eficiência e a confiabilidade na construção de corpora e no desenvolvimento de sistemas robustos em PLN, contribuindo para a sistematização de processos anotativos e sua integração com soluções automáticas.

Palavras–chave

Anotação de corpus. Processamento de Linguagem Natural. Qualidade de dados. Padrões reutilizáveis. LLMs. Decomposição de tarefas.

Abstract

Inuzuka, Marcelo Akira.
. Goiânia-GO, 2025. 293p. PhD. Thesis. Programa de Pós-graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

Corpus annotation is essential for training Natural Language Processing (NLP) models, yet it faces challenges such as high cognitive complexity, annotator inconsistency, and elevated costs. This thesis proposes task decomposition as a methodological strategy to modularize complex NLP processes, promoting greater conceptual clarity, scalability, and reproducibility. Initially focused on Argument Mapping, the research redirected its scope due to the infeasibility of the original task, concentrating on the identification of reusable patterns applicable to annotation and automation stages.

Guidelines, a hierarchical decomposition algorithm, and artifacts such as annotated datasets and the Argmap platform — which supports collaborative annotation with quality control — were developed. The approach was validated through three empirical case studies: hashtag segmentation, keyphrase curation, and annotation of argumentative structures. Results demonstrate that decomposition improves consistency among agents (human or automatic), guideline clarity, and automation feasibility.

The thesis also introduces the Recruiter–Selector architectural pattern, which structures tasks into two independent modules — candidate generation and final selection — applicable to both annotation workflows and algorithms based on Large Language Models (LLMs). It concludes that decomposition driven by reusable patterns enhances efficiency and reliability in corpus construction and the development of robust NLP systems, contributing to the systematization of annotation processes and their integration with automatic solutions.

Keywords

Corpus annotation. Natural Language Processing. Data quality. Reusable patterns. LLMs. Task decomposition.

Sumário

Lista de Siglas

BRAT Brat Rapid Annotation Tool. 126, 188

CFC Curadoria de Frases-Chave. 72, 138, 158–161, 171, 174, 175, 187, 188, 195
CoT Chain of Thought. 71
CT Classificação de Tópicos. 103, 138, 158, 159, 171–175, 179, 187

DAG Grafo Acíclico Dirigido. 68

GPT Generative Pre-trained Transformer. 65, 86, 87, 93, 95–97, 99

IAA Inter-Annotator Agreement. 107, 108, 113–115, 153, 167, 168, 170, 179, 188
IAGT Identificação Automática de Gêneros Textuais. 145–149, 153, 156, 157

LLM Large Language Model. 12–14, 20–22, 26, 29, 50, 51, 54, 55, 64–68, 70–73, 76–78, 85, 107, 110, 111, 114, 115, 159, 188, 200, 201

MAM1 Mapeamento de Argumentos Monodocumento. 16, 123, 124, 134, 137, 138, 186
MAMD Mapeamento de Argumentos Multidocumento. 118, 123, 186, 188, 189

PLN Processamento de Linguagem Natural. 10, 20, 21, 25, 27, 31, 48–50, 55, 56, 64, 66–68, 70–72, 77, 78, 84, 85, 87, 98, 100, 101, 115, 116, 139, 144, 147, 171, 188–192, 194, 198, 199, 202

RAG Retrieval-Augmented Generation. 54, 71, 204

SCT Segmentação e Classificação de Tópicos. 123, 138, 174, 175, 187

CAPÍTULO 1

Introdução

A produção contínua de conhecimento em áreas como ciência, direito e política gera, diariamente, grandes volumes de textos argumentativos. Esses textos sustentam hipóteses, fundamentam decisões e estruturam propostas, constituindo uma base essencial para o avanço de debates e práticas sociais. Apesar dos avanços recentes em computação, a maior parte desse conteúdo ainda é transmitida por meio de linguagem natural não estruturada, o que impõe esforço considerável para leitura, análise e síntese em escala.

No contexto do PLN, diversas técnicas têm sido desenvolvidas para estruturar esse conteúdo, com destaque para a sumarização argumentativa. A efetividade dessas técnicas, no entanto, depende de *corpora* anotados com camadas semânticas profundas e consistentes, os quais ainda são escassos. A construção desses recursos impõe desafios metodológicos relacionados à clareza conceitual, à confiabilidade entre anotadores e à escalabilidade do processo.

Mesmo em casos amplamente utilizados, como o UKP Sentential [Stab et al. 2018], observa-se a necessidade de centenas de horas de trabalho humano qualificado, o que evidencia a complexidade da tarefa e a importância de estratégias que favoreçam sua modularização. A decomposição de tarefas surge, nesse contexto, como abordagem promissora para lidar com tais desafios, sobretudo quando aliada ao uso de padrões reutilizáveis e às capacidades emergentes de modelos de linguagem em larga escala (LLMs).

Esta tese investiga a decomposição de tarefas como eixo estruturante para o aprimoramento da qualidade da anotação de *corpus* em PLN. Parte-se da hipótese de que a divisão sistemática de tarefas complexas em subtarefas cognitivamente mais simples, orientada por padrões explícitos e compatível com arquiteturas de LLMs, favorece a clareza conceitual, a reprodutibilidade e a viabilidade de automação. Os fundamentos e objetivos da pesquisa são detalhados nas seções a seguir.

1.1 Contextualização e Justificativa do Problema

A anotação de *corpus* constitui etapa fundamental para o desenvolvimento de sistemas baseados em aprendizado supervisionado no PLN. Ao fornecer exemplos rotulados por especialistas, os *corpora* anotados permitem a modelagem e a avaliação de tarefas como classificação, extração de entidades, reconhecimento de relações e estruturação discursiva. No entanto, a construção desses recursos impõe desafios metodológicos recorrentes, especialmente em domínios que exigem anotações interpretativas, como a argumentação, a análise de sentimentos ou a categorização pragmática. Para a trajetória completa que motivou o redirecionamento da pesquisa e a análise da tarefa original de mapeamento de argumentos, ver o **Apêndice A**.

Tarefas anotativas com maior grau de complexidade cognitiva tendem a apresentar altos custos operacionais, baixa reprodutibilidade e significativa variabilidade entre anotadores. Fatores como ambiguidade conceitual, sobrecarga de instruções e ausência de modularização dificultam a execução eficiente e confiável dessas tarefas. Estudos anteriores apontam que a divisão de tarefas complexas em subtarefas cognitivamente mais simples pode mitigar tais dificuldades, favorecendo a clareza conceitual, o foco da atenção e a avaliação segmentada da qualidade.

A aplicação sistemática da decomposição de tarefas, entretanto, ainda carece de uma base metodológica consolidada no contexto da anotação de *corpus*. Em geral, as práticas de decomposição permanecem *ad hoc*, conduzidas por decisões informais ou pela intuição dos pesquisadores envolvidos. A ausência de uma estrutura conceitual para orientar tais divisões compromete a reusabilidade dos métodos e limita seu potencial de generalização para novos contextos e domínios.

Esta tese parte da premissa de que a formalização de padrões reutilizáveis de decomposição pode contribuir significativamente para a melhoria da qualidade da anotação em PLN. Por meio da identificação de padrões recorrentes, da análise de seus efeitos sobre a clareza e a consistência anotativa, e da articulação entre abordagens humanas e automáticas — como o uso de LLMs —, busca-se estabelecer um arcabouço teórico-metodológico que sistematize o processo de decomposição em tarefas de anotação complexas. Tal abordagem visa ampliar a viabilidade, a reprodutibilidade e a escalabilidade da construção de *corpora* anotados em diferentes cenários do PLN.

1.2 Questões de Pesquisa

Para investigar a hipótese de que a decomposição de tarefas pode favorecer a qualidade da anotação de *corpus* em PLN, esta tese foi guiada pelas seguintes questões:

Q1. Como decompor tarefas complexas de anotação em subtarefas cognitivamente mais simples e passíveis de automação?
Q2. Quais padrões são úteis para organizar o fluxo de anotação e promover qualidade, reprodutibilidade e automação?
Q3. Como avaliar os efeitos da decomposição no processo anotativo e na qualidade dos dados resultantes?

1.3 Objetivos

Esta tese tem como objetivo geral:

> Investigar como a decomposição de tarefas pode contribuir para a anotação de *corpus* em Problemas de Linguagem Natural, promovendo clareza conceitual, qualidade de anotação, reprodutibilidade e viabilidade de automação.

A investigação parte de estudos de caso empíricos que envolvem tarefas complexas de anotação, com foco na análise dos desafios envolvidos, na sistematização de padrões recorrentes e na exploração de mecanismos de apoio baseados em modelos de linguagem de grande porte (LLMs).

Os objetivos específicos são:

- Investigar a viabilidade da decomposição de tarefas de anotação em subtarefas mais simples;
- Identificar padrões de projeto úteis à organização de tarefas anotativas;
- Avaliar os efeitos da decomposição sobre a qualidade da anotação e a reprodutibilidade;
- Desenvolver ferramentas para apoiar fluxos de anotação baseados em decomposição.
- Gerar e disponibilizar *corpora* anotados com qualidade verificável

1.4 Conjecturas Fundamentais

A fundamentação teórica desta tese é guiada por quatro conjecturas interligadas, que refletem pressupostos cognitivos e metodológicos sobre a natureza dos problemas complexos, a estruturação da anotação e sua relação com a aprendizagem automática. Essas conjecturas não são formuladas como hipóteses testáveis em sentido estrito, mas como proposições plausíveis que orientam a construção conceitual e prática da pesquisa.

Conjectura 1 (Compreensão Hierárquica) *Todo sistema complexo hierárquico é, até certo limite, decomponível em partes compreensíveis por seres humanos. A complexidade de um problema pode ser analisada em termos da quantidade e da organização hierárquica de suas subtarefas, sendo o esforço cognitivo proporcional à profundidade e à extensão dessa estrutura.*

Conjectura 2 (Linguagem de Padrões) *A compreensão humana de tarefas complexas depende de representações simbólicas. Nomear padrões recorrentes nessas tarefas alavanca a capacidade de raciocínio, abstração e comunicação, constituindo uma linguagem operacional que facilita a estruturação, o reuso e a automação de processos.*

Conjectura 3 (Anotação como Modelagem) *Dado um processo de anotação que produza exemplos consistentes e representativos, é possível treinar modelos de aprendizado de máquina capazes de executar a tarefa-alvo com desempenho satisfatório. Assim, a anotação opera como forma de modelagem conceitual da tarefa, sendo tanto meio de formalização quanto ponte para sua automatização.*

Conjectura 4 (Primitivas de Anotação) *A decomposição de uma tarefa atinge um ponto de parada quando as subtarefas resultantes se tornam suficientemente simples para serem executadas com qualidade consistente entre diferentes anotadores, baseadas em confiança mútua e entendimento compartilhado. Essas unidades mínimas são chamadas de primitivas de anotação, e representam o limite funcional da decomposição, além do qual há risco de perda de significado ou aumento desnecessário da fragmentação.*

Essas conjecturas fornecem as bases conceituais que sustentam a abordagem adotada nesta tese: a decomposição de tarefas orientada por padrões reutilizáveis como estratégia para produzir anotações de alta qualidade, interpretáveis por humanos e adequadas à posterior automatização por modelos de aprendizado de máquina.

1.5 Procedimentos Metodológicos

Esta tese adota uma abordagem metodológica mista, combinando construção de artefatos com investigação empírica qualitativa. Nesse contexto, cinco metodologias principais foram utilizadas:

- **Estudo de Caso**: metodologia qualitativa aplicada para investigar em profundidade fenômenos contextuais e complexos [Yin 2015]. Nesta tese, foram conduzidos três estudos de caso empíricos, documentados nos Capítulos 5 (Segmentação de *Hashtags*), 6 (Curadoria de Frases-Chave) e 7 (Decomposição de Tarefas de Anotação

de *Corpus*). Cada estudo de caso apresenta o problema específico, os artefatos desenvolvidos, os experimentos conduzidos e a análise dos resultados, com foco na decomposição de tarefas e nos padrões de projeto observados.

- **Pesquisa em Ciência do *Design* (*Design Science Research* – DSR)**: metodologia voltada à construção de artefatos úteis e cientificamente validados [Dresch Daniel Pacheco Lacerda 2015]. Como descrito no Apêndice A, a DSR foi utilizada como inspiração geral para o ciclo iterativo de desenvolvimento e avaliação dos artefatos (como *datasets*, algoritmos e *guidelines*), especialmente nos primeiros estágios da pesquisa. A estrutura clássica da DSR (problema, objetivos, *design*, demonstração, avaliação e comunicação) foi observada, ainda que aplicada de forma flexível no contexto empírico dos estudos de caso.
- **Teoria Fundamentada (*Grounded Theory* – GT)**: abordagem qualitativa indutiva que visa gerar teorias a partir de dados empíricos [Charmaz 2009]. Nesta tese, a GT foi aplicada à análise de padrões recorrentes nos processos de anotação e nas decisões tomadas em reuniões de adjudicação, contribuindo para a formulação de categorias e padrões reutilizáveis no contexto de anotação de *corpus*.
- **Revisão Sistemática da Literatura (*Systematic Literature Review* – SLR)**: técnica rigorosa para mapeamento do estado da arte sobre tópicos específicos [Kitchenham 2004]. Foi aplicada de forma pontual, especialmente para fundamentar decisões metodológicas e justificar escolhas de tarefas e métricas.
- **Revisão de Escopo (*Scoping Review* – SR)**: abordagem utilizada para realizar levantamentos preliminares em áreas com pouca literatura consolidada [Arksey e O'Malley 2005]. A SR foi útil para explorar campos emergentes e orientar etapas exploratórias de projeto, quando uma SLR completa não se justificava.

O Capítulo 2 apresenta detalhadamente os procedimentos adotados.

1.6 Organização do Texto

A estrutura da tese está organizada da seguinte forma:

- **Capítulo 2 — Metodologia**
  Apresenta o arcabouço metodológico adotado, detalhando as abordagens utilizadas, incluindo estudos de caso, *Design Science Research* e Teoria Fundamentada.
- **Capítulo 3 — Revisão da Literatura**
  Realiza uma síntese crítica sobre os temas centrais da tese: Qualidade de Anotação, Padrões de Projeto e Decomposição de Tarefas.
- **Capítulo 4 — Decomposição de Tarefas em Problemas de Linguagem Natural**
  Propõe uma teoria de decomposição de tarefas voltada a Problemas em Linguagem

Natural, com base em uma taxonomia conceitual e evidências empíricas extraídas de anotações complexas.

- **Capítulo 5 — Segmentação de Hashtags**
  Primeiro estudo de caso empírico, no qual a metodologia de decomposição de tarefas é aplicada à segmentação de *hashtags*, com ênfase em ordenação, reconstrução e agrupamento.
- **Capítulo 6 — Curadoria de Frases-Chave**
  Segundo estudo de caso, dedicado à seleção e agrupamento de frases-chave para suporte à classificação de tópicos. Apresenta artefatos, métricas e análise da tarefa segundo o padrão Recrutador-Selecionador.
- **Capítulo 7 — Decomposição de Tarefas de Anotação de Corpus**
  Terceiro estudo de caso, voltado à análise de tarefas de anotação complexas e sua divisão em subtarefas. São discutidas estratégias de controle de qualidade e padrões de adjudicação. A Figura A.1 ilustra a decomposição final resultante.
- **Capítulo 8 — Considerações Finais**
  Retoma os objetivos e questões de pesquisa, sintetiza as contribuições obtidas e propõe direções para trabalhos futuros.

1.7 Contribuições

Esta tese contribui para a área de PLN ao propor uma abordagem sistemática para a decomposição de tarefas baseada em padrões reutilizáveis, aplicada à construção e avaliação de artefatos em múltiplos contextos. As contribuições podem ser organizadas em três eixos complementares:

- **Contribuições conceituais e teóricas:**
  - Formulação de uma abordagem para decomposição de tarefas como ferramenta para modularização de processos complexos em PLN, com ênfase na anotação de *corpus*.
  - Proposição de quatro conjecturas fundamentais que sustentam a base epistemológica da tese, articulando aspectos cognitivos, linguísticos e computacionais da decomposição.
  - Introdução do conceito de *primitivas de anotação* como limite funcional da decomposição e critério de granularidade anotativa.
  - Formalização de padrões arquiteturais de decomposição, com destaque para o padrão Recrutador–Selecionador, aplicável à organização de fluxos de subtarefas.
- **Contribuições metodológicas:**

  - Aplicação da metodologia de estudo de caso múltiplo para analisar tarefas reais com diferentes estruturas e domínios, incluindo anotação argumentativa, curadoria de frases-chave e segmentação de *hashtags*.
  - Obtenção do estado da arte em segmentação de *hashtags* via abordagem *zero-shot* baseada em padrões linguísticos e arquiteturais, demonstrando o potencial da decomposição para estruturar tarefas mesmo sob ausência de dados anotados.
  - Desenvolvimento e documentação de artefatos como *guidelines*, *datasets* anotados, métricas de qualidade e ferramentas de suporte.
  - Sistematização de padrões reutilizáveis de decomposição, com critérios de abstração, aplicabilidade e reuso.

- **Contribuições tecnológicas e práticas:**
  - Integração de padrões de decomposição com modelos de linguagem de grande porte (LLMs), visando à automação parcial de subtarefas anotativas.
  - Desenvolvimento de fluxos semiautomáticos assistidos por LLMs, avaliados quanto à clareza, confiabilidade e alinhamento com diretrizes humanas.
  - Proposição de diretrizes para o uso responsável e eficaz de LLMs na decomposição e suporte à anotação em tarefas complexas.

CAPÍTULO 2

Metodologia

Este capítulo apresenta a estrutura metodológica adotada nesta tese de doutorado, composta por uma abordagem principal — o *Estudo de Caso* — e por duas abordagens complementares: a *Design Science Research* (DSR) e a *Teoria Fundamentada* (TFST). A combinação dessas abordagens foi concebida com o objetivo de investigar, modelar e validar padrões de projeto voltados à qualidade de anotação de *corpus* em tarefas de PLN.

A estratégia metodológica foi organizada de modo a articular quatro componentes principais:

1. **Revisões de literatura e análise conceitual**, conduzidas nos Capítulos 3 e 4, com o objetivo de delimitar o espaço de problema, identificar conceitos centrais e apoiar a formulação das categorias iniciais da pesquisa;
2. **Estudos de Caso Empíricos**, documentados nos Capítulos 5, 6 e 7, os quais possibilitam a observação e análise de padrões de projeto e estratégias de decomposição em contextos reais;
3. **Construção de Artefatos**, como *datasets*, *guidelines*, modelos e ferramentas computacionais, orientada pelos princípios da DSR;
4. **Análise Qualitativa Indutiva**, baseada na Teoria Fundamentada Sociotécnica, para extração de padrões e hipóteses a partir dos dados dos processos de anotação.

A seguir, são descritas , detalhadamente, cada uma das abordagens e procedimentos metodológicos que compõem o percurso da pesquisa:

2.1 **Revisão de Literatura como Procedimento Metodológico**: Explica como as revisões teóricas foram conduzidas de forma estruturada para apoiar a formulação das categorias e hipóteses da pesquisa.

2.2 **Estudo de Caso**: Fundamenta o uso do estudo de caso como abordagem central e descreve os critérios adotados para condução dos capítulos empíricos.

2.3 ***Design* Science Research**: Apresenta a aplicação da DSR na construção dos artefatos desenvolvidos e empregados nos estudos de caso.

2.4 **Teoria Fundamentada Sociotécnica**: Detalha o uso da TFST como referencial qualitativo para a organização e análise de dados empíricos.

2.1 Revisão de Literatura como Procedimento Metodológico

As revisões de literatura realizadas nesta tese foram concebidas como etapas metodológicas fundamentais para a delimitação do espaço de problema, formulação de hipóteses e estruturação das categorias analíticas iniciais. Diferentemente das revisões sistemáticas formais descritas por autores como Kitchenham [Kitchenham 2004], não foi seguido um protocolo rigoroso de inclusão e exclusão. Em vez disso, adotou-se uma abordagem seletiva, recursiva e orientada por propósito, com foco em *surveys* e artigos de referência amplamente citados, muitas vezes localizados por meio da exploração de cadeias de citações em publicações centrais.

Essa revisão investigou conceitos relacionados a três pilares fundamentais da tese:

1. **Qualidade de Anotação de Corpus**, com foco em diretrizes de anotação, confiabilidade entre anotadores, estratégias de adjudicação e controle de qualidade;
2. **Padrões Reutilizáveis**, incluindo padrões linguísticos, padrões de projeto computacional, padrões de processo e sua aplicação em tarefas de PLN;
3. **Decomposição de Tarefas**, especialmente no contexto de aprendizado de máquina e anotação de *corpus*, com atenção a estruturas hierárquicas, operações cognitivas e mecanismos de divisão funcional.

Esse processo foi desenvolvido ao longo de dois capítulos específicos:

- No Capítulo 3, intitulado **Revisão da Literatura**, foi realizado um levantamento seletivo de publicações que abordam a qualidade de anotação de *corpus*, práticas de anotação colaborativa, padrões de anotação e projetos existentes voltados à confiabilidade e consistência dos dados anotados.
- No Capítulo 4, intitulado **Decomposição de Tarefas em Problemas de Linguagem Natural**, foi realizada uma análise conceitual de teorias e modelos de decomposição de tarefas, com base na literatura em PLN, aprendizado de máquina e linguística. Essa revisão fundamenta a proposta de uma nova abordagem de decomposição baseada em padrões linguísticos identificáveis por grandes modelos de linguagem (LLMs), e embasa a formulação de dois aportes originais desta tese: (i) o padrão de projeto **Recrutador-Selecionador**; e (ii) uma metodologia de decomposição de tarefas voltada à anotação de *corpus*.

Ambas as revisões foram conduzidas de forma iterativa e articulada às demais fases da pesquisa, contribuindo diretamente para a definição das hipóteses centrais, estruturação de *guidelines*, fundamentação dos padrões propostos e organização dos capítulos empíricos.

2.2 Estudo de Caso

A abordagem de estudo de caso foi adotada como metodologia central desta tese, por sua adequação à investigação empírica de fenômenos complexos inseridos em contextos reais [Yin 2015]. Segundo Stake [Stake 1995], estudos de caso podem ser classificados como intrínsecos, instrumentais ou coletivos. Esta tese adota a abordagem de estudo de caso coletivo, na qual múltiplos casos empíricos são analisados de forma coordenada com o objetivo de compreender um fenômeno mais amplo.

Os três estudos de caso apresentados nesta tese foram selecionados por cobrirem tarefas distintas de anotação, com graus variados de complexidade e natureza operacional, mas que compartilham o objetivo comum de avaliar práticas que impactam a qualidade do *corpus* anotado. Além de observar a aplicação de padrões de projeto específicos, os estudos também analisam estratégias de decomposição de tarefas baseadas em padrões linguísticos, especialmente aqueles identificáveis ou reaproveitáveis por meio de LLMs.

Capítulo 5 Segmentação de Hashtags: propõe e avalia uma arquitetura baseada no padrão Recrutador-Selecionador, aplicada à segmentação de *hashtags* compostas.

Capítulo 6 Curadoria de Frases-Chave: decompõe a tarefa de curadoria em subtarefas cognitivas (agrupamento, filtragem e seleção) e avalia abordagens manuais e automáticas.

Capítulo 7 Decomposição de Tarefas de Anotação de Corpus: analisa práticas de adjudicação, estabilidade anotacional e padrões operacionais a partir de dados reais de anotação.

2.3 *Design* Science Research

A abordagem de *Design Science Research* (DSR) foi empregada para orientar a construção iterativa de artefatos utilizados nos estudos empíricos da tese. Foram seguidas as atividades propostas por Peffers et al. [Peffers et al. 2007], incluindo a identificação de problemas, definição de objetivos, desenvolvimento, demonstração, avaliação e comunicação de artefatos.

Os artefatos desenvolvidos incluem: *guidelines* como métodos formais de anotação; *datasets* e modelos como representações e instanciações; ferramentas computacionais que operacionalizam processos de anotação e avaliação. Esses elementos foram integrados aos estudos de caso e avaliados em contextos reais.

2.4 Teoria Fundamentada

A Teoria Fundamentada [Glaser e Strauss 2017] foi adotada como referencial para a análise qualitativa de dados empíricos produzidos durante a pesquisa. Embora não tenha sido aplicada em sua forma metodológica completa, a vertente Sociotécnica [Hoda 2022] foi utilizada como estrutura conceitual para organizar a interpretação dos dados em torno de três eixos: qualidade de anotação, padrões de projeto e decomposição de tarefas.

A análise foi conduzida com base em evidências como versões de *guidelines*, interações entre anotadores e decisões de adjudicação. Os achados resultantes sustentam a formulação dos padrões apresentados e a proposta metodológica de decomposição baseada em práticas reais de anotação.

A combinação das abordagens descritas neste capítulo — estudo de caso como estratégia central, *Design Science Research* para construção de artefatos, revisão de literatura orientada por propósitos analíticos e Teoria Fundamentada Sociotécnica como referencial qualitativo — fornece a base metodológica para o desenvolvimento desta tese. Cada uma dessas abordagens contribui de forma complementar para a articulação entre teoria e prática, permitindo tanto a formulação de padrões conceituais quanto sua validação empírica em contextos reais de anotação. No capítulo seguinte, apresenta-se a revisão da literatura que fundamenta os pilares conceituais da pesquisa, com foco na qualidade de anotação de *corpus*, padrões reutilizáveis e decomposição de tarefas.

CAPÍTULO 3

Revisão da Literatura

No estágio básico da TFST, apresentado no Capítulo 2, foram identificados os conceitos fundamentais que serão importantes para a compreensão de toda a pesquisa. Neste capítulo, apresentaremos uma revisão limitada da literatura, que envolve estes conceitos fundamentais que servirão de base teórica para os capítulos seguintes.

Proponentes da Teoria Fundamentada recomendam uma **exposição limitada a literatura** [Stol, Ralph e Fitzgerald 2016], em vez de começar com uma revisão da literatura abrangente. O objetivo desta recomendação é desenvolver inicialmente uma mente aberta (*open-mindedness*) e assim evitar vieses de confirmação. Assim, evita-se apegar a teorias preconcebida e buscar o desenvolvimento de novas teorias originais a partir das observações empíricas. Neste sentido, esta revisão 'enxuta' da literatura (*lean literature review*) descreveu [Hoda 2022] e foi concentrada em *surveys* e artigos referenciados por estes, visando conhecer o estado da arte dos assuntos mais importantes, dominar conceitos fundamentais e identificar lacunas de pesquisa.

3.1 Qualidade de Anotação

Apesar de ser um assunto relevante para a avaliação, validação e treinamento de modelos de aprendizado de máquina, o problema da qualidade de anotação de *corpus* é um assunto de pesquisa muito recente. A string de busca '`"corpus annotation" or "dataset annotation" or quality`' aplicada no Google Acadêmico resulta em somente um *survey*. Trata-se do extenso trabalho de [Klie, Castilho e Gurevych 2024], ainda sem revisão por pares.

A **qualidade da anotação** de um *corpus* é crucial para garantir a precisão, generalização e consistência dos modelos de PLN (PLN). Anotações bem feitas reduzem vieses, melhoram a reprodutibilidade dos experimentos e evitam retrabalho, economizando tempo e recursos. Além disso, um *corpus* de alta qualidade é essencial para validar hipóteses científicas, aumentar a confiança da comunidade e dos usuários, e garantir o bom desempenho de sistemas de PLN em ambientes de produção, evitando erros que possam comprometer a confiabilidade dos resultados e aplicações.

Segundo Klie et al (2024), a qualidade de anotação dos dados é fundamental para o desenvolvimento de modelos de aprendizado de máquina precisos e confiáveis, mas muitos dos conjuntos de dados amplamente utilizados contêm erros ou vieses. Apesar de existirem **guidelines** para a criação de conjuntos de dados de alta qualidade, ainda não foi realizada uma análise em grande escala sobre as práticas de gerenciamento de qualidade em ***datasets*** de linguagem natural. Em seguida, os autores analisaram 591 artigos científicos que apresentam conjuntos de dados textuais, com foco em aspectos como gerenciamento de anotadores, **acordo inter-anotadores**, adjudicação e validação de dados. Embora a maioria dos trabalhos aplique boas práticas de qualidade, 30% das publicações apresentam gerenciamento deficiente, com problemas frequentes relacionados ao cálculo de *acordo inter-anotadores* e taxas de erro de anotação.

Um ponto importante antes de analisar a qualidade de um conjunto de *corpus* é defini-la *a priori*. [Klie, Castilho e Gurevych 2024] definiu a *qualidade de anotação* como uma combinação de 4 aspectos:

- **Estabilidade** consiste em seus resultados não variarem com o tempo, ou seja, dado um fenômeno, ele deve ser anotado de forma semelhante independente se foi anotado mais cedo mais tarde. A instabilidade pode, por exemplo, ocorrer devido a descuido, distrações ou cansaço, mudança nas diretrizes de anotação ou até mesmo aprendizado por meio da prática.
- **Reprodutibilidade** reflete se um processo de criação de conjunto de dados é reproduzível, ou seja, se diferentes anotadores ainda puderem fornecer os mesmos resultados com a mesma documentação de projeto referente ao processo, às diretrizes e ao esquema.
- **Precisão** ocorre quando as anotações e os textos criados durante o processo são precisos se estiverem de acordo com as diretrizes e o resultado desejado.
- **Imparcialidade** descreve o grau em que os artefatos criados estão livres de erros sistemáticos e não aleatórios (viés).

[Klie, Castilho e Gurevych 2024] também argumentam que a *concordância entre anotadores*, medida através de métricas, é um valor indireto que estima a confiabilidade e que reflete a estabilidade, reprodutibilidade e precisão. Com esta definição em mente, os autores identificaram boas práticas que dão suporte à gestão de qualidade de anotação, identificando 22 métodos, que foram agrupados em 5 categorias: processo de anotação, gerenciamento dos anotadores, estimativa de qualidade, melhoria de qualidade e adjudicação. Os métodos divididos em categorias são ilustrados na Figura 3.1.

Um ponto também muito importante apontado pelo trio de autores é o constante uso arbitrário das faixas de valores de concordância entre anotadores para medir a confiabilidade da anotação. Por exemplo, [Landis e Koch 1977] rotulou faixas de valores

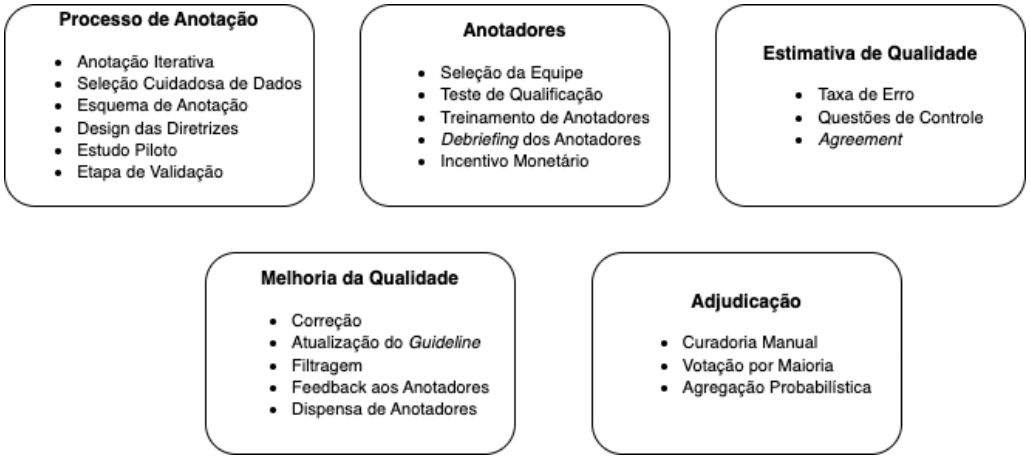


Figura 3.1: *Métodos considerados como boas práticas, adaptado de [Klie, Castilho e Gurevych 2024]*

de Kappa de Cohen (K_c) com 0,01 - 0,20 concordância leve, 0,21 - 0,40 concordância justa, 0,41 - 0,60 concordância moderada, 0,61 - 0,80 concordância substancial, 0,81 - 1,00 concordância quase perfeita.

Além desta métrica, existem diversas outras, específicas para o tipo de situação, tal como o Alfa de Krippendorf (α). Em [Krippendorff 2004] o autor desta métrica declarou:

> *A concordância é o que medimos;* ***confiabilidade*** *é o que desejamos inferir a partir dela.*

Neste trabalho, depois de apontar vários problemas do uso de sua métrica para inferir a **confiabilidade dos dados**, Krippendorff apontou várias considerações e condições prévias para poder aplicar essa inferência.

[Klie, Castilho e Gurevych 2024] também cita vários autores que afirmam que escolher um **nível de concordância** considerado bom o suficiente não é trivial. Há diversas razões para tal afirmação: não há um *nível de concordância* aceitável universal para todas as situações; também não há uma métrica aplicável para todo tipo de tarefa; as métricas variam conforme o tamanho das instâncias observadas ou pela quantidade de anotadores; o limiar (*threshold*) pode variar em função da subjetividade da tarefa; etc. Finalmente, sugere que o valor de confiabilidade de um trabalho seja comparado com tarefas similares, caso seja possível.

Outro ponto relevante é como a **subjetividade** influencia na qualidade de anotação, uma vez que não se trata de um erro e sim de itens que são naturalmente dependentes de perspectivas pessoais e não em verdades universais. [Palomaki, Rhinehart e Tseng 2018] trata tais itens como variações aceitáveis e sugere algumas estratégias no *design* da tarefa, tal como aceitar que múltiplos rótulos sejam considerados corretos para um item classificado. Por outro lado, [Reidsma e Akker 2008]

sugere evitar o descarte de dados subjetivos identificando-os para serem classificados por modelos treinados nesses subconjuntos.

Apesar do extenso estudo, [Klie, Castilho e Gurevych 2024] não analisaram aspectos relacionados à qualidade de anotação associados ao uso de ferramentas de anotação. A proposta dessa tese é analisar não somente os eixos estrutural e gerencial, mas também aspectos mais amplos, como padrões de projeto. Por isso, neste capítulo, foi feito um levantamento de diversos tipos de padrões que serão apresentados na Seção 3.2 seguinte.

3.2 Padrões

O livro *A Pattern Language* [Alexander, Ishikawa e Silverstein 1977] é uma obra seminal que propõe uma abordagem sistemática para a resolução de problemas complexos de *design*, baseada em padrões observados e testados no mundo real. A obra apresenta 253 padrões de arquitetura em diversas escalas, abrangendo desde casas até cidades.

Os autores propõem uma linguagem onde os elementos fundamentais são os padrões, inter-relacionados e combináveis entre si, oferecendo escalabilidade e flexibilidade de aplicação para diversos contextos voltados para o bem-estar humano. O termo **linguagem de padrões** foi escolhido por representar uma gramática de soluções que, assim como uma linguagem natural, permite a combinação e recombinação de elementos para formar composições adaptáveis e coerentes. Embora inicialmente formulado no domínio da Arquitetura, o conceito de linguagem de padrões influenciou profundamente outros campos, como Engenharia de *Software* [Gamma et al. 1995, p. 357], Interação Humano-Computador [Dearden e Finlay 2006], Educação [Sharp e Eckstein 2003], entre outros.[1]

Segundo os autores, um padrão pode ser definido como:

> *Cada **padrão** descreve um problema que ocorre repetidamente em nosso ambiente e, em seguida, descreve o núcleo da solução para esse problema, de tal forma que você pode usar essa solução um milhão de vezes, sem nunca fazer da mesma maneira duas vezes.* [Alexander, Ishikawa e Silverstein 1977, p. X]

Essa concepção é aprofundada em *The Timeless Way of Building* [Alexander 1979], que complementa a fundamentação teórica do uso de padrões. Três ideias centrais podem ser destacadas:

[1] Uma síntese abrangente dos domínios de aplicação pode ser consultada na Wikipedia: https://en.wikipedia.org/wiki/Pattern_language#Application_domains.

- **Padrões vivos resolvem sistemas de forças**: soluções efetivas resultam do equilíbrio de múltiplas forças em um ambiente físico ou social, promovendo harmonia e bem-estar [Alexander 1979, p. 134].
- **A linguagem de padrões parte de um núcleo gerador**: tal como um embrião que se desdobra a partir de um conjunto genético inicial, as construções urbanas se desenvolvem por atos criativos mediados por uma linguagem compartilhada [Alexander 1979, p. xiii].
- **Soluções são singulares, mesmo que derivadas do mesmo padrão**: como os contextos nunca se repetem exatamente, a aplicação de um mesmo padrão pode gerar formas distintas [Alexander 1979, p. 147].

A **linguagem de padrões** é descrita como um sistema de regras que permite às pessoas moldarem seus próprios ambientes:

> *Uma linguagem de padrão dá a cada pessoa que a utiliza o poder de criar uma variedade infinita de edifícios novos e exclusivos, assim como sua linguagem comum lhe dá o poder de criar uma variedade infinita de frases [...]. Cada* ***padrão*** *é uma regra que descreve o que você deve fazer para gerar a entidade que ele define [...]. É nesse sentido que o sistema de padrões forma uma linguagem* [Alexander 1979, p. 167–183].

Um exemplo concreto é dado na descrição de casas de pedra do sul da Itália, cujos padrões arquitetônicos – como salas quadradas, arcos, abóbadas e superfícies caiadas – compõem uma linguagem que possibilita tanto individualidade quanto coerência nas construções locais [Alexander 1979, p.188].

Concluindo, Alexander [Alexander, Ishikawa e Silverstein 1977, Alexander 1979] criou um arcabouço teórico para a aplicação de soluções recorrentes a problemas complexos, estabelecendo um modelo de representação que equilibra generalidade e adaptabilidade. Sua influência se estendeu a diversas disciplinas, dando origem, por exemplo, ao conceito de **padrões de projeto** na Engenharia de *Software*, tema da próxima subseção.

3.2.1 Padrões de Projeto

Inspirados na abordagem de Alexander, [Gamma et al. 1995] introduziram os **padrões de projeto** no domínio da Engenharia de *Software*. Esses padrões sistematizam boas práticas no desenvolvimento orientado a objetos, promovendo reuso e facilitando a comunicação entre desenvolvedores.

Cada padrão é descrito por quatro componentes essenciais:

- **Nome**: cria um vocabulário comum entre os profissionais.

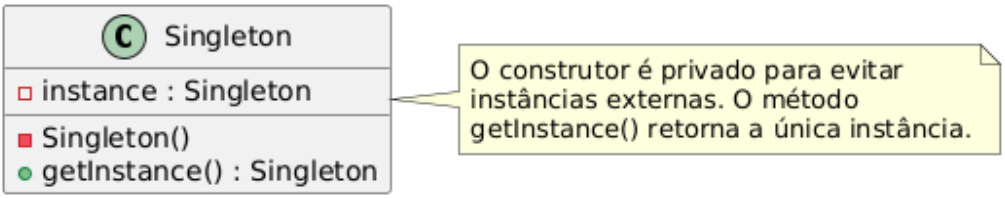


Figura 3.2: *Padrão de projeto Singleton, utilizado para fornecer uma única instância de uma Classe*

- **Problema**: define o contexto e as situações em que o padrão se aplica.
- **Solução**: apresenta os elementos envolvidos e seus relacionamentos, de forma abstrata.
- **Consequências**: discute os impactos da adoção do padrão, como flexibilidade e desempenho.

O catálogo clássico de [Gamma et al. 1995] organiza 23 padrões em três categorias:

- **Padrões criacionais**: tratam da instanciação de objetos (e.g., *Singleton*, Figura 3.2).
- **Padrões estruturais**: organizam a composição entre classes e objetos.
- **Padrões comportamentais**: regulam a comunicação entre objetos e a atribuição de responsabilidades.

A abordagem foi posteriormente expandida por [Fowler 2013], que catalogou padrões voltados a domínios específicos como aplicações empresariais. Os padrões são agrupados em categorias como Arquitetura (e.g., *Layered Architecture*), Lógica de Domínio (e.g., *Domain Model*), Persistência (e.g., *Data Mapper*), Distribuição (e.g., *DTO*), Apresentação (e.g., *Front Controller*), entre outros.

Além disso, [Shaw e Garlan 1996] introduziram o conceito de **estilos arquiteturais**, que estabelecem princípios de organização e interconexão entre componentes de *software*. Exemplos incluem o estilo *pipe-and-filter*, cliente-servidor e REST [Fielding e Taylor 2002], amplamente adotado no *design* de APIs *Web*.

Conclui-se que os padrões de projeto, em suas diversas formas e níveis de abstração, constituem um arcabouço fundamental para o desenvolvimento de sistemas complexos. Sua utilização promove não apenas reuso e qualidade, mas também a criação de uma linguagem compartilhada entre os profissionais da área.

Além dos domínios já consolidados como Arquitetura e Engenharia de *Software*, o conceito de padrões tem sido estendido a áreas emergentes como aprendizado de máquina, sistemas interativos e práticas colaborativas com humanos no circuito. Um panorama dessas aplicações contemporâneas — incluindo padrões de processo, padrões para engenharia de sistemas de aprendizado de máquina e padrões para sistemas com

intervenção humana (Human-in-the-Loop) — é apresentado no Anexo B.1, com exemplos e taxonomias que evidenciam o potencial de reutilização e adaptação de padrões nesses contextos.

3.3 Decomposição de Tarefas

A **decomposição de tarefas** é um conceito central em diversas teorias que buscam compreender e organizar processos complexos, oferecendo um quadro sistemático para dividir problemas amplos em partes menores e inter-relacionadas. Sob uma perspectiva teórica, essa abordagem permite a formalização de processos cognitivos, organizacionais e computacionais, estabelecendo uma base para análises estruturais e funcionais mais aprofundadas. Nesta seção, serão discutidos os fundamentos conceituais e as definições que sustentam a decomposição de tarefas, destacando sua relevância enquanto constructo teórico. Além disso, serão examinados seus desdobramentos epistemológicos e metodológicos, com foco em como essa técnica molda a compreensão de problemas complexos em diferentes domínios e sustenta práticas acadêmicas, como o *design* de modelos e processos analíticos em áreas como o processamento de linguagem natural.

3.3.1 Decomponibilidade

A **decomponibilidade**, conceito amplamente discutido por Herbert Simon em sua teoria sobre a arquitetura de sistemas complexos [Simon 1962], é a propriedade que viabiliza a divisão de sistemas ou problemas em partes menores e manejáveis, tornando-se essencial para a **decomposição de tarefas**. Simon destacou que sistemas altamente decomponíveis possuem componentes relativamente independentes, o que facilita sua análise e modificação. Essa relação é fundamental, pois a decomponibilidade determina a eficácia da segmentação, garantindo que os componentes resultantes sejam funcionais e contribuam para a solução do todo.

O conceito de **sistema** tem raízes etimológicas no grego antigo *sýstēma*, que significa "conjunto organizado"ou "todo composto por partes", destacando desde sua origem a ideia de elementos interconectados que formam uma unidade funcional. Ludwig von Bertalanffy, um dos principais teóricos na definição moderna de sistema, descreveu-o como "complexo de elementos em interação"[Bertalanffy 1969, p. 33]. Em sua "Teoria Geral dos Sistemas", Bertalanffy enfatizou que sistemas podem ser abertos ou fechados, dependendo de sua interação com o ambiente externo, e destacou que o comportamento do sistema não pode ser compreendido apenas pela soma de suas partes, mas sim pelas interações entre elas. Essa definição amplia a aplicação do conceito para diversas áreas do

conhecimento, como biologia, sociologia, cibernética e ciência da computação, servindo como base para o estudo de estruturas complexas.

Simon introduz sua obra explicando a missão das Ciências Naturais: "a tarefa central de uma ciência natural é tornar o maravilhoso comum: mostrar que a complexidade, vista corretamente, é apenas uma máscara para a simplicidade; encontrar um padrão oculto no caos aparente", disse [Simon 2019, p. 1, tradução nossa]. Enquanto as Ciências Naturais lidam com **sistemas naturais**, como ecossistemas e organismos, que surgem espontaneamente na natureza, os **sistemas artificiais**, como máquinas e organizações, são projetados pelo ser humano com objetivos específicos. Para teorizar sobre esses sistemas, Herbert Simon escreveu a obra sobre as Ciências do Artificial [Simon 2019]. Herbert Simon destaca que os *sistemas artificiais* são moldados por finalidades e restrições ambientais, frequentemente inspirados nos naturais. [Simon 2019, p. 5] assim distingue os dois sistemas: 1. Coisas artificiais são sintetizadas por seres humanos. 2. As coisas artificiais podem imitar as aparências das coisas naturais. 3. As coisas artificiais podem ser caracterizadas em termos de funções, objetivos e adaptação. 4. As coisas artificiais são frequentemente discutidas, especialmente quando estão sendo projetadas, em termos de requisitos e descrições.

Sobre o aspecto funcional das coisas artificiais, [Simon 2019, p. 5] afirma: "*O cumprimento do propósito ou a adaptação a um objetivo envolve uma relação entre três termos: o* ***propósito ou objetivo****, o* ***caráter do artefato*** *e o* ***ambiente*** *no qual o* ***artefato*** *atua.*". Por exemplo, o *propósito* dos relógios é informar a hora, mas existem diversos tipos com *características* distintas, projetados para *ambientes* específicos. Um relógio de pêndulo, por exemplo, funciona bem fixo sobre uma lareira, mas enfrentaria sérios problemas em um navio sujeito a balanços constantes. Já um relógio solar é eficiente em locais ensolarados, mas seria inútil durante o inverno no Ártico.

Outro conceito importante na teoria proposta por [Simon 2019, p. 6] é considerar artefatos como **interfaces** entre o **ambiente externo** e o **ambiente interno**, em tradução nossa:

> *...um artefato pode ser considerado um ponto de encontro - uma "****interface****" nos termos atuais - entre um* ***ambiente "interno"****, a substância e a organização do próprio artefato, e um* ***ambiente "externo"****, o ambiente em que ele opera...Observe que essa maneira de ver os artefatos se aplica igualmente bem a muitas coisas que não são feitas pelo homem - a todas as coisas que, de fato, podem ser consideradas adaptadas a alguma situação; e, em particular, aplica-se aos sistemas vivos que evoluíram por meio das forças da evolução orgânica... Análogo ao papel desempenhado pela seleção natural na biologia evolutiva é o papel desempenhado pela racionalidade nas ciências do comportamento humano...*

Simon também propôs a aplicação do conceito de interface na **predição do comportamento** de coisas artificiais e naturais (tradução nossa):

> *...a primeira vantagem de dividir o ambiente externo do interno ao estudar um sistema adaptativo ou artificial é que, muitas vezes, podemos **prever o comportamento** a partir do conhecimento dos objetivos do sistema e de seu ambiente externo, com apenas suposições mínimas sobre o ambiente interno...Em muitos casos, o fato de um determinado sistema atingir uma determinada meta ou adaptação depende apenas de algumas características do ambiente externo e não dos detalhes desse ambiente. Os biólogos estão familiarizados com essa propriedade dos sistemas adaptativos sob o rótulo de homeostase...De uma forma ou de outra, o projetista isola o sistema interno do ambiente, de modo que seja mantida uma relação invariável entre o sistema interno e a meta, independente de variações em uma ampla faixa da maioria dos parâmetros que caracterizam o ambiente externo.*

Outro conceito também relevante é o **ambiente da tarefa** e sua relação com a **predição do comportamento**:

> *...A descrição de um artifício em termos de sua organização e funcionamento - sua interface entre os ambientes interno e externo - é um dos principais objetivos da atividade de invenção e design...O ambiente externo determina as condições para a realização da meta. Se o sistema interno for projetado adequadamente, ele será adaptado ao ambiente externo, de modo que seu comportamento será determinado em grande parte pelo comportamento desse último...Para prever como ele se comportará, precisamos apenas perguntar: "Como um sistema racionalmente projetado se comportaria nessas circunstâncias?" O comportamento assume a forma do **ambiente da tarefa**."*

Com base nesses conceitos fundamentais introduzidos no primeiro capítulo de sua obra, Simon desenvolve sua teoria em outros capítulos, envolvendo várias áreas interdisciplinares, tais como a Computação, Economia Comportamental [2], Psicologia Cognitiva, Ciência do *Design*, etc. O capítulo final tem como título "A Arquitetura da Complexidade: Sistemas Hierárquicos", nele Simon conceitua **sistemas complexos**, **sistemas hierárquicos**, e **subsistema elementar** que são tratados mais profundamente em um artigo anterior do mesmo autor, [Simon 1962, tradução nossa] assim descreve esses dois conceitos:

[2] Através da teoria de racionalidade limitada (bounded rationality), Simon ganhou o prêmio Nobel de 1978, mais informações em https://www.nobelprize.org/uploads/2018/06/simon-lecture.pdf

> *"...a grosso modo, por **sistema complexo** [3] entendo aquele composto por um grande número de partes que interagem de forma não simples... frequentemente a complexidade assume a forma de hierarquia - o sistema complexo sendo composto de subsistemas que, por sua vez, têm seus próprios subsistemas, e assim por diante...Por **sistema hierárquico**, ou hierarquia, entendo um sistema composto de subsistemas inter-relacionados, cada um deles, por sua vez, hierárquico em sua estrutura até chegarmos a algum nível mais baixo de subsistema elementar. Na maioria dos sistemas da natureza, é um tanto arbitrário o ponto em que deixamos o particionamento e quais subsistemas consideramos elementares...Em um tipo de pesquisa biológica, uma célula pode ser tratada como um **subsistema elementar**; em outro, uma molécula de proteína; em outro ainda, um resíduo de aminoácido."*

Para ilustrar o conceito de *sistemas hierárquicos*, [Simon 1962, p. 469, tradução nossa] apresentou quatro tipos de sistemas - social, biológico, físico e simbólico:

> *"...Um tipo de hierarquia que é frequentemente encontrado nas **ciências sociais**: uma organização formal. Empresas, governos, universidades, todos têm uma estrutura claramente visível de partes dentro de partes...A estrutura hierárquica dos **sistemas biológicos** é um fato bem conhecido. Tomando a célula como o bloco de construção, encontramos células organizadas em tecidos, tecidos em órgãos e órgãos em sistemas. Descendo a partir da célula, subsistemas bem definidos - por exemplo, núcleo, membrana celular, microssomos, mitocôndrias e assim por diante...A estrutura hierárquica de muitos **sistemas físicos** é igualmente clara. ... No nível microscópico, temos partículas elementares, átomos, moléculas e macromoléculas. No nível macroscópico, temos sistemas de satélites, sistemas planetários, galáxias...Uma classe muito importante de sistemas foi omitida em meus exemplos até agora: **sistemas de produção simbólica humana**. Um livro é uma hierarquia no sentido em que estou usando esse termo. Geralmente é dividido em capítulos, os capítulos em seções, as seções em parágrafos, os parágrafos em sentenças, as sentenças em cláusulas e frases, as cláusulas e frases em palavras. "*

Para analisar a decomposição de aplicações em sistemas computacionais [Courtois 1977] utilizou métodos analíticos e numéricos denominados **decomponibilidade quase-completa**, que possibilitaram analisar o desempenho de sistemas complexos

[3] Simon ponderou que há dois tipos de sistemas complexos: organizados e desorganizados e que seus estudos tratam do primeiro tipo, sendo que definições mais formais podem ser consultadas em [Weaver 1948]

	A1	A2	A3	B1	B2	C1	C2	C3
A1	-	100	-	2	-	-	-	-
A2	100	-	100	1	1	-	-	-
A3	-	100	-	-	2	-	-	-
B1	2	1	-	-	100	2	1	-
B2	-	1	2	100	-	-	1	2
C1	-	-	-	2	-	-	100	-
C2	-	-	-	1	-	100	-	100
C3	-	-	-	-	2	-	100	-

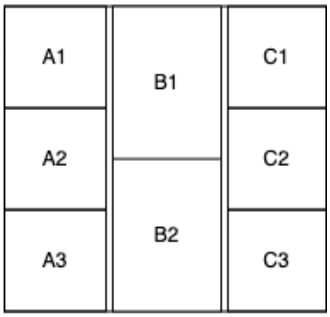


Figura 3.3: *Adaptado de [Simon 1962]. Um sistema hipotético quase decomponível, que consiste na representação da troca de calor em ambientes isolados por paredes (linhas duplas) ou divisórias (linhas simples), sendo que quanto menos isolados maior é o número (100); e quanto mais isolado menor é o valor (1 ou 2). Al, A2 e A3 podem ser interpretados como cubículos em um cômodo, B1 e B2 como cubículos em um segundo cômodo e C1, C2 e C3 como cubículos em um terceiro. As entradas da matriz são os coeficientes de difusão de calor entre os cubículos.*

como sistemas de filas de uso de um recurso, como o gerenciamento de memórias com diversos níveis de desempenho distintos. Coube a [Courtois 1977] aplicar esses métodos em diversos outros domínios, notadamente na Economia; e assim cunhou o termo **quase-decomponibilidade**: *"... Nos sistemas hierárquicos, podemos distinguir entre as interações entre os subsistemas, por um lado, e as interações dentro dos subsistemas, ou seja, entre as partes desses subsistemas, por outro. As interações nos diferentes níveis podem ser, e muitas vezes serão, de diferentes ordens de magnitude. Em uma organização formal, geralmente haverá mais interação, em média, entre dois funcionários que são membros do mesmo departamento do que entre dois funcionários de departamentos diferentes. Em substâncias orgânicas, as forças intermoleculares geralmente são mais fracas do que as forças moleculares, e as forças moleculares são mais fracas do que as forças nucleares."*. A Figura 3.3 ilustra um exemplo de sistema quase decomponível.

O termo "quase" em *quase-decomponibilidade* enfatiza que os subsistemas de um sistema complexo não são completamente independentes, mas "quase independentes", com interações fortes dentro de cada subsistema e interações mais fracas entre eles. Isso reflete a conectividade inerente entre partes do sistema, permitindo uma análise simplificada e modular, sem ignorar a existência de trocas residuais. Esse conceito reconhece a complexidade realista dos sistemas, admitindo que as interações entre subsistemas podem variar em intensidade dependendo da escala ou do contexto, o que torna a decomposição aproximada e adaptável.

Herbert Simon distingue **descrições de estado** e **descrições de processo** para analisar *sistemas complexos*. As *descrições de estado* são estáticas e capturam a configuração atual do sistema em um momento específico, como a posição de peças em um tabuleiro de xadrez ou a temperatura de um sistema físico. Já as *descrições de processo* são dinâmicas e explicam como o sistema evolui de um estado para outro, como as regras que governam os movimentos no xadrez ou os mecanismos de uma reação química. Ambas são complementares: os estados mostram "onde estamos", e os processos explicam "como chegamos aqui" ou "para onde vamos". [Simon 1962] explica como reduzir o grau de complexidade de sistemas através da simplificação, enfatizando que o caminho é através de descrições que exploram a sua redundância: "O grau de complexidade ou simplicidade de uma estrutura depende fundamentalmente da maneira como a descrevemos. A maioria das estruturas complexas encontradas no mundo é extremamente redundante, e podemos usar essa redundância para simplificar sua descrição. Mas para usá-la, para obter a simplificação, precisamos encontrar a representação correta.".

Esses conceitos – *quase decomponibilidade* e *representação* – também estão relacionados à **compreensão humana da complexidade**. Simon estabelece essa interconexão da seguinte forma:

> *"Se você pedir a uma pessoa que desenhe um objeto complexo - por exemplo, um rosto humano -, ela quase sempre procederá de **forma hierárquica**. Em primeiro lugar, ela fará o contorno do rosto. Em seguida, adicionará ou inserirá características: olhos, nariz, boca, orelhas, cabelo. Se for solicitado a detalhar mais, ele começará a desenvolver detalhes para cada uma das características - pupilas, pálpebras, cílios para os olhos e assim por diante - até atingir os **limites de seu conhecimento** anatômico. Suas informações sobre o objeto são organizadas de forma hierárquica na memória, como um sumário de tópicos...as subpartes pertencentes a diferentes partes interagem apenas de forma agregada - os detalhes de sua interação podem ser ignorados...ao estudar a interação de duas nações, não precisamos estudar em detalhes as interações de cada cidadão da primeira com cada cidadão da segunda...O fato, então, de muitos sistemas complexos terem uma **estrutura hierárquica quase decomponível** é um fator facilitador importante que nos permite entender, descrever e até mesmo "ver" esses sistemas e suas partes."*

Nesta seção, foram apresentados conceitos fundamentais sobre a decomponibilidade de sistemas complexos. Por meio deles, foi possível compreender aspectos essenciais, como a distinção entre *sistemas naturais* e *artificiais*, a recorrência da *hierarquização em sistemas complexos*, o método de *quase decomponibilidade* – que facilita a análise das interações entre as diversas partes e subpartes de um sistema hierárquico – e, por fim,

a importância das *descrições de estado e de processo* na *compreensão humana* desses sistemas.

3.3.2 Cognição e Tarefas Humanas

A relação entre **tarefa e cognição humana** é um tema central na **ciência cognitiva**,pois permite compreender como as pessoas processam informações, tomam decisões e executam ações em diferentes contextos. Nesse domínio, uma tarefa é definida como um objetivo a ser alcançado, envolvendo a interação de processos mentais, como atenção, memória, percepção e raciocínio, com as condições impostas pelo ambiente. O objetivo desta seção é explorar como o conceito de tarefa pode ser usado para investigar os mecanismos cognitivos subjacentes ao comportamento humano, destacando sua relevância tanto para a compreensão teórica da cognição quanto para aplicações práticas, como na arquitetura de sistemas, na psicologia e no *design* de sistemas interativos.

Na *ciência cognitiva*, a memória é geralmente dividida em curto prazo, longo prazo e de trabalho, com funções distintas, mas interdependentes. A **memória de curto prazo** mantém informações por períodos breves, sendo essencial para tarefas imediatas, como lembrar um número de telefone [Atkinson 1968, p. 111]. Já a **memória de longo prazo** armazena informações por períodos extensos, incluindo fatos, eventos e habilidades. A **memória de trabalho**, por sua vez, combina armazenamento e processamento temporário, sendo crucial para atividades cognitivas complexas, como raciocínio e resolução de problemas. Esses sistemas interagem continuamente, formando a base para o aprendizado e o comportamento humano.

Uma teoria também importante na memorização de informações é o ***chunking***. Este termo foi cunhado por George Miller, e consiste em um processo de agrupar unidades significativas da memória (*chunks*), formado pela junção de um conjunto de agrupamentos ou blocos já formados na memória e sua fusão em uma unidade maior. O *chunking* implica a capacidade de construir essas estruturas recursivamente, levando assim a uma organização hierárquica da memória. Através deste processo mental, o ser humano é capaz de lidar com a capacidade limitada da *memória de curto prazo*, estimada em 7 ± 2 itens [Miller 1956]. Por exemplo, em vez de memorizar o dígito individual de um número de telefone '556212563478' é melhor visualizá-lo através de fragmentos (*chunks*): '+55 62 1256-3478'.

Outra teoria também relevante é a **Lei de potência da prática**, comprovada empiricamente por [Seibel 1963], que afirma que o desempenho em uma tarefa melhora consistentemente com a prática, seguindo uma relação matemática dada por: $T = T_0 \cdot N^{-b}$ onde T é o tempo necessário para executar a tarefa após N repetições, T_0 é o tempo inicial gasto na execução, e b é uma constante positiva que reflete a taxa de

aprendizado. À medida que N aumenta, o tempo T diminui de forma não linear, com as maiores melhorias ocorrendo nas primeiras repetições. Segundo [Newell 1994, p.7], a teoria de *chunking* ajuda a explicar a Lei de Potência da Prática. Ele argumenta que as pessoas agrupam informações em blocos (*chunks*) continuamente à medida que ganham experiência, e esses blocos tornam o desempenho mais eficiente quando são relevantes para a tarefa. Inicialmente, a prática acelera o aprendizado, pois novos blocos úteis são criados rapidamente, levando a uma melhoria exponencial. Com o tempo, blocos mais complexos e gerais, que aparecem menos frequentemente, tornam-se menos úteis, e o impacto da prática diminui, desacelerando o aprendizado. Esse padrão de aprendizado, embora não seja exatamente uma lei de potência, se aproxima dela, com ganhos rápidos no início e uma desaceleração progressiva depois. Assim, o *chunking* é um fator subjacente que contribui para essa dinâmica, possibilitando a formação de estruturas cognitivas mais eficientes com a prática.

Além do conceito de *chunking*, há dois conceitos muito importantes na teoria de Simon e Newell. O **ambiente da tarefa** refere-se ao contexto externo e objetivo do problema, incluindo os objetos, regras e condições que o definem, enquanto o **espaço de problema** é a representação interna e abstrata desse ambiente, construída pelo agente (humano ou máquina) para resolver o problema. O *espaço de problema* é formado por estados (configurações possíveis) e operadores (ações que podem ser realizadas), e sua eficácia depende de como o ambiente da tarefa é representado e processado. Por exemplo, no jogo de xadrez, o *ambiente da tarefa* inclui o tabuleiro e as regras, enquanto o *espaço de problema* abrange todas as configurações possíveis do tabuleiro e os movimentos válidos das peças. A complexidade do *espaço de problema* pode ser gigantesca, dado o grande número de posições possíveis que o tabuleiro pode ter após várias jogadas, o que torna a busca por uma solução altamente desafiadora. Em resumo, o *ambiente da tarefa* é a realidade externa, e o *espaço de problema* é a maneira como essa realidade é interpretada e abordada para encontrar uma solução.

Outro conceito central é a **busca** nas teorias de Allen Newell, pois explica como humanos e sistemas artificiais exploram um *espaço de problema* para resolver desafios e aprender. Na visão de Newell, a resolução de problemas envolve navegar nesse espaço, onde cada configuração representa um *estado do problema*, e o objetivo é encontrar o caminho que conecta a situação inicial à solução desejada [Newell 1994, p. 10]. Esse conceito foi aplicado no ***General Problem Solver*** **(GPS)**, um sistema pioneiro de inteligência artificial que utilizava métodos *heurísticos* para simular a solução de problemas de maneira similar ao pensamento humano. A busca também está relacionada ao aprendizado, especialmente no processo de formação de *chunks*, envolvendo a recuperação de experiências prévias com tarefas análogas [Simon e Newell 1971]. Além disso, ela reflete a flexibilidade cognitiva humana, permitindo a adaptação a ambientes dinâmicos e deci-

sões sob incerteza. Assim, a busca conecta cognição, aprendizado e inteligência artificial, sendo essencial para entender o comportamento inteligente.

Além do conceito de busca, [Simon e Newell 1971] também explicaram o papel da **linguagem na representação dos espaços de problemas** (tradução própria):

> *"... a pesquisa sobre solução de problemas começou a deixar de perguntar como as buscas são conduzidas em espaços de problemas, um assunto sobre o qual adquirimos uma compreensão considerável, para perguntar como os espaços de problemas - representações internas de problemas - são construídos nas mentes humanas. No entanto, o tema da representação interna vincula a pesquisa sobre solução de problemas a duas outras áreas importantes da psicologia: percepção e psicolinguística. As informações chegam ao solucionador de problemas humano principalmente na forma de **declarações em linguagem natural** e exibições visuais. Para que as informações sejam trocadas entre essas fontes externas e a mente, elas devem ser codificadas e decodificadas. As informações, conforme representadas externamente, devem ser transformadas para corresponder às **representações** nas quais são mantidas internamente. É muito difícil imaginar quais seriam essas transformações enquanto tivermos acesso apenas às representações externas, e não às internas. É um pouco como criar um programa para traduzir do inglês para o idioma X, quando ninguém nos diz nada sobre o idioma X."*

Entre as décadas de 1950 e 1980, esses conceitos emergiram em paralelo com o rápido avanço da computação eletrônica. Nesse contexto, teorias foram testadas e avaliadas por meio de modelos e arquiteturas projetados para executar tarefas envolvendo aprendizado de máquina. Durante esse período, surgiram diversas abordagens teóricas para a resolução de problemas sem a necessidade de programar soluções diretamente, formando o campo que viria a ser conhecido como **Inteligência Artificial (IA)**. Duas dessas abordagens ganharam destaque: a **IA simbólica** e a **IA conectivista**. A *IA simbólica* se baseava na representação de conhecimento por meio de símbolos, regras e lógica formal, enquanto a *IA conectivista* utilizava redes de conexões inspiradas no funcionamento de neurônios biológicos, dando origem às redes neurais. Ambas buscavam desenvolver **sistemas de processamento de informação**, uma metáfora central para compreender o funcionamento da mente humana. Esse conceito, inspirado no funcionamento dos computadores, propunha que a mente humana pudesse ser vista como um sistema capaz de receber, processar, armazenar e recuperar informações, de maneira análoga a um computador.

Entre as décadas de 1980 e 2000, com o amadurecimento dos conceitos previamente estabelecidos, o foco em estratégias de **raciocínio** e aprendizado computacional

cresceu exponencialmente, consolidando subcampos distintos dentro da *inteligência artificial (IA)*. Nesse período, o **aprendizado de máquina** tornou-se um componente central, oferecendo métodos que permitiram aos sistemas computacionais identificar padrões e generalizar conhecimentos a partir de dados, sem depender exclusivamente de regras programadas manualmente. [Mitchell 1997] definiu aprendizado de máquina da seguinte forma:

> **Definição 3.1** *Diz-se que um programa de computador aprende com a experiência E com relação a alguma classe de tarefas T e medida de desempenho P, se seu desempenho em tarefas em T, conforme medido por P, melhora com a experiência E.*

A *IA Simbólica*, predominantemente baseada no **método dedutivo**, explorava o uso de regras explícitas e estruturas formais para realizar inferências lógicas. Em contrapartida, a *IA Conectivista*, amplamente fundamentada no **método indutivo**, utilizava redes neurais para aprender diretamente de exemplos, simulando o raciocínio humano em tarefas como classificação, previsão e reconhecimento de padrões. Essa complementaridade metodológica impulsionou o surgimento de abordagens híbridas, promovendo avanços significativos na capacidade das máquinas de resolver problemas complexos.

Na Seção 3.3.3 seguinte, serão apresentados diversos métodos para decomposição de tarefas para aprendizado de máquina utilizando principalmente o *método indutivo*, que aprende indutivamente através de experiências baseadas em exemplos ou instâncias produzidas principalmente por meio de anotação de dados.

3.3.3 Métodos de Decomposição de Tarefas

Por ser uma atividade importante na compreensão humana de sistemas complexos, a **decomposição de tarefas** é utilizada em diversos contextos, assumindo definições e conotações variadas. Outras subáreas da computação que lidam com decomposição de tarefas são: Planejamento Automático [Ghallab, Nau e Traverso 2004], Controle Inteligente [Åström e McAvoy 1992], Sistemas Multi-Agentes [Jiang e Matsubara 2014], dentre outros. Nesta seção, são apresentados esses conceitos em diversos domínios de interesse, tais como: Computação em Geral, Mineração de Dados e Aprendizado de Máquina. Nosso foco de estudos é nos últimos domínios citados, especificamente no aprendizado supervisionado, no qual a atividade de anotação de dados é fundamental. No contexto de tarefas de classificação, [Rokach 2006] assim a definiu:

> **Definição 3.2** *A ideia da metodologia de decomposição para tarefas de classificação é dividir uma tarefa de classificação complexa em várias subtarefas mais simples e gerenciáveis que possam ser resolvidas usando os* ***métodos***

> ***de indução*** *existentes e, em seguida, unir suas soluções para resolver o problema original.*

Segundo [Rokach 2006], a *decomposição de tarefas* não apenas reduz a complexidade das tarefas, mas também melhora a compreensão e a modularidade dos modelos, diminui o tempo de treinamento, possibilita o uso de técnicas paralelas e oferece flexibilidade para a combinação de modelos (*ensembles*). Além disso, essa abordagem facilita o processamento de grandes volumes de dados ao subdividi-los em conjuntos menores, o que pode, potencialmente, aprimorar a acurácia preditiva [Kusiak 2000].

Taxonomia de Rokach

No intuito de sistematizar os diversos métodos de decomposição de tarefas, [Rokach 2006] propôs uma taxonomia, conforme ilustrado na Figura 3.4. Com base nessa tipologia, será apresentada uma análise dos diferentes métodos de decomposição de tarefas de aprendizado de máquina nos contextos de aplicação mencionados anteriormente. O primeiro critério de decomposição está associado ao termo **conceito**, que possui um nível de abstração adequado para englobar vários elementos de uma base de conhecimento em inteligência artificial, como categorias, tuplas, funções, hipóteses, relações, entre outros. [Michie 1995] destaca a relevância do papel de *conceito*[4] na inteligência artificial (tradução própria):

> *A Inteligência Artificial é a ciência da* ***engenharia de conceitos****. Queremos que as máquinas aceitem e usem os* ***conceitos das pessoas****. Queremos que elas descubram novos* ***conceitos*** *e os comuniquem às pessoas. Também queremos que elas unam* ***conceitos complexos*** *e, combinando dados de observação com conhecimentos pré-existentes, criem novas teorias importantes para a ciência e a tecnologia.*

Os **métodos de decomposição** propostos por [Rokach 2006] baseiam-se em **conceitos originais**, isto é, operam diretamente sobre as tuplas ou atributos dos conjuntos de dados, sem realizar transformações nos mesmos. Em contraste, outras abordagens trabalham com **conceitos intermediários**, que utilizam dados abstraídos a partir dos originais. Nesta análise, iniciaremos pela apresentação dos métodos de decomposição fundamentados em *conceitos originais*.

[4] Um método notável no aprendizado de máquina é o *Aprendizado de Conceitos*, que se refere ao processo de identificar ou inferir uma definição geral de um conceito a partir de exemplos específicos. Em outras palavras, consiste no aprendizado de categorias ou classes com base em exemplos positivos e negativos. Para mais detalhes, veja [Mitchell 1997, p. 32].

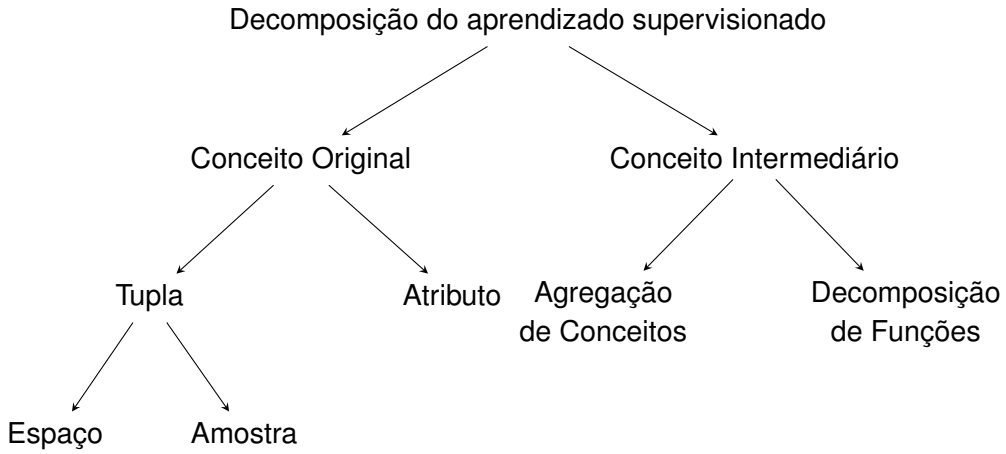


Figura 3.4: *Taxonomia de Métodos de Decomposição proposta por [Rokach 2006]. O primeiro critério envolve a divisão de conceitos originais e intermediários.*

Análise Geral dos Métodos de Decomposição

Nesta seção, foram analisados diversos *métodos de decomposição de tarefas*, conforme discutidos nas subseções anteriores (ver Seção B.2) e sistematizados na Tabela 3.1. A análise foi fundamentada, em parte, na taxonomia proposta por [Rokach 2006], que contempla propriedades como *mutualidade exclusiva* e *método de aquisição da estrutura*. Adicionalmente, foram incorporados dois métodos relevantes não incluídos na taxonomia original ([Bhargava 1999]; [Blei, Ng e Jordan 2003]), e ampliado o conjunto de propriedades analisadas. Os métodos listados na tabela foram descritos em maior detalhe, com exemplos e discussões complementares, no **Anexo B.2**.

Tipo de Conceito	**Base de Decomposição**	**Método**	**ED**	**ME**	**MAE**	**SAC**	**DA**	**TAM**
Original	Atributo (*Feature*)	[Kusiak 2000]	SC	S	M	MD	I	RS
		[Bay 1999]	SC	N	Arb	AM	TI	KNN
		[Bhargava 1999]*	SC	N	Iz	MD	S	AG
	Tupla (Espaço)	[Ramamurti e Ghosh 1999]	A	N	Iz	AM	TI	NNet
		[Sommerfield 1997]	A	S	Iz	AM	TI	NB
	Amostra (de Treinamento)	[Ali e Pazzani 1996]	SC	N	Arb	AM	TI	FOIL
		[Domingos 1996]	SC	S	Arb	AM	TI	RISE
		[Rokach, Maimon e Arad 2005]	SC	S	Iz	AM	TI	KNN
Intermediário	Agregação	[Anand et al. 1995]	SM	N	Arb	AM	TI	NNet
		[Buntine 2000]	G	N	M	PLN	TI	GM
		[Blei, Ng e Jordan 2003]*	G	N	Iz	PLN	TI	LDA
	Função	[Michie 1995]	A	S	M	SE	TI	ILP
		[Zupan et al. 1999]	A	S	Iz	AM	TI	HINT

Tabela 3.1: *Tabela com informações sobre os métodos de decomposição de tarefas analisados e suas propriedades*

A seguir, descrevem-se as propriedades utilizadas na análise:

- **Estrutura de Decomposição (ED)**: Tipo de estrutura de dados utilizada para

representar a decomposição da tarefa. Valores possíveis incluem: Subconjuntos (SC), Árvore (A), Grafo (G) e Sub-modelos (SM).

- **Mutualidade Exclusiva (ME)**: Indica se os conceitos utilizados na decomposição são mutuamente exclusivos. Valores: Sim (S), Não (N).
- **Método de Aquisição da Estrutura (MAE)**: Modo de obtenção da estrutura de decomposição. Valores: Manual (M), Pré-definido (PD), Arbitrário (Arb), Induzido (Iz).
- **Subárea da Computação (SAC)**: Área de pesquisa em que o método foi originalmente proposto ou aplicado. Valores: Mineração de Dados (MD), Aprendizado de Máquina (AM), PLN (PLN), Sistemas Especialistas (SE).
- **Domínios de Aplicação (DA)**: Domínios práticos em que o método foi aplicado. Valores: Indústria (I), Tecnologia da Informação (TI), Saúde (S).
- **Técnicas de Aprendizado de Máquina (TAM)**: Técnicas utilizadas no método ou referenciadas como base, tais como: Rough Set (RS), k-Nearest Neighbor (KNN), Redes Neurais (NNet), Naive Bayes (NB), Algoritmos Genéticos (AG), FOIL, RISE, Modelos Gráficos (GM), Árvores de Classificação e Regressão (CART), Latent Dirichlet Allocation (LDA), Programação Lógica Indutiva (ILP) e HINT.

Os resultados apresentados na Tabela 3.1 revelam algumas tendências relevantes:

- Quando a *base de decomposição* consiste em atributos ou amostras de treinamento, predominam estruturas baseadas em subconjuntos (SC), com ou sem mutualidade exclusiva. Em contraste, métodos baseados em tuplas ou conceitos intermediários tendem a adotar estruturas hierárquicas mais elaboradas, como árvores (A), grafos (G) ou submodelos (SM).
- Para todas as bases de decomposição analisadas, observa-se a existência de pelo menos um método com estrutura adquirida por indução (Iz).
- A subárea de Mineração de Dados (MD) emprega majoritariamente métodos baseados em atributos para identificação de padrões de interesse, enquanto Aprendizado de Máquina (AM) cobre uma gama mais ampla de bases e técnicas. Por sua vez, o PLN (PLN) tem explorado fortemente a agregação de conceitos, em geral utilizando estruturas de grafo e métodos probabilísticos.
- A maioria dos métodos apresenta aplicações generalistas na área de Tecnologia da Informação (TI), embora também haja registros de uso específico em domínios como Indústria (I) e Saúde (S).

Embora os resultados apresentados não tenham pretensão de exaustividade, devido à limitação do escopo temporal (1995–2005) e à ausência de amostragem sistemática, eles oferecem uma visão abrangente das estratégias de decomposição disponíveis até

o início do século XXI. Como discutido nas próximas seções, os avanços em Aprendizado Profundo e Modelos de Linguagem de Grande Escala impulsionaram novas formas de decomposição, abordadas em detalhes na sequência.

3.3.4 Decomponibilidade em LLMs

As reflexões desenvolvidas nesta seção são fundamentadas pelo apêndice B.3, que apresenta uma visão detalhada da arquitetura dos LLMs, servindo de base conceitual para a análise de suas limitações e do papel da decomposição de tarefas em seu aprimoramento.

A ascensão dos LLMs revolucionou o PLN, permitindo avanços significativos em geração e compreensão de texto. No entanto, esses sistemas ainda enfrentam desafios em tarefas que exigem raciocínio simbólico [Qian et al. 2022], desambiguação semântica [Moraes et al. 2024] e processamento contextual refinado. [Cambria et al. 2017] destacam a importância da decomposição de tarefas para lidar com essas limitações, organizando o PLN de forma mais estruturada.

A metáfora da *sentiment suitcase* ilustra essa abordagem, mostrando que a análise de sentimentos não é uma tarefa única, mas uma mala que contém 15 subtarefas essenciais, como reconhecimento de entidades e extração de aspectos [Rana e Cheah 2016], dentre outros. Embora LLMs possam processar essa "mala" como um todo, a falta de segmentação explícita pode comprometer a precisão e a interpretabilidade. Dessa forma, a decomposição de tarefas melhora a transparência dos modelos e potencializa o uso eficiente das capacidades emergentes das LLMs. Nesse sentido, [Cambria et al. 2023] considera a decomposição de tarefas como um dos sete pilares fundamentais para o futuro da inteligência artificial, essencial para a construção de sistemas mais robustos, interpretáveis e eficazes.

Nos últimos anos, diversos *surveys* identificaram as capacidades e limitações dos Modelos de LLMs. Entre os desafios apontados, a metodologia de decomposição de tarefas surge como uma abordagem promissora para mitigar problemas estruturais desses modelos, organizando a execução de atividades complexas em etapas mais bem definidas. A seguir, são destacadas quatro limitações dos LLMs, identificadas em *surveys* específicos sobre o tema, que podem ser mitigadas por essa metodologia:

- **Dificuldade de Generalização**: LLMs necessitam de *fine-tuning* para adaptação a novos domínios devido à sua limitada capacidade de transferência de conhecimento [Hadi et al. 2023].

- **Dificuldade em Raciocínio Lógico**: Modelos de linguagem apresentam limitações ao realizar inferências complexas, pois tendem a basear suas respostas em correlações estatísticas [Asher et al. 2023].

- **Alucinação de Informação**: LLMs frequentemente geram respostas imprecisas ou infundadas, pois operam sem mecanismos de verificação da veracidade [Qian et al. 2022].

- **Explicabilidade Limitada**: A interpretabilidade dos LLMs é reduzida, dificultando a auditoria de suas decisões, especialmente em aplicações críticas [Kumar 2024].

Ao longo desta seção, será realizada uma análise das técnicas voltadas à mitigação dessas limitações. Por fim, essas restrições serão revisitadas, e uma síntese geral será apresentada, destacando como as técnicas da metodologia de decomposição de tarefas podem contribuir para sua mitigação.

Um dos principais desafios dos LLMs é a *capacidade de generalização*. Segundo [Hupkes et al. 2023], essa habilidade refere-se à transferência de representações, conhecimentos e estratégias para novas situações, permitindo que os modelos operem de forma robusta e confiável em dados diferentes dos usados no treinamento. Um aspecto fundamental desse processo é a *generalização composicional*, que serve de base para a *aprendizagem composicional* [Sinha, Premsri e Kordjamshidi 2024] . Essa habilidade possibilita que os modelos recombinem elementos previamente aprendidos para interpretar ou gerar novas estruturas linguísticas, sendo essencial para capturar a flexibilidade e a criatividade da linguagem humana.

A *aprendizagem composicional* pode ser caracterizada em diferentes facetas, com destaque para *sistematicidade* (*systematicity*) e *produtividade* (*productivity*) [Sinha, Premsri e Kordjamshidi 2024]. A primeira refere-se à capacidade dos modelos de recombinar elementos conhecidos em novas configurações, permitindo a interpretação e geração de expressões inéditas. Já a *produtividade* diz respeito à generalização para estruturas de maior complexidade ou comprimento do que as vistas no treinamento, garantindo que os modelos consigam lidar com *inputs* mais extensos sem perda de coerência.

Para avaliar a capacidade de generalização dos LLMs, [Dziri et al. 2023] testou seu desempenho em *tarefas composicionais* (*compositional tasks*), ou seja, tarefas que exigem a decomposição de problemas em subetapas e a síntese dessas etapas em uma resposta precisa. Para modelar essas tarefas, o autor as representou como *grafos computacionais* (*computational graphs*), isto é, grafos dirigidos acíclicos onde cada nó representa valores numéricos e as arestas correspondem a funções matemáticas, todas convergindo para um nó final que contém o resultado. Com essa representação, cada operação aritmética gera um grafo estático que modela a execução do algoritmo. Por exemplo, a operação 7×49 resulta no valor 343, sendo representada por múltiplos caminhos computacionais, onde o mais longo possui seis etapas, servindo assim como medida de profundidade do raciocínio.

Para mitigar os desafios impostos pela profundidade do raciocínio em tarefas multi-etapas (*multihop*), pode-se empregar a técnica de *scratchpad* [Nye et al. 2021], que

consiste em instruir ou treinar um modelo para registrar explicitamente suas etapas intermediárias, funcionando como um bloco de rascunho auxiliar, semelhante à estratégia usada por seres humanos ao resolver problemas matemáticos complexos. *Nossa proposta é simples: permitir que o modelo produza uma sequência arbitrária de tokens intermediários, que chamamos de scratchpad, antes de produzir a resposta final. Por exemplo, em problemas de adição, o scratchpad contém os resultados intermediários de um algoritmo de adição longo padrão. Para treinar o modelo, codificamos as etapas intermediárias do algoritmo como texto e usamos o treinamento supervisionado padrão* [Nye et al. 2021]. Veja a Figura 3.5 sobre como o scratchpad é usado.

Além da tarefa aritmética, [Dziri et al. 2023] testaram outros dois problemas: o *Einstein's Puzzle*, um quebra-cabeça lógico que exige o preenchimento de uma tabela com base em uma lista de restrições; e um problema de programação dinâmica, que consiste em identificar a subsequência de números que maximiza uma determinada função, sob a restrição de não adjacência. Embora o uso do *scratchpad* tenha melhorado o desempenho, os experimentos evidenciaram que os *transformers* não aprendem a estruturar o raciocínio de maneira composicional. Em vez disso, tendem a reconhecer padrões locais e a estabelecer correspondências entre subgrafos previamente observados durante o treinamento. Além desses experimentos conduzidos por [Dziri et al. 2023], [Qian et al. 2022] e [Nogueira, Jiang e Lin 2021] também obtiveram resultados semelhantes em estudos similares.

Sem Scratchpad

Input:
```
2 9 + 5 7
```

Target:
```
8 6
```

Com Scratchpad

Input:
```
2 9 + 5 7
```

Target:
```
<scratch>
2 9 + 5 7 , C: 0
2 + 5 , 6 C: 1 # added 9 + 7 = 6, carry 1
, 8 6 C: 0 # added 2 + 5 + 1 = 8, carry 0
</scratch>
0 8 6
```

Output:
```
8 6
```

Figura 3.5: *Exemplo de duas instâncias para treinamento de soma de inteiros: a primeira sem scratchpad e a segunda com scratchpad. Os comentários (marcados com #) são adicionados para fins de clareza e não fazem parte do target.*

A técnica de *scratchpad* foi uma das técnicas pioneiras de *prompting* que deu origem a diversas outras, gerando uma família de técnicas semelhantes denominada Cadeia de Raciocínio (*Chain-of-Thought*) [Wei et al. 2023]..

Essa técnica consiste na geração de uma sequência de passos intermediários de raciocínio, permitindo que modelos de linguagem realizem tarefas complexas de forma mais estruturada. O principal benefício da Cadeia de Raciocínio é a capacidade de decompor problemas multi-etapas em subetapas mais manejáveis, permitindo um melhor aproveitamento dos recursos computacionais em tarefas que demandam maior esforço cognitivo.

Além disso, a estruturação do raciocínio fornece uma janela interpretável sobre o comportamento do modelo, facilitando a identificação de erros no processo de inferência. Essa abordagem demonstrou eficácia em uma ampla gama de tarefas, incluindo problemas matemáticos, raciocínio baseado em senso comum e manipulação simbólica. Seu funcionamento pode ser potencializado pelo *few-shot prompting*, uma técnica na qual o modelo é apresentado a um pequeno número de exemplos de entrada e saída antes de gerar sua própria resposta. Diferente do *zero-shot prompting*, onde não há exemplos fornecidos, e do *fine-tuning*, que exige ajustes nos pesos do modelo, o *few-shot prompting* permite que modelos de grande porte adquiram rapidamente padrões de raciocínio a partir de poucos exemplos, tornando-se uma estratégia eficaz para elicitar cadeias de raciocínio sem a necessidade de re-treinamento extenso.

Além da técnica clássica de Cadeia de Raciocínio, diversas variantes foram desenvolvidas. Vários *surveys* foram produzidos com o objetivo de catalogar técnicas utilizadas, principalmente, para aprimorar o raciocínio passo a passo em modelos de linguagem. A seguir, descrevemos algumas dessas abordagens:

- **Least-to-Most (LtM) Prompting** - Método que começa resolvendo subtarefas simples antes de lidar com problemas mais complexos, permitindo uma abordagem incremental e hierárquica [Xia et al. 2024].
- **Chain-of-Code (CoCode)** - Estratégia que transforma problemas em componentes de código, aproveitando a estrutura programática para melhorar a coerência e exatidão dos raciocínios gerados [Xia et al. 2024].
- **Chain-of-Logic** - Técnica que aplica regras lógicas formais para decompor problemas complexos em proposições mais simples, garantindo inferências estruturadas e verificáveis [Xia et al. 2024].
- **Chain-of-Event** - Método utilizado para dividir tarefas, como sumarização multi-documento, em eventos discretos, aprimorando a contextualização e a precisão do conteúdo gerado [Xia et al. 2024].
- **Question Decomposition** - Estratégia que decompõe perguntas complexas em sub-perguntas mais gerenciáveis, promovendo uma abordagem modular e incremental

para a inferência [Chu et al. 2024].

- **Selection-Inference** - Método que identifica subproblemas relevantes dentro de uma questão maior, facilitando a inferência passo a passo e aumentando a precisão da resposta [Yu et al. 2023].

Para mitigar o problema das alucinações geradas por LLMs, uma técnica se destacou como bem-sucedida, criada por [Lewis et al. 2021]: a Geração Aumentada por Recuperação (Retrieval Augmented Generation, Retrieval-Augmented Generation (RAG)). Essa abordagem aprimora a geração de texto ao integrar um mecanismo de recuperação de informações externas, garantindo respostas mais precisas e reduzindo a propagação de informações factualmente incorretas. Segundo [Hu e Lu 2024], RAG é definido da seguinte forma:

> *"A Geração Aumentada por Recuperação (RAG) melhora de forma eficiente o desempenho dos modelos de linguagem generativa ao integrar técnicas de recuperação de informações. Ela resolve um desafio crítico enfrentado por modelos generativos autônomos: a tendência de produzir respostas que, embora plausíveis, podem não estar fundamentadas em fatos. Ao recuperar informações relevantes de fontes externas, o RAG reduz significativamente a incidência de alucinações ou respostas factualmente incorretas, aumentando assim a confiabilidade e a riqueza do conteúdo gerado."*

Apesar de seu potencial, RAG ainda enfrenta desafios significativos. Como apontado por [Gao et al. 2024], [Yu et al. 2024] e [Zhao et al. 2024], a precisão na recuperação de informações continua sendo um obstáculo, pois documentos irrelevantes ou desatualizados podem comprometer a qualidade das respostas. Além disso, a fusão e contextualização das fontes recuperadas podem gerar inconsistências ou respostas contraditórias. Questões de eficiência computacional também emergem, uma vez que o processo de recuperação adiciona custos de tempo e processamento. Por fim, [Yu et al. 2024] destacam a dificuldade em avaliar RAG de forma sistemática, dado que métricas tradicionais nem sempre capturam adequadamente a relação entre recuperação e geração. Assim, pesquisas atuais buscam aprimorar mecanismos de reranking, recuperação adaptativa e modelos híbridos de avaliação, visando maximizar a utilidade do RAG em aplicações práticas.

Além do RAG, que integrou técnicas de Recuperação de Informações e Geração de Conteúdo por LLM, surgiram propostas inovadoras para integrar diversas áreas da computação com LLMs, com destaque para três áreas:

- **LLM-Based Algorithms**: Segundo Chen et al. [Chen et al. 2024], um *LLM-Based Algorithm* é definido como um algoritmo que contém uma ou mais chamadas para

modelos de linguagem como componentes essenciais, sendo projetado para depender criticamente das capacidades dos LLMs. Essa abordagem combina a execução de algoritmos tradicionais com chamadas estratégicas a LLMs, estruturadas em grafos computacionais que facilitam a decomposição de tarefas.

- **Síntese de programas**: LLMs demonstraram capacidade crescente na geração automática de código, impulsionando técnicas de programação assistida [Chen, Tworek e Jun Heewoo 2021], síntese indutiva de programas [Ellis, Pu e Solar-Lezama Armando 2021] e até mesmo descoberta automática de novas abordagens algorítmicas [Austin et al. 2021]. Modelos como Codex [Chen, Tworek e Jun Heewoo 2021] e AlphaCode [Li, Wang e Wu Jian 2022] ilustram o potencial dos LLMs na escrita, depuração e explicação de código-fonte.
- **LLM-Based Agents**: Segundo Huang et al. [Huang et al. 2024], os agentes baseados em LLMs emergiram como uma abordagem promissora para planejamento autônomo, combinando percepção do ambiente, raciocínio e tomada de decisões em múltiplas etapas. O planejamento desses agentes pode ser categorizado em cinco abordagens principais: decomposição de tarefas, seleção de múltiplos planos, uso de módulos externos, reflexão e refinamento, e planejamento baseado em memória.

A decomposição de tarefas emerge como uma estratégia fundamental para mitigar as limitações das LLMs, especialmente em desafios que exigem raciocínio simbólico, desambiguação semântica e processamento contextual refinado. Ao dividir tarefas complexas em subtarefas estruturadas, essa abordagem reduz a sobrecarga cognitiva do modelo, tornando o processamento mais interpretável e preciso. Um exemplo disso é a metáfora da *sentiment suitcase*, que demonstra como a análise de sentimentos pode ser desmembrada em múltiplas subtarefas, como reconhecimento de entidades e extração de aspectos, permitindo um tratamento mais granular e eficaz. Dessa forma, a segmentação explícita viabiliza um melhor controle sobre a qualidade da saída gerada pelos modelos, tornando-os mais confiáveis e alinhados às necessidades específicas de cada aplicação em PLN.

CAPÍTULO 4

Decomposição de Tarefas em Problemas de Linguagem Natural

Neste capítulo, é apresentada a metodologia de decomposição de tarefas em Problemas de Linguagem Natural.

Segue um resumo de cada uma das seções a seguir:

4.1 **Representações Linguísticas**: Apresenta as principais formas de representação das línguas, com ênfase na gramática generativista e analisando duas subáreas da linguística: a sintaxe e a semântica;

4.2 **Teoria Baseada no Uso**: Apresenta a Teoria Baseada em Uso e Gramática de Construção, que juntos fornecem bases teóricas para compreensão de padrões na linguística.

4.4 **Decomposição de Tarefas por Padrões**: Propõe uma metodologia de decomposição de tarefas no PLN baseada em padrões linguísticos.

4.5 **Padrão Arquitetural Recrutador-Selecionador**: Propõe um padrão de projeto que segue a metodologia de decomposição de tarefas.

4.6 **Decomposição de Tarefas na Anotação de Corpus**: Propõe um pseudo-código aplicável em decomposição de tarefas de anotação de *corpus*.

4.1 Representações Linguísticas

Na seção 3.3.1, foi discutida a importância da representação de sistemas complexos para que sua decomposição seja viabilizada. Nesta seção, será apresentada uma visão geral sobre o conceito de representações linguísticas, ou seja, sobre como uma língua pode ser representada de forma simbólica ou estrutural.

A linguística é uma área científica ampla e diversificada, na qual diferentes níveis de análise são abrangidos, conforme ilustrado na Figura 4.1, que organiza suas principais subáreas. Nesta visão geral, será dado foco às subáreas de sintaxe e semântica, consideradas fundamentais para que a estrutura e o significado de uma linguagem sejam representados.

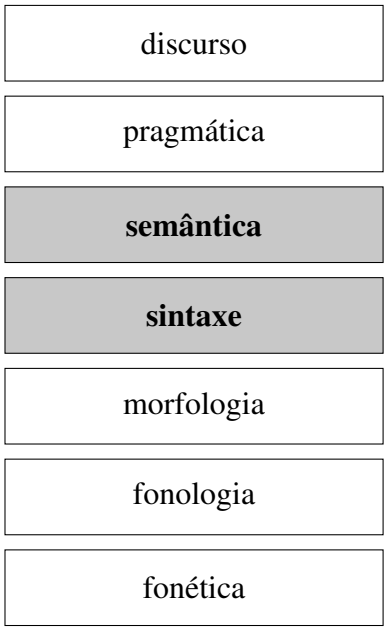


Figura 4.1: *Subáreas da linguística, adaptado de [Caseli e Nunes 2024, p.13]*

No campo da linguística, o estudo das estruturas linguísticas está frequentemente associado à aquisição da linguagem, sendo marcado por duas abordagens teóricas distintas. A primeira é o **generativismo**, desenvolvido por Chomsky em 1957, que se baseia em dois conceitos fundamentais: a **Gramática Gerativa** [Chomsky 2002] e a **Gramática Universal** [Chomsky 1965]. Essa teoria propõe formalismos matemáticos para modelar a sintaxe da linguagem natural.

A segunda abordagem, inicialmente conhecida como **Linguística Cognitiva** e posteriormente consolidada como **Teoria Baseada no Uso**, enfatiza a dimensão simbólica da linguagem, argumentando que a gramática emerge do uso de símbolos linguísticos significativos e das experiências cognitivas humanas.

Nesta seção, apresentaremos a teoria gerativa e suas principais contribuições para o estudo da linguagem. Na seção seguinte, discutiremos como a *Teoria Baseada no Uso* auxilia na compreensão dos padrões linguísticos.

A *gramática gerativa* contribuiu significativamente aos formalismos matemáticos ao modelar a linguagem como um sistema de regras formais que operam sobre cadeias de símbolos, permitindo uma descrição precisa e abstrata da estrutura linguística. Uma de suas maiores contribuições foi a **Hierarquia de Chomsky**, que classifica gramáticas em quatro níveis (regulares, livres de contexto, sensíveis ao contexto e recursivamente enumeráveis), cada um com diferentes graus de complexidade. Essa hierarquia influenciou profundamente a ciência da computação, sendo usada na definição de linguagens formais, na construção de compiladores e no desenvolvimento de expressões regulares e autômatos. Assim, a gramática gerativa conectou os estudos de linguagem natural com os

fundamentos teóricos da computação. A Figura 4.2 ilustra um exemplo de como regras gramaticais podem ser usadas para gerar a frase "*the man hit the ball*" em uma árvore sintaticamente correta.

1. Sentence → NP + VP
2. NP → T + N
3. VP → Verb + NP
4. T → *the*
5. N → *man, ball*, etc.
6. Verb → *hit, took*, etc.

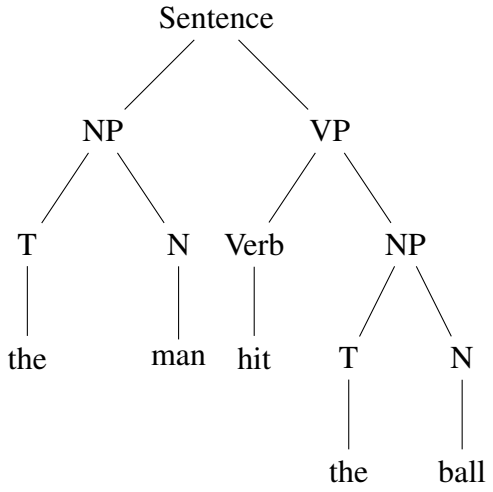


Figura 4.2: *Exemplo de regras gramaticais e da árvore sintática correspondente para uma frase gerada com base nessas regras, apresentadas lado a lado. [Chomsky 2002, p.26]*

A *Gramática Universal* é outro conceito central na teoria de aquisição de linguagem de Chomsky. Ele argumenta que os seres humanos nascem com um conjunto inato de princípios linguísticos, pois as experiências obtidas por meio da exposição a estímulos, como as falas (*utterances*) direcionadas a recém-nascidos, seriam insuficientes para explicar a capacidade de adquirir linguagem. [Smith 1999, p. 45] descreve esse conceito da seguinte forma (tradução nossa):

> *Chomsky é famoso por argumentar que a aquisição do primeiro idioma é um exemplo da "pobreza do estímulo", em que acabamos sabendo mais do que está presente nos enunciados aos quais somos expostos. Há uma "enorme lacuna entre os dados disponíveis e o estado alcançado, uma característica de todo crescimento e desenvolvimento"... No domínio da linguagem, isso pode ser ilustrado pela convergência de intuições sobre frases que os falantes nunca encontraram antes. Nós nos baseamos nessa convergência na discussão sobre "John speaks fluently English", presumindo, esperamos que corretamente, que você concordaria com nosso julgamento. Esses exemplos levantam várias questões...a sequência de "Pronome Verbo Advérbio Substantivo" é inferivelmente não inglesa porque não ocorre...Exemplos como esses nunca são ensinados em sala de aula e são muito raros em textos ou conversas normais. Se isso for verdade..., então ficamos com o problema de explicar como chegamos às intuições que temos. A resposta de Chomsky é*

> *que temos esse conhecimento como uma função conjunta de ter adquirido os itens lexicais do inglês e de tê-los incorporado em uma estrutura fornecida pela Gramática Universal (UG), o conjunto de princípios linguísticos que recebemos ao nascer em virtude de sermos humanos... Uma olhada em qualquer livro didático mostra que meio século de pesquisa em sintaxe gerativa revelou inúmeros exemplos desse tipo e, ao mesmo tempo, removeu a explicação alternativa plausível de que essas coisas nos são ensinadas. Se estiver correto, isso demonstra que devemos atribuir grande parte do conhecimento que temos ao estado inicial, à UG, e não ao efeito do input linguístico ao qual estamos diretamente expostos. Em resumo, ele é inato.*

As teorias de Chomsky, além de contribuírem para o formalismo por meio de regras e árvores sintáticas, serviram de base para grandes projetos de anotação. Um dos mais influentes é o **Penn Treebank (PTB)** [Marcus, Marcinkiewicz e Santorini 1993], que, em sua primeira versão, não apenas anotou cada **sintagma** (*phrase*) da árvore sintática com rótulos estruturais, mas também adicionou **anotações morfossintáticas** (*Part-of-Speech tags*) a 4,5 milhões de palavras do inglês americano. Esse processo de anotação atribui rótulos gramaticais a cada palavra (folha da árvore), indicando sua categoria sintática.

A tarefa de *POS-tagging* consiste em rotular sentenças com classes gramaticais, como substantivo (`NN`), adjetivo (`JJ`), preposição (`PP`) e verbo (`VB`), entre outras. O PTB utiliza uma notação parentética (*bracketed notation*) para representar sua estrutura sintática. Por exemplo, a sentença “This has some logic.” é anotada como:

```
(S (NP-SBJ (DT This)) (VP (VBZ has) (NP (DT some) (NN logic))) (. .))
```

Essa notação equivale à árvore sintática ilustrada na Figura 4.3.

A representação de sentenças no *PTB* é obtida por meio da **análise de constituintes** e, portanto, constrói uma árvore de constituência [Caseli e Nunes 2024, p. 129]. Outro tipo de análise sintática é a **análise de dependências**, baseada na gramática de dependência [Caseli e Nunes 2024, p. 133].

Além do *PTB*, outro conjunto notável de *treebanks* é o **Universal Dependencies (UD)** [Marneffe et al. 2021]. Enquanto o **Penn Treebank (PTB)** adota uma abordagem baseada em **sintagmas** (*phrases*), agrupando palavras hierarquicamente em estruturas como **sintagmas nominais** (`NP`) e **sintagmas verbais** (`VP`), o **UD** emprega um modelo fundamentado em **dependências gramaticais**, descrevendo relações diretas entre palavras, conforme ilustrado na Figura 4.4.

Além das **relações de dependência**, o UD também anota as categorias gramaticais (*POS tags*), porém em um conjunto reduzido de **apenas 17 categorias universais**

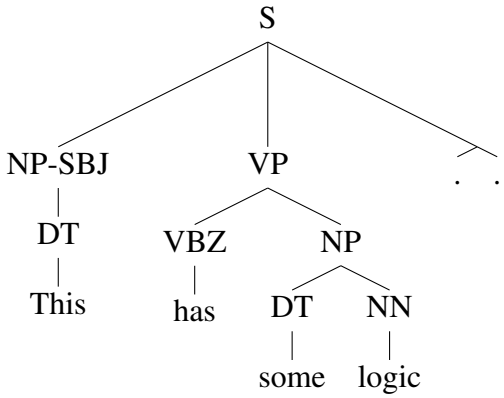


Figura 4.3: *Diagrama de árvore correspondente à anotação do Penn Treebank. As tags* `NP-SBJ`, `VP` *e* `NP` *representam categorias sintáticas (nós intermediários), enquanto as tags* `DT`, `VBZ` *e* `NN` *indicam classes gramaticais das palavras.*

(UPOS), priorizando simplicidade, consistência entre línguas e compatibilidade com diferentes sistemas de anotação. Para capturar especificidades linguísticas, o UD introduz **features** (*atributos ou propriedades*) opcionais, que complementam os rótulos UPOS. Por exemplo, na Figura 4.4, o nó "gato"receberia o rótulo `NOUN` e as *features* `Gender=Masc` e `Number=Sing`, indicando que se trata de um **substantivo masculino singular**.

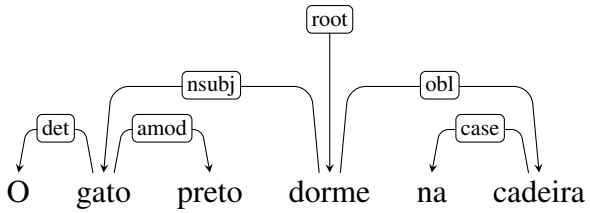


Figura 4.4: *Árvore de dependências da frase "O gato preto dorme na cadeira". O nó raiz geralmente é o verbo da sentença. Cada palavra é representada por um nó que possui relações de dependências entre outros nós.*

Outro aspecto relevante das árvores sintáticas é que elas representam a estrutura formal das sentenças, mas não o seu conteúdo. Em outras palavras, capturam a sintaxe, mas não a semântica. Por exemplo, a frase *"Colorless green ideas sleep furiously."* [Chomsky 2002, p. 15], ilustrada na Figura 4.5, é sintaticamente bem formada, mas semanticamente ininteligível.

Segundo [Caseli e Nunes 2024, p. 165], a semântica estuda o significado de palavras e frases. No entanto, atribuir um sentido claro e preciso a uma palavra não é uma tarefa trivial. Por exemplo, o verbo *tomar* pode assumir diferentes significados dependendo do contexto em que é empregado, como em "tomar um susto", "tomar ciência" e "tomar cuidado".

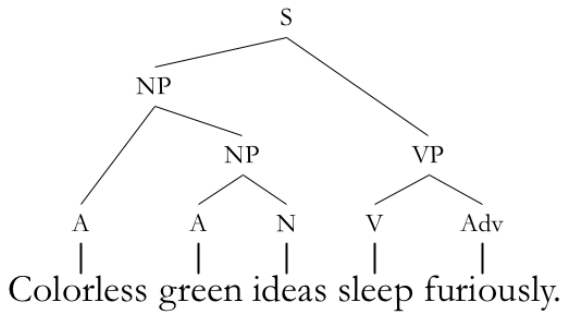


Figura 4.5: *Exemplo de árvore sintática correta, mas sem sentido semântico*

Na linguística, [Caseli e Nunes 2024, p. 166] afirma que há duas perspectivas concorrentes: **perspectiva representacional ou essencialista**, que considera que significado e palavra são entidades distintas, sendo que a primeira fornece forma para "hospedar"o significado; e a **perspectiva pragmática radical**, que defende que o significado é provisório e instável, decorrente de situações concretas e dependerá do uso, do contexto, do tempo, do espaço, de quem fala.

Na esteira da *perspectiva representacional*, há diversas formas de representações semânticas. Segundo [Caseli e Nunes 2024, p. 170 e p. 190], elas podem ser categorizadas em duas principais abordagens: **semântica com técnicas simbólicas**, que envolve o uso de regras e representações formais explícitas para processar e compreender textos em linguagem natural; e a **semântica distribucional**, que considera que palavras com um contexto linguístico semelhante tendem a ter significados similares ou aproximados.

Na abordagem *semântica com técnicas simbólicas*, há uma arquitetura típica para o **Entendimento da Linguagem Natural** (*Natural Language Understanding – NLU*) [Caseli e Nunes 2024, p. 170], composta por dois principais componentes. O primeiro componente é a **Base de Conhecimento**, que consiste em um repositório centralizado e processável por máquina que armazena informações, regras e procedimentos para capturar e representar conhecimento geral ou de um domínio específico. Duas formas de representação estruturada do conhecimento são utilizadas: redes (e.g., OWL, RDF, *WordNet*) e frames (e.g., *FrameNet*). E o segundo componente é o **Sistema Lógico** que é responsável pelo raciocínio formal, composto por uma **representação lógica** e um **motor de inferência**. O significado é representado de forma abstrata por meio da matemática e da lógica formal. Exemplos incluem Lógica Descritiva, Lógica de Primeira Ordem, Programação em Lógica (PROLOG) e Lógicas Intensionais. Com esses dois componentes, uma arquitetura de *NLU* pode realizar inferências e responder a perguntas com base no raciocínio lógico e no conhecimento armazenado.

4.2 Teoria Baseada no Uso

Outra teoria que ganhou destaque na década de 1980 foi a Linguística Cognitiva, mais especificamente a ***Teoria Baseada no Uso*** (***usage-based theory, UBT***). Essa teoria contrapõe as ideias de Chomsky e defende que a linguagem é adquirida através das experiências de seu uso. De acordo com [Tomasello 2003, p. 5], essa teoria pode ser definida da seguinte forma (tradução nossa):

> *"As teorias baseadas no uso sustentam que a essência da linguagem é sua dimensão simbólica, sendo a gramática um derivado. A capacidade de se comunicar com espécies simbolicamente (convencionalmente, intersubjetivamente) é uma adaptação biológica específica da espécie. Mas, ao contrário da gramática gerativa e de outras abordagens formais, nas abordagens baseadas no uso, a dimensão gramatical da linguagem é um produto de um conjunto de processos históricos e ontogenéticos chamados coletivamente de gramaticalização. Quando os seres humanos usam símbolos para se comunicarem uns com os outros, encadeando-os em sequências, surgem padrões de uso que se consolidam em construções gramaticais... Em vez de conceber as regras linguísticas como procedimentos algébricos para combinar palavras e morfemas que não contribuem para o significado, essa abordagem concebe as construções linguísticas como símbolos linguísticos significativos, uma vez que nada mais são do que os padrões nos quais os símbolos linguísticos significativos são usados na comunicação"*

Seguindo a esteira da *teoria baseada no uso*, há um conceito fundamental que são as **construções**, que propõem modelos linguísticos nos quais a forma e o significado estão integrados, ou seja, não há uma distinção rígida entre léxico e gramática. [Lacerda 2017] assim sumariza a **Gramática de Construções** e sua abordagem baseada em construções:

> *"A Gramática de Construções busca a(s) motivação(ões) para cada construção estudada. A motivação pode ser encontrada em aspectos da aquisição da língua, princípios de gramaticalização, demandas discursivas, princípios icônicos ou princípios gerais de categorização...Há diferentes modelos linguísticos que seguem a abordagem construcional. Eles se unem em torno dos seguintes princípios gerais, compartilhados por todos: a unidade básica da gramática é a construção; a estrutura semântica é projetada diretamente na estrutura sintática; a língua, como outros sistemas cognitivos, é uma rede de nós e elos entre os nós; as associações entre esses nós são representadas na forma de hierarquias de herança; a estrutura da língua é moldada pelo uso"*

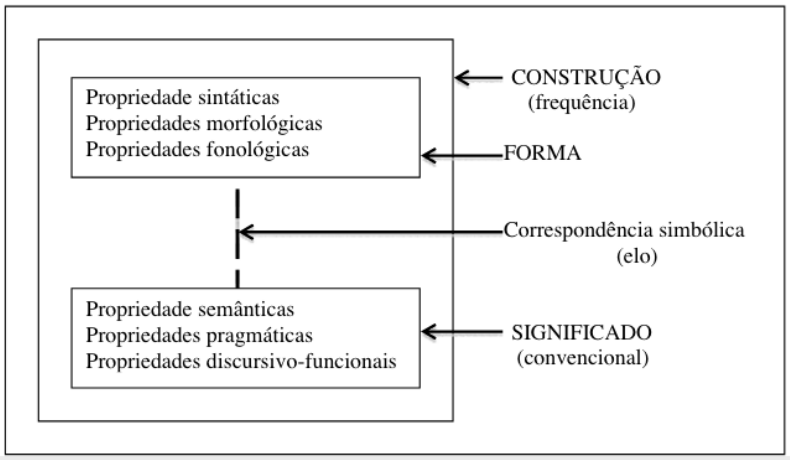


Figura 4.6: *Estrutura simbólica de uma construção proposto por [Croft 2001, p. 18], adaptado por [Lacerda 2017].*

[Croft 2001] propôs um modelo de construção, conforme ilustrado na Fig. 4.6. Segundo o autor, as *construções* são, essencialmente, unidades simbólicas que estabelecem um pareamento entre forma e significado. Ele define o termo *significado* como a representação de todos os aspectos convencionalizados da função de uma construção, os quais podem incluir não apenas as propriedades da situação descrita pelo enunciado, mas também características do discurso em que o enunciado está inserido e da situação pragmática dos interlocutores. Por exemplo, uma construção como *"Que gato lindo!"* pode ser utilizada para expressar a surpresa do locutor.

Um exemplo clássico de *construção*, no qual forma e significado estão interligados, é discutido por [Kay e Fillmore 1999]: a construção *"What's X doing Y?"* (WXDY). Esse *template* pode ser utilizado para expressar surpresa ou reprovação diante de uma situação inesperada. Exemplos incluem:

- *"What's this fly doing in my soup?"* (expressando surpresa sobre a presença da mosca na sopa);
- *"What's that dog doing on the table?"* (reprovando a presença do cachorro em um local inadequado).

Em uma análise da *Teoria Baseada no Uso*, [Ibbotson 2013] aponta que, embora a combinação de palavras em uma sentença possa gerar milhões de possibilidades, a ocorrência real dessas combinações segue uma distribuição desigual, com algumas formas sendo muito mais frequentes do que outras. Por exemplo, se a frase *"I like your green cheese"* permitisse a substituição de cada uma de suas cinco palavras por 20 alternativas, haveria 20^5 (3.200.000) possíveis frases de cinco palavras. Essa redundância na linguagem facilita a recuperação do significado mesmo na presença de ruído na comunicação, conforme demonstrado por [Shannon 1951]. Por exemplo, com um pouco de esforço, a frase "xvxn whxn thx sxgnxl xs nxxsy" pode ser compreendida, e esse

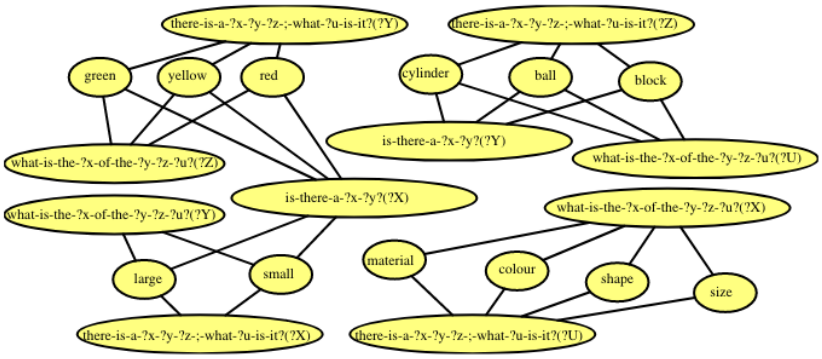


Figura 4.7: *Rede categórica de construções obtida semi-automaticamente*

princípio se aplica tanto a letras e palavras individuais quanto a sequências maiores de palavras.

Ibbotson também argumenta que a **composicionalidade da linguagem** influencia a aprendizagem, pois grande parte do conhecimento linguístico é baseada em exemplos específicos usados na comunicação cotidiana. Estudos sobre a fala dirigida à criança mostram que um pequeno conjunto de moldes (*construções*), como *Where's the X?* (*Where's the ball?*), *I wanna X* (*I wanna play*), *It's a X* (*It's a cat*), e *Put X here* (*Put the toy here*), representa uma parcela significativa da linguagem adquirida. Essas estruturas previsíveis auxiliam as crianças na identificação e generalização de padrões linguísticos, permitindo que construam novas sentenças a partir de esquemas já conhecidos, um processo amplamente apoiado por pesquisas sobre aquisição da linguagem.

Gramáticas de construção são produzidas a partir de observações de uso. Existem diversas formas de categorização e organização de construções, sendo uma delas a representação por meio de **redes categóricas**. Por exemplo, [Doumen, Beuls e Eecke 2024] construiu, de forma semiautomática, uma *rede categórica* a partir de um conjunto de dados composto por perguntas e respostas sobre um cenário com objetos tridimensionais. A Fig. 4.7 ilustra um fragmento dessa representação emergente, derivada das perguntas observadas.

A Gramática de Construção, fundamentada na *Teoria Baseada no Uso*, oferece uma perspectiva dinâmica sobre a organização da linguagem, enfatizando a relação entre forma e significado em unidades simbólicas recorrentes. A evidência empírica demonstra que **padrões linguísticos** emergem a partir da experiência de uso e são organizados em redes complexas de construções, moldadas por fatores cognitivos, pragmáticos e discursivos. Com o avanço dos modelos de PLN, particularmente os LLMs, tornou-se possível investigar até que ponto essas construções são capturadas e representadas por arquiteturas baseadas em aprendizado profundo. A próxima seção explora essa questão, analisando as evidências de que LLMs internalizam *padrões linguísticos* e reproduzem

estruturas emergentes da linguagem natural em seus processos de geração e inferência.

4.3 Representações de Padrões Linguísticos em *LLMs*

A compreensão de como modelos de LLM representam e processam informações linguísticas tem sido alvo de diversas pesquisas recentes. Há evidências empíricas de que esses modelos constroem padrões linguísticos de forma hierárquica, refletindo diferentes níveis de estrutura linguística, desde a morfologia até a semântica e o raciocínio textual.

Estudos como os de [Hewitt e Manning 2019] demonstram que modelos como BERT capturam árvores de dependência sintática de forma latente. Esses modelos aprendem a agrupar palavras de acordo com suas funções sintáticas, sem supervisão explícita. Técnicas de projeção vetorial mostram que as relações entre palavras seguem distribuições espaciais que refletem estruturas gramaticais conhecidas.

Além disso, pesquisas indicam que as camadas inferiores dos modelos são especializadas em representações morfológicas, codificando aspectos como gênero, número e tempo verbal. Isso sugere que os modelos desenvolvem uma hierarquia de representações semelhante à encontrada em abordagens linguísticas formais.

A análise das representações internas dos modelos revela que palavras semanticamente relacionadas tendem a ocupar regiões similares no espaço latente dos embeddings. [Jawahar, Sagot e Seddah 2019] mostram que as camadas superiores de BERT e Generative Pre-trained Transformer (GPT) codificam relações semânticas complexas, como sinonímia e hiponímia, sem a necessidade de aprendizado supervisionado específico.

Pesquisas como [Clark et al. 2019] analisaram os pesos das cabeças de atenção em Transformers, revelando que algumas delas correspondem diretamente a relações sintáticas. Por exemplo, certas cabeças aprendem a identificar sujeitos e objetos de uma sentença, enquanto outras focam em identificar modificadores e adjuntos.

Essa distribuição de atenção sugere que os modelos internalizam padrões linguísticos emergentes, reforçando a ideia de que a estrutura da linguagem está embutida de forma distribuída ao longo das camadas do modelo.

Além disso, experimentos de *probing* demonstram que informações linguísticas podem ser extraídas de camadas específicas dos modelos, sugerindo que as redes neurais capturam não apenas padrões de coocorrência estatística, mas também regularidades estruturais subjacentes à linguagem natural.

As evidências apresentadas indicam que os LLMs desenvolvem representações linguísticas de forma emergente e estruturada, capturando desde características morfológicas até padrões sintáticos e semânticos. Embora essas representações não sejam idênticas às formações cognitivas humanas, elas demonstram que os modelos de linguagem

conseguem abstrair e organizar informações de maneira hierárquica, refletindo padrões fundamentais da linguagem natural.

4.4 Decomposição de Tarefas por Padrões

A **decomposição de tarefas por padrões em LLM** é uma abordagem estruturada que busca modularizar tarefas complexas ao identificar e reutilizar *padrões linguísticos recorrentes* para segmentar um problema em subtarefas mais gerenciáveis. Essa metodologia é fundamentada na observação de que fenômenos linguísticos seguem estruturas previsíveis, como *construções sintáticas*, *padrões discursivos* e *agrupamentos semânticos*, permitindo que a decomposição seja guiada por regularidades naturais da linguagem. Ao adotar essa estratégia, torna-se possível *melhorar a qualidade da anotação de corpus*, *aumentar a interpretabilidade dos processos de PLN* e *facilitar a reutilização de componentes* em diferentes aplicações, desde anotação automática até treinamento de modelos de aprendizado de máquina. Além disso, essa abordagem promove um fluxo mais eficiente de processamento, permitindo que subtarefas sejam tratadas de forma *incremental*, *hierárquica* ou *paralela*, reduzindo a complexidade computacional e garantindo maior consistência nos resultados. Dessa forma, a decomposição de tarefas por padrões representa uma metodologia híbrida entre *Teoria Baseada no Uso*, *Métodos de Indução* e *LLM-Based Algorithms*, tornando-se um método robusto para estruturar e otimizar soluções em PLN.

4.4.1 Diferenças entre *Datasets* Tabulares e *Corpus* de Texto

Na Seção 3.3, foi apresentada a taxonomia de métodos de decomposição proposta por [Rokach 2006]. Nesta seção, discute-se a diferença entre os tipos de dados utilizados: enquanto Rokach analisou principalmente métodos aplicados a dados tabulares, a decomposição proposta neste trabalho foca em *corpora* de texto, que apresentam diferenças estruturais significativas. Seguindo a taxonomia de Rokach, a metodologia de decomposição de tarefas em PLN adotada aqui enquadra-se na categoria de *decomposição de conceito intermediário*, sendo mais especificamente caracterizada como *decomposição por padrões linguísticos*, conforme ilustrado na Figura 4.8.

A escolha de abordagens para aprendizado de máquina varia significativamente conforme a estrutura dos dados, impactando diretamente os métodos de decomposição de tarefas. Em problemas envolvendo *datasets tabulares*, os dados são organizados em formato de tabela, onde cada linha representa uma instância e cada coluna corresponde a um atributo específico. Métodos de decomposição para esse tipo de dado foram explorados por [Rokach 2006] na Seção 3.3.3, na qual as relações entre atributos são explicitamente

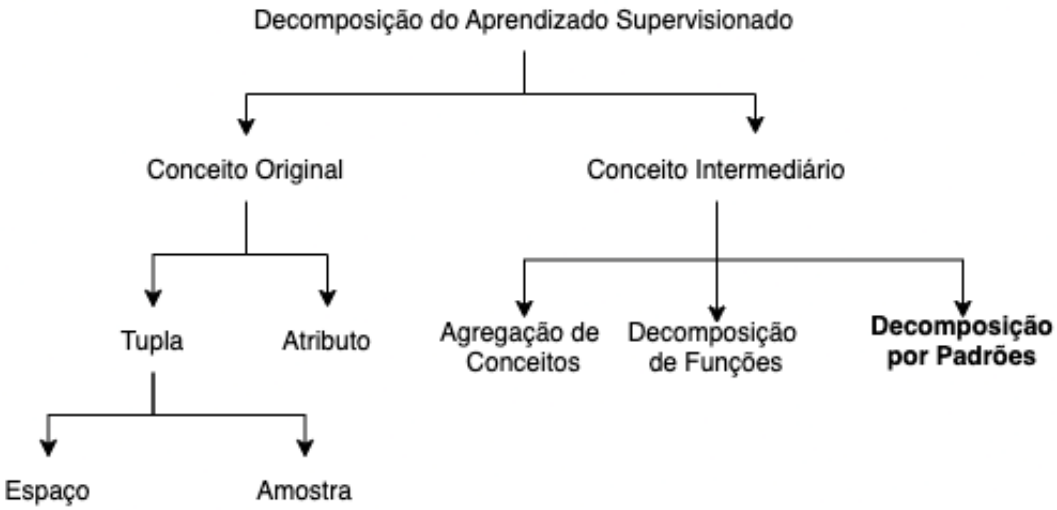


Figura 4.8: *Taxonomia de Métodos de Decomposição proposta por [Rokach 2006], com a inclusão da Decomposição por Padrões dentro do conceito intermediário.*

definidas, permitindo a aplicação de técnicas tradicionais, como *árvores de decisão* e *métodos de indução*, exemplificados pelos algoritmos *ID3* e *CART*. Já em *corpora* textuais, a estrutura é inerentemente hierárquica e sequencial, apresentando padrões latentes que exigem modelos mais sofisticados, como *Transformers* e LLMs. A compreensão dessas diferenças estruturais é essencial para o desenvolvimento de estratégias eficazes de decomposição de tarefas em PLN, garantindo que as características intrínsecas de cada tipo de dado sejam corretamente exploradas.

A Tabela 4.1 apresenta uma comparação entre tarefas de aprendizado de máquina em conjuntos de dados tabulares e *corpora* textuais, destacando suas diferenças fundamentais. Enquanto os conjuntos tabulares possuem estrutura fixa e relações bem definidas entre atributos, os *corpora* textuais apresentam padrões latentes e dependências semânticas entre tokens, exigindo abordagens mais avançadas, como modelos baseados em LLMs. A compreensão dessas diferenças é essencial para a definição de métodos adequados de decomposição de tarefas em PLN.

A distinção entre conjuntos de dados tabulares e *corpora* textuais evidencia a necessidade de estratégias específicas para a decomposição de tarefas em aprendizado de máquina. Enquanto os métodos tradicionais aplicados a dados tabulares se beneficiam de relações explícitas entre atributos e de técnicas consolidadas, como árvores de decisão e métodos de indução, os desafios inerentes ao processamento de texto exigem abordagens mais avançadas, capazes de capturar padrões latentes e dependências contextuais. Dessa forma, a decomposição de tarefas em PLN, fundamentada na *decomposição por padrões linguísticos*, emerge como uma solução estruturada e adaptável para lidar com as complexidades dos *corpora* textuais. Para formalizar essa abordagem, a próxima seção apresenta

Característica	Conjunto de Dados Tabular	*Corpus* de Textos
Formato dos Dados	Estruturado (tabelas com colunas e linhas bem definidas)	Sequencial (texto contínuo composto por tokens)
Representação de Entradas	Vetores numéricos ou nominais (*features* específicas por amostra)	Sequências de *tokens* ou *embeddings* de palavras
Relações Entre Atributos	Independentes ou correlacionados explicitamente	Latentes e contextuais entre *tokens*
Padrões a Serem Aprendidos	Regras explícitas, correlações diretas entre colunas	Relações sintáticas e semânticas emergentes
Modelos Comuns	Árvores de decisão, regressão logística, redes neurais tabulares	*LLMs*, *Transformers*, *RNNs*, *Word2Vec*
Pré-processamento	Normalização, imputação de valores faltantes	*Tokenização*, remoção de *stopwords*, *stemming/lemmatização*
Interpretação dos Resultados	Mais direta e interpretável	Dependente do contexto e mais difícil de explicar
Desafios	Lidar com valores ausentes e ruído nos dados	Capturar significado semântico e dependências contextuais
Estrutura Hierárquica	Organizado por colunas distintas	Hierárquico, envolvendo palavras, frases e parágrafos
Tipo de Aprendizado	Baseado em atributos independentes	Baseado em dependências sequenciais e contextuais

Tabela 4.1: *Comparação entre aprendizado de máquina em conjuntos de dados tabulares e corpus de textos*

a definição matemática da decomposição de tarefas, estabelecendo um modelo baseado em grafos direcionados acíclicos (DAGs) que permite estruturar e modularizar processos de PLN de forma rigorosa e sistemática.

4.4.2 Definição Matemática

A *decomposição de tarefas por padrões em PLN (PLN)* pode ser formalizada como um *grafo acíclico dirigido (Grafo Acíclico Dirigido (DAG) - Directed Acyclic Graph)*, onde cada nó representa uma subtarefa e as arestas definem a relação de dependência entre elas. Essa estrutura permite capturar a modularidade e a organização hierárquica da decomposição.

Seja uma *tarefa complexa* representada por uma função:

$$F : X \to Y$$

onde:

- X é o conjunto de entradas (por exemplo, textos brutos, frases ou tokens),
- Y é o conjunto de saídas esperadas (por exemplo, anotações, rótulos ou textos processados),

- F define a transformação da entrada X para a saída Y.

A *decomposição de tarefas* consiste em dividir F em um conjunto de *subtarefas* $T = \{t_1, t_2, ..., t_n\}$, organizadas como um *grafo acíclico dirigido*:

$$G = (V, E)$$

onde:

- $V = T$ é o conjunto de nós do grafo, onde cada nó t_i representa uma subtarefa,
- $E \subseteq T \times T$ é o conjunto de arestas, onde uma aresta (t_i, t_j) indica que a saída de t_i é entrada para t_j (isto é, t_i precede t_j).

Cada subtarefa t_i é uma função parcial definida como:

$$t_i : X_i \to Y_i$$

onde $X_i \subseteq X$ e $Y_i \subseteq Y$ podem ser subconjuntos específicos da entrada e da saída.

A relação de dependência entre as subtarefas é dada por um *operador de composição* C, que combina os resultados das subtarefas para reconstruir a função original F:

$$F(x) = C(t_1(x_1), t_2(x_2), ..., t_n(x_n))$$

onde:

- x_i representa a entrada processada por cada subtarefa t_i,
- C pode representar *composição sequencial*, *paralela* ou *hierárquica*, dependendo da estrutura do grafo G.

A decomposição pode assumir diferentes formas, dependendo da estrutura do grafo G:

1. **Decomposição Sequencial (*Pipeline*)**
   Se G é um caminho linear $t_1 \to t_2 \to ... \to t_n$, então cada subtarefa recebe a saída da anterior, formando uma cadeia:

   $$F(x) = (t_n \circ t_{n-1} \circ \cdots \circ t_1)(x)$$

2. **Decomposição Paralela**
   Se G contém múltiplos nós de entrada sem dependências entre si, temos tarefas paralelas cujos resultados são combinados por um operador de agregação C:

   $$F(x) = C(t_1(x), t_2(x), ..., t_n(x))$$

3. **Decomposição Hierárquica**
   Se G é uma árvore, cada nó t_i pode ser recursivamente decomposto em subtarefas menores, criando uma estrutura multinível.

4.4.3 Quase-decomponibilidade e a Decomposição de Tarefas

A organização da linguagem natural pode ser analisada sob a perspectiva da *quase-decomponibilidade*, um conceito discutido na Seção 3.3.1, que descreve sistemas complexos nos quais as partes interagem de maneira modular, mas ainda mantêm certo grau de interdependência. Esse princípio tem sido explorado na modelagem de sistemas cognitivos e linguísticos, apontando indícios de que a linguagem pode não ser completamente decomponível em unidades independentes, mas sim estruturada em componentes interconectados por relações hierárquicas e padrões recorrentes. No contexto computacional, a quase-decomponibilidade sugere que, embora seja viável segmentar tarefas em subtarefas menores, essas unidades podem manter dependências que precisam ser consideradas para preservar a coerência e a eficiência do processamento linguístico.

Essa ideia se alinha diretamente à *Teoria Baseada no Uso* (*usage-based theory, UBT*), que argumenta que a estrutura da linguagem emerge da experiência de uso e da repetição de construções linguísticas em diferentes contextos, conforme discutido na Seção 4.2. Em vez de postular uma gramática fixa e rigidamente segmentada, a UBT propõe que padrões linguísticos surgem como unidades simbólicas flexíveis, organizadas de acordo com sua frequência e funcionalidade no discurso. A interseção entre quase-decomponibilidade e UBT sugere que a linguagem pode ser representada como uma rede de construções parcialmente independentes, nas quais elementos estruturais mantêm interdependências semânticas e pragmáticas, ao mesmo tempo em que permitem um certo grau de modularidade no processamento.

No contexto do PLN (PLN), a *decomposição de tarefas por padrões linguísticos* opera sob esses mesmos princípios. Modelos de PLN frequentemente segmentam tarefas complexas, como análise sintática, reconhecimento de entidades nomeadas e inferência textual, em subtarefas menores. No entanto, essa decomposição não pode ser arbitrária, pois a estrutura linguística impõe restrições que precisam ser preservadas para que o sistema funcione de maneira eficaz. A identificação de padrões linguísticos recorrentes serve como uma estratégia para definir pontos naturais de segmentação, garantindo que a decomposição respeite as relações latentes entre os componentes do texto.

Além disso, modelos como *Large Language Models* (LLMs) demonstram evidências de internalizar esses padrões ao capturar distribuições estatísticas e regularidades estruturais da linguagem natural. Como os LLMs são treinados em larga escala sobre dados reais da língua, sua capacidade de representar construções quase-decomponíveis

sugere que a decomposição de tarefas em PLN pode ser informada diretamente pelas representações emergentes desses modelos. Essa abordagem permite uma integração entre princípios linguísticos e técnicas computacionais, possibilitando que a decomposição de tarefas por padrões linguísticos seja refinada de maneira dinâmica à medida que os modelos capturam novas regularidades na linguagem.

4.4.4 Padrões de Decomposição e sua Influência na Componibilidade das LLMs

A decomposição de tarefas tem sido abordada como uma estratégia para mitigar limitações dos *Modelos de Linguagem de Grande Escala* (LLMs), especialmente no que se refere à generalização, ao raciocínio estruturado e à interpretabilidade. Conforme discutido na Seção 3.3.4, os desafios enfrentados por esses modelos, como dificuldades em inferências multi-etapas e tendência a gerar respostas inconsistentes, podem ser parcialmente contornados ao estruturar suas operações em padrões de decomposição.

Uma das estratégias analisadas é a Chain of Thought (CoT), que permite a explicitação de passos intermediários no raciocínio do modelo. Como destacado anteriormente, essa técnica se desdobrou em variantes como *Least-to-Most Prompting* e *Question Decomposition*, demonstrando potencial para melhorar a capacidade dos LLMs de lidar com tarefas complexas. A segmentação explícita das operações permite que os modelos organizem suas respostas de forma mais previsível, tornando a inferência menos dependente de padrões estatísticos superficiais.

Outra abordagem explorada foi a dos *LLM-Based Algorithms*, que estruturam tarefas como grafos computacionais, permitindo uma organização hierárquica do raciocínio. Esse método sugere que, ao integrar operações algorítmicas aos modelos, é possível tornar sua execução mais eficiente e componível, favorecendo a reutilização de estratégias previamente aprendidas. Entretanto, a eficácia dessa abordagem pode variar conforme a complexidade da tarefa e a necessidade de adaptação dos modelos a novos contextos.

Além disso, a decomposição de tarefas tem sido aplicada em agentes baseados em LLMs (*LLM-Based Agents*), os quais necessitam estruturar seu planejamento e tomada de decisão em múltiplas etapas. Como descrito na Seção 3.3.4, esses agentes podem se beneficiar da segmentação hierárquica de suas tarefas, permitindo um melhor controle sobre a execução de processos que envolvem múltiplas interações. No entanto, desafios permanecem quanto à capacidade dos modelos de aplicar essa estruturação de forma sistemática em ambientes dinâmicos.

A Seção 3.3.4 também apresentou técnicas voltadas à redução de erros e ao aprimoramento da auditabilidade dos modelos, como a RAG. Ao integrar mecanismos de recuperação de informações externas ao processo de geração, essa abordagem busca

mitigar problemas como alucinações de conteúdo e inconsistências factuais. No entanto, sua efetividade depende de fatores como a precisão na recuperação dos documentos e a integração coerente dessas informações na resposta gerada.

Embora padrões de decomposição possam contribuir para tornar os LLMs mais estruturados e componíveis, sua aplicação ainda apresenta desafios e exige investigações adicionais. A segmentação de tarefas pode facilitar a organização do raciocínio e tornar as inferências dos modelos mais transparentes, mas a viabilidade dessa abordagem varia conforme o contexto da aplicação e a complexidade das tarefas envolvidas.

Assim, o uso de padrões de decomposição representa uma alternativa promissora para aprimorar a organização e previsibilidade dos LLMs. No entanto, conforme discutido na Seção 3.3.4, a eficácia dessas técnicas ainda depende de fatores como a capacidade dos modelos de internalizar essas estruturas e sua aplicabilidade em cenários diversos. Pesquisas futuras poderão fornecer uma compreensão mais precisa sobre seus benefícios e limitações em diferentes domínios.

4.5 Padrão Arquitetural Recrutador-Selecionador

Na seção 3.2.1 (Padrões de Projeto), foram explorados diversos padrões arquiteturais amplamente utilizados no contexto da Engenharia de *Software* e que oferecem soluções estruturadas para problemas de organização e processamento de dados. Inspirado por essas abordagens, o padrão *Recrutador-Selecionador* emergiu como uma solução inovadora durante os estudos de caso apresentados nos Capítulos 5 (Segmentação de *Hashtags*) e 6 (Curadoria de Frases-Chave (CFC)). Esses estudos de caso, focados na anotação de *corpus* para PLN, revelaram a necessidade de um padrão que equilibrasse a exploração eficiente de um espaço de problema com a seleção refinada de soluções candidatas, especialmente em tarefas que exigem o reconhecimento e a classificação de padrões linguísticos.

O padrão *Recrutador-Selecionador* apresenta semelhanças com abordagens em diversas áreas, refletindo sua natureza versátil e adaptável. Em Recursos Humanos (RH), o padrão espelha o processo de recrutamento, onde candidatos são inicialmente filtrados (*Recrutador*) com base em critérios individuais, como qualificações, e depois avaliados comparativamente (*Selecionador*) para seleção final, assemelhando-se à geração e classificação de $\mathcal{C}$. Em Inteligência Artificial (IA), ele se assemelha a pipelines de aprendizado de máquina que combinam geração de hipóteses (e.g., *feature* extraction) com seleção de modelos, especialmente em tarefas de PLN onde LLMs ajudam a identificar padrões linguísticos em $2^{\mathcal{P}}$. Nos Sistemas de Recomendação, o *Recrutador* atua como um gerador de itens candidatos (similar à etapa de *candidate generation*), enquanto o *Selecionador* refina a lista com base em preferências do usuário, comparável à classificação em $\mathcal{S}^*$. Na

arquitetura *Map/Reduce*, a decomposição de $\mathcal{P}$ em $2^{\mathcal{P}}$ pelo *Recrutador* reflete a fase *Map*, e a classificação pelo *Selecionador* é análoga à fase *Reduce*, que agrega e organiza resultados. Por fim, em Algoritmos Genéticos, o padrão ecoa a geração de populações iniciais (*Recrutador*) e a seleção de indivíduos mais aptos (*Selecionador*) com base em funções de aptidão, similar à aplicação de E e $-$, destacando sua aplicabilidade em domínios que requerem exploração e refinamento estruturados.

Esta seção está estruturada em quatro subseções que detalham o padrão *Recrutador-Selecionador* sob diferentes perspectivas. Inicialmente, a Subseção 4.5.1 apresenta uma definição matemática rigorosa do padrão, modelando o espaço de problema como uma sequência de tokens e formalizando as funções do *Recrutador* e do *Selecionador*. Em seguida, a Subseção 4.5.2 explora as primitivas de decomposição de tarefas e manipulação de padrões que sustentam a geração de candidatos, detalhando operações como seleção, composição e reconhecimento. A Subseção 4.5.3 oferece uma visualização esquemática do fluxo do padrão, ilustrando a conexão entre os componentes por meio de um diagrama claro e intuitivo. Por fim, a Subseção 4.5.4 analisa a eficácia do padrão, destacando como a integração de LLMs nas primitivas pode superar a explosão combinatória do espaço de problema, promovendo maior eficiência e qualidade nas anotações. Juntas, essas subseções fornecem uma visão abrangente do padrão, preparando o terreno para sua aplicação prática nos estudos de caso subsequentes.

4.5.1 Definição Matemática do Padrão Recrutador-Selecionador

O padrão **Recrutador-Selecionador** é formalizado como um processo estruturado em um espaço de problema definido por uma sequência de tokens. Considere $\mathcal{P} = (t_1, t_2, \ldots, t_n)$ o espaço de problema, onde cada t_i é um token em uma sequência ordenada, com relações latentes entre os tokens (e.g., dependências sintáticas ou semânticas). O conjunto $2^{\mathcal{P}}$ representa todas as possíveis subsequências de $\mathcal{P}$. O espaço total de soluções possíveis é denotado por $\mathcal{S}$, e o objetivo é obter um conjunto de soluções classificadas $\mathcal{S}^*$, derivado de $\mathcal{S}$.

O *Recrutador* é definido como uma função de geração de candidatos:

$$R : 2^{\mathcal{P}} \to \mathcal{C},$$

onde $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ é o conjunto de candidatos, sendo $\mathcal{C} \subseteq \mathcal{S}$. Cada candidato $c \in \mathcal{C}$ é uma solução inicial gerada a partir de uma subsequência $X \in 2^{\mathcal{P}}$, avaliada individualmente por um critério de qualificação $E : \mathcal{S} \to \mathbb{R}$. O critério E mede propriedades intrínsecas de cada candidato (e.g., consistência, relevância interna), produzindo um valor que pode ser

adaptado a inclusão/exclusão por meio de um limiar $\theta \in \mathbb{R}$. Formalmente:

$$\mathcal{C} = \{c \in \mathcal{S} \mid \exists X \in 2^{\mathcal{P}}, c = r(X) \text{ e } E(c) \geq \theta\},$$

onde $r : 2^{\mathcal{P}} \to \mathcal{S}$ mapeia subsequências a candidatos em $\mathcal{S}$, e $E(c) \geq \theta$ indica qualificação suficiente para inclusão, sendo θ um parâmetro opcional que pode variar conforme o contexto. O *Recrutador* explora $2^{\mathcal{P}}$ e gera $\mathcal{C}$ com base em E, focando em avaliações individuais sem comparações entre candidatos.

O *Selecionador* é modelado como uma função de classificação:

$$S_e : \mathcal{C} \to \mathcal{S}^*,$$

que mapeia o conjunto de candidatos $\mathcal{C}$ para o conjunto de soluções classificadas $\mathcal{S}^*$. Aqui, $\mathcal{S}^*$ associa cada candidato a um rótulo, com base em um conjunto de critérios de qualidade $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$, que podem incluir avaliações comparativas entre candidatos (e.g., relevância relativa, coerência global) ou propriedades individuais. Formalmente:

$$\mathcal{S}^* = S_e(\mathcal{C}) = \{(c_j, l_j) \mid c_j \in \mathcal{C}, l_j = f_\Gamma(c_j)\},$$

onde $f_\Gamma : \mathcal{C} \to \mathcal{L}$ é uma função de classificação que atribui a cada candidato c_j um rótulo $l_j \in \mathcal{L}$ (e.g., alta/baixa qualidade), sendo $\mathcal{L}$ o conjunto de rótulos possíveis, e f_Γ é determinada por Γ. O *Selecionador* organiza $\mathcal{C}$ em $\mathcal{S}^*$, aplicando critérios mais amplos, frequentemente comparativos, distintos de E.

O padrão *Recrutador-Selecionador* é descrito pela composição das funções R e S_e:

$$\mathcal{S}^* = S_e(R(2^{\mathcal{P}})),$$

representando um processo em duas etapas: o *Recrutador* gera $\mathcal{C} \subseteq \mathcal{S}$ a partir de $2^{\mathcal{P}}$, qualificando candidatos com base em E aplicado a padrões intrínsecos, e o *Selecionador* classifica esses candidatos em $\mathcal{S}^*$ usando Γ, que pode envolver comparações. Essa formalização separa a qualificação individual no *Recrutador*, adaptável a limiares, da classificação comparativa no *Selecionador*.

4.5.2 Primitivas de Decomposição de Tarefas e Manipulação de Padrões

No contexto do padrão *Recrutador-Selecionador*, a exploração do espaço de problema $\mathcal{P} = (t_1, t_2, \ldots, t_n)$ pelo *Recrutador* é realizada por meio de primitivas que decompõem tarefas e manipulam padrões para gerar o conjunto de candidatos $\mathcal{C} \subseteq S$. Essas primitivas operam sobre $2^{\mathcal{P}}$, o conjunto de todas as subsequências de $\mathcal{P}$, e contribuem para

a avaliação individual dos candidatos com base no critério de qualificação $E : S \to \mathbb{R}$, que pode ser ajustado por um limiar θ para inclusão em $\mathcal{C}$. As principais primitivas envolvidas são:

- **Seleção**: Identificação de subsequências em $2^{\mathcal{P}}$ com base em propriedades intrínsecas relevantes. Formalmente, é uma função $S_e : 2^{\mathcal{P}} \to 2^{\mathcal{P}}$ que filtra subsequências conforme um critério auxiliar $E_s : \mathcal{P} \to \mathbb{R}$ (e.g., frequência de tokens), contribuindo para a geração de candidatos qualificados por E.
- **Composição**: Combinação de subsequências em $2^{\mathcal{P}}$ para formar candidatos mais complexos. Modelada como $C_o : 2^{\mathcal{P}} \times 2^{\mathcal{P}} \to 2^{\mathcal{P}}$, esta primitiva une subestruturas (e.g., tokens adjacentes) em padrões que são mapeados por $r : 2^{\mathcal{P}} \to S$ e avaliados por E.
- **Agrupamento**: Organização de subsequências semelhantes em $2^{\mathcal{P}}$ em categorias. Definida como $G : 2^{\mathcal{P}} \to \mathcal{G}$, onde $\mathcal{G}$ é um conjunto de grupos (e.g., baseados em similaridade semântica), facilita a identificação de padrões para inclusão em $\mathcal{C}$.
- **Decomposição**: Divisão de uma subsequência $X \in 2^{\mathcal{P}}$ em componentes menores. Representada por $D : 2^{\mathcal{P}} \to 2^{\mathcal{P}}$, permite analisar subestruturas individuais, cujos candidatos resultantes são qualificados por E.
- **Reconhecimento**: Detecção de padrões recorrentes em $2^{\mathcal{P}}$. Formalizada como $R_e : 2^{\mathcal{P}} \to \mathcal{M}$, onde $\mathcal{M}$ é um conjunto de padrões reconhecidos (e.g., n-gramas frequentes), apoia a geração de $\mathcal{C}$ ao destacar relações latentes.
- **Estruturação**: Organização de subsequências em $2^{\mathcal{P}}$ em uma forma hierárquica. Definida como $T : 2^{\mathcal{P}} \to \mathcal{H}$, onde $\mathcal{H}$ é uma estrutura ordenada, refina os candidatos para avaliação por E.

Essas primitivas são aplicadas pelo *Recrutador* para transformar $2^{\mathcal{P}}$ em $\mathcal{C}$, conforme $R : 2^{\mathcal{P}} \to \mathcal{C}$. Elas exploram as relações latentes em $\mathcal{P}$ (e.g., dependências sintáticas) e geram candidatos que são individualmente qualificados por E, como $E(c) \geq \theta$, sem envolver comparações entre candidatos. O *Selecionador*, por sua vez, opera sobre $\mathcal{C}$ com $S_e : \mathcal{C} \to \mathcal{S}^*$, utilizando Γ para classificar os candidatos, frequentemente de forma comparativa. Assim, as primitivas fornecem uma base estruturada para a geração de $\mathcal{C}$, alinhando-se ao objetivo do padrão de produzir $\mathcal{S}^*$ de maneira sistemática.

4.5.3 Representação Esquemática do Padrão Recrutador-Selecionador

O funcionamento do padrão *Recrutador-Selecionador* pode ser visualizado por meio de um diagrama esquemático, conforme apresentado na Figura 4.9. Este diagrama ilustra o fluxo linear do processo, desde o espaço de problemas até o conjunto final de

soluções classificadas, destacando as funções do *Recrutador* como gerador de candidatos e do *Selecionador* como classificador de candidatos.

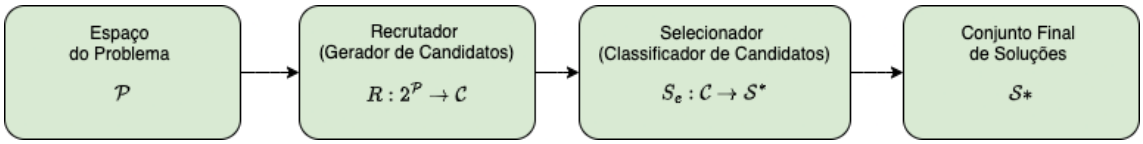


Figura 4.9: *Diagrama esquemático do padrão Recrutador-Selecionador, mostrando o fluxo do espaço de problema $\mathcal{P}$ para o conjunto final de soluções $\mathcal{S}^*$. O Recrutador ($R : 2^{\mathcal{P}} \to \mathcal{C}$) gera candidatos $\mathcal{C} \subseteq \mathcal{S}$ a partir das subsequências de $\mathcal{P}$, enquanto o Selecionador ($S_e : \mathcal{C} \to \mathcal{S}^*$) classifica os candidatos em $\mathcal{S}^*$.*

O diagrama representa o espaço de problema $\mathcal{P} = (t_1, t_2, \ldots, t_n)$ como uma sequência de tokens, que é explorada pelo *Recrutador* para gerar o conjunto de candidatos $\mathcal{C} \subseteq \mathcal{S}$ a partir de todas as subsequências possíveis em $2^{\mathcal{P}}$. O *Recrutador*, denotado por $R : 2^{\mathcal{P}} \to \mathcal{C}$, utiliza um critério de qualificação $E : \mathcal{S} \to \mathbb{R}$ para avaliar individualmente cada candidato, combinando-o opcionalmente com um limiar θ para determinar sua inclusão. Em seguida, o *Selecionador*, representado por $S_e : \mathcal{C} \to \mathcal{S}^*$, classifica os candidatos em $\mathcal{S}^*$ com base no conjunto de critérios $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$, que pode incluir avaliações comparativas entre os candidatos.

A composição $\mathcal{S}^* = S_e(R(2^{\mathcal{P}}))$ é refletida no fluxo linear do diagrama, que conecta o espaço de problema às soluções classificadas por meio das duas etapas principais. As setas indicam a direção do processo, enquanto os rótulos (*Espaço do Problema*, *Recrutador (Gerador de Candidatos)*, *Selecionador (Classificador de Candidatos)*, *Conjunto Final de Soluções*) clarificam o papel de cada componente, enfatizando a separação entre a geração e qualificação inicial no *Recrutador* e a classificação final no *Selecionador*.

4.5.4 Eficácia do Padrão Arquitetural Recrutador-Selecionador

A eficácia do padrão arquitetural *Recrutador-Selecionador* reside em sua capacidade de estruturar a resolução de problemas complexos, como aqueles envolvendo espaços de problema extensos, de maneira eficiente e escalável. No contexto de uma sequência de *tokens* $\mathcal{P} = (t_1, t_2, \ldots, t_n)$ com relações latentes, o espaço de todas as subsequências $2^{\mathcal{P}}$ pode apresentar uma explosão combinatória, cujo tamanho cresce exponencialmente com n (i.e., 2^n), tornando a exploração exaustiva inviável em cenários de alta dimensionalidade. O padrão mitiga esse desafio ao integrar primitivas que aproveitam LLMs para capturar padrões linguísticos latentes, otimizando a geração e classificação de candidatos.

As primitivas de decomposição, composição e reconhecimento, aplicadas pelo *Recrutador* ($R : 2^{\mathcal{P}} \to \mathcal{C}$), podem ser aprimoradas por LLMs treinados em vastos *corpora* linguísticos. Esses modelos identificam padrões intrínsecos (e.g., sintaxe, semântica,

coocorrências) de forma probabilística, reduzindo a necessidade de explorar todas as combinações em $2^{\mathcal{P}}$. Por exemplo, um LLM pode priorizar subsequências que exibem alta probabilidade de formar frases coerentes ou relações semânticas significativas, guiando o critério de qualificação $E : \mathcal{S} \to \mathbb{R}$ para selecionar candidatos $\mathcal{C} \subseteq \mathcal{S}$ de maneira direcionada. Essa abordagem diminui a complexidade computacional, substituindo a busca exaustiva por uma exploração informada, baseada no conhecimento latente dos LLMs.

Além disso, a integração de LLMs nas primitivas enriquece a avaliação do *Selecionador* ($S_e : \mathcal{C} \to \mathcal{S}^*$). Os modelos podem fornecer embeddings ou pontuações contextuais que refinam os critérios $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$, permitindo classificações mais precisas e comparativas entre candidatos. Por exemplo, a coerência global ou a relevância semântica de um candidato pode ser avaliada em relação a um *corpus* de referência, otimizando o conjunto final de soluções $\mathcal{S}^*$. Essa sinergia entre primitivas e LLMs não apenas supera a explosão combinatória, mas também melhora a qualidade das soluções, especialmente em tarefas de PLN como anotação de *corpus*.

A eficácia do padrão foi observada em estudos de caso, como a segmentação de *hashtags* e a curadoria de frases-chave, onde a utilização de LLMs nas primitivas reduziu significativamente o tempo de processamento e aumentou a precisão das anotações. A capacidade de adaptar o limiar θ em E e os critérios $-$ com base em saídas de LLMs permite uma flexibilidade adicional, tornando o padrão robusto frente a diferentes tamanhos de $\mathcal{P}$ e níveis de complexidade. Assim, o *Recrutador-Selecionador*, ao incorporar LLMs, oferece uma solução escalável e eficiente, superando os limites impostos pela explosão combinatória e promovendo soluções de alta qualidade em contextos linguísticos diversos.

4.5.5 Semelhanças com Outros Padrões e Contexto Inspirador

O padrão *Recrutador-Selecionador* foi inspirado por uma metáfora do mundo real: o processo de recrutamento e seleção em Recursos Humanos (RH). Nesse contexto, o *Recrutador* reflete a etapa inicial de triagem, onde candidatos são avaliados individualmente com base em critérios de qualificação (e.g., experiência, habilidades), semelhante à geração de $\mathcal{C} \subseteq \mathcal{S}$ a partir de $2^{\mathcal{P}}$ com o critério $E : \mathcal{S} \to \mathbb{R}$. Já o *Selecionador* corresponde à fase de seleção final, onde os candidatos são comparados e classificados (e.g., entrevistas, testes), análoga à classificação de $\mathcal{C}$ em $\mathcal{S}^*$ usando Γ. Essa metáfora foi fundamental para estruturar o padrão, pois captura a essência de equilibrar a geração ampla de possibilidades com uma avaliação refinada, uma necessidade comum em tarefas de PLN.

Embora o padrão tenha sido desenvolvido no contexto de PLN, uma revisão da

literatura não identificou trabalhos que aplicassem diretamente um padrão equivalente para tarefas linguísticas. No entanto, padrões com estruturas semelhantes foram encontrados em outras áreas da computação, evidenciando a aplicabilidade interdisciplinar do *Recrutador-Selecionador*. Em Inteligência Artificial (IA), o padrão ecoa pipelines de aprendizado de máquina, onde a geração de hipóteses (e.g., extração de características) é análoga ao *Recrutador*, e a seleção de modelos (e.g., validação cruzada) reflete o *Selecionador*, especialmente em tarefas de PLN que utilizam Large Language Models (LLMs) para explorar padrões em $2^{\mathcal{P}}$. Nos Sistemas de Recomendação, o *Recrutador* é similar à etapa de *candidate generation*, que produz um conjunto inicial de itens, enquanto o *Selecionador* se assemelha ao *ranking*, classificando itens com base em preferências do usuário, comparável à produção de $\mathcal{S}^*$. Na arquitetura *Map/Reduce*, a decomposição de $\mathcal{P}$ em $2^{\mathcal{P}}$ pelo *Recrutador* espelha a fase *Map*, e a classificação pelo *Selecionador* é análoga à fase *Reduce*, que agrega resultados em $\mathcal{S}^*$. Por fim, em Algoritmos Genéticos, o padrão se alinha à geração de populações iniciais (*Recrutador*) e à seleção de indivíduos mais aptos (*Selecionador*) com base em funções de aptidão, refletindo a aplicação de E e Γ.

A ausência de padrões diretamente comparáveis em PLN sugere que o *Recrutador-Selecionador* é uma contribuição inovadora para o campo, preenchendo uma lacuna ao adaptar conceitos de outras áreas da computação para tarefas linguísticas. A metáfora de RH, combinada com a estrutura modular do padrão, permite que ele seja adaptado a diferentes domínios, desde a anotação de *corpus* até problemas mais amplos que envolvam exploração e refinamento de soluções. Essa versatilidade, aliada à sua capacidade de integrar LLMs para superar a explosão combinatória de $2^{\mathcal{P}}$, como discutido na Subseção 4.5.4, posiciona o padrão como uma ferramenta promissora para avanços em PLN e além.

4.6 Decomposição de Tarefas na Anotação de Corpus

A decomposição de tarefas emergiu como uma estratégia fundamental para enfrentar os desafios da anotação na tarefa de *mapeamento de argumentos*, cuja resolução automatizada era o objetivo inicial da pesquisa. A ausência de *datasets* anotados impôs a necessidade de construir *corpora* do zero, mas os primeiros experimentos revelaram um problema crítico: a **baixa confiabilidade da anotação**, evidenciada por uma concordância negativa ou nula entre anotadores. Como a concordância é uma medida indireta de confiabilidade, esse resultado indicava que a tarefa, tal como inicialmente formulada, apresentava inconsistências na interpretação das diretrizes ou na aplicabilidade do esquema de anotação, tornando inviável um processo de anotação direta.

Sob a ótica de [Simon 1962] apresentada na Seção 3.3.1, a dificuldade em garantir a confiabilidade da anotação pode ser explicada pela *complexidade da tarefa*,

uma vez que sistemas complexos tendem a exigir maior esforço cognitivo para serem compreendidos e manipulados. Simon argumenta que a complexidade de um sistema não está apenas na quantidade de elementos que o compõem, mas na maneira como esses elementos interagem e são representados cognitivamente. Além disso, o autor destaca que a compreensão humana da complexidade é limitada pelos **limites do conhecimento da tarefa**, ou seja, a capacidade de um indivíduo lidar com um problema depende diretamente da forma como as informações são organizadas e apresentadas. Quando uma tarefa exige um nível de abstração excessivo ou apresenta múltiplas interpretações, a tomada de decisão se torna inconsistente, comprometendo a confiabilidade do processo.

Para mitigar esse problema, a decomposição da anotação em **subtarefas mais objetivas e bem delimitadas** foi essencial. Em vez de solicitar que os anotadores realizassem a tarefa globalmente — o que levava a inconsistências e variações difíceis de controlar —, foram definidas etapas distintas, como a *curadoria de frases-chave*, a *detecção do gênero textual*, a *segmentação e a classificação de tópicos*. Essas subtarefas foram submetidas a processos rigorosos de avaliação de qualidade, garantindo maior estabilidade e reprodutibilidade dos dados gerados.

Além de melhorar a confiabilidade da anotação, a decomposição da tarefa também se alinha ao conceito de **arquitetura da complexidade** de [Simon 1962], no qual a resolução de problemas pode ser facilitada quando estruturada em níveis hierárquicos de menor complexidade. Simon argumenta que sistemas complexos podem ser melhor compreendidos e manipulados quando organizados de forma hierárquica, pois isso permite que cada componente seja analisado individualmente, sem que o sistema como um todo se torne incompreensível. Dessa forma, a decomposição de tarefas não apenas melhorou a precisão e a imparcialidade das anotações, como também possibilitou um refinamento contínuo das diretrizes, tornando o processo mais robusto e escalável para futuras etapas do *mapeamento de argumentos*.

4.6.1 Algoritmo de Decomposição Hierárquica de Tarefas

A decomposição de tarefas baseada na concordância entre anotadores é um método iterativo que permite o refinamento progressivo da anotação, garantindo maior confiabilidade nos dados. Esse processo segue um ciclo de avaliação e ajuste das diretrizes (*guidelines*), permitindo a segmentação da tarefa sempre que a concordância entre anotadores indicar instabilidade.

Algoritmo 4.1: Decomposição Hierárquica de Tarefas de Anotação

Entrada: $\mathcal{T} = (\mathcal{G}, I, Q, \mathcal{F})$, onde:

$\mathcal{G}$ – Conjunto de diretrizes (*guideline* com instruções básicas de anotação);

I – Conjunto de instâncias da tarefa (inicialmente não anotadas);

Q – Métrica de confiabilidade (inicialmente nula);

$\mathcal{F}$ – Conjunto de subtarefas (inicialmente vazio).

$\mathcal{A}$ – Conjunto de anotadores.

A **Função** `DecomporTarefa`$(T, \mathcal{A})$**:**

- $I_a \leftarrow \varnothing$; `// Conjunto de instâncias anotadas`
- $\mathcal{C} \leftarrow \varnothing$; `// Coleção das métricas de concordância`
- **enquanto** $(\mathcal{L} \leftarrow$ `ObterPróximoLote`$(I)) \neq \varnothing$ **faça**
  - $\mathcal{L}_a \leftarrow$ `Anotar`$(\mathcal{L}, \mathcal{A}, \mathcal{G})$;
  - $I_a \leftarrow I_a \cup \mathcal{L}_a$;
  - $\mathcal{G} \leftarrow$ `AtualizarGuideline`$(\mathcal{G}, \mathcal{L}_a)$;
  - $c \leftarrow$ `ComputarConcordância`$(\mathcal{L}_a)$;
  - $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$;
  - $Q \leftarrow$ `ComputarConfiança`$(\mathcal{C})$;
  - **se** `ChecarEstabilidade`$(\mathcal{C})$ **então**
    - **retorne** `CriarNó`$(\mathcal{G}, I_a, Q, \varnothing)$;
  - $\Delta \leftarrow$ `AvaliarEfeito`$(\mathcal{C})$;
  - **se** $\Delta \leq 0$ ***e*** `ChecarDecomponibilidade`(T) **então**
    - $\mathcal{F} \leftarrow \varnothing$;
    - $\mathcal{S} \leftarrow$ `IdentificarSubtarefas`(T);
    - **para** ***cada*** $s \in \mathcal{S}$ **faça**
      - $\mathcal{F} \leftarrow \mathcal{F} \cup \{$`DecomporTarefa`$(s, \mathcal{A})\}$;
    - **retorne** `CriarNó`$(\mathcal{G}, I_a, Q, \mathcal{F})$;
- **retorne** `CriarNó`$(\mathcal{G}, I_a, Q, \mathcal{F})$;

O Algoritmo 4.1 sistematiza o processo de anotação de dados de forma iterativa e hierárquica, buscando aprimorar tanto a confiabilidade das anotações quanto a adequação das diretrizes. Para tanto, duas dimensões estatísticas são consideradas: *estabilidade* e *efeito acumulado*.

Estabilidade A estabilidade refere-se à consistência das medições de concordância entre os diferentes lotes de anotações. Em outras palavras, as variações na métrica de concordância devem ser pequenas e não sistemáticas ao longo do tempo. Por exemplo, considere que, para cada lote i (com $i = 1, \ldots, L$), existam n_i observações de concordância

C_{ij} (para $j = 1, \ldots, n_i$). Para cada lote, calcula-se a média:

$$\bar{C}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} C_{ij},$$

e a variância σ_i^2. Um indicador simples de estabilidade é o coeficiente de variação (CV):

$$CV_i = \frac{\sigma_i}{\bar{C}_i}.$$

Se os valores de CV_i permanecerem consistentemente abaixo de um limiar predefinido θ (por exemplo, $\theta = 0.1$), conclui-se que as medições são estáveis. Dessa forma, a função `ChecarEstabilidade(`$\mathcal{C}$`)` verifica se não há ganhos adicionais na concordância entre os anotadores. Quando essa condição é satisfeita, entende-se que o processo de anotação se estabilizou e a tarefa pode ser considerada completa, não necessitando de novos ajustes.

Efeito Acumulado O efeito acumulado quantifica a magnitude das variações na métrica de concordância entre lotes sucessivos. Suponha que, para cada lote i, a média de concordância seja $\bar{C}_i$. A diferença entre dois lotes consecutivos é definida por:

$$\Delta_i = \bar{C}_{i+1} - \bar{C}_i, \quad i = 1, \ldots, L-1.$$

Para padronizar essa diferença e avaliar sua relevância, utiliza-se o *Cohen's d*. Se s_i e s_{i+1} são os desvios padrões dos lotes i e $i+1$, e n_i e n_{i+1} os respectivos tamanhos amostrais, o desvio padrão combinado s_p é calculado por:

$$s_p = \sqrt{\frac{(n_i - 1)s_i^2 + (n_{i+1} - 1)s_{i+1}^2}{n_i + n_{i+1} - 2}},$$

e o tamanho do efeito é dado por:

$$d_i = \frac{\bar{C}_{i+1} - \bar{C}_i}{s_p}.$$

Valores de d_i próximos de zero (por exemplo, $d_i < 0.2$) indicam que as variações entre os lotes são pequenas. A função `AvaliarEfeito(`$\mathcal{C}$`)` utiliza essa análise para determinar se as atualizações no *guideline* estão produzindo efeitos consistentemente negativos ou nulos. Quando esse cenário é identificado, significa que as mudanças não estão promovendo melhorias na concordância, sinalizando a necessidade de decompor a tarefa em subtarefas para reavaliar e ajustar as diretrizes.

Funções que exploram padrões linguísticos são aquelas que envolvem decomposição de tarefas baseadas em análise de padrões linguísticos.

- A função `ChecarDecomponibilidade(`T`)` avalia se a tarefa T pode ser decomposta em subtarefas menores, baseando-se nos princípios apresentados na Seção 3.3.1. Nessa seção, argumenta-se que um sistema complexo é decomponível quando possui uma estrutura hierárquica, e isso é particularmente relevante para a linguagem natural, que é caracterizada por padrões linguísticos recorrentes e relações sintáticas e semânticas bem definidas. Portanto, essa função deve explorar a organização hierárquica inerente às estruturas linguísticas para determinar se uma tarefa pode ser dividida em partes menores de forma significativa, garantindo que a decomposição preserve a coerência e a interpretabilidade do processo de anotação.
- A função `IdentificarSubtarefas(`T`)` complementa a função anterior ao realizar a segmentação efetiva da tarefa em subtarefas mais manejáveis. Assim como `ChecarDecomponibilidade(`T`)`, essa função se baseia em padrões linguísticos para definir divisões apropriadas dentro da estrutura da linguagem. Métodos de agrupamento de instâncias, análise de padrões morfossintáticos e heurísticas baseadas em relações semânticas podem ser empregados para identificar subtarefas distintas dentro de T. A principal diferença em relação à função anterior é que, enquanto `ChecarDecomponibilidade(`T`)` verifica se a decomposição é viável, `IdentificarSubtarefas(`T`)` determina a melhor maneira de dividir a tarefa em segmentos coerentes e funcionalmente relevantes.

Detalhamento das Funções Auxiliares Para preencher as lacunas relativas aos detalhes do pseudocódigo, apresentam-se a seguir os objetivos e possíveis implementações das funções auxiliares:

- **`ObterPróximoLote(`**I**`)`**: Essa função é responsável por selecionar e retornar um subconjunto de instâncias ainda não anotadas a partir do conjunto I. A seleção pode ser realizada de forma sequencial, aleatória ou por meio de algum critério de prioridade, garantindo a cobertura completa das instâncias durante o processo iterativo.
- **`AtualizarGuideline(`**$\mathcal{G}, \mathcal{L}_a$**`)`**: Após a anotação de um lote, é comum que surjam ambiguidades ou novas interpretações que demandem ajustes nas diretrizes. Essa função incorpora o feedback do lote anotado $\mathcal{L}_a$ para refinar e atualizar o conjunto de diretrizes $\mathcal{G}$, de modo a melhorar a consistência entre os anotadores nas próximas iterações.
- **`ComputarConcordância(`**$\mathcal{L}_a$**`)`**: Aqui é calculada a métrica de concordância entre os anotadores para o lote atual $\mathcal{L}_a$. Dependendo do contexto e da natureza

da tarefa, pode-se empregar medidas estatísticas como o coeficiente de Cohen's kappa, a correlação ou outras métricas que quantifiquem o grau de acordo entre as anotações.

- **`ComputarConfiança(`**$\mathcal{C}$**`)`**: infere um valor de confiança a partir das métricas de concordância acumuladas no conjunto $\mathcal{C}$, conforme descrito na Seção 3.1. É importante considerar que, nas primeiras rodadas de anotação, a concordância entre anotadores pode ser baixa devido a ambiguidades iniciais e ao ajuste das diretrizes. No entanto, com a progressão do processo iterativo, a tendência natural é que a concordância se estabilize à medida que as diretrizes se tornam mais claras e os anotadores se adaptam às regras da anotação. Assim, a função deve empregar uma estratégia que evite conclusões precipitadas a partir de valores iniciais baixos e que permita a confiança ser ajustada dinamicamente conforme a estabilidade do sistema aumenta.
- **`Anotar(`**$\mathcal{L}, \mathcal{A}, \mathcal{G}$**`)`**: Essa função executa a anotação do lote $\mathcal{L}$ pelos anotadores $\mathcal{A}$ utilizando as diretrizes atuais $\mathcal{G}$. O resultado é o conjunto $\mathcal{L}_a$ de instâncias já anotadas, que servirá como base para a atualização das diretrizes e para o cálculo das métricas de concordância.
- **`CriarNó(`**$\mathcal{G}, I_a, Q, \mathcal{F}$**`)`**: Ao final do processo ou de uma etapa de decomposição, essa função gera um nó que encapsula o estado atual da anotação. Esse nó inclui as diretrizes atualizadas $\mathcal{G}$, as instâncias já anotadas I_a, a métrica de confiabilidade Q e, se aplicável, o conjunto de subtarefas $\mathcal{F}$.

Integração no Fluxo do Algoritmo Após a atualização das diretrizes e o acúmulo das métricas de concordância em $\mathcal{C}$, as funções `ChecarEstabilidade(`$\mathcal{C}$`)` e `AvaliarEfeito(`$\mathcal{C}$`)` são invocadas. A primeira verifica se o processo de anotação se estabilizou — ou seja, se não há ganhos adicionais na concordância entre os anotadores, sinalizando que a tarefa está completa. Já a segunda analisa se as atualizações realizadas estão produzindo efeitos consistentemente negativos ou nulos, o que indica a necessidade de decompor a tarefa em subtarefas para promover uma nova avaliação e ajuste das diretrizes.

Essa abordagem integrada permite um monitoramento contínuo do processo de anotação, garantindo que os ajustes sejam realizados de forma dinâmica para manter a confiabilidade dos dados e a eficácia do sistema.

CAPÍTULO 5

Segmentação de *Hashtags*

Neste capítulo, apresenta-se um caso de uso da decomposição de tarefas, explorando especificamente a aplicação do padrão *Recrutador-Selecionador*, proposto na Seção 4.5, na tarefa de segmentação de *hashtags*.

A seguir, apresenta-se um resumo das seções deste capítulo:

5.1 **Introdução**: Definem-se a tarefa de segmentação de *hashtags*, seu contexto de aplicação e um histórico do processo de pesquisa desenvolvido ao longo de quatro artigos.

5.2 **Principais Contribuições**: Descrevem-se as diretrizes gerais da metodologia utilizada, os resultados alcançados e as contribuições do estudo.

5.3 **Solução Proposta**: Detalha-se a forma como o padrão Recrutador-Selecionador foi aplicado ao caso de uso.

5.4 **Experimentos e Resultados**: Apresentam-se e analisam-se os resultados experimentais obtidos na visão da arquitetura proposta.

5.5 **Conclusão**: Discutem-se as implicações da aplicação do padrão Recrutador-Selecionador e a metodologia de decomposição de tarefas, destacando as principais conclusões do estudo.

5.1 Introdução

A **segmentação de palavras** é definida como a tarefa de inserção de espaços entre palavras quando estes não estão explicitamente indicados no texto. Um caso específico dessa tarefa é a **segmentação de hashtags**, em que palavras artificialmente concatenadas para formar uma frase-chave são separadas. Essa técnica é geralmente associada a mensagens em redes sociais. *Hashtags* são amplamente utilizadas em mídias sociais, e sua segmentação é frequentemente aplicada como uma etapa de pré-processamento antes da utilização de modelos de PLN em tarefas como análise de sentimentos, detecção de discurso de ódio e identificação de eventos.

As *hashtags* geralmente não seguem convenções padronizadas da linguagem escrita, apresentando frequentemente erros ortográficos, neologismos e entidades nomeadas previamente desconhecidas. Embora algumas *hashtags* possam ser segmentadas com facilidade, uma parte substancial delas requer modelos com bom desempenho de generalização, capazes de lidar de forma robusta com palavras fora do vocabulário que não foram vistas durante o treinamento. Alguns exemplos que ilustram esses desafios incluem: #aamirkhan (‘Aamir Khan’), um ator e cineasta de Bollywood; #fangtasyisland (‘Fangtasy Island’), um erro ortográfico de ‘Fantasy Island’; e #nooootttt (‘nooootttttt’), uma variação da palavra ‘not’.

Embora hoje a separação entre palavras pareça natural nas línguas ocidentais, historicamente isso nem sempre foi o caso. No latim clássico, por exemplo, a escrita era contínua, sem espaçamento entre palavras — um estilo conhecido como *scriptio continua*, como ilustrado na Figura 5.1. Foi somente com a evolução da língua falada para a escrita e a crescente disseminação de textos manuscritos que o espaçamento passou a ser adotado, principalmente entre os séculos XIII e XIV, como discutido por Saenger [Saenger 1997]. Em contrapartida, línguas orientais como o chinês e o japonês não incorporaram essa convenção, mantendo até hoje a escrita sem separação explícita entre palavras. Por isso, a tarefa de segmentação é especialmente crítica em contextos orientais, sendo essencial para o correto funcionamento de modelos de PLN. Em línguas de origem latina, apesar da adoção histórica do espaçamento, a segmentação torna-se novamente relevante em contextos específicos onde esse espaçamento é artificialmente suprimido — como ocorre com *hashtags* em redes sociais — justificando o estudo e desenvolvimento de métodos automáticos para esse fim.

5.1.1 Processo de Construção da Solução

Este estudo de caso relata um processo contínuo de pesquisa e experimentação, resultando em diversas contribuições emergentes publicadas em quatro artigos. O primeiro artigo [Inuzuka, Rocha e Nascimento 2020] apresenta uma revisão sistemática sobre o problema da segmentação de *hashtags* (SH), na qual foram analisados 69 trabalhos selecionados a partir de um total de 771 coletados. Os resultados indicaram uma lacuna na disponibilidade de ferramentas, conjuntos de dados e algoritmos específicos para línguas ocidentais.

No segundo artigo [Rodrigues et al. 2020], foram exploradas técnicas de transferência e adaptação de conhecimento, destilando um LLM, o BERT Base, a um modelo menor (BERT Mini), que foi otimizado para a tarefa de SH. Esse modelo superou duas abordagens anteriores: uma baseada em CNN-BiLSTM-CRF [Ma, Ganchev e Weiss 2018] e outra em LSTM [Doval e Gómez-Rodríguez 2019].

Figura 5.1: *Página do Codex Vaticanus, o mais antigo manuscrito completo da Bíblia, todo em scriptio continua. Figura em domínio público em [Wikipedia 2017]*

A terceira publicação [Resplande et al. 2020] contribuiu com a criação de um *dataset* para a tarefa de SH em português. Além disso, utilizou um modelo GPT-2 *small* sem ajuste fino (*fine-tuning*), alcançando um desempenho de concordância próximo ao padrão-ouro de anotação humana, com um coeficiente Kappa de 0,898.

Os três artigos anteriores forneceram a base empírica para a construção do quarto artigo[Rodrigues et al. 2021], que atualmente representa o estado da arte na tarefa de SH, comprovado por [Kodali et al. 2022] em um novo *dataset* chamado Hashset.

5.2 Principais Contribuições

Os principais trabalhos anteriores que exploraram a segmentação de *hashtags* incluem Maddela et al. [Maddela, Xu e Preoţiuc-Pietro 2019], Doval et al. [Doval e Gómez-Rodríguez 2019] e Çelebi et al. [Celebi e Özgür 2016, Celebi e Özgür 2018]. Embora essas pesquisas tenham investigado o uso de modelos treinados especificamente para segmentação de *hashtags*, até onde se sabe, nenhum estudo avaliou o desempenho *zero-shot* de modelos *Transformer* pré-treinados nessa tarefa.

Neste trabalho, uma estrutura de segmentação de *hashtags* foi proposta com base

na combinação de dois modelos de linguagem pré-treinados, GPT-2 [Radford et al. 2018] e BERT [Devlin et al. 2019], empregando pesquisa de *beam search* e reclassificação (*reranking*). Foi demonstrado que essa abordagem atinge desempenho de estado da arte em conjuntos de dados de segmentação de *hashtags*, superando abordagens anteriores baseadas em engenharia de características e modelos treinados especificamente para esse domínio.

As principais contribuições deste estudo são:

- **Foi apresentada a primeira abordagem *zero-shot* baseada em Transformers para segmentação de *hashtags***, com a obtenção de resultados de estado da arte no conjunto de dados TEST-BOUN [Celebi e Özgür 2016]. Até onde se sabe, esta é a primeira investigação sobre a aplicação do GPT-2 [Radford et al. 2018] a uma tarefa de segmentação de palavras, além de ser a primeira aplicação de modelos *Transformer* para segmentação de *hashtags*.
- **Foi desenvolvido um método *zero-shot* para segmentação de *hashtags* multilíngue**, possibilitando a integração transparente em *pipelines* de PLN, como análise de sentimento em mídias sociais.
- **Foi disponibilizada a implementação como código aberto**, garantindo a reprodutibilidade dos experimentos e facilitando a adoção da segmentação de *hashtags* em aplicações reais.

Com este estudo, foi demonstrado que modelos de linguagem de propósito geral podem ser eficazes para segmentação de *hashtags* multilíngue sem a necessidade de treinamento supervisionado.

5.3 Solução Proposta

Com base na definição formal e na motivação conceitual do padrão Recrutador–Selecionador (Seção 4.5), esta seção descreve sua implementação na tarefa de segmentação de *hashtags*. A solução proposta consiste em um *framework* composto por dois módulos principais — *Segmentador* e *Reordenador* — que operam de forma encadeada para gerar, pontuar e reordenar candidatos, conforme os princípios de decomposição modular da tarefa.

O framework desenvolvido é composto por dois módulos: Segmentador (*Segmenter*) e Reordenador (*Re-ranker*). O papel do *Segmentador* é gerar uma lista de candidatos alvo C_i a partir da *hashtag* h e produzir uma pontuação s_i para cada um dos candidatos. Em cada candidato C_i, um ou vários caracteres delimitadores podem ser inseridos em diferentes posições em h. No *pipeline*, o *Reordenador* recebe essa lista de candidatos e gera uma nova pontuação s'_i para cada um deles. Como última operação do

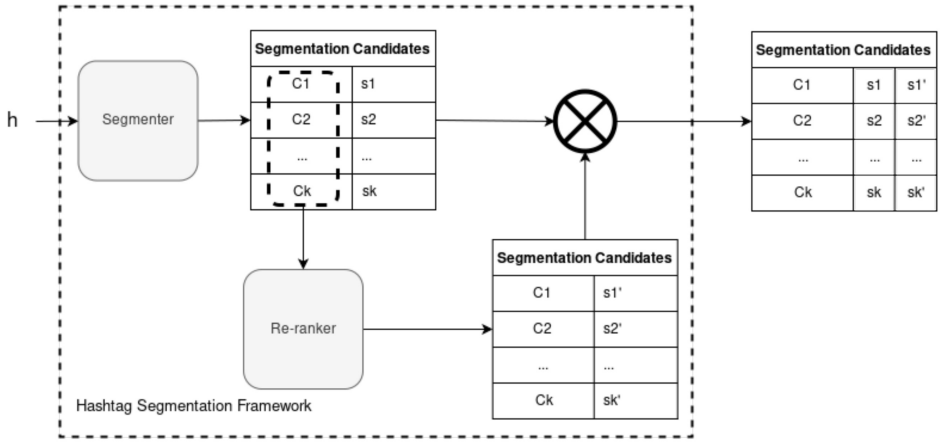


Figura 5.2: *Nosso framework proposto. Dada uma hashtag h, um módulo Segmenter propõe candidatos de segmentação C_i com escores s_i, que são reordenados pelo Re-ranker, que calcula novos escores s'_i para cada C_i.*

pipeline, as listas produzidas pelos dois módulos anteriores são combinadas, gerando uma lista de candidatos C_i associados às pontuações s_i e s'_i, provenientes do *Segmentador* e do *Reordenador*, respectivamente.

Nesta seção, nosso *framework* proposto é explicado de maneira modular, do nível mais alto ao mais baixo de abstração. Inicialmente, analisamos apenas a entrada e saída de cada módulo, como pode ser visto na Figura 5.2. No nível mais alto de abstração, o *framework* recebe uma *hashtag* h e produz uma lista de k candidatos segmentados, cada um com duas pontuações: s_i e s'_i, onde $1 \leq i \leq k$. Os detalhes das entradas e saídas de cada módulo são explicados a seguir.

Arquitetura

A arquitetura utilizada na solução possui dois módulos principais: **Segmentador** e **Reordenador**, que segue o padrão *Recrutador-Selecionador* (vide Seção 4.5). O *Segmentador* ou *Recrutador* tem o papel de gerar uma lista de candidatos segmentados C_i a partir da *hashtag* h e atribuir uma pontuação s_i a cada um dos candidatos. Cada candidato C_i consiste na adição de um ou mais caracteres delimitadores a h. Na sequência do *pipeline*, o *Reordenador* ou *Selecionador* recebe essa lista de candidatos e calcula uma nova pontuação s'_i para cada um deles. A última operação do *pipeline* combina as listas produzidas pelos dois módulos anteriores, gerando uma lista de candidatos C_i, cada um associado às pontuações s_i e s'_i, provenientes do *Segmentador* e do *Reordenador*, respectivamente.

Segmentador

Nossa abordagem é baseada principalmente em modelagem de linguagem e busca em feixe (*beam search*). Um modelo de linguagem aprende uma distribuição de probabilidade sobre sequências de texto $Y = (y_1, \cdots, y_N)$ de um determinado comprimento finito N, onde cada y_i é uma palavra ou token. Essa distribuição é comumente fatorada de forma *auto-regressiva*, conforme a equação abaixo:

$$P(Y;\theta) = \prod_{i=1}^{N} P(y_i|y_{<i};\theta), \tag{5-1}$$

onde θ representa os parâmetros do modelo.

A inferência com este modelo pode ser realizada com um algoritmo de *busca gulosa* (*greedy search*), no qual se seleciona a palavra y_i de maior probabilidade a cada passo, dada a sequência de palavras anteriores $y_{<i}$. A busca em feixe é um algoritmo heurístico padrão para decodificação de sequências em modelos como o modelo de linguagem autoregressivo da Eq. 5-1. Ela expande a busca gulosa com um *feixe* de tamanho k, onde, a cada passo i, são computadas as k palavras y_i de maior probabilidade e armazenadas em feixes distintos. Para cada sequência candidata, repete-se o processo para o próximo passo temporal $i+1$, gerando k^2 candidatos, dos quais apenas os k de maior probabilidade são mantidos.

Uma das principais vantagens de tratar a segmentação de palavras como um problema de modelagem de linguagem é a possibilidade de transferência de aprendizado *zero-shot* [Radford et al. 2019], permitindo o uso de um modelo de linguagem pré-treinado sem necessidade de refinamento para a tarefa específica.

Algoritmo HSBS

A busca em feixe pode ser vista como um algoritmo guloso que constrói uma árvore de busca de forma ampla-primeiro (*breadth-first*), expandindo apenas os nós com melhor pontuação segundo uma função de custo. Descrições detalhadas de como esse algoritmo pode ser aplicado à segmentação de palavras foram apresentadas por Doval e Gómez-Rodríguez [Doval e Gómez-Rodríguez 2019] e Zhang e Clark [Zhang e Clark 2011]. Baseamo-nos nessas ideias gerais, mas adotamos um método de expansão diferente e incorporamos uma nova função de custo para poda de ramos na árvore.

Para descrever nosso algoritmo de busca em feixe para segmentação de *hashtags*, definimos as seguintes estruturas de dados:

- $H = \langle c_1, c_2, \ldots, c_n \rangle$ representa uma *hashtag* com n caracteres c_i, onde $n \geq 2$.

- $S = \langle c_1, d_1, c_2, d_2, \ldots, c_{n-1}, d_{n-1}, c_n \rangle$ representa uma segmentação de H, onde $d_i = \varepsilon$ indica ausência de delimitador e $d_i = \square$ representa um delimitador. Para referenciar uma posição j em S, usamos S_j, onde $1 \leq j \leq (2 \times n) - 1$.
- $T = \langle S_1, S_2, \ldots, S_j \rangle$ é uma árvore de candidatos de segmentação.
- $D = \{\langle S_1, s_1 \rangle, \langle S_2, s_2 \rangle, \ldots, \langle S_j, s_j \rangle\}$ é um dicionário de candidatos segmentados e suas respectivas pontuações s_i.

Definimos também cinco funções que operam sobre essas estruturas:

- $Gerar(H)$: gera S a partir de uma *hashtag* H, inicialmente sem delimitadores.
- $Comprimento(S)$: retorna o tamanho de S considerando todos os caracteres.
- $ContarEspacos(S)$: retorna o número de delimitadores $\square$ em S.
- $Pontuar(T)$: calcula a pontuação s_i de cada nó S_i em T.
- $Selecionar(D, top_k)$: seleciona os top_k melhores candidatos com base em D.

Algoritmo 5.1: *Hashtag* Segmentation Beam Search (HSBS)

Entrada: H – *Hashtag* a ser segmentada;
e – Número de expansões;
top_k – Número máximo de segmentos mantidos.

Função `HSBS`(H, e, top_k)**:**

- $D \leftarrow \varnothing$; `// Conjunto de segmentos`
- $S \leftarrow$ `Gerar`(H) ; `// Segmentação inicial`
- $T \leftarrow \{S\}$; `// Conjunto de candidatos`
- **para** $t \leftarrow 1$ ***até*** e **faça**
  - $T \leftarrow$ `Expandir`(T, t);
  - $D \leftarrow$ `Pontuar`(T);
  - $T \leftarrow$ `Podar`(D, top_k);
- $D \leftarrow$ `Pontuar`(T);
- **retorne** D;

Função `Expandir`(T, t)**:**

- $T_{exp} \leftarrow \varnothing$; `// Conjunto expandido`
- **para** ***cada*** $S \in T$ **faça**
  - **se** `ContarEspacos`$(S) \geq t - 1$ **então**
    - $l \leftarrow$ `Comprimento`(S);
    - $j \leftarrow 2$;
    - **enquanto** $j < l$ **faça**
      - **se** $S_j \neq \square$ **então**
        - $S_j \leftarrow \square$;
        - $T_{exp} \leftarrow T_{exp} \cup \{S\}$;
      - $j \leftarrow j + 2$;
- **retorne** T_{exp};

Função `Podar`(D, top_k)**:**

- $T_{pruned} \leftarrow \varnothing$; `// Conjunto podado`
- $T_{topk} \leftarrow$ `Selecionar`(D, top_k);
- **para** ***cada*** $S \in T_{topk}$ **faça**
  - $T_{pruned} \leftarrow T_{pruned} \cup \{S\}$;
- **retorne** T_{pruned};

O algoritmo de busca em feixe para segmentação de *hashtags* (*HSBS*) é apresentado no Algoritmo 5.1. Em resumo, o pseudocódigo consiste em três etapas: (1) inicializar a árvore de segmentação T com a *hashtag* H e expandi-la, (2) calcular as pontuações de todos os nós da árvore e (3) manter apenas os top_k melhores candidatos para a próxima

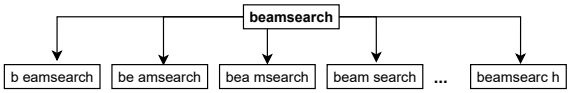
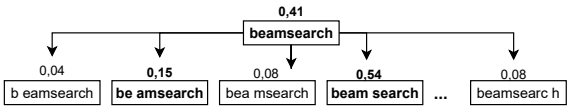
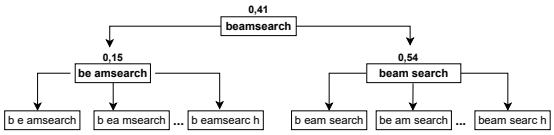
Pseudocódigo	Simulacão
$t = 1$ $T_{expanded} = Expandir(T)$	
$D = Pontuar(T_{expanded})$	
$T_{pruned} = Podar(D,3)$ $t = 2$ $T_{expanded}$ $Expandir(T_{pruned})$	= 

Tabela 5.1: *Simulação do algoritmo de busca em feixe em segmentação de hashtags (HSBS).*

iteração.

A Tabela 5.1 ilustra a execução do Algoritmo 5.1 e apresenta três etapas: na primeira etapa (iteração $t = 1$), a *hashtag 'beamsearch'* é expandida em 9 candidatos (*'b eamsearch'*, *'be amssearch'*, $\cdots$, *'beamsearc h'*); na segunda etapa, geramos o dicionário D calculando a pontuação de cada candidato; por fim, os três melhores candidatos (*'beamsearch'*, *'be amsearch'* e *'beam search'*) são selecionados com pontuações de 0,41, 0,15 e 0,54, respectivamente. Na iteração $t = 2$, os nós folha *'be amsearch'* e *'beam search'* são expandidos.

Reordenador

O *Reordenador* recebe a lista de candidatos gerada pelo *segmentador*, contendo as top_k melhores segmentações selecionadas pelo algoritmo de busca em feixe. Em seguida, ele atribui uma nova pontuação a cada candidato. Em nossa implementação, utilizamos o modelo BERT [Devlin et al. 2018] para realizar essa reordenação, por meio do método de pontuação de linguagem mascarada (*masked language model scoring*), conforme definido por Salazar et al. [Salazar et al. 2020].

Um ponto importante a ser observado é a relevância deste módulo na arquitetura da solução. O *segmentador* emprega modelos de linguagem ***autoregressivos***, cujas pontuações são computadas pela biblioteca `lm-scorer`[1] e utilizadas na função `Pontuar` (vide Algoritmo HSBS 5.1). Como mostra a Equação 5-1, nesses modelos a probabilidade de cada *token* depende apenas dos anteriores, o que pode introduzir um viés direcional da esquerda para a direita (*left-to-right bias*) [Salazar et al. 2020].

[1] Disponível em: https://github.com/simonepri/lm-scorer

Já no *reordenador*, adotamos um modelo ***autoencoder*** (ou bidirecional), que utiliza uma máscara durante o treinamento para prever cada *token* com base em seu contexto completo – tanto à esquerda quanto à direita. Essa abordagem caracteriza os modelos da família *BERT-like*, cuja técnica foi originalmente apresentada por Devlin et al. [Devlin et al. 2019]. Em contraste, modelos *autoregressivos*, como os da família *GPT-like* [Radford et al. 2018], são unidirecionais, prevendo o próximo *token* a partir dos anteriores.

A decisão de utilizar um modelo *autoregressivo* no *segmentador* foi fundamentada em evidências empíricas, especificamente na melhoria de desempenho observada ao substituir o BERT pelo GPT-2 no terceiro trabalho, conforme relatado na Seção 5.1.1. Por outro lado, a adoção de um modelo *autoencoder* no *reordenador* foi motivada pela observação de que, embora o candidato correto frequentemente estivesse entre os 10 primeiros retornados pelo *segmentador*, nem sempre era classificado como o melhor.

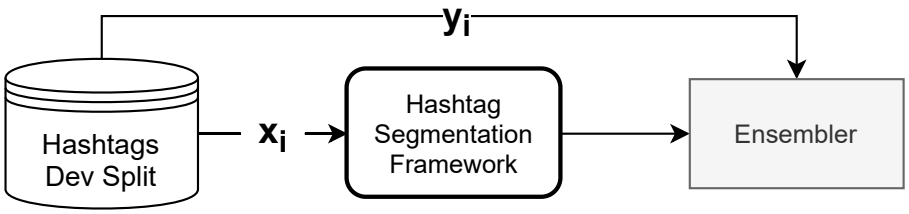
Essa limitação foi mitigada com a técnica de pontuação proposta por Salazar et al. [Salazar et al. 2020], implementada na ferramenta `mlm-scoring`[2]. A pontuação baseada em *pseudo-log-likelihood* (PLL) mostrou-se superior à baseada em modelos *GPT-like*, especialmente em tarefas como reconhecimento automático de fala (ASR) e tradução automática neural (NMT). Segundo os autores, essa vantagem decorre da maior capacidade dos modelos *BERT-like* em capturar a aceitabilidade linguística, o que foi evidenciado por experimentos no conjunto BLiMP [Warstadt et al. 2023], projetado especificamente para avaliar esse aspecto.

A atuação do *reordenador* aprimora significativamente a qualidade da seleção de candidatos. No entanto, observou-se que em determinados casos os modelos de linguagem subjacentes aos módulos *segmentador* e *reordenador* apresentavam julgamentos conflitantes quanto à qualidade das segmentações. Esse cenário motivou a criação de um módulo adicional, responsável por combinar as saídas desses dois componentes, com o objetivo de melhorar a precisão final da solução. A próxima seção descreve esse módulo: o *Combinador*.

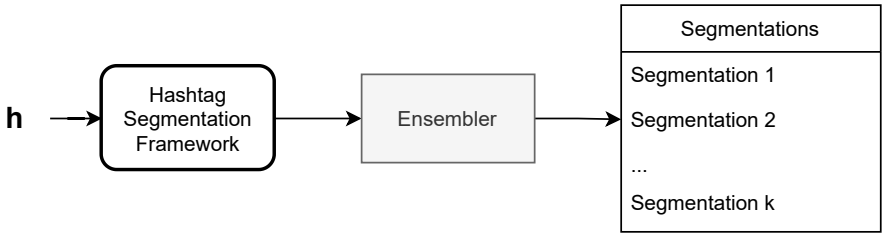
Combinador (*Ensembler*)

Um combinador (*ensembler*) básico foi implementado como um módulo destacável, aplicado ao final da *pipeline* do *framework*. Esse componente tem como objetivo combinar os *rankings* fornecidos pelo segmentador e pelo reordenador em um único *ranking* final. O *combinador* não foi integrado ao núcleo do *framework*, uma vez que diversas abordagens podem ser utilizadas para a combinação dos *rankings*. Idealmente, múltiplas

[2]Disponível em: https://github.com/awslabs/mlm-scoring



(a) Treinamento do Ensembler



(b) *Pipeline* de produção

Figura 5.3: *No treinamento do Ensembler (a), x_i e y_i referem-se a uma hashtag e sua segmentação a partir de uma instância i do conjunto Hashtag Dev Split. No pipeline de produção (b), uma hashtag h pode ser segmentada em k candidatos, onde $k \geq 1$.*

estratégias de combinação devem ser consideradas no tratamento do problema de segmentação de *hashtags*.

Nos experimentos apresentados neste artigo, foi utilizado um *combinador* simples como linha de base. Dado os dois principais candidatos selecionados pelo *segmentador*, c_1 e c_2, e as funções de pontuação f_S e f_R associadas ao *segmentador* e ao *reordenador*, respectivamente, a função de decisão f_E, que caracteriza o *combinador*, é definida da seguinte forma:

$$f_E(c_1, c_2) = \alpha \cdot |f_S(c_1) - f_S(c_2)| - \beta \cdot |f_R(c_1) - f_R(c_2)| \tag{5-2}$$

A função de decisão produz o ranking final considerando apenas dois candidatos. Para qualquer *hashtag* e suas segmentações candidatas, caso f_E seja positivo, o *ranking* do combinador será composto pelos dois principais candidatos selecionados pelo *segmentador*. Caso contrário, os mesmos candidatos serão considerados, porém ordenados conforme definido pelo reordenador.

Os parâmetros α e β correspondem a hiperparâmetros responsáveis por ponderar as diferenças absolutas entre as pontuações dos candidatos. Ambos são representados por números reais no intervalo $[0, 1]$ e são otimizados por meio de busca em grade (*grid search*) sobre um conjunto de desenvolvimento (conforme ilustrado na Figura 5.3).

Durante a busca em grade, os valores de α e β são selecionados de forma a maximizar o *F-score* obtido no conjunto de desenvolvimento. Após a otimização, o

combinador é considerado pronto para ser integrado a um *pipeline* de produção, atuando como módulo responsável por combinar as saídas dos componentes *segmentador* e *reordenador* do *framework* de segmentação de *hashtags*.

Aplicação do Padrão Recrutador–Selecionador na Arquitetura

O *framework* proposto para segmentação de *hashtags* é uma implementação concreta do padrão arquitetural **Recrutador–Selecionador** (vide Seção 4.5). Nesse padrão, uma tarefa é decomposta em duas etapas principais: geração de candidatos (recrutamento) e seleção dos melhores candidatos (seleção). Essa separação permite modularizar o raciocínio computacional em fases distintas, facilitando tanto a manutenção quanto a extensão do sistema.

No contexto da segmentação de *hashtags*, o papel do **Recrutador** é desempenhado pelo módulo *Segmentador*. Esse componente é responsável por explorar sistematicamente as possíveis segmentações de uma *hashtag* h por meio do algoritmo *Hashtag Segmentation Beam Search* (HSBS), descrito na Seção 5.3. Cada segmentação candidata C_i é gerada com base na inserção de delimitadores em diferentes posições da sequência de caracteres da *hashtag*. Essas alternativas são pontuadas com base em sua probabilidade segundo um modelo de linguagem, produzindo assim uma lista de pares $\langle C_i, s_i \rangle$.

O papel do **Selecionador** é então assumido pelo módulo *Reordenador*, que recebe os k melhores candidatos do *Segmentador* e os reavalia utilizando um modelo de linguagem contextualizado (BERT), empregando a técnica de *masked language model scoring* (vide Seção 5.3). Essa reavaliação resulta em novas pontuações s'_i, que complementam os escores originais atribuídos pelo recrutador.

Por fim, o *pipeline* do *framework* integra esses dois momentos — geração e seleção — de forma encadeada, produzindo uma lista final de candidatos associados às pontuações s_i e s'_i. A flexibilidade do padrão Recrutador–Selecionador é ainda evidenciada na inclusão de um módulo opcional de combinação de pontuações (*combinador*), que pode aplicar funções de decisão sobre s_i e s'_i para definir o *ranking* final.

Assim, a arquitetura do *framework* implementa de forma explícita e funcional o padrão Recrutador–Selecionador, promovendo modularidade, reaproveitamento e clareza na decomposição da tarefa de segmentação de *hashtags*.

5.4 Experimentos e Resultados

Dois modelos de linguagem pré-treinados e publicamente disponíveis foram comparados: o BERT (*bert-large-uncased-whole-word-masking*) e o GPT-2 (*gpt2-large*). A análise teve como objetivo avaliar o desempenho desses modelos quando empregados como segmentadores em um cenário estritamente *zero-shot*, ou seja, sem qual-

	Test-STAN						Test-BOUN					
	[Celebi e Özgür 2018]		GPT-2		BERT		[Celebi e Özgür 2018]		GPT-2		BERT	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
N = 1	**82.9**	**80.4**	72.2	75.9	43.1	41.9	**93.2**	**90.0**	89.9	85.2	62.3	57.1
N = 2	**92.9**	**91.6**	90.7	90.2	47.8	49.6	96.2	94.4	**97.9**	**97.0**	47.7	49.5
N = 5	94.4	93.2	**97.4**	**97.6**	55.4	60.6	96.6	94.8	**99.7**	**99.6**	75.0	75.6
N = 10	94.4	93.2	**98.8**	**99.1**	62.9	69.3	96.6	94.8	**99.7**	**99.6**	79.2	80.6

Tabela 5.2: *Avaliação do tipo oracle em dois conjuntos de dados de segmentação de hashtags: Test-STAN e Test-BOUN, onde um resultado é considerado correto se a segmentação de referência (gold standard) aparecer entre as N segmentações com maior pontuação ($N = \{1,2,5,10\}$). Três modelos de segmentação de hashtags foram comparados: o modelo proposto por [Celebi e Özgür 2018], GPT-2 e BERT — sendo os dois últimos modelos de linguagem pré-treinados usados em modo zero-shot estrito, sem qualquer re-treinamento ou ajuste. O melhor entre os três modelos em cada conjunto de teste foram destacados em negrito.*

quer reprocessamento ou re-treinamento. Para tanto, um algoritmo de busca em feixe (*beam search*) foi implementado e aplicado aos conjuntos de dados apresentados por [Celebi e Özgür 2018].

Experimentos com o Segmentador

A fim de garantir comparabilidade com estudos anteriores, foi seguido integralmente o procedimento de avaliação proposto por [Celebi e Özgür 2018], considerando-se um resultado como correto quando a segmentação de referência (padrão-ouro) encontrava-se entre as N segmentações com maior pontuação. Foram comparados os valores de *F-score* e acurácia correspondentes às N melhores segmentações geradas por cada modelo.

Conforme evidenciado na Tabela 5.2, o BERT revelou-se ineficaz como segmentador, uma vez que, mesmo ao se considerar os 10 principais candidatos, os resultados obtidos não superaram os alcançados por trabalhos anteriores [Celebi e Özgür 2018] nos conjuntos *Test-STAN* e *Test-BOUN*.

Embora o GPT-2 não tenha superado o modelo de linguagem proposto por [Celebi e Özgür 2018] quando apenas o candidato com maior pontuação foi considerado, um desempenho significativamente superior foi observado quando múltiplos candidatos foram incluídos na avaliação, posicionando o GPT-2 como o modelo de melhor desempenho entre todos os analisados (ver Tabela 5.2). De acordo com os experimentos conduzidos por [Maddela, Xu e Preoţiuc-Pietro 2019], os resultados do GPT-2 considerando

Conjunto de Dados	Arquitetura	Não Supervisionado?	F-1	Acurácia
Test-Stanford	Microsoft Word Breaker [Maddela, Xu e Preoţiuc-Pietro 2019]	Não	84.6	83.6
	Çelebi et al. [Celebi e Özgür 2018]	Não	82.9	80.4
	Çelebi et al. + engenharia de atributos	Não	**90.2**	88.5
	Maddela et al. + engenharia de atributos	Não	89.8	**91.0**
	Segmentador (GPT-2) → Reordenador (BERT), $\alpha = 0.0, \beta = 1.0$	Sim	51.9	45.2
	Segmentador (GPT-2) → Reordenador (BERT), $\alpha = 0.2, \beta = 0.1$	Sim	85.7	84.3
Test-BOUN	Microsoft Word Breaker [Celebi e Özgür 2018]	Não	84.4	86.2
	Çelebi et al. [Celebi e Özgür 2018]	Não	93.2	90.0
	Çelebi et al. + engenharia de atributos	Não	94.9	92.9
	Segmentador (GPT-2) → Reordenador (BERT), $\alpha = 0.0, \beta = 1.0$	Sim	72.7	62.3
	Segmentador (GPT-2) → Reordenador (BERT), $\alpha = 0.2, \beta = 0.1$	Sim	**95.6**	**93.4**

Tabela 5.3: *As pontuações F1 e de acurácia obtidas pela nossa estrutura foram avaliadas nos conjuntos de dados Test-Stanford [Celebi e Özgür 2016] e Test-BOUN [Celebi e Özgür 2016]. Resultados anteriores reportados na literatura foram utilizados para fins de comparação. A busca pelos melhores hiperparâmetros foi realizada no conjunto de validação ($\alpha = 0.2, \beta = 0.1$), e o uso do BERT para reordenação foi explorado de forma não supervisionada ($\alpha = 0.0, \beta = 1.0$). A baseline Microsoft Word Breaker foi superada no Test-Stanford, e novos resultados estado-da-arte foram alcançados no conjunto de dados Test-BOUN.*

os 10 principais candidatos aproximam-se do desempenho humano. De modo geral, ao se comparar os resultados de [Celebi e Özgür 2018] com os obtidos pelo GPT-2, observa-se um aumento no *F-score* de 94,4% para 98,8% no *Test-STAN* e de 96,6% para 99,7% no *Test-BOUN* ($N = 10$).

Experimentos com o Reordenador

Tendo sido demonstrada a eficácia do GPT-2 como segmentador na Seção 5.4, os esforços subsequentes foram direcionados ao reordenamento de seus candidatos. Na Tabela 5.3, são descritos os experimentos conduzidos com duas abordagens distintas para a reordenação dos dois principais candidatos selecionados pelo GPT-2. Em ambas as abordagens, a decisão entre a ordem original atribuída pelo GPT-2 e a nova ordenação proposta pelo BERT é realizada com base nas pontuações dos candidatos e nos pesos α e β.

A primeira abordagem consiste em uma linha de base na qual se assume, de forma sistemática, a confiabilidade do reranqueamento proposto pelo BERT. No módulo de combinação (*Ensembler*), tal estratégia pode ser implementada pela atribuição de $\alpha = 0$ e $\beta = 1$, ou ainda por qualquer outro valor arbitrário positivo para β.

Na segunda abordagem, os valores de α e β são determinados por meio de busca em grade (*grid search*) aplicada ao conjunto de desenvolvimento de cada um dos *datasets*, a saber, *Dev-Stanford* e *Dev-BOUN*.

Os resultados apresentados na Tabela 5.3 indicam que a adoção irrestrita da reordenação fornecida pelo BERT resulta em desempenho inferior até mesmo ao *baseline* padrão do Microsoft Word Breaker. Por outro lado, quando os pesos α e β são ajustados com base nos dados de desenvolvimento, observa-se que o *framework* proposto supera o *baseline* do Word Breaker no conjunto de teste *Test-Stanford*, sendo superado apenas por soluções que empregam engenharia de atributos (*feature engineering*).

Por fim, destaca-se que, no conjunto de teste *Test-BOUN*, é atingido um novo estado da arte, mesmo na ausência de qualquer forma de engenharia de atributos (vide Tabela 5.3), superando-se, inclusive, a combinação anterior de modelo de linguagem e engenharia de atributos proposta por [Celebi e Özgür 2018].

Resultados no Hashformer com o *Dataset* HashSet

Com o objetivo de avaliar a robustez e a capacidade de generalização da arquitetura *Hashformer*, foram realizados experimentos utilizando o *dataset HashSet*. Esse *dataset* foi desenvolvido de forma independente por terceiros, conforme apresentado no trabalho de [Kodali et al. 2022]. O *HashSet* se destaca por apresentar maior diversidade linguística e complexidade semântica em comparação aos *datasets* tradicionalmente utilizados para a tarefa de segmentação de *hashtags*, como *STAN* e *BOUN*.

O *HashSet* é composto por duas partes: *HashSet-Manual*, contendo 1.901 *hashtags* anotadas manualmente, e *HashSet-Distant*, com mais de 330 mil *hashtags* segmentadas automaticamente por regras baseadas em *camel case*. Os experimentos foram conduzidos utilizando o subconjunto manual, que representa um desafio significativo por conter maior proporção de entidades nomeadas (74,4%) e tokens não ingleses (12,4%), além de *hashtags* mais longas e com maior número médio de segmentos.

A ferramenta *hashformer* analisada nesta tese foi confirmada como o estado da arte (SOTA) consistentemente em quase todos *benchmarks* de segmentação de *hashtags*, como *STAN-Dev*, *STAN-Largee* e *BOUN*, superando modelos anteriores como o *Multi-task Pairwise Neural Ranking* (MPNR). A única exceção foi no *dataset Stan-Small*, em poucos pontos percentuais (1% a 3%), conforme pode ser visto na Tabela 5.4. Nesses *datasets*, o *Hashformer* apresentou os melhores desempenhos de acurácia em múltiplos valores de *top-n*, demonstrando sua eficácia em ambientes controlados e amplamente utilizados pela comunidade de PLN.

Arquitetura	Conjunto de Dados	n=1	n=2	n=5	n=7	n=9	n=10
MPNR	BOUN	81,60	88,09	90,29	90,69	90,69	90,69
	STAN-Dev	73,12	78,16	81,92	82,71	82,81	82,81
	STAN-Small	82,76	86,19	86,82	86,82	86,82	86,82
	STAN-Large	63,78	73,10	74,73	74,75	74,75	74,75
	HashSet-Manual	41,93	45,98	47,50	47,71	47,71	47,71
Hashformer	BOUN	83,68	87,69	91,39	99,00	99,30	99,30
	STAN-Dev	80,04	84,49	90,02	98,72	99,51	99,60
	STAN-Small	80,05	85,11	88,90	97,11	97,38	97,38
	STAN-Large	72,17	75,74	79,25	85,38	85,82	85,86
	HashSet-Manual	56,71	68,54	78,22	91,53	94,00	94,37

Tabela 5.4: *Desempenho do MPNR e do Hashformer em diferentes conjuntos de dados, em termos de acurácia top-n*

No entanto, ao ser aplicado ao *HashSet-Manual*, o *Hashformer* apresentou desempenho inferior, com 56,71% de acurácia para *top-1*, em comparação a 80,05% em *STAN-Small* e 83,68% em *BOUN*. A análise de erros revelou que mais de 77% das *hashtags* incorretamente segmentadas continham entidades nomeadas, sugerindo que esse fator impacta negativamente a segmentação.

Em resumo, embora o *Hashformer* apresente desempenho SOTA em benchmarks tradicionais, sua eficácia diminui diante da complexidade do *HashSet*, evidenciando a necessidade de abordagens mais robustas e sensíveis à variabilidade linguística presente em ambientes reais.

5.5 Conclusão

Este capítulo apresentou um estudo de caso centrado na tarefa de segmentação de *hashtags*, cujo principal objetivo foi investigar a aplicabilidade do padrão arquitetural Recrutador–Selecionador, conforme definido na Seção 4.5. A decomposição da tarefa em duas etapas principais — geração e seleção de candidatos — foi operacionalizada por meio de três módulos interdependentes: *segmentador*, *reordenador* e *combinador*. Essa organização modular permitiu explicitar, de forma clara, os critérios adotados em cada fase do processo, favorecendo a reusabilidade e a extensibilidade da solução proposta.

A arquitetura desenvolvida integra modelos de linguagem com diferentes características: o GPT-2, de natureza autoregressiva, foi empregado na geração inicial de candidatos; o BERT, com arquitetura bidirecional, foi utilizado para reavaliação das segmentações; e um módulo combinador foi responsável por arbitrar entre os rankings produ-

zidos por ambos. Essa composição permitiu o uso de estratégias de inferência *zero-shot*, sem ajuste fino ou supervisão direta, evidenciando a robustez do padrão em contextos de baixa disponibilidade de dados anotados.

Os resultados experimentais indicaram que a solução proposta supera abordagens anteriores em diversos *benchmarks* clássicos, alcançando resultados de estado da arte no conjunto Test-BOUN. Adicionalmente, sua aplicação ao conjunto HashSet permitiu avaliar os limites da abordagem frente à maior diversidade linguística e presença de entidades nomeadas, apontando oportunidades para investigação futura, como a integração de conhecimento externo ou mecanismos adaptativos.

De modo geral, este estudo contribui para a validação prática do padrão Recrutador–Selecionador, demonstrando seu potencial para guiar a decomposição arquitetural de tarefas de PLN em cenários reais. A estrutura modular adotada favorece não apenas a explicitação do raciocínio computacional envolvido, mas também a avaliação sistemática de variações no processo de geração e seleção, em conformidade com os princípios delineados na Seção 4.5.

CAPÍTULO 6

Curadoria de Frases-Chave

Neste capítulo, é apresentado um caso de uso para uma tarefa inédita de PLN, denominada *curadoria de frases-chave*. Especificamente, é explorada a aplicação do padrão *recrutador-selecionador*, proposto na Seção 4.5, no contexto dessa tarefa.

A seguir, é apresentado um resumo de cada uma das seções subsequentes:

- **6.1 Introdução**: são apresentadas a definição, o contexto de aplicação da tarefa de *curadoria de frases-chave*, as linhas gerais da metodologia utilizada, os resultados alcançados e as contribuições;
- **6.2 Qualidade de curadoria de frases-chave**: são apresentados critérios de avaliação de qualidade de frases-chave;
- **6.3 Aplicação do Padrão Recrutador-Selecionador**: é apresentada a forma como o padrão Recrutador-Selecionador foi aplicado no caso de uso;
- **6.4 Experimentos**: são descritas a montagem e a avaliação dos experimentos realizados;
- **6.5 Conclusão**: é realizada a análise e conclusão da aplicação do padrão Recrutador-Selecionador e da metodologia de decomposição de tarefas.

6.1 Introdução

Com base no padrão arquitetural Recrutador-Selecionador, proposto na Seção 4.5, esta seção apresenta sua aplicação prática no contexto da tarefa de *curadoria de frases-chave*, uma atividade inédita de PLN concebida com o objetivo de mitigar a baixa qualidade média das *keyphrases* geradas automaticamente. Essa tarefa é caracterizada por sua natureza altamente decomponível, sendo composta por duas etapas distintas: (i) extração automática de um grande volume de *keyphrases* e (ii) seleção manual ou supervisionada de um subconjunto de alta qualidade. Essa divisão é alinhada diretamente à estrutura proposta pelo padrão Recrutador-Selecionador, em que a etapa de recrutamento é realizada por ferramentas automatizadas e a etapa de seleção é conduzida por especialistas humanos ou por modelos orientados por curadoria.

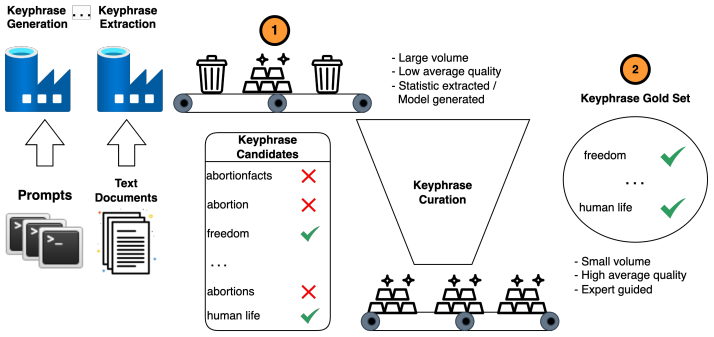


Figura 6.1: *Tarefa de Curadoria de Keyphrases: (1) são recebidas keyphrases de ferramentas de geração e extração automática, geralmente com qualidade média baixa; (2) como resultado, um pequeno conjunto de keyphrases de alta qualidade é produzido, com participação humana ou orientação supervisionada.*

A investigação da tarefa foi motivada pelo desafio observado em estudos anteriores relacionados à classificação de tópicos em segmentos de texto argumentativo. Nesses estudos, a geração de *keyphrases* foi tratada como uma etapa preliminar essencial, sendo detectado que, apesar da geração de centenas de *keyphrases*, a seleção de um subconjunto pequeno e de alta qualidade apresentava alto grau de complexidade e subjetividade. A necessidade de garantir concordância entre anotadores, evitar ambiguidades e reduzir a carga cognitiva tornou evidente a importância de se aplicar um mecanismo estruturado de curadoria, como o previsto no padrão proposto.

Para que uma análise abrangente de grandes volumes de textos argumentativos seja viabilizada, torna-se essencial uma exploração inicial das ideias centrais abordadas nesses conteúdos. Essa exploração pode ser facilitada pela aplicação de técnicas de extração automática de *keyphrases* (*AKE*), as quais permitem identificar termos recorrentes, jargões, possíveis vieses, agrupamentos temáticos e contrastes argumentativos. No entanto, como relatado por Hasan e Ng [Hasan e Ng 2014], os resultados de ferramentas de AKE frequentemente apresentam baixa qualidade média, devido a erros como *supergeneralização* e *redundância*. A curadoria, neste contexto, é introduzida como uma camada complementar orientada à seleção, com vistas à obtenção de um subconjunto mais confiável e útil para tarefas posteriores, como a anotação de tópicos.

Conforme ilustrado na Figura 6.1, a tarefa de curadoria de *keyphrases* foi concebida de modo a alinhar-se diretamente à arquitetura do padrão Recrutador-Selecionador: a entrada da tarefa consiste em um conjunto extenso de *keyphrases* geradas automaticamente (recrutamento), e a saída desejada é um conjunto reduzido e de alta qualidade, obtido por meio da intervenção humana qualificada (seleção). Essa divisão funcional permite a modularização da tarefa, o isolamento de erros por camada e a aplicação de métricas distintas de avaliação para cada etapa.

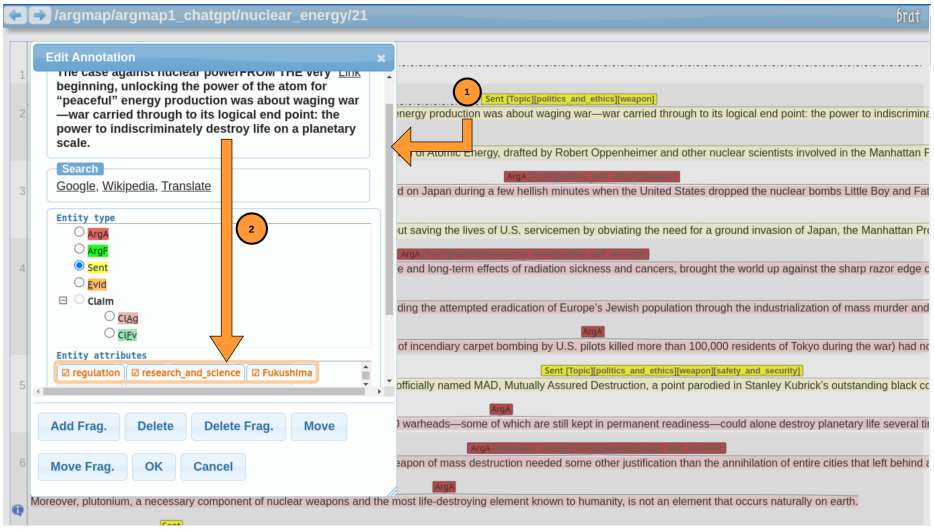


Figura 6.2: *Classificação de Tópicos (CT) na plataforma Brat: (1) um tópico frasal – que inicia o segmento de texto – é selecionado; (2) a sentença é anotada com três keyphrases: "regulation", "research and science" e "Fukushima", escolhidas a partir de um conjunto de keyphrases disponível.*

A aplicação do padrão também se estende à tarefa de categorização semântica de segmentos textuais, conforme apresentado na Figura 6.2. Neste caso, a curadoria de *keyphrases* é utilizada como insumo para a anotação de tópicos em textos argumentativos, tarefa que também pode ser beneficiada pela decomposição baseada no padrão Recrutador-Selecionador. Especificamente, cada segmento textual é anotado por meio da seleção de 1 a 3 *keyphrases* a partir de um conjunto previamente curado, assegurando consistência semântica e coerência temática entre os segmentos e os rótulos atribuídos.

A situação mencionada é abordada por meio da solução proposta, em que a decomposição da tarefa — orientada pelo padrão Recrutador-Selecionador — permite que a curadoria seja realizada com maior controle de qualidade, maior transparência no processo de anotação e possibilidade de escalabilidade por meio da automação parcial ou total das etapas de recrutamento e seleção.

6.2 Qualidade de curadoria de frases-chave

Dando continuidade à aplicação do padrão Recrutador-Selecionador, a etapa de seleção de frases-chave foi orientada por critérios rigorosos de qualidade. A literatura especializada tem oferecido uma variedade de critérios recomendados para que a qualidade de palavras-chave possa ser avaliada. Um compêndio dessas recomendações foi compilado por Firoozeh et al. [Firoozeh et al. 2020], tendo servido como base para a construção da diretriz de anotação adotada nesta tarefa.

Dez propriedades abstratas, denominadas **propriedades de saliência (keyness)**, foram delineadas para assegurar a qualidade de expressões-chave em seu contexto. Essas propriedades foram organizadas por Firoozeh em três categorias: informacionais — **exaustividade**, **especificidade**, **minimalidade**, **imparcialidade** e **representatividade**; morfológicas — **correção formal** (*well-formedness*) e **referencialidade** (*citationess*); e baseadas em domínio — **conformidade** (*conformity*), **homogeneidade** e **univocidade** (*univocity*).

Na adaptação dessas propriedades ao processo de curadoria proposto nesta tese, observou-se que elas poderiam ser reorganizadas segundo três níveis linguísticos: morfossintático, semântico e pragmático. A Tabela 6.1 apresenta cada uma das propriedades de saliência com suas respectivas definições, organizadas por nível.

No nível morfossintático, as propriedades de *correção formal* e *referencialidade* foram destacadas. A primeira garante que a ortografia, o espaçamento e a estrutura das expressões-chave estejam formalmente corretos. A segunda assegura que a lematização e a flexão estejam adequadamente ajustadas. Com base nessas propriedades, foi estruturada a primeira etapa da curadoria, na qual uma filtragem e adaptação morfossintática das expressões geradas automaticamente é realizada, visando estabilizá-las formalmente antes que se considerem seus significados ou contextos [Manning e Schütze 1999]. Para essa etapa, quatro critérios objetivos foram definidos: ortografia correta, estrutura sintática, tipo gramatical e preferência por formas no singular.

Nível	Propriedade de Saliência	Definição
Morfossintático	Correção Formal	Prevê a ortografia correta, o espaçamento e a qualidade formal geral das palavras-chave.
Morfossintático	Citação	Garante a lematização precisa e a adequada flexão e classificação das expressões-chave.
Semântico	Homogeneidade	Preferência observada entre seus sinônimos ou candidatos similares dentro do conjunto de expressões-chave.
Semântico	Minimalidade	Cada expressão-chave deve ter um significado único quando comparada a outras expressões-chave do conjunto.
Pragmático	Exaustividade	Os conjuntos de expressões-chave devem abranger todos os temas de importância informacional dentro do *corpus* textual.
Pragmático	Representatividade	Cada expressão-chave deve representar com precisão os principais aspectos dos textos no *corpus*.
Pragmático	Conformidade	Cada expressão-chave deve estar em conformidade com a terminologia ou jargão do seu domínio.
Pragmático	Univocidade	Cada expressão-chave deve ser inequívoca em relação a outros termos dentro do seu domínio.
Pragmático	Especificidade	Adequação das expressões-chave em relação ao domínio específico de seu conjunto de dados em oposição a outros domínios.
Pragmático	Imparcialidade	Ausência de vieses, impressões opinativas ou parcialidade nas expressões-chave.

Tabela 6.1: *Propriedades de saliência e suas definições por nível linguístico*

Durante o processo de curadoria, foi observada uma tensão recorrente entre as propriedades de *minimalidade* e *representatividade*. Essa relação inversa tornou impraticável a maximização simultânea de ambas. Frases-chave muito genéricas apresentaram representatividade excessiva, englobando outras expressões mais específicas. Por outro

lado, quando a minimalidade foi acentuada em demasia, as expressões resultantes se mostraram pouco relevantes, sendo frequentemente ignoradas durante a anotação. Concluiu-se, portanto, que o melhor desempenho da curadoria é alcançado por meio de um equilíbrio entre essas duas propriedades.

Por fim, as propriedades de *especificidade* e *imparcialidade*, também pertencentes ao nível pragmático, exigem atenção especial, pois sua adequação pode ultrapassar os limites do conjunto de dados originalmente considerado. Essas propriedades asseguram não apenas a qualidade local da anotação, mas também sua adaptabilidade e robustez frente a contextos discursivos variados, inclusive em domínios distintos ou em futuras aplicações da anotação.

6.3 Aplicação do Padrão Recrutador-Selecionador

Dando continuidade à operacionalização do padrão Recrutador-Selecionador, a tarefa de curadoria de frases-chave foi formalmente definida e estruturada em subtarefas, de forma a possibilitar que cada etapa fosse analisada individualmente, com controle e transparência sobre os resultados obtidos. Essa abordagem orientada a tarefas permite que os critérios de qualidade apresentados na seção anterior sejam sistematicamente aplicados em cada fase do processo.

A partir dos critérios qualitativos previamente descritos, uma diretriz de anotação foi seguida pelos anotadores, com o objetivo de assegurar a produção de anotações de alta qualidade. Em termos gerais, a tarefa de curadoria de frases-chave foi formalizada da seguinte maneira:

Definição 6.1 *Seja $K = \{k_1, k_2, \ldots, k_n\}$ um conjunto de n frases-chave, obtidas por extração automática ou indicadas por especialistas humanos. A tarefa de curadoria de frases-chave consiste na seleção de um subconjunto $S \subset K$, com $|S| = s$ e $s < n$, tal que cada elemento $k_i \in S$ satisfaça um conjunto de critérios de qualidade linguística, semântica e pragmática previamente definidos — incluindo correção formal, representatividade, especificidade, imparcialidade, entre outros. O objetivo da curadoria é maximizar a qualidade informacional do conjunto S, reduzindo redundâncias e assegurando cobertura temática adequada ao domínio em questão.*

A partir dessa definição, e com base em sucessivas reuniões com os anotadores especialistas, a tarefa foi decomposta em três subtarefas principais, conforme ilustrado na Figura 6.3.

- **Agrupamento de Frases-Chave (KC)** — Dada uma lista $N = [K_1, K_2, \ldots, K_n]$ composta por n frases-chave, são construídos subconjuntos $C = \{C_1, C_2, \ldots, C_c\}$, em que

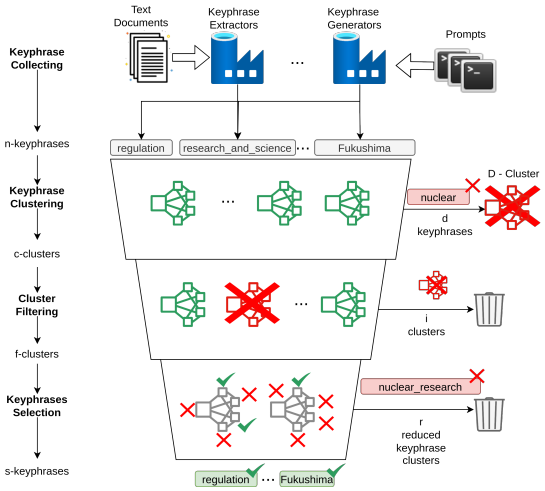


Figura 6.3: *Fluxo de subtarefas da curadoria de frases-chave*

cada C_i representa um agrupamento de frases-chave semanticamente semelhantes, com $0 < c < n$. Durante esse processo, frases malformadas são identificadas e descartadas, sendo alocadas em um agrupamento separado $D = \{K_1, K_2, \ldots, K_d\}$.

- **Filtragem de Agrupamentos (CF)** — Nessa etapa, um subconjunto $F \subseteq C$ contendo f agrupamentos é selecionado para prosseguir no fluxo de processamento, onde $0 < f < c$. Os agrupamentos restantes, considerados irrelevantes ou redundantes, são excluídos e reunidos em um subconjunto I. Assim, C é particionado em F (agrupamentos válidos) e I (agrupamentos ignorados).
- **Seleção de Frases-Chave (KS)** — Para cada agrupamento $C_i \in F$, são selecionadas entre 1 e x frases-chave representativas, formando-se conjuntos $S = \{S_1, S_2, \ldots, S_s\}$, com $S_i \subseteq C_i$. Essa seleção tem por objetivo isolar as expressões mais relevantes e informativas para representar cada agrupamento.

Cada subtarefa foi desenhada para que a aplicação dos critérios de qualidade pudesse ser realizada de forma modular. No nível de entrada, o parâmetro n deve ser suficientemente elevado para que a propriedade de *exaustividade* possa ser satisfeita. Durante a etapa de agrupamento, a *homogeneidade* semântica é priorizada, ao passo que, na filtragem, critérios como *relevância*, *minimalidade* e *conformidade com o domínio* são considerados para refinar os agrupamentos resultantes. Por fim, na etapa de seleção, aspectos como *correção formal* e *representatividade* tornam-se centrais para a escolha das melhores expressões.

Essa decomposição baseada em subtarefas reflete diretamente a lógica do padrão Recrutador-Selecionador: a geração de um conjunto amplo e ruidoso de frases (recrutamento) é seguida de uma filtragem seletiva, conduzida com base em critérios explícitos de qualidade (seleção). Dessa forma, obtém-se um fluxo orientado por tarefas, capaz de promover tanto controle quanto escalabilidade no processo de curadoria.

6.3.1 Avaliação

A avaliação da tarefa foi conduzida com base em uma abordagem comparativa entre métodos humanos e automatizados, utilizando um conjunto de dados de referência (*gold standard*) anotado por especialistas. Inicialmente, esse conjunto *gold* foi criado com base em múltiplas anotações manuais, permitindo que o grau de concordância entre os anotadores fosse avaliado para cada subtarefa.

Uma vez estabelecido o *gold standard*, os desempenhos dos métodos automatizados puderam ser avaliados de forma sistemática, por meio da comparação entre suas saídas e as anotações de referência. Além disso, uma análise extrínseca foi conduzida, permitindo que se observasse o impacto da curadoria de frases-chave em tarefas subsequentes. Com isso, não apenas a qualidade intrínseca das anotações foi avaliada, mas também sua utilidade em contextos aplicados.

6.4 Experimentos

Nesta seção, apresenta-se primeiramente a experiência de montagem e anotação do *corpus* de frases-chave, seguida pela descrição do processo de avaliação dos resultados. O ambiente experimental foi inicialmente configurado e os dados foram coletados por meio de ferramentas de extração e geração de frases-chave, conforme ilustrado no fluxo da Figura 6.3, na etapa denominada *Coleta de Frases-Chave*.

Em seguida, o conjunto de dados foi anotado por meio da execução das três subtarefas do fluxo: *Agrupamento de Frases-Chave (KC)*, *Filtragem de Agrupamentos (CF)* e *Seleção de Frases-Chave (KS)*.

Para avaliar os resultados dos experimentos de anotação, foram conduzidas três análises: (i) a Concordância entre anotadores (*Inter-Annotator Agreement (IAA)*), apresentada na Seção 6.4.3, foi mensurada para cada subtarefa; (ii) realizou-se uma comparação de desempenho entre três modelos de LLMs aplicados ao conjunto de dados *gold*; e (iii) conduziu-se uma avaliação extrínseca comparando a curadoria humana com métodos automáticos na tarefa de classificação de tópicos.

6.4.1 Preparação dos Experimentos

Para a coleta da lista *N* de frases-chave, foram testadas cinco ferramentas: **Rake**, **Spacy/Textrank**, **Yake (Y)**, **Keybert (K)** e **ChatGPT**. A última é uma ferramenta de geração de frases-chave, e os prompts utilizados para gerar 60 frases-chave estão listados na Tabela 6.2, divididas em três grupos de 20 frases cada: argumentos *a favor* (P) e *contra* (A) um determinado *tópico*, além das principais *entidades nomeadas* (N) citadas nos

argumentos sobre o mesmo tópico. As demais ferramentas realizam extração automática de frases-chave, processando os textos e retornando as expressões mais relevantes.

Após análise empírica e aplicação de critérios de controle de qualidade, foram selecionadas três ferramentas para compor os conjuntos de frases-chave: **ChatGPT**, **Yake** e **Keybert**.

Para alimentar o *Yake* e o *Keybert* com textos para extração de frases-chave, foram utilizados 400 documentos textuais baixados e pré-processados a partir das URLs do conjunto de dados *UKP Sentential* [Stab e Gurevych 2014]. Esse *dataset* está organizado em 8 tópicos controversos, com 50 documentos cada. Para cada tópico, foram extraídas 40 frases-chave por ferramenta de extração *zero-shot*, processando-se o texto concatenado dos documentos do respectivo tópico. Assim, obteve-se, para cada tópico, um total de $n = 140$ frases-chave de entrada para os experimentos da curadoria.

Tipo	Prompt utilizado
Pro (P)	List the 20 main keyphrases of arguments in favor of {topic}, limiting the number of keywords to 3
Against (A)	List the 20 main keyphrases of arguments against {topic}, limiting the number of keywords to 3
NER (N)	What are the 20 most frequently cited named entities in arguments about {topic}?

Tabela 6.2: *Prompts utilizados para geração de frases-chave com ChatGPT. Pro (a favor), Against (contra) e NER (entidades nomeadas).*

6.4.2 Anotação do Conjunto de Dados

A anotação do conjunto de dados para curadoria de frases-chave seguiu o seguinte procedimento para cada subtarefa do fluxo: (a) dois especialistas realizam a anotação da subtarefa; (b) um terceiro especialista atua na adjudicação, resolvendo as discrepâncias e atribuindo um rótulo final. Assim, cada instância anotada foi avaliada por, no mínimo, três especialistas independentes.

Todos os especialistas possuem ao menos dois anos de experiência em pesquisa com anotação textual, sendo um com mestrado em Ciência da Computação, outro com formação em Direito e Engenharia de *Software*, e o terceiro com formação em Linguística.

Para estabelecer boas práticas no processo de anotação, seis tópicos foram discutidos em reuniões preparatórias. Para fins de anotação de teste, foram selecionados dois tópicos: *"death penalty"* e *"marijuana legalization"*.

6.4.3 Avaliação da Concordância entre Anotadores

A IAA foi avaliada individualmente para cada subtarefa, utilizando métricas apropriadas a cada contexto.

Para a subtarefa de *Agrupamento de Frases-Chave (KC)*, foram utilizadas as métricas *Adjusted Rand Index (ARI)*, *Normalized Mutual Information (NMI)* e *Fowlkes-Mallows Score (FMS)*.

A avaliação foi feita com base na similaridade entre os agrupamentos C' e C'', anotados por dois especialistas, conforme a definição formal de Meilă [Meilă 2007]. Tais métricas estão implementadas na biblioteca *Scikit-Learn* [Pedregosa et al. 2011].

Na subtarefa de *Filtragem de Agrupamentos (CF)*, a concordância foi calculada com base na similaridade entre dois conjuntos de agrupamentos, F' e F'', anotados independentemente. Os conjuntos comparados foram da forma $\{C'_1, C'_2, ..., C'_f\}$ e $\{C''_1, C''_2, ..., C''_f\}$, onde cada C_j representa um subconjunto de frases-chave.

Por exemplo, dados os agrupamentos $\{\{a,b\},\{c\},\{d,e,f\}\}$ e $\{\{c,f\},\{b,d\},\{g\}\}$, a concordância foi mensurada considerando a distância entre subconjuntos, utilizando como base a *distância de Jaccard*. A métrica adotada foi o *Alfa de Krippendorff* com distância de Jaccard [Krippendorff 2011], por meio da biblioteca *NLTK* [Loper e Bird 2002]. No exemplo, as distâncias entre $\{d,e,f\}$ e os subconjuntos do segundo anotador foram *0.75*, *0.75* e *1.0*, respectivamente, resultando em um Alfa de Krippendorff de *-0.09*.

A subtarefa de *Seleção de Frases-Chave (KS)* consistiu na escolha de x frases-chave por anotador, a partir de cada agrupamento C_i pertencente a um conjunto adjudicado $A = \{C_1, C_2, ..., C_f\}$.

Como exemplo, considere o conjunto $A = [\{a,g\},\{b,d,i,f\},\{h,j,k,l\}]$, com seleções feitas por dois anotadores: $S' = [\{a,g\},\{d,i\},\{l\}]$ e $S'' = [\{g\},\{d,i\},\{j\}]$. A concordância foi calculada usando o *Kappa de Fleiss* [Fleiss 1971], apropriado para múltiplos juízes e categorias. O valor obtido para este exemplo, utilizando a biblioteca *NLTK* [Loper e Bird 2002], foi de *0.43*.

Com base na experimentação, foram definidos os seguintes parâmetros para cada subtarefa: - Para *KC*, o número de agrupamentos adequado foi $c = 32$; - Após a *CF*, os agrupamentos foram reduzidos para $f = s = 16$; - Para *KS*, a escolha de $x = 2$ frases-chave por agrupamento mostrou-se adequada.

Subtarefa	Concordância	Death penalty	Marijuana legalization	Média
Agrupamento de Frases-chave (KC)	Adjusted Rand Index (ARI)	0.34	0.39	0.36
	Fowlkes-Mallows Score (FMS)	0.38	0.41	0.39
	Normalized Mutual Information (NMI)	0.50	0.51	0.50
Filtragem de Agrupamentos (CF)	Krippendorff's Alpha with Jaccard Distance (KAJ)	0.38	0.26	0.32
Seleção de Frases-Chave (KS)	Kappa de Fleiss	0.63	0.65	0.64

Tabela 6.3: *Concordância entre anotadores por subtarefa*

Os resultados da Tabela 6.3 indicam níveis variados de concordância entre os anotadores, em especial valores baixos nas primeiras subtarefas. Alguns fatores que podem ter influenciado esse desempenho incluem: (i) ausência de uma ferramenta madura

de suporte à anotação; (ii) inexistência de benchmarks para comparação direta; e (iii) inexperiência relativa dos anotadores na execução da tarefa.

Para mitigar tais limitações, estão em andamento as seguintes ações: desenvolvimento de uma ferramenta dedicada de anotação, expansão do conjunto de dados com novos textos e tópicos, e melhorias contínuas na diretriz de anotação. Espera-se que tais ajustes aumentem a reprodutibilidade e a confiabilidade do processo, permitindo análises futuras mais robustas.

6.4.4 Comparação com Ferramentas Automáticas

Esta seção apresenta uma análise comparativa do desempenho de dois Modelos de Linguagem de Grande Porte (LLMs) — *ChatGPT 3.5* e *Gemini 1.5-flash* — na tarefa de *Curadoria Automática de Frases-Chave*, utilizando como referência um conjunto de dados *gold* previamente elaborado.

Para orientar os modelos na execução da tarefa, foi aplicada a técnica de *Chain-of-Thought* [Kojima et al. 2023], com o intuito de guiar o raciocínio durante o processo de seleção. A sequência de prompts utilizada com os LLMs é detalhada na Tabela 6.4. Nessa abordagem, os modelos foram solicitados a selecionar 16 frases-chave a partir de uma lista com 140 opções. A Tabela 6.5 apresenta os resultados obtidos para os tópicos *"marijuana legalization"* e *"death penalty"*.

A análise dos resultados foi conduzida sob duas perspectivas complementares: uma avaliação qualitativa, centrada em critérios linguísticos, e uma avaliação quantitativa, com base na correspondência entre os conjuntos gerados e o conjunto *gold*.

Sequência	Prompt
1	Could you curate a list of keyphrases given next?
2	Curate 16 keyphrases from this list of 140 keyphrases:
3	[The user have to copy-and-paste the mentioned list of 140 keyphrases...]

Tabela 6.4: *Sequência de prompts utilizada com o ChatGPT para curadoria de frases-chave. O primeiro prompt (1) inicia a interação; o segundo (2) define a tarefa e a quantidade a ser selecionada; no terceiro prompt (3), o usuário fornece a lista de frases-chave. As respostas intermediárias dos modelos foram ignoradas.*

Análise Qualitativa

Conforme detalhado na Tabela 6.5, tanto o *ChatGPT* quanto o *Gemini* apresentaram desempenho satisfatório em relação aos aspectos morfossintáticos, cujas propriedades estão indicadas por **(M)** em cada frase-chave. No entanto, nos níveis **semântico** e **pragmático**, representados por **(S)** e **(P)**, respectivamente, o desempenho foi significativamente inferior quando comparado ao conjunto *gold*.

#	Marijuana Legalization			Death Penalty		
	ChatGPT	Gemini	Gold	ChatGPT	Gemini	Gold
1	**Safety and quality control (15)**	**Health risks (10)**	Advocacy groups	**Cost-effectiveness (2)**	**Cost-effectiveness (2)**	Amnesty International, Human Rights Watch
2	Consumer choice (12)	Consumer choice (12)	cost savings on public funds	**Rehabilitation potential (12)**	**Discrimination (6)**	**Cost-effectiveness**
3	Criminal justice reform (4)	Economic growth (7)	drug abuse	**Social order (14)**	**Flawed justice system (5)**	Crime Prevention, Removal of Dangerous individuals
4	Economic growth (7)	Harm reduction (15)	drug laws	Moral retribution (11)	**Human rights (8)**	criminology
5	Harm reduction (15)	Medicinal benefits (14)	drug policy	Recidivism prevention (12)	**Rehabilitation potential (12)**	homicide, murdering
6	Medicinal benefits (14)	Public safety concerns (15)	drug related violence and crime	Accountability	Moral objections (11)	**Discrimination**
7	Social equity (6)	*Tourism and industry boost (S)*	economic aspects	Deterrence	Public safety (14)	**Flawed Justice System**
8	Medical research opportunities (14)	*Job creation (S)*	effects on personal and mental development	*Disproportionate impact on marginalized communities (M)*	Deterrence	**Human Rights**
9	*Tourism and industry boost (S)*	*Tax revenue (S)*	gateway drug	*Possibility of executing the innocent (M)*	Retribution	Innocence Project
10	*Job creation (S)*	*Alleviation of chronic pain (S)*	**health risks**	*Crime prevention (S)*	*Possibility of executing the innocent (M)*	Irreversibility
11	*Cost savings in law enforcement (P)*	*Addiction potential (S)*	illicit drug	*Proportional punishment (S)*	*Cruel and unusual punishment (P)*	morality
12	*Reduction of opioid use (P)*	*Impaired cognitive function (S)*	individual freedom	*Protection of innocent lives (S)*	*Violation of international norms (P)*	**Rehabilitation Potential**
13	*Reducing drug-related violence (P)*	*Negative effects on youth (S)*	law enforcement agencies	*Violation of international norms (P)*	*Closure for victims' families (S, P)*	public opinion leaders
14	*Less burden on the judicial system (M,S)*	*Reduction of opioid use (P)*	medicinal	*Closure for victims' families (S, P)*	*Focus on prevention and rehabilitation (S, P)*	**Social order**
15	*Alleviation of chronic pain (S, P)*	*Conflict with federal law (P)*	**safety and quality control**	*Focus on prevention and rehabilitation (S, P)*	*Justice(S, P)*	Supreme Court, U.S. Department of Justice
16	*Treatment for specific conditions (e.g., epilepsy) (M, S, P)*	*Treatment for specific conditions (e.g., epilepsy) (M, S, P)*	social and moral concerns	*Justice (S, P)*	*Innocence (M, S, P)*	Symbolic value

Tabela 6.5: *Análise qualitativa comparativa dos conjuntos de frases-chave curadas por LLMs com base em critérios linguísticos. As frases-chave em **negrito** estavam presentes no conjunto gold, com o respectivo número de frases-chave gold (#), sendo interpretadas como de alta qualidade. As frases-chave sublinhadas não apresentaram erros e foram consideradas de boa qualidade; o número correspondente da frase-chave gold também é informado quando faz parte do mesmo agrupamento. As frases-chave em itálico apresentaram falhas em propriedades linguísticas. As propriedades são denotadas como (M - morfossintática; S - semântica; P - pragmática).*

Por exemplo, as expressões *medicinal benefits* e *alleviation of chronic pain* pertencem ao mesmo campo semântico e, portanto, deveriam estar agrupadas, de forma que apenas uma delas fosse selecionada como representante. No estado atual, a presença de múltiplas frases-chave semanticamente redundantes tende a gerar confusão para os anotadores, pois mais de uma opção pode ser considerada adequada para qualquer trecho que discuta o potencial analgésico da maconha — o que impacta negativamente a concordância entre anotadores.

Em relação às questões de ordem **pragmática**, observou-se um número considerável de frases-chave com baixa **representatividade** em relação ao conteúdo do *corpus*.

Algumas expressões, como *justice*, são excessivamente amplas e genéricas, tornando sua aplicação ambígua e, por vezes, desprovida de significado claro. Por outro lado, frases como *cost savings in law enforcement* podem ser relevantes, mas são pouco representativas do conjunto, aparecendo com baixa frequência e, portanto, em desvantagem frente a outras frases-chave mais recorrentes.

Além disso, identificou-se a presença de frases tendenciosas, marcadas por juízos de valor explícitos. Um exemplo é *cruel and unusual punishment*, cuja formulação inclui adjetivos e advérbios subjetivos, comprometendo a imparcialidade e a neutralidade da anotação.

Análise Quantitativa

Métrica	Marijuana Legalization		Death Penalty		Média
	ChatGPT	Gemini	ChatGPT	Gemini	
Jaccard w/ Exact Match	**0,03**	**0,03**	0,10	**0,19**	0,09
Dice w/ Exact Match	**0,06**	**0,06**	0,19	**0,31**	0,15
Jaccard w/ *Cluster* Match	**0,33**	0,23	0,19	**0,28**	0,26
Dice w/ *Cluster* Match	**0,50**	0,38	0,31	**0,44**	0,41

Tabela 6.6: *Resultado da curadoria de frases-chave realizada pelo ChatGPT e Gemini. As métricas utilizadas foram os índices de Jaccard e Dice, considerando uma correspondência exata com a frase-chave gold (exact match) ou a correspondência de uma frase-chave de boa qualidade pertencente ao mesmo agrupamento de uma frase-chave gold (cluster match).*

A Tabela 6.6 apresenta os resultados de desempenho na tarefa de curadoria de frases-chave realizada pelo *ChatGPT* e pelo *Gemini*. O *Índice de Jaccard* e o *Coeficiente de Dice* foram utilizados como métricas de avaliação, uma vez que a tarefa consiste na comparação da similaridade entre conjuntos de frases-chave.

Observa-se que a correspondência exata entre as frases-chave geradas e as do conjunto *gold* é bastante limitada, com índices variando entre 0,03 e 0,06 no tópico *"marijuana legalization"*, o que evidencia o desafio da tarefa.

Para uma análise mais abrangente, também foram consideradas correspondências com base em **similaridade semântica**, isto é, quando as frases pertencem ao mesmo agrupamento temático. Neste caso, as comparações foram restritas às frases-chave classificadas como de boa qualidade. Essa abordagem permite uma avaliação mais realista da utilidade dos conjuntos gerados, levando em conta a diversidade e a variação linguística natural da linguagem.

6.4.5 Avaliação Extrínseca

Neste experimento, foi conduzida uma avaliação extrínseca no contexto da classificação multirrótulo de tópicos. O desempenho da tarefa foi comparado com base na concordância entre anotadores, utilizando a métrica *Alfa de Krippendorff*, em conjunto com a medida de distância de *Jaccard*. A hipótese subjacente é que conjuntos de frases-chave de maior qualidade tendem a reduzir a ambiguidade e a aumentar o nível de concordância entre os anotadores.

Além dos conjuntos curados manualmente, foram selecionadas, para fins comparativos, as 20 principais frases-chave geradas automaticamente por cada ferramenta avaliada.

A avaliação foi realizada com base em quatro documentos textuais, totalizando 32 segmentos a serem classificados. Para cada segmento, os anotadores deveriam selecionar de uma a três frases-chave a partir de um conjunto com 16 opções. A tarefa foi executada por seis anotadores, majoritariamente estudantes de graduação em Ciência da Computação ou Linguística. O custo estimado por segmento foi de $0,65, com base em uma taxa horária média de $5,21 e tempo médio de análise de oito minutos por segmento.

Foram selecionados dois tópicos — *abortion* e *nuclear energy* —, cada um com dois documentos. Assim, os quatro documentos (*D*) foram combinados com quatro conjuntos distintos de frases-chave (*KS*), resultando em 16 combinações *D+KS*. Para garantir uma avaliação equilibrada, cada combinação foi atribuída a três anotadores distintos (*P*), totalizando 48 tarefas de anotação (oito por anotador).

Os resultados dessa avaliação estão apresentados na Tabela 6.8. O conjunto de frases-chave curado manualmente apresentou o melhor desempenho, com uma média geral de 0,32, mantendo consistência entre os documentos. O conjunto gerado pelo *ChatGPT* obteve média de 0,18, enquanto os demais métodos automatizados, *KeyBERT* e *YAKE*, apresentaram resultados significativamente inferiores, ambos com média de 0,06. O ganho de concordância obtido com o uso do conjunto gerado pelo *ChatGPT* foi de 0,14, conforme detalhado na Tabela 6.7.

Método	**Ganho de Concordância (IAA)**	
ChatGPT	0,14	76,64%
KeyBERT	0,26	416,76%
YAKE	0,27	500,17%

Tabela 6.7: *Ganho de Concordância entre Anotadores (IAA) ao utilizar o conjunto de frases-chave curado manualmente, em comparação com os conjuntos gerados automaticamente.*

Os resultados evidenciam que a qualidade do conjunto de frases-chave influencia diretamente a concordância entre anotadores. O desempenho inferior dos métodos automatizados, especialmente *KeyBERT* e *YAKE*, pode ser atribuído a diversos fatores, como

Keyword Set	Topic	Doc	a1	a2	a3	a1-a2	a1-a3	a2-a3	Average	General Average
Human Curation	abortion	29	p6	p2	p3	**0.30**	0.22	0.29	**0.27**	0.32
		44	p6	p2	p3	0.24	**0.57**	0.13	**0.31**	
	nuclear energy	0	p5	p1	p4	**0.38**	0.15	0.26	**0.26**	
		21	p5	p1	p4	**0.52**	0.39	0.41	**0.44**	
ChatGPT	abortion	29	p2	p1	p4	0.16	0.18	0.10	0.15	0.18
		44	p2	p1	p4	0.25	0.37	0.18	0.27	
	nuclear energy	0	p5	p6	p3	0.10	0.16	0.01	0.09	
		21	p5	p6	p3	0.31	0.25	0.11	0.22	
KeyBERT	abortion	29	p5	p3	p4	0.04	0.16	0.11	0.10	0.06
		44	p5	p3	p4	-0.07	0.18	-0,12	0.00	
	nuclear energy	0	p6	p2	p1	0.07	0.10	0.06	0.07	
		21	p6	p2	p1	0.06	-0.02	0.19	0.08	
YAKE	abortion	29	p5	p6	p1	0.22	0.23	0.20	0.22	0.05
		44	p5	p6	p1	-0.09	-0.17	0.00	-0.09	
	nuclear energy	0	p2	p3	p4	0.03	0.01	-0.02	0.00	
		21	p2	p3	p4	0.12	-0.01	0.13	0.08	

Tabela 6.8: *Comparação da Concordância entre Anotadores (IAA) utilizando o Alfa de Krippendorff com a distância de Jaccard como métrica de similaridade [Artstein e Poesio 2008].*

a ausência de controle semântico, redundâncias e a presença de frases pouco representativas ou excessivamente genéricas. Esses fatores dificultam a tarefa de seleção pelos anotadores, resultando em decisões mais dispersas e, consequentemente, menor índice de concordância.

No caso do *ChatGPT*, observa-se uma melhora substancial em relação aos métodos tradicionais, o que sugere que LLMs, quando guiados por técnicas como *Chain-of-Thought*, são capazes de gerar frases-chave mais coerentes e informativas. Ainda assim, o conjunto curado manualmente mantém uma vantagem considerável, especialmente por apresentar maior equilíbrio entre representatividade, especificidade e neutralidade semântica — aspectos que favorecem decisões mais consistentes por parte dos anotadores.

Esses achados reforçam a importância da curadoria humana, sobretudo em contextos onde a qualidade da anotação tem impacto direto na validade de tarefas supervisionadas, como classificação ou treinamento de modelos. Também destacam o potencial de uso combinado entre curadoria humana e ferramentas baseadas em LLMs para alcançar um melhor custo-benefício em cenários reais de anotação.

6.5 Conclusão

Este capítulo apresentou uma abordagem estruturada para a tarefa de *curadoria de frases-chave*, fundamentada na metodologia de decomposição de tarefas e orientada pelo padrão arquitetural *Recrutador–Selecionador*. A tarefa foi segmentada em três subtarefas principais — *Agrupamento (KC)*, *Filtragem (CF)* e *Seleção (KS)* —, cada uma operando com critérios explícitos de qualidade, definidos para garantir controle mais rigoroso da anotação.

A decomposição proposta permitiu isolar e compreender os desafios específicos de cada subtarefa, resultando em valores de *concordância entre anotadores (IAA)* que variaram de forma coerente com a complexidade cognitiva envolvida: *0,4–0,5* para *KC*, *0,3* para *CF* e *0,6* para *KS*. Esses níveis de concordância são comparáveis aos relatados em estudos similares, como os conduzidos por *Ishita et al. (2010)*, e demonstram a viabilidade da tarefa mesmo em cenários de alta subjetividade. A adoção da ferramenta de anotação em desenvolvimento — inspirada em práticas de adjudicação e feedback em tempo real — promete elevar ainda mais a estabilidade e a precisão dessas anotações, contribuindo para a consolidação de uma linguagem comum entre anotadores.

Do ponto de vista extrínseco, os resultados demonstraram ganhos expressivos na qualidade da anotação. Houve um aumento médio de *0,14 a 0,27* pontos no *Alfa de Krippendorff* em tarefas de classificação multilabel, indicando um ganho relativo de até *500%* em comparação com os piores métodos automáticos (como o *Yake*) e *76%* sobre o melhor gerador (ChatGPT). Tais achados reforçam o papel central da curadoria humana como fator crítico para a melhoria da confiabilidade dos dados.

A comparação entre métodos automáticos também revelou limitações importantes dos modelos de linguagem de grande porte (LLMs), que, apesar de superarem extratores tradicionais em termos de cobertura e fluência, apresentaram falhas semânticas recorrentes, sobretudo quanto à *univocidade* das frases. As métricas de similaridade — como o *Índice de Jaccard* e o *Coeficiente de Dice* — evidenciaram a distância entre os sistemas automáticos e o conjunto de referência, com valores de correspondência exata entre *0,03 e 0,31*, dependendo do tópico avaliado. O uso de uma métrica de correspondência semântica baseada em agrupamentos corrigiu parcialmente essa limitação, elevando os escores para até *0,50*.

Como contribuição adicional, foram disponibilizados um conjunto de dados anotados, um guia de critérios de qualidade e uma métrica original de avaliação de agrupamentos. Esses artefatos não apenas viabilizam a replicação dos experimentos, como também fortalecem o ecossistema de ferramentas e recursos voltados à anotação de *corpus* em PLN.

Em síntese, a tarefa de curadoria de frases-chave demonstrou ser altamente adequada à aplicação da metodologia de decomposição de tarefas e do padrão Recrutador–Selecionador, oferecendo evidências sólidas da importância de estruturas modulares e controladas para garantir a qualidade de anotação. Os achados aqui relatados fortalecem a proposta desta tese de desenvolver uma teoria fundamentada em padrões de projeto aplicáveis ao contexto de anotação de *corpus*, com potencial de impacto amplo e reutilizável em diferentes domínios do PLN.

CAPÍTULO 7

Decomposição de Tarefas de Anotação de *Corpus*

Este capítulo apresenta o terceiro estudo de caso desta tese, dedicado à análise da decomposição de tarefas complexas de anotação de *corpus*. O foco recai sobre estratégias que viabilizam a divisão de tarefas originalmente inviáveis — em termos de custo, tempo ou confiabilidade — em subtarefas mais controláveis, especialmente em contextos de anotação argumentativa.

A metodologia adotada envolve ciclos iterativos de anotação, adjudicação e refinamento de diretrizes, ancorados em métricas de concordância entre anotadores. Por meio dessa abordagem, foram definidos critérios objetivos para a reformulação de tarefas, com base em padrões linguísticos e evidências empíricas, conforme discutido no Capítulo 4.

A seguir, é apresentado um resumo de cada uma das seções subsequentes:

- **7.1** **Introdução**: apresenta um resumo de como a decomposição de tarefas foi aplicada no caso de uso;
- **7.2** **Mapeamento de Argumentos Monodocumento**: descreve a reformulação da tarefa original de mapeamento de argumentos em uma variante monodocumento com estrutura em camadas;
- **7.3** **Segmentação e Classificação de Tópicos**: introduz uma tarefa complementar focada na segmentação textual e anotação temática, destacando os impactos da heterogeneidade textual;
- **7.6** **Análise dos Resultados**: fornece uma análise geral dos efeitos da decomposição nas métricas de concordância e viabilidade anotativa;
- **7.7** **Conclusão**: sintetiza os principais achados do estudo de caso e suas implicações para a modelagem de tarefas anotativas em PLN.

7.1 Introdução

A decomposição de tarefas de anotação de *corpus* emergiu nesta pesquisa como uma estratégia central para lidar com os desafios de confiabilidade observados na tarefa original de Mapeamento de Argumentos. Conforme discutido na Seção 4.6, a baixa concordância entre anotadores, aliada à complexidade interpretativa da tarefa, inviabilizou abordagens diretas de anotação. Esse cenário evidenciou a necessidade de estruturar o processo em etapas mais objetivas e manejáveis.

Neste contexto, este capítulo explora a aplicação prática do algoritmo de decomposição hierárquica de tarefas, apresentado na Seção 4.6.1. O algoritmo considera duas dimensões estatísticas — a estabilidade da concordância e o efeito acumulado das mudanças nas diretrizes — como gatilhos para a reavaliação e segmentação da tarefa em subtarefas mais simples e bem definidas.

A proposta é inspirada na perspectiva de [Simon 1962] sobre a arquitetura da complexidade, segundo a qual sistemas complexos tendem a evoluir a partir da integração hierárquica de subsistemas mais simples, relativamente autônomos e estáveis ao longo do tempo. Ao aplicar esse princípio à anotação linguística, buscou-se tornar o processo mais robusto, reprodutível e escalável, por meio de ciclos iterativos de refinamento baseados em evidências empíricas extraídas diretamente da atividade de anotação. Mais detalhes sobre essa abordagem podem ser encontrados na Seção 3.3.1.

7.1.1 Tarefa Original: Mapeamento de Argumentos Multidocumento

Em diversas áreas do conhecimento, como ciência, direito e política, a produção textual diária resulta em milhares de documentos argumentativos que sustentam hipóteses, justificam decisões e defendem posicionamentos. Para promover o avanço do conhecimento em determinada área, torna-se imprescindível o acesso a produções anteriores, de modo a evitar esforços redundantes e estimular a formulação de ideias originais. No entanto, apesar dos avanços computacionais observados nas últimas décadas, a maior parte desse conhecimento permanece comunicada em linguagem natural não estruturada, o que impõe um ônus considerável à análise humana, proporcional à quantidade de documentos disponíveis.

Nesse contexto, técnicas de *sumarização automática de argumentos* emergem como uma alternativa promissora para apoiar a análise de grandes volumes de texto, ao fornecer representações estruturadas que sintetizam a argumentação presente em múltiplas fontes. Contudo, a construção de sistemas desse tipo requer, como etapa preliminar, a disponibilização de conjuntos de dados anotados manualmente por especialistas, que funcionem como padrão-ouro para o treinamento e a avaliação de modelos de aprendi-

zado de máquina. Esses conjuntos ainda são escassos ou insuficientemente anotados para a complexidade da tarefa, o que torna necessário o investimento em processos de anotação manual — atividades que, por sua vez, demandam protocolos rigorosos de controle de qualidade, a fim de evitar vieses e garantir a validade científica dos resultados.

A tarefa original que motivou o desenvolvimento desta tese foi a de **mapeamento de argumentos multidocumento** (doravante referida pela sigla **Mapeamento de Argumentos Multidocumento (MAMD)**), definida da seguinte forma: *dado um conjunto de documentos não estruturados sobre um mesmo tópico, construir uma árvore de argumentos que sumarize a argumentação presente nos textos*.

O principal componente dessas árvores argumentativas são os **argumentos**, definidos como pares compostos por uma **premissa** e uma **conclusão**, cujo objetivo é persuadir ou justificar um posicionamento frente a determinado tema. As conclusões expressam hipóteses, opiniões ou julgamentos, enquanto as premissas oferecem justificativas ou evidências que sustentam tais conclusões. Quando articulados em um texto, esses pares constituem uma **estrutura argumentativa**, que pode ser representada graficamente por meio de um *mapa de argumentos*.

A Figura 7.1 ilustra um exemplo de mapa argumentativo construído a partir de comentários públicos sobre o Projeto de Lei nº 5198/2020, que propõe a proibição do uso de linguagem neutra em instituições de ensino e concursos públicos. Cada nó do grafo representa uma **unidade argumentativa (UA)**, classificada como premissa (P) ou conclusão (C). As unidades em vermelho indicam argumentos contrários à proposição central, enquanto as em verde expressam apoio. As arestas indicam relações de suporte entre UAs (P→P, P→C, C→C).

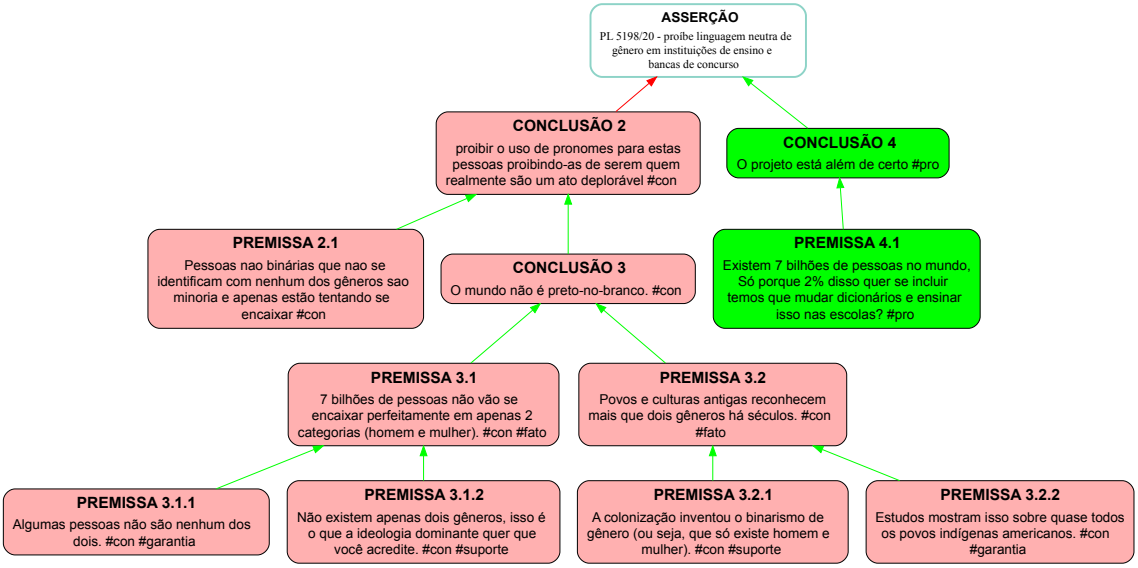


Figura 7.1: *Exemplo de diagrama de argumentação construído a partir de comentários sobre o PL 5198/2020*

A avaliação da qualidade dos mapas de argumentos pode ser conduzida com base em critérios tradicionalmente empregados em tarefas de sumarização automática, tanto na variante monodocumento [Kryscinski et al. 2019] quanto na multidocumento [Verma e Om 2019]. Os principais critérios incluem:

- **Relevância**: seleção de conteúdos centrais e informativos presentes nas fontes originais.
- **Consistência**: manutenção do alinhamento factual entre o resumo gerado e os textos-fonte.
- **Fluência**: qualidade linguística das sentenças, no nível gramatical e estilístico.
- **Coerência**: articulação lógica entre as unidades textuais do mapa, garantindo a construção de uma narrativa compreensível.
- **Cobertura**: representação adequada dos principais tópicos abordados nos documentos de entrada.
- **Não-redundância**: supressão de repetições, assegurando concisão sem perda de conteúdo.

Para além desses aspectos técnicos, é fundamental considerar a **eficácia comunicativa da visualização** do mapa gerado. A representação gráfica da argumentação visa não apenas sintetizar o conteúdo, mas também apoiar a compreensão, promover *insights* e facilitar a exploração interativa por parte dos usuários. O campo de pesquisa em *visualização de argumentos* tem demonstrado resultados positivos em contextos educacionais e na promoção do pensamento crítico [Ortiz 2007], especialmente com o suporte de ferramentas como o portal colaborativo Kialo[1] e o framework Argdown[2]. Assim, a avaliação de um mapa de argumentos deve também levar em conta sua clareza visual, a organização dos elementos e sua adequação à tarefa de comunicar de forma compreensível e persuasiva a estrutura argumentativa subjacente.

7.1.2 Metodologia de Anotação Baseada em Padrões

Esta tese adota uma abordagem baseada em padrões com o objetivo de estruturar o processo de anotação de dados linguísticos. O termo "padrão", neste contexto, refere-se a métodos recorrentes documentados na literatura, associados à gestão da qualidade de anotação.

Conforme discutido na Seção 3.1, a qualidade da anotação pode ser analisada a partir de quatro dimensões principais: estabilidade, reprodutibilidade, precisão e imparcialidade. Os padrões incorporados nesta metodologia correspondem a métodos descritos

[1] https://www.kialo.com/
[2] https://argdown.org/

por [Klie, Castilho e Gurevych 2024], os quais têm como objetivo influenciar positivamente essas dimensões ao longo do processo de anotação.

Esses padrões foram organizados por Klie et al. em cinco categorias: processo de anotação, gerenciamento de anotadores, estimativa de qualidade, melhoria da qualidade e adjudicação. A proposta desta tese é integrar esses padrões ao planejamento e à execução das tarefas de anotação, promovendo maior controle sobre os fatores que impactam a confiabilidade dos dados gerados.

O processo adotado combina diferentes padrões em etapas sucessivas, conforme descrito a seguir:

- **Planejamento:** seleção criteriosa dos dados, definição do esquema de anotação, elaboração e validação do *guideline*, e realização de estudo piloto;
- **Formação de equipe:** recrutamento da força de trabalho, aplicação de testes de qualificação, treinamento dos anotadores e coleta de depoimentos sobre o processo;
- **Execução:** anotação independente, uso de questões de controle e cálculo de métricas de concordância interanotador;
- **Correção e refinamento:** revisão do *guideline*, correção de anotações, filtragem de anotações inconsistentes, remoção de anotadores com baixo desempenho e fornecimento de *feedback* individual;
- **Adjudicação:** curadoria manual e agregação de anotações por meio de votação majoritária ou métodos probabilísticos.

A metodologia adotada não pressupõe um fluxo único ou fixo. A escolha e a combinação dos padrões são determinadas pelas características da tarefa de anotação, como o grau de subjetividade, a disponibilidade de recursos humanos e o uso pretendido do *corpus*. O objetivo é tornar o processo mais transparente, auditável e ajustado a critérios de qualidade previamente definidos.

Nas seções seguintes, são apresentadas aplicações desta metodologia em diferentes contextos de anotação, com ênfase na sua viabilidade e nos resultados obtidos.

Processo de anotação

Para anotar cada camada de anotação, foi aplicada uma metodologia dividida em duas equipes: equipe de anotadores especialistas e equipe de anotadores treinados. Os anotadores especialistas são responsáveis pela definição de regras de anotação, treinamento de anotadores e manutenção da qualidade de anotação. Os anotadores treinados formam uma equipe mais numerosa e que realiza um volume maior de anotações, seguindo as regras disponibilizadas em um *guideline*.

A qualidade de anotação é aferida pelas métricas de concordância entre anotadores. No nosso processo, dois anotadores anotam cada dado e as discordâncias são

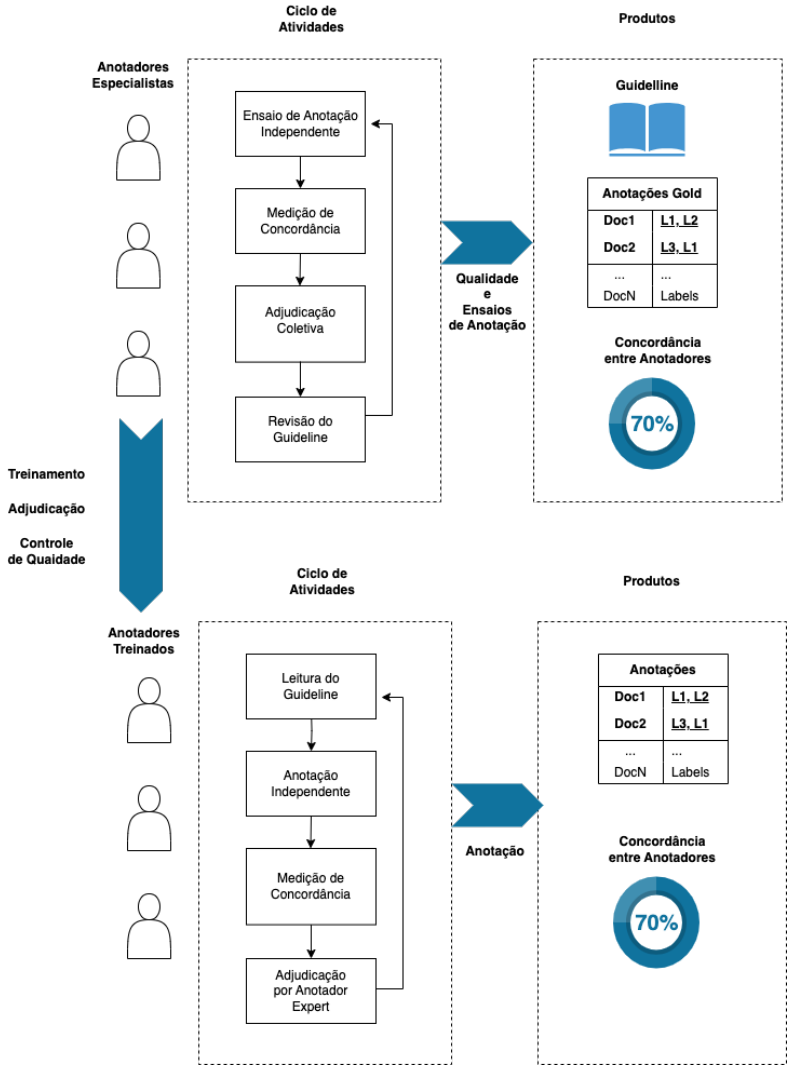


Figura 7.2: *fluxo de atividades de anotação*

resolvidas por um terceiro anotador denominado adjudicador. Assim, garante-se maior qualidade por meio do ganho de precisão e impessoalidade pela redundância de anotação e análise de pelo menos 3 avaliações de cada dado.

O processo de anotação de cada camada de anotação é ilustrado no diagrama da figura 7.2.

O ciclo de atividades dos anotadores especialistas começa com ensaios de anotação independente, onde um conjunto de documentos previamente atribuídos é anotado. Quando a anotação do conjunto de documentos é finalizada, a medida de concordância entre eles é calculada. O valor calculado indica o nível de divergência dos documentos e aqueles que tiverem um índice maior são avaliados para discussão em reunião - o qual denominamos de adjudicação coletiva - para se realizarem diagnósticos e resolver dissensos. No final deste ciclo, os consensos são registrados como novas regras do *guideline*. O produto deste ciclo é o *guideline* revisado e anotações padrão *gold*, que possuem uma concordância entre anotadores medida antes da adjudicação coletiva.

Para capacitar novos anotadores, realizamos treinamentos sobre o conteúdo do *guideline* e avaliamos o desempenho deles através de anotações independentes que são depois comparadas com a anotação *gold*. Se o desempenho do anotador em um lote de documentos (geralmente 10 documentos) for acima de um valor típico - geralmente 0,5 - o anotador é considerado treinado.

O ciclo de atividades dos anotadores treinados inicia com as anotações em conformidade do *guideline* para fins de consulta e apoio. Os anotadores anotam independentemente um do outro - sem consultas entre si - e então, no final das anotações, uma medida de concordância entre dois anotadores é calculada. No final, um anotador *expert* (especialista) adjudica as anotações e, assim, uma anotação padrão *gold* final é produzida.

7.1.3 Organização dos experimentos

A partir da decomposição de tarefas detalhada na Figura 7.3, as próximas seções apresentam os experimentos realizados para validar a aplicação prática formalizada no Algoritmo 4.1. Os experimentos visam analisar a eficácia da decomposição no contexto da anotação de *corpora*.

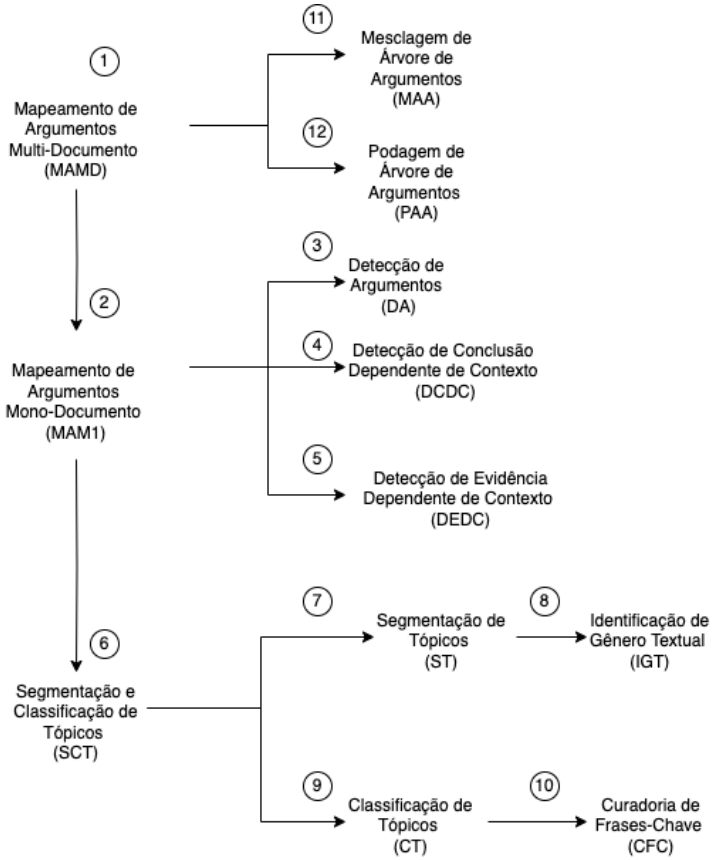


Figura 7.3: *Árvore de decomposição em subtarefas da tarefa de Mapeamento de Argumentos.*

A decomposição de tarefas pode ser interpretada como uma **busca em profundidade em árvore** (*Depth-First Search - DFS*), conforme ilustrado na Figura 7.4. A tarefa original, representada pela raiz da árvore, é o MAMD, que foi decomposta em três subtarefas: MAM1, Mesclagem de Árvores de Argumentos (MAA) e Podagem de Árvores de Argumentos (PAA). Como as subtarefas MAA e PAA dependem de árvores de argumentos previamente construídas, a subtarefa MAM1 precisou ser desenvolvida inicialmente.

Seguindo a ordem da busca em profundidade, 12 subtarefas — incluindo a própria MAMD — foram desenvolvidas total ou parcialmente, conforme ilustrado na Figura A.1, levando em conta as relações de dependência. Dessa forma, subtarefas que não dependem dos resultados de outras, como os nós-folha 8 e 10, puderam ser concluídas primeiro.

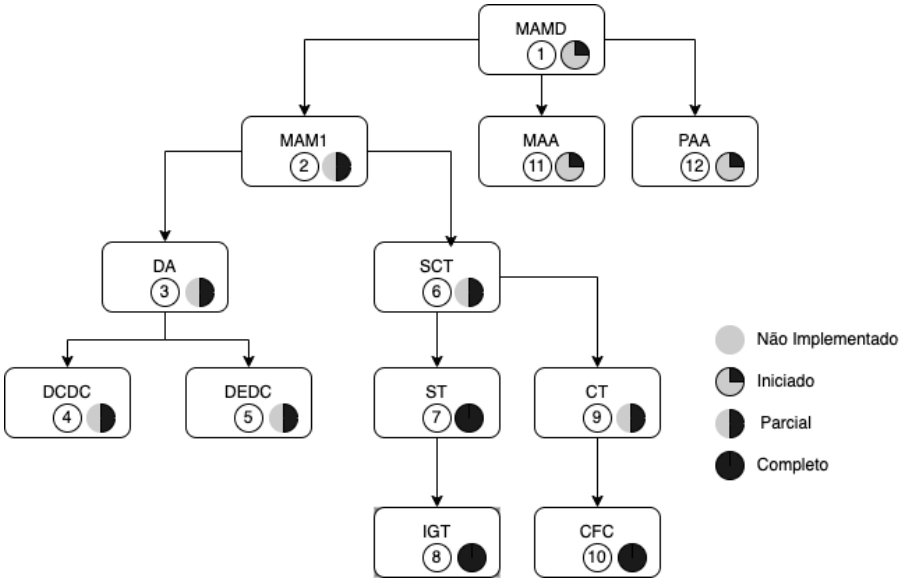


Figura 7.4: *Árvore de subtarefas representada como uma busca em profundidade (Depth-First Search - DFS) com respectivos status de anotação.*

Camadas de anotação

Nas seções seguintes, são apresentados os experimentos realizados nas principais subtarefas — MAM1, Segmentação e Classificação de Tópicos (SCT) e MAMD — nessa ordem. Como evidenciado na Figura 7.3, existe uma relação de dependência entre essas subtarefas principais: MAMD depende de MAM1, que por sua vez depende de SCT. Essa configuração revela um padrão de camadas dependentes, no qual cada camada superior depende dos resultados consolidados na camada imediatamente inferior.

7.2 Mapeamento de Argumentos Monodocumento (MAM1)

A primeira tarefa anotada foi o MAM1. O objetivo foi identificar unidades argumentativas (como conclusões e evidências) em textos individuais, estabelecendo uma base para representações mais complexas em etapas posteriores. Esta seção descreve o processo de seleção do *corpus* original e o pré-processamento necessário para viabilizar a anotação contextualizada.

7.2.1 Busca por *Dataset* Original

Inicialmente, realizou-se uma busca por *datasets* previamente anotados que pudessem ser adequados à tarefa. Foram considerados *corpora* de diferentes domínios — saúde, jurídico e conhecimentos gerais. No entanto, os conjuntos da área da saúde, como o *CORD-19*[3], e da área jurídica foram descartados devido ao elevado custo de anotação especializada, conforme também apontado por [Inuzuka et al. 2020].

A partir dessa triagem, o foco passou a recair sobre textos de conhecimentos gerais. Foram analisados quatro *datasets* principais:

- *Dr Inventor* [Lauscher, Glavaš e Ponzetto 2018]
- *DebateSum* [Roush e Balaji 2020]
- *Argument Annotated Essays* [Stab e Gurevych 2014]
- *UKP Sentential* [Stab et al. 2018]

Cada *dataset* foi avaliado segundo critérios como quantidade de documentos, tipo de segmentação, domínio temático, granularidade de anotação, disponibilidade pública e grau de dependência de contexto. A Tabela 7.1 resume essas características.

[3] https://allenai.org/data/cord-19

Característica	Dr Inventor	DebateSum	UKP Sentential	Persuasive Essays
Quantidade de Documentos	40	187.328	400	90
Divisão por tópicos	Não	Tópicos diversos sem quantidade definida	8 tópicos controversos com 50 documentos cada	Não
Tipo de documento	Artigos científicos	Documentos diversos	Páginas *Web*	Redações
Domínio de aplicação	Ciência da Computação	Conhecimentos gerais (debates)	Conhecimentos gerais	Escrita acadêmica
Granularidade	Span	Span	Sentenças	Span
Disponibilidade	Público	Público	Público	Público
***Guideline* disponível**	Sim	Não	Não	Sim
Anotação disponibilizada	Componentes e relações argumentativas	Trechos de texto (tópico e evidências) sem distinção estruturada	Sentenças rotuladas como argumento pró, contra e neutro	Componentes e relações argumentativas
Dependência de contexto	Sim, texto completo	Sim, texto completo	Não, sentença isolada	Sim, texto completo

Tabela 7.1: *Características dos datasets avaliados para a escolha do corpus base*

Dentre os conjuntos analisados, o *UKP Sentential* destacou-se pela organização temática em tópicos controversos e pela granularidade em nível de sentenças, com rótulos previamente aplicados. Essas características facilitaram sua adaptação como *corpus* base.

7.2.2 Pré-processamento do UKP Sentential

Apesar de sua estrutura vantajosa, o UKP Sentential apresenta uma limitação: as sentenças estão isoladas, sem o texto completo original. Para viabilizar uma abordagem mais contextualizada, foi necessário um pré-processamento que permitisse a reconstrução dos documentos.

As principais etapas desse processo foram:

- **Rastreamento das fontes originais:** os metadados do *corpus* foram utilizados para recuperar os documentos completos por meio de links arquivados no *Internet Archive* [4].
- **Reconstrução textual:** os conteúdos foram convertidos para o formato texto puro (`.txt`), com remoção de elementos de navegação e formatação HTML.
- **Mapeamento das sentenças anotadas:** as sentenças rotuladas foram localizadas nos textos por meio de algoritmos de correspondência textual aproximada.
- **Segmentação e alinhamento:** foram aplicadas heurísticas de segmentação morfossintática e realinhamento para resolver divergências entre as sentenças originais e os textos reconstruídos.

[4] Disponíveis em https://archive.org/

O *corpus* resultante preserva o contexto original das sentenças, servindo como base estruturada para as etapas posteriores de anotação.

7.2.3 Detecção de Argumentos (DA)

A tarefa de Detecção de Argumentos (DA), conforme definida por [Stab et al. 2018], consiste na identificação de sentenças argumentativas que expressem posicionamentos favoráveis ou contrários a um tópico específico, bem como sentenças não argumentativas. No referido estudo, foram anotadas 25.000 sentenças, categorizadas em três classes: *pro*, *against* e *no-argument*, correspondendo, respectivamente, a sentenças com posicionamento favorável, contrário e ausência de conteúdo argumentativo.

A construção do *corpus* foi conduzida por meio da seleção aleatória de oito tópicos socialmente controversos, extraídos de um portal de debates: aborto, clonagem, controle de armas, energia nuclear, legalização da maconha, pena de morte, renda mínima e uso de uniforme escolar. A coleta textual foi realizada por meio de buscas no mecanismo Google, sendo consideradas as 50 primeiras páginas com cópias arquivadas no portal Archive[5], posteriormente convertidas em texto puro.

A amostragem consistiu na extração aleatória de sentenças contendo, no mínimo, três palavras e pelo menos um verbo. O *corpus* resultante compreende 25.492 sentenças, das quais 14.353 foram rotuladas como `no-argument`, 4.944 como `support argument` e 6.195 como `oppose argument`, distribuídas em 50 documentos por tópico, totalizando 400 documentos anotados.

Considerando que o objetivo subsequente da tarefa de Mapeamento de Argumentos é a construção de estruturas hierárquicas de argumentação (árvores argumentativas) a partir de documentos completos, tornou-se necessário recuperar os textos integrais associados às sentenças previamente anotadas. Esse processo envolveu não apenas a recuperação textual a partir das URLs originais, mas também a contextualização das sentenças rotuladas em seu ambiente textual. Assim, foi produzido um conjunto de dados derivado, no qual os rótulos do *corpus* original — atribuídos a sentenças descontextualizadas — foram mapeados para suas respectivas ocorrências nos documentos completos.

Para a visualização e anotação dos documentos enriquecidos com contexto, inicialmente utilizou-se a ferramenta Brat Rapid Annotation Tool (BRAT), que permite a inspeção visual das sentenças rotuladas em seu entorno textual. A adoção do *dataset UKP Sentential* e a recuperação dos rótulos associados possibilitaram a reutilização de anotações previamente existentes, reduzindo significativamente o custo cognitivo e o esforço manual exigido para a execução da próxima subtarefa da cadeia de anotação.

[5] Disponível em `https://archive.org`

7.2.4 Detecção de Conclusão Dependente de Contexto (DCDC)

Nesta etapa, foi possível contar com cerca de 30% a 60% de argumentos previamente identificados com um dos três rótulos apresentados na seção anterior (*pro*, *against*, *no_argument*). Também foram observados rótulos aplicados incorretamente, em sua maioria devido à ausência de contexto durante a anotação. Por exemplo, no *UKP Sentential*, algumas sentenças descontextualizadas foram rotuladas como *pro*, mas, ao serem analisadas em seu contexto original, revelaram-se não argumentativas, exigindo a reclassificação para *no_argument*. Embora essas falhas não inviabilizem o reaproveitamento dos rótulos, elas evidenciam a importância do contexto na análise argumentativa.

Além da relevância do contexto, a tarefa de mapeamento exigia a identificação de dois componentes argumentativos principais: a conclusão (*claim*) e a evidência (*evidence*). Um argumento é composto, no mínimo, por uma evidência que apoia uma conclusão ou outra evidência, formando um diagrama ou árvore argumentativa, conforme ilustrado na Figura 7.5.

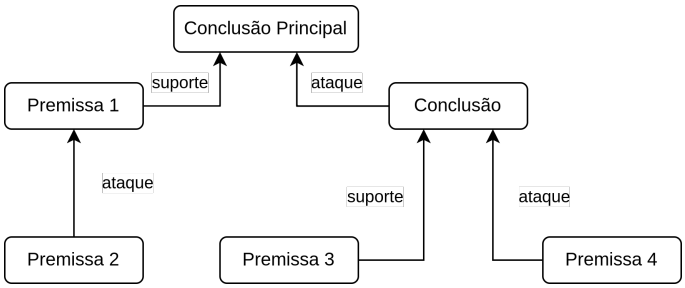


Figura 7.5: *Estrutura de argumentos composta por quatro premissas, uma conclusão intermediária e uma conclusão principal, com relações de ataque e suporte entre os componentes argumentativos.*

Note, na Figura 7.5, que uma ordem mais conveniente para a anotação de árvores argumentativas é iniciar pela detecção de conclusões, pois as premissas ou evidências são dependentes delas, ao terem como função central sustentá-las.

Considerando essa prioridade e a importância do contexto para a análise argumentativa, a teoria proposta por [Levy et al. 2014] foi adotada como base conceitual para a tarefa de Detecção de Conclusão Dependente de Contexto (DCDC).

Segundo essa abordagem, a DCDC é definida como a tarefa de identificar, em textos livres, *afirmações gerais e bem formuladas que apoiam ou contestam diretamente um tópico previamente definido*, levando em consideração o contexto em que estão inseridas [Levy et al. 2014]. Diferentemente da detecção de conclusões genéricas, a DCDC exige que a relevância da afirmação seja avaliada em relação ao tema-alvo, o que implica desafios adicionais, como desambiguação semântica e a seleção de trechos com força argumentativa. Essa definição forneceu diretrizes mais objetivas para a anotação de

conclusões no *corpus*, auxiliando na distinção entre afirmações conclusivas e sentenças apenas descritivas, genéricas ou fora de contexto.

A Tabela 7.2 apresenta exemplos e contraexemplos de conclusões dependentes de contexto (CDCs), com base nos critérios estabelecidos por [Levy et al. 2014]. Sentenças como S1, S2 e S3 são consideradas CDCs válidas, enquanto as demais falham por falta de generalidade, por repetirem o tópico ou por não estabelecerem relação argumentativa clara.

ID	Sentença	CDC
S1	Violent video games can increase children's aggression	**V**
S2	Video game publishers unethically train children in the use of weapons	**V**
S3	Violent games affect children positively.	**V**
S4	Video game addiction is excessive or compulsive use of computer and video games that interferes with daily life	**X**
S5	Violent TV shows just mirror the violence that goes on in the real world.	**X**
S6	Violent video games should not be sold to children	**X**
S7	"Doom" has been blamed for nationally covered school shooting. Este candidato falha no critério da generalidade, uma vez que foca em um vídeo game em específico.	**X**

Tabela 7.2: *Exemplos e contraexemplos de conclusões dependentes de contexto (CDCs) com base no tópico "The sale of violent video games to minors should be banned".*

Além de pelo menos uma conclusão, um argumento também é composto por uma evidência ou premissa. A próxima seção trata justamente deste componente complementar dos argumentos.

7.2.5 Detecção de Evidência Dependente de Contexto (DEDC)

Complementarmente ao trabalho de [Levy et al. 2014], [Rinott et al. 2015] propõem a tarefa de *Detecção de Evidência Dependente de Contexto* (DEDC), que consiste em identificar, em textos não estruturados, trechos que sustentem diretamente uma afirmação (*claim*) dentro do contexto de um tópico específico. Uma evidência é considerada válida quando, além de ser linguisticamente coerente e bem formulada, estabelece uma relação semântica clara com a afirmação e é relevante ao tópico em discussão. Essa definição pressupõe que a utilidade da evidência está condicionada tanto ao conteúdo da afirmação quanto ao enquadramento temático do debate.

A Tabela 7.3 apresenta exemplos e contraexemplos de evidências dependentes de contexto (EDCs). Para que uma EDC seja identificada, pressupõe-se a existência de um contexto composto pela combinação entre um tópico e uma conclusão, formando o que anteriormente chamamos de CDC. As sentenças S1, S2 e S3 são consideradas evidências válidas por sustentarem diretamente a Conclusão A no contexto do tópico proposto, compondo a CDC A. Já as sentenças S4 e S5, embora contenham informações factuais, não oferecem suporte relevante à CDC A. A sentença S6, por sua vez, apenas reafirma a conclusão, sem apresentar nova informação, o que a descaracteriza como evidência. Por fim, a CDC B, formada pelo mesmo tópico e por uma conclusão oposta (Conclusão B), expressa um ponto de vista contrário à CDC A e é sustentada pela sentença S7, que constitui uma EDC válida nesse novo contexto.

Tópico: Use of performance enhancing drugs (PEDs) in professional sports	
Conclusão A: PEDs can be harmful to athletes' health	
S1: A 2006 study examined 320 athletes for psychiatric side effects induced by anabolic steroid use. The study found a higher incidence of mood disorders in these athletes compared to a control group.	V
S2: The International Agency for Research on Cancer classifies androgenic steroids as "Probably carcinogenic to humans."	V
S3: Rica Reinisch, a triple Olympic champion and world record-setter at the Moscow Games in 1980, has suffered numerous miscarriages and recurring ovarian cysts following drug abuse.	V
S4: The UN estimates that there are more than 50 million regular users of heroin, cocaine and synthetic drugs.	X
S5: FDA does not approve ibuprofen[2] for babies younger than six months due to risk of liver damage.	X
S6: Doping can ultimately damage your health.	X
Conclusão B: Use of PED is in line with the spirit of sport	
S7: Professor Savulescu, a philosopher and bioethicist, believes that biological manipulation embodies the sports spirit: the capacity to improve ourselves on the basis of reason and judgment.	V

Tabela 7.3: *Exemplos e contraexemplos de evidências dependentes de contexto (EDC). A marcação V/X indica se o candidato é uma evidência dependente de contexto (EDC) para o par tópico-conclusão (CDC A e CDC B). Adaptado de [Rinott et al. 2015].*

A abordagem baseada na identificação de conclusões e evidências dependentes de contexto permitiu estruturar critérios objetivos para a anotação argumentativa. No entanto, verificou-se que muitas construções argumentativas seguem padrões linguísticos recorrentes, sinalizados por indicadores discursivos. Esses padrões passaram a ser considerados uma alternativa viável para a detecção de argumentos de forma mais sistemática. A próxima subseção apresenta essa abordagem baseada em padrões, aplicada como es-

tratégia complementar na tarefa de mapeamento de argumentos, de forma integrada e contextualizada, com foco na anotação de sentenças, e não mais de trechos isolados de texto.

7.2.6 Detecção de Argumentos por Padrões

A abordagem de [Levy et al. 2014] baseou-se na detecção separada de conclusões e das evidências que as sustentam, tratando-as como tarefas distintas. Como estratégia alternativa, foi adotada uma abordagem empírica baseada em indicadores discursivos que, quando combinados com uma premissa e uma conclusão, formam uma construção argumentativa — em consonância com a Teoria Baseada no Uso apresentada na Seção 4.2.

Indicadores Discursivos

A detecção de padrões argumentativos com base em indicadores discursivos demonstrou-se uma abordagem eficaz para o mapeamento de argumentos. Com base no trabalho de [Gao et al. 2022], foi elaborada uma lista com 14 indicadores de discurso voltados à introdução de conclusões, apresentada na Tabela 7.4. Esses indicadores expressam conclusões, relações de consequência ou de sumarização.

Conclusão	**Consequência**	**Sumarização**
so	as a result	to sum up
conclude that	hence	in short
thus	consequently	–
we may deduce	–	–
in conclusion	–	–
points to the conclusions	–	–
proves that	–	–
therefore	–	–
may be inferred	–	–

Tabela 7.4: *Indicadores de Claims (Conclusão)*

Além desses, foram identificados indicadores relacionados a premissas ou evidências, conforme apresentado na Tabela 7.5. Esses elementos discursivos sinalizam relações de contraste, adição ou causa.

A Tabela 7.6 apresenta indicadores que podem ser utilizados tanto para conclusões quanto para premissas, atuando como conectores versáteis no discurso argumentativo.

Com base na Teoria Baseada no Uso, foram identificadas construções argumentativas compostas por premissas (P), conclusões (C) e indicadores discursivos (I). A partir de [Gao et al. 2022, Apêndice C], foram definidas quatro estruturas linguísticas principais: **PIC** (Premissa → Indicador → Conclusão), **CIP** (Conclusão → Indicador → Pre-

Contraste	Adição	Causa
But	And	deduced
On the contrary	in addition	due to
However	besides	given that
Whereas	moreover	for
–	furthermore	since
–	what's more	researchers found that
–	–	indicated by
–	–	is supported by
–	–	this can be seen from

Tabela 7.5: *Indicadores de Premissa (Evidências)*

Explicação		Exemplificação	
implies	because	for example	or instance
assuming that	in light of	as	as indicated by
in that	in view of	as shown	derived from
accordingly	clearly	follows that	entails
it should be clear	it is highly probable that	–	–
indicates that	it follows that	–	–
shows that	suggests that	–	–

Tabela 7.6: *Indicadores discursivos de ambos: conclusões e premissas*

missa), **IPC** (Indicador, Premissa → Conclusão) e **ICP** (Indicador → Conclusão → Premissa). A Tabela C.1 apresenta exemplos dessas estruturas, com exceção da forma **ICP**, omitida da tabela por ocorrer em apenas dois casos: *Here is why C: P.* e *In support of C, P.*

Além das construções em uma única sentença, também foram identificadas estruturas compostas por duas sentenças, conforme ilustrado na Tabela C.2. Todas essas formas foram registradas no *guideline*[6] para apoiar os anotadores na identificação de argumentos.

Seguindo uma abordagem integrada, adotou-se um esquema de anotação no qual conclusões e premissas são anotadas simultaneamente. Nesta abordagem, os rótulos são aplicados em sentenças, e não em trechos isolados (*spans*) de texto. Assim, uma mesma sentença pode conter simultaneamente uma conclusão e uma premissa. As próximas subseções apresentam esse esquema de anotação, considerando sentenças simples — contendo apenas conclusões ou premissas — e sentenças compostas, que combinam ambos os elementos argumentativos.

[6] Acessível em https://argmap.inf.ufg.br/guideline/appendix2/

Construções: Conclusões Simples ou Compostas

Foram adotados dois rótulos principais para categorizar conclusões: **ClaimFavor** e **ClaimAgainst**. O primeiro é aplicado quando a sentença expressa um posicionamento favorável ao tópico; o segundo, quando o posicionamento é contrário. Cada rótulo pode ser enriquecido com uma propriedade adicional: **Supported** ou **Rebuttal**. A propriedade `Supported` indica que a conclusão é sustentada por uma evidência ou por outra conclusão, ainda dentro da mesma sentença. Já a propriedade `Rebuttal` caracteriza construções em que uma declaração inicial é desafiada ou refutada por um argumento oposto, também presente na mesma sentença.

Dessa forma, uma conclusão pode receber um dos seguintes seis rótulos:

- **ClaimAgainst**: conclusão simples, sem suporte explícito, contrária ao tópico.
  - **SupportedClaimAgainst**: conclusão contrária ao tópico, sustentada por evidência ou outra conclusão.
  - **RebuttalClaimAgainst**: conclusão contrária ao tópico, construída por meio de uma refutação explícita de uma ideia favorável.
- **ClaimFavor**: conclusão simples, sem suporte explícito, favorável ao tópico.
  - **SupportedClaimFavor**: conclusão favorável ao tópico, sustentada por evidência ou outra conclusão.
  - **RebuttalClaimFavor**: conclusão favorável ao tópico, construída por meio de uma refutação explícita de uma ideia contrária.

Construções: Evidências

Para a anotação de evidências, foram definidos quatro rótulos, com base em observações empíricas:

- **Definition**: define ou esclarece termos relevantes para a conclusão.
- **QuantitativeData**: evidencia quantitativa que aumenta a verossimilhança da conclusão por meio de dados estatísticos, comparações ou proporções.
- **Event**: relata eventos com marco temporal e efeitos bem definidos, utilizados para sustentar a conclusão.
- **EvidenceOther**: evidência que não se enquadra nas categorias anteriores.

Relações entre Unidades Argumentativas

Para representar relações entre unidades argumentativas (UAs), foram definidos três tipos de rótulos: **group**, **converge** e **diverge**. Cada um desses rótulos estabelece uma relação entre duas sentenças anotadas.

O rótulo **group** indica co-dependência semântica. É utilizado quando duas ou mais sentenças, individualmente incompletas, formam um enunciado coeso ao serem agrupadas. Um exemplo típico é a paráfrase, em que sentenças são semanticamente equivalentes. Outros casos incluem perguntas-resposta (questões-ganchos), complementações por especificação ou generalização. Na prática, o rótulo `group` agrega duas sentenças como uma única entidade argumentativa composta. Exemplos dessa anotação foram documentados no manual de diretrizes, disponível no Anexo E.

A relação **converge** indica que duas sentenças apresentam posicionamentos compatíveis, complementares ou mutuamente reforçadores sobre o mesmo tópico. Elas podem abordar o tópico sob diferentes enfoques, mas mantêm coerência argumentativa. São consideradas unidades independentes, conectadas por compartilharem a mesma linha de raciocínio.

> `(Tópico)` "Pena de Morte"
> `(ClaimFavor)` "A pena de morte é uma forma eficaz de dissuadir crimes graves."
> `(QuantitativeData)` "Estudos mostram que países com pena de morte têm menores taxas de homicídio."
> Relações:
> `(QuantitativeData) → converge → (ClaimFavor)`
> `(ClaimFavor) → converge → (Tópico)`

Neste exemplo, a sentença rotulada como `ClaimFavor` expressa uma conclusão favorável ao tópico "pena de morte" e, por isso, se conecta a ele via `converge`. A sentença rotulada como `QuantitativeData` reforça essa conclusão com uma evidência quantitativa, estabelecendo também uma relação de convergência.

A relação **diverge** indica oposição argumentativa. Duas sentenças divergentes apresentam posições contraditórias, incompatíveis ou excludentes sobre o mesmo tema. A divergência pode ocorrer por meio de refutação explícita, pressupostos conflitantes ou conclusões opostas. Assim como em `converge`, as sentenças são independentes, mas sua conexão expressa contraste argumentativo.

> `(Tópico)` "Pena de Morte"
> `(ClaimFavor)` "A pena de morte reduz os custos do sistema penal."
> `(SupportedClaimAgainst)` "A pena de morte é mais cara que a prisão perpétua, devido aos longos processos judiciais."
> Relações:
> `(SupportedClaimAgainst) → diverge → (ClaimFavor)`
> `(ClaimFavor) → converge → (Tópico)`

Neste exemplo, a sentença `ClaimFavor` apresenta uma conclusão favorável ao tópico, e a sentença `SupportedClaimAgainst` introduz uma evidência empírica que contradiz diretamente esse argumento. A relação entre elas é rotulada como `diverge`, pois expressa conflito argumentativo.

7.2.7 Resultados Parciais da Anotação da Subtarefa MAM1

O esquema de anotação proposto para as unidades argumentativas — conclusões e premissas — e para as relações entre elas foi construído a partir de múltiplos ciclos de ensaio: anotação independente, medição de concordância entre anotadores, adjudicação coletiva e revisão contínua do *guideline*, conforme ilustrado na Figura 7.2. Além desse esquema, foram desenvolvidos outros artefatos, incluindo: (1) um manual de diretrizes da tarefa [7] e (2) uma ferramenta própria para anotação de árvores argumentativas, denominada Argmap, descrita no Anexo F.

Módulo de Anotação de Argumentos

Com o objetivo de viabilizar a anotação de argumentos, foi desenvolvido um módulo específico para essa finalidade. Um dos primeiros passos consistiu no carregamento do conjunto de dados descrito na Seção 7.2.3. A Figura 7.6 apresenta o documento "00", do tópico "aborto", com todas as sentenças marcadas com os rótulos `Sent` (sem marcação argumentativa), `ArgA` (argumento contra) e `ArgF` (argumento a favor). As marcações `ArgA` e `ArgF` foram importadas do *dataset* UKP Sentential.

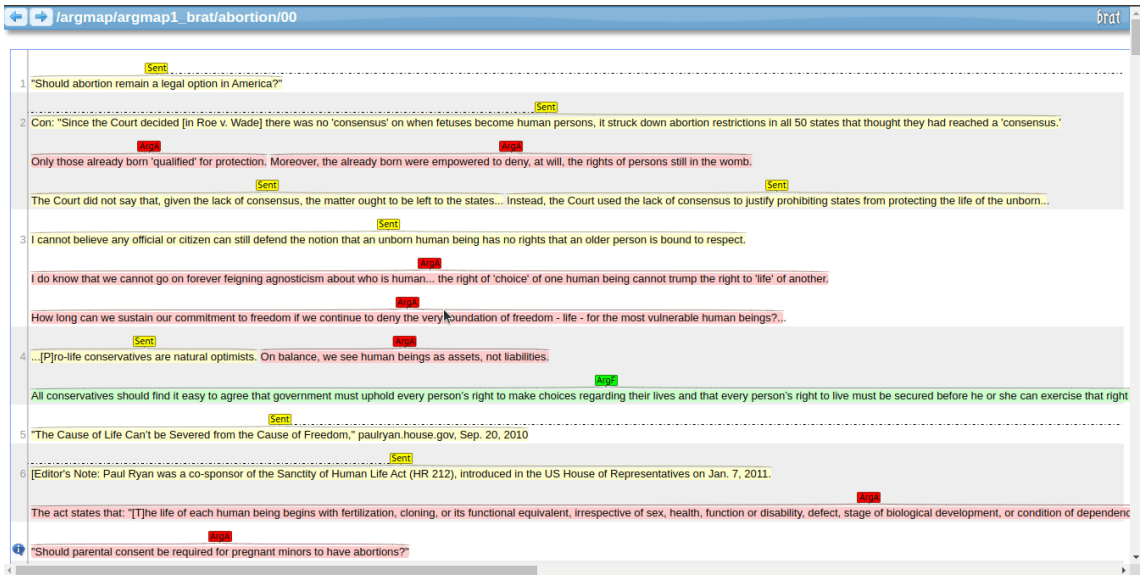


Figura 7.6: *Visualização de documento aberto*

[7] Acessível em: https://argmap.inf.ufg.br/guideline

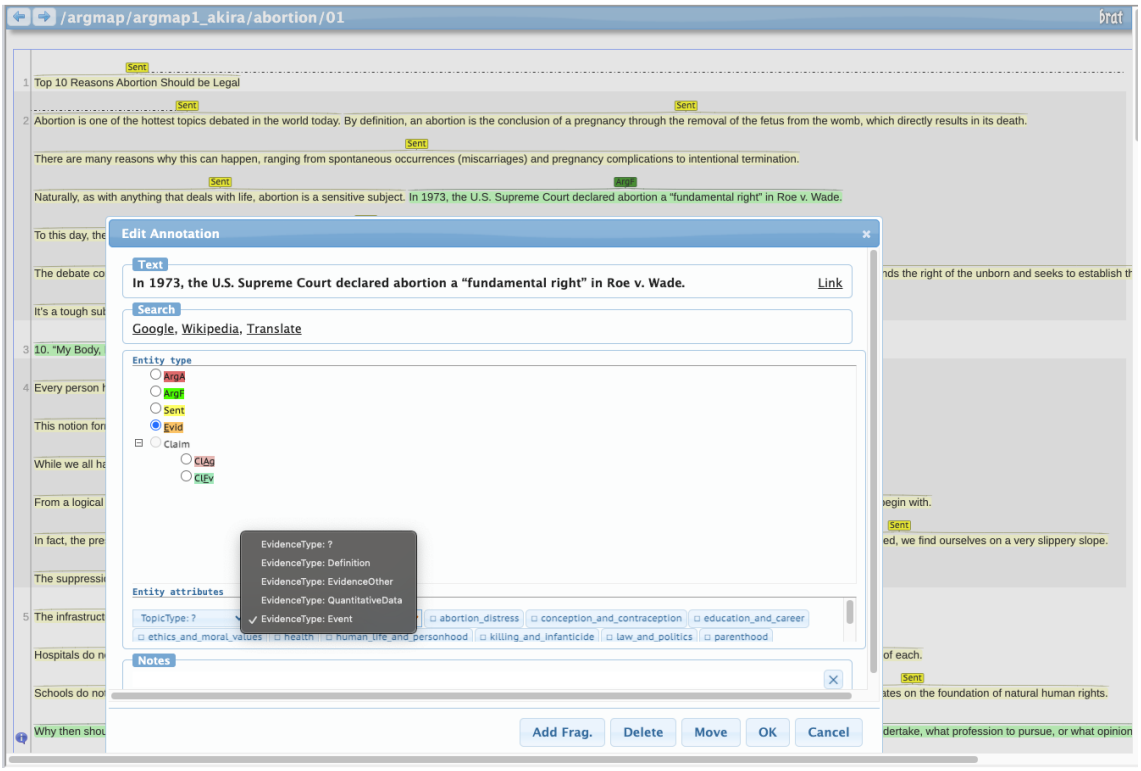


Figura 7.7: *Atribuição do rótulo `Evid` e de um atributo a uma sentença*

Com todas as sentenças do texto previamente rotuladas, não foi necessário que o anotador realizasse a identificação inicial das unidades argumentativas. Dessa forma, sua principal tarefa passou a ser a substituição dos rótulos `ArgA`, `ArgF` ou `Sent` por um dos três rótulos utilizados na classificação final: `Evid` (Evidência), `ClAg` (Conclusão contra) ou `ClF` (Conclusão a favor). Em seguida, dependendo do tipo de rótulo atribuído, o anotador deveria aplicar um atributo correspondente à entidade. Por exemplo, na Figura 7.7, uma sentença rotulada originalmente como `ArgF` foi alterada para `Evid`, com o atributo `Event`.

Outra funcionalidade importante implementada na ferramenta foi a aplicação de relações entre entidades argumentativas. A Figura 7.8 ilustra a seleção de uma relação entre uma sentença rotulada como `Evid` e outra como `ClFv`. À direita da interface de edição, é exibido em tempo real o estado atual da árvore argumentativa, composta pela raiz (tópico `Aborto`), um nó `ClaimFavor` e um nó `Evidence` que o apoia. Os identificadores T1 e T4 correspondem às sentenças envolvidas.

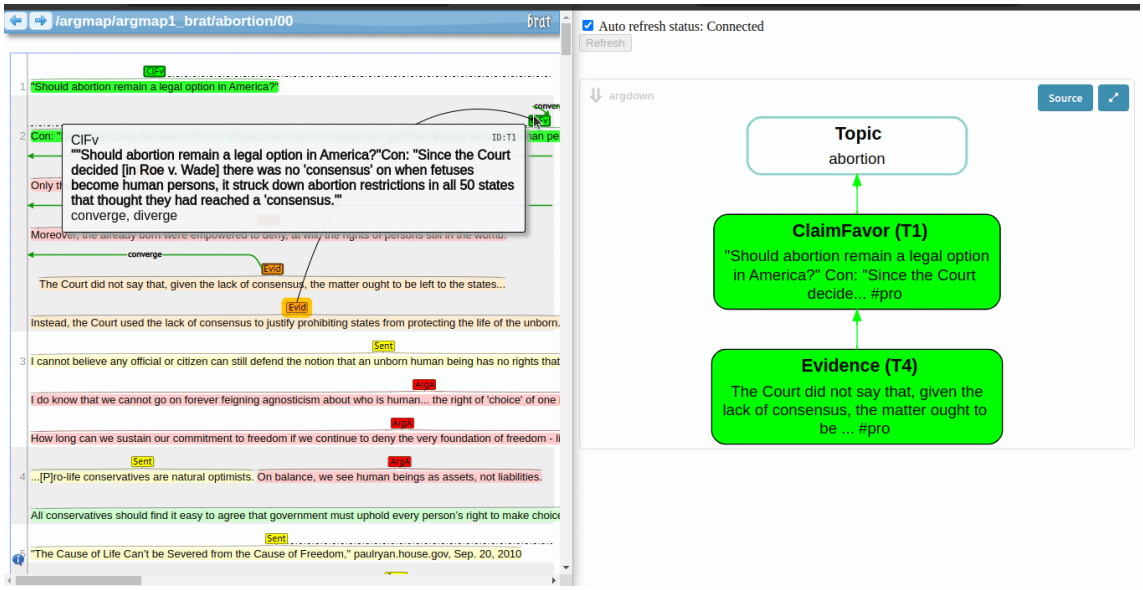


Figura 7.8: *Seleção do argumento de origem e destino da relação*

Na Figura 7.9, é demonstrada a escolha do tipo de relação a ser aplicada entre as sentenças. Neste exemplo, foi selecionada a opção `converge`, indicando que a evidência (T5) oferece suporte à conclusão (T1).

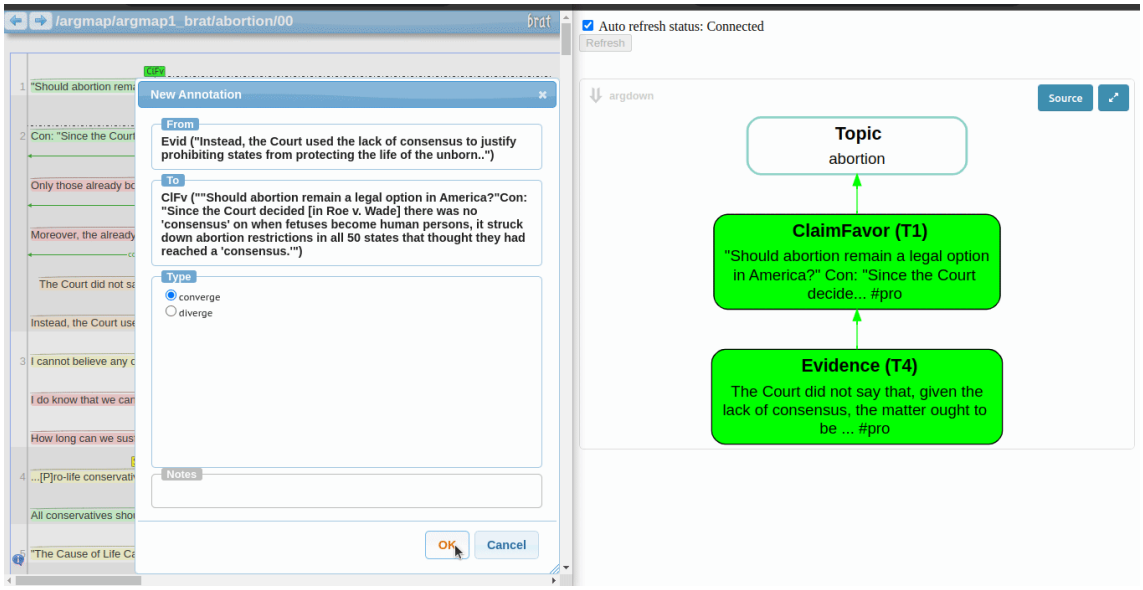


Figura 7.9: *Seleção do tipo de relação entre os argumentos selecionados*

Por fim, conforme mostrado na Figura 7.10, o diagrama correspondente à marcação realizada é automaticamente renderizado na interface, refletindo a estrutura atualizada da árvore, com a adição do novo nó `Evidence (T5)`. A atualização do diagrama ocorre em tempo real.

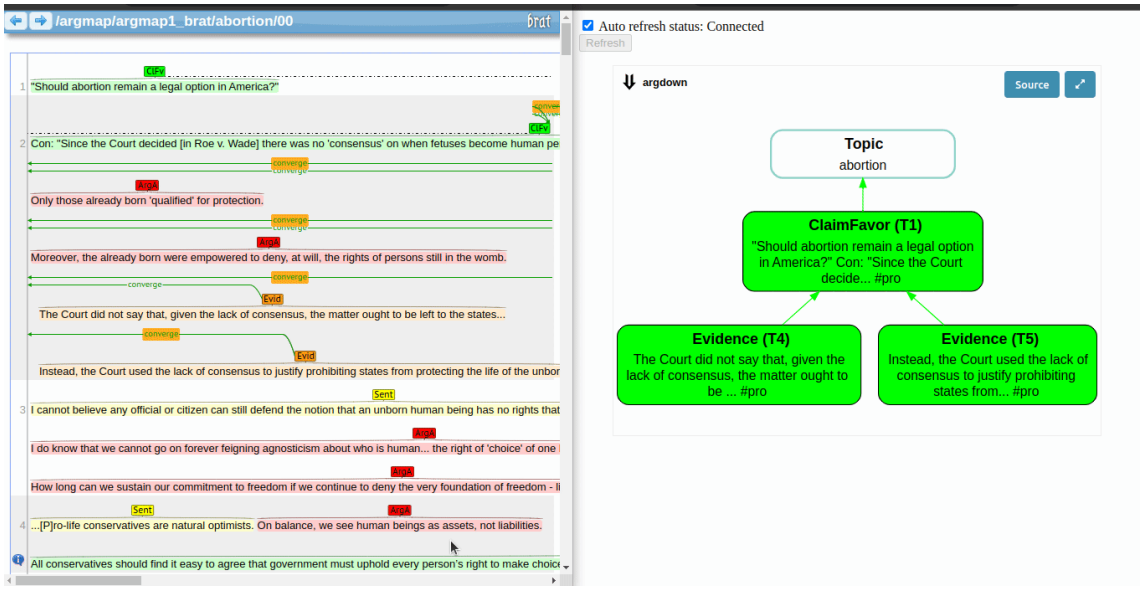


Figura 7.10: *Visualização do resultado da anotação*

A funcionalidade principal do Argmap é a edição de árvores argumentativas. No entanto, outras funcionalidades foram implementadas com o intuito de melhorar a qualidade da anotação, tais como: (1) checagem em tempo real de inconsistências — por exemplo, quando um rótulo **ArgA** ou **ArgF** permanece sem alteração; (2) ferramenta de linha de comando (CLI) para administração — como a atribuição de documentos para anotação por diferentes usuários, entre outras.

Necessidade da Segmentação de Tópicos

Seguindo os passos definidos no Algoritmo 4.1, foram observados avanços iniciais na concordância de anotação (c) da tarefa de MAM1 (MAM1). No entanto, após diversos ciclos, as medidas de concordância entre os anotadores (C) permaneceram instáveis. As atualizações no *guideline*, que inicialmente contribuíram para ganhos de concordância (Δ), passaram a apresentar efeitos reduzidos, indicando baixa confiabilidade em determinados casos.

A partir da análise qualitativa de divergências, identificou-se que a maioria dos casos de baixa concordância ocorria em documentos longos, com mais de 40 argumentos anotados. Esses documentos, embora abordassem um tópico específico — como energia nuclear, por exemplo —, frequentemente apresentavam variações temáticas sutis ao longo do texto, como mudanças entre subtemas relacionados à regulamentação, acidentes ou terrorismo. Quando essas mudanças de enquadramento não eram percebidas pelos anotadores, surgiam divergências na delimitação e estruturação dos argumentos.

Como resposta a esse problema, foram realizados ensaios em que os textos foram segmentados previamente por subtópicos. Essa segmentação permitiu delimitar melhor o escopo de cada trecho, reduzindo ambiguidades interpretativas. A aplicação dessa

estratégia resultou em ganhos de concordância nos segmentos tratados, com impacto positivo na concordância geral por documento.

Com base nessa experiência, o grupo de anotadores especialistas (ver Figura 7.2) deliberou pela necessidade de uma etapa dedicada à identificação e segmentação de tópicos e subtópicos. Essa decisão motivou a formulação da nova tarefa de **Segmentação e Classificação de Tópicos (SCT)**, que será apresentada na próxima seção.

7.3 Segmentação e Classificação de Tópicos (SCT)

Nesta etapa, a segmentação de tópicos foi realizada manualmente por anotadores humanos, com base em critérios linguísticos e discursivos definidos nas diretrizes da anotação. Essa decisão metodológica foi necessária devido à ausência de marcações tópicas explícitas no *corpus Argmap*, bem como à heterogeneidade dos textos, que inviabilizava a aplicação direta de métodos automáticos ou supervisionados de segmentação.

A tarefa de SCT foi incorporada ao fluxo do MAM1 como uma subtarefa intermediária essencial para delimitar e enquadrar o contexto temático em que se inserem as unidades argumentativas (UAs). Essa necessidade tornou-se particularmente evidente na análise de documentos extensos, nos quais a subtarefa de Detecção de Argumentos (DA) resultava na identificação de mais de quarenta UAs, dificultando a interpretação relacional e comprometendo a qualidade da anotação.

Ao promover a segmentação temática e a organização hierárquica dos tópicos abordados no texto, a SCT atua como uma estratégia de apoio à decomposição argumentativa, permitindo circunscrever com maior precisão os espaços de problema e os subcontextos em que as UAs se articulam. Essa organização temática prévia viabiliza não apenas a redução da carga cognitiva sobre os anotadores, como também contribui para uma interpretação mais robusta e fundamentada das relações argumentativas, ao oferecer uma moldura contextual coerente para a aplicação de rótulos relacionais e funcionais nas etapas subsequentes da tarefa.

As próximas seções detalham as etapas que compõem a tarefa de SCT. A Seção 7.3.1 descreve a *Segmentação de Tópicos* (ST), responsável por dividir o documento em segmentos temáticos coerentes. Em seguida, a Seção 7.3.2 apresenta a *Identificação de Gênero Textual* (IGT), empregada para identificar páginas *Web* com estrutura discursiva atípica — tais como propagandas, apresentações (*SlideShare*), entre outras — nas quais a segmentação de tópicos tende a falhar ou gerar resultados inconsistentes. A Seção 7.3.4 aborda a *Classificação de Tópicos* (CT), que consiste na atribuição de múltiplos rótulos semânticos a cada segmento previamente identificado. Por fim, a Seção 7.3.3 descreve a CFC, etapa fundamental para a CT, uma vez que fornece um conjunto de frases-chave

de alta qualidade que subsidia a tarefa de classificação multirrótulo, garantindo maior precisão e coerência na rotulação temática dos segmentos.

7.3.1 Segmentação de Tópicos (ST)

A segmentação de tópicos é adotada nesta pesquisa como uma etapa de suporte à anotação em documentos extensos, motivada pela baixa concordância observada entre anotadores nesse tipo de situação. A análise qualitativa revelou que tais documentos frequentemente transitam por subtópicos ao longo de sua progressão discursiva, o que impacta diretamente na interpretação de unidades argumentativas cuja identificação depende de informações distribuídas ao longo do texto. Essa característica se mostrou particularmente relevante no contexto das tarefas de Detecção de Conclusão Dependente de Contexto (DCDC) e Detecção de Evidência Dependente de Contexto (DEDC), conforme discutido nas Seções 7.2.4 e 7.2.5. A segmentação temática permite isolar regiões textuais coesas, reduzindo ambiguidades interpretativas e favorecendo uma análise mais localizada, de modo a tornar a anotação mais precisa e menos sujeita a variações decorrentes da sobrecarga cognitiva imposta por documentos longos e discursivamente densos.

A segmentação de tópicos é amplamente reconhecida como uma técnica fundamental para a organização de documentos extensos em unidades semanticamente coesas, favorecendo tanto a interpretação humana quanto a aplicação de modelos computacionais com restrições de contexto, como os baseados em *transformers* [Yu et al. 2023]. Também atua como etapa intermediária relevante em *pipelines* de PLN, contribuindo para o desempenho em tarefas como sumarização, classificação e recuperação de informações [Wang et al. 2017]. Além disso, a segmentação tem sido associada à modelagem da coerência textual, ao permitir a identificação de transições tópicas que refletem mudanças discursivas relevantes [Yu et al. 2023]. Por fim, sua aplicação favorece a navegabilidade e a análise contextual em tarefas sensíveis à organização temática, como a análise de sentimentos ou de estruturas argumentativas [Adebayo, Caro e Boella 2016].

Escopo e Definição da Tarefa

Segmentação de tópicos é uma subárea da segmentação de texto voltada à identificação de limites entre unidades temáticas coerentes. Diferente da segmentação de texto convencional, que pode se basear em critérios estruturais (como sentenças ou parágrafos), a segmentação de tópicos exige a detecção de mudanças semânticas, sendo fundamentada em variações de **coesão lexical** e **continuidade discursiva**.

A análise de *coesão lexical*, nesse sentido, oferece um recurso empírico para identificar a progressão temática no texto. O uso de correferência permite observar como entidades são retomadas, generalizadas ou reformuladas ao longo da estrutura discursiva,

contribuindo para a manutenção ou transição de tópicos. No Anexo G, essa relação é explorada por meio da descrição de mecanismos anafóricos, catafóricos e nominais, evidenciando como padrões de retomada lexical podem ser utilizados na delimitação de segmentos temáticos.

A segmentação de tópicos pode ser classificada em duas abordagens principais: linear e hierárquica. A segmentação hierárquica organiza o texto em múltiplos níveis de tópicos e subtópicos, formando uma estrutura aninhada. Já a segmentação linear divide o texto em segmentos contíguos e não sobrepostos, cada um representando uma unidade tópica coerente. Conforme discutido por [Yu et al. 2023], embora ambas sejam relevantes, a segmentação linear é mais adequada para tarefas que exigem delimitação clara de tópicos em sequência. Esta pesquisa adota a segmentação linear, com foco na identificação de limites tópicos em documentos extensos a partir de critérios de coerência e similaridade semântica entre sentenças adjacentes.

A segmentação de tópicos pode ser combinada com a classificação de tópicos multirótulo, em que rótulos previamente definidos são atribuídos a cada segmento textual de forma não exclusiva. Quando aplicadas conjuntamente, essas tarefas podem se beneficiar mutuamente, especialmente na detecção de *continuidade discursiva*. A classificação multirótulo permite identificar sobreposições temáticas entre segmentos adjacentes, funcionando como um indicativo de coerência tópica. Na prática, se dois segmentos consecutivos compartilham o mesmo conjunto de rótulos, isso sugere ausência de mudança de tópico e, portanto, a segmentação entre eles pode ser considerada inválida. A principal contribuição dessa abordagem ocorre no processo de anotação, pois exige que o anotador reflita sobre os principais assuntos abordados em cada segmento, atribuindo-lhes rótulos apropriados, além de observar com atenção quando um subtópico é introduzido e outro deixa de ser desenvolvido. A tarefa de classificação de tópicos será tratada em maior detalhe na Seção 7.3.4.

A abordagem adotada neste trabalho para segmentação e classificação de tópicos é supervisionada, com base em exemplos anotados manualmente. Embora existam alternativas não supervisionadas, como os métodos baseados em modelagem de tópicos (ver Seção B.2.5), a opção por um método supervisionado se justifica pelo potencial da tarefa de anotação em gerar insights e achados relevantes sobre a estrutura tópica dos textos. Além disso, uma análise exploratória revelou alta heterogeneidade entre os documentos, resultado esperado devido à estratégia de coleta — baseada nas primeiras páginas retornadas por um mecanismo de busca — o que limita a eficácia de abordagens não supervisionadas que assumem maior homogeneidade textual.

A tarefa de segmentação de tópicos é formulada neste trabalho como um problema de classificação binária por sentença, conforme proposto por [Yu et al. 2023]. Dado um documento representado como uma sequência de sentenças $\mathbf{d} = [s_1, s_2, \ldots, s_n]$, o ob-

jetivo é prever uma sequência de rótulos binários $\mathbf{y} = [y_1, y_2, \ldots, y_n]$, em que $y_i = 1$ indica que a sentença s_i marca o fim de um segmento. Essa formulação permite treinar modelos supervisionados com base em dados anotados, utilizando representações contextuais para capturar mudanças de tópico ao longo do texto.

Desafios em *Corpus* Heterogêneo

Grande parte dos *datasets* amplamente utilizados para a tarefa de segmentação de tópicos — como o *WIKI-727K* [Koshorek et al. 2018], o *WikiSection* [Arnold et al. 2019], *Elements* e *Cities* [Chen et al. 2009] — é derivada de fontes textuais homogêneas, predominantemente enciclopédicas, como a Wikipedia. Esses *corpora* compartilham características estruturais importantes: apresentam segmentações explícitas por meio de seções e subseções, uma escrita coesa e bem organizada, além de estarem alinhados a um estilo discursivo mais formal e informativo. Essas propriedades facilitam tanto a segmentação automática quanto a extração de padrões estáveis para o treinamento supervisionado de modelos.

Além da uniformidade textual, esses *corpora* também se destacam por seu *alto volume de dados*, sendo os mais relevantes da literatura em termos de escala. O *WIKI-727K*, por exemplo, conta com 727.746 documentos, sendo o maior *corpus* supervisionado já criado para segmentação de tópicos. O *WikiSection* possui 23.129 documentos, segmentados automaticamente por seções de domínio. Mesmo os *corpora* menores, como *Elements* (118 documentos) e *Cities* (100 documentos), mantêm uma estrutura segmentável e aproveitam o estilo enciclopédico da Wikipedia. Em todos os casos, a anotação foi realizada de forma *automática*, explorando diretamente as marcações de seções nos próprios artigos.

Em contraste, a análise exploratória do *corpus Argmap*, desenvolvido nesta tese, revelou um cenário substancialmente mais complexo. Composto por 400 documentos de diversos *gêneros textuais*, o *corpus* apresenta alta heterogeneidade de estilo, tamanho e formatação, frequentemente sem segmentação tópica explícita. Diferentemente dos *corpora* enciclopédicos, cuja segmentação foi viabilizada automaticamente com base em pistas estruturais claras, o *corpus Argmap* exigiu um processo de anotação manual, com base em critérios linguísticos e discursivos interpretativos. Essa abordagem demanda maior atenção analítica e esforço humano, tornando o processo mais trabalhoso e custoso.

Além disso, elementos visuais como tabelas, listas hierárquicas, imagens e comentários embutidos estavam frequentemente presentes nos textos, o que impôs desafios adicionais à segmentação e à anotação manual. Isso se agravou pelo fato de que a raspagem *web* (*web scraping*) utilizada para recuperar o conteúdo das páginas indicadas no *corpus* original (*UKP Sentential*) não lida adequadamente com tais estruturas visuais — especialmente quando organizadas em formato tabular ou em camadas HTML hierárqui-

cas. Como consequência, o texto extraído apresentava uma linearidade artificial e confusa, frequentemente misturando fragmentos disjuntos de conteúdo, o que tornava mais difícil identificar mudanças tópicas ou interpretar o fluxo argumentativo de forma coesa.

Outro ponto importante refere-se ao pré-processamento: nos *corpora* baseados em Wikipedia, há uma etapa clara de remoção de elementos não textuais (como blocos informativos (*infoboxes*), figuras e listas), que contribui para a fluidez textual. Já no *Argmap*, isso não foi possível, justamente por depender de raspagem a partir de fontes *web* com estrutura inconsistente e muitas vezes orientada à exibição visual, não textual.

Corpus	Estratégia de Anotação	Docs	Observações
WIKI-727K	Automática (seções da Wikipedia)	727K	Maior *corpus* supervisionado. Segmentação baseada em estrutura hierárquica de artigos.
WikiSection	Automática (domínios com seções)	23K	Subconjuntos `en_disease` e `en_city`. Ênfase em coerência local e tópicos de domínio.
WIKI-50	Amostra manual do WIKI-727K	50	Usado para avaliação qualitativa e testes com humanos.
Elements	Automática (Wikipedia – elementos químicos)	118	Estrutura padronizada com base em seções típicas de artigos científicos.
Cities	Automática (Wikipedia – cidades)	100	Similar ao *Elements*, com foco em descrições geográficas e históricas.
Choi (2000)	Sintética (concatenação de trechos)	920	Criado artificialmente a partir de segmentos desconexos do *corpus* Brown.
Manifesto	Manual	5	Textos políticos com variação estrutural e baixa segmentação explícita.
Argmap	Manual (gêneros diversos, sem estrutura clara)	400	*Corpus* heterogêneo desenvolvido nesta tese. Inclui textos com variação estilística e estrutural.

Tabela 7.7: *Comparação entre corpora utilizados em segmentação de tópicos. WIKI-727K [Koshorek et al. 2018], WikiSection [Arnold et al. 2019], Elements e Cities [Chen et al. 2009] são corpora enciclopédicos construídos automaticamente a partir da Wikipedia. Choi (2000) [Choi 2000] e Manifesto [Glavaš, Nanni e Ponzetto 2016] representam fontes não enciclopédicas com estratégias sintéticas ou manuais. Argmap é corpus manual desenvolvido nesta tese.*

Para ilustrar essas diferenças, a Tabela 7.7 apresenta um panorama comparativo dos principais *corpora* de segmentação de tópicos, indicando origem, estratégia de anotação e tamanho. Essa distinção evidencia a necessidade de metodologias mais adaptativas para lidar com *corpora* heterogêneos. Enquanto a estrutura formal e segmentada dos textos enciclopédicos favorece a construção de *datasets* em larga escala com anotação automatizada, *corpora* como o *Argmap* exigem abordagens manuais, mais refinadas e sensíveis aos desafios de descontinuidade, multimodalidade e variação discursiva.

Considerando a heterogeneidade do *corpus Argmap*, a identificação do gênero textual de cada documento torna-se um recurso relevante para lidar com variações de estilo e organização discursiva. O reconhecimento do gênero permite antecipar padrões estruturais e argumentativos característicos, contribuindo para estratégias mais informadas de segmentação e anotação. A seção a seguir apresenta os principais trabalhos sobre Identificação de Gênero Textual (IGT), com ênfase em suas abordagens metodológicas e aplicações em contextos multigênero.

7.3.2 Identificação de Gênero Textual (IGT)

A identificação automática de gêneros textuais constitui uma etapa relevante em tarefas de PLN aplicadas a *corpora* heterogêneos, contribuindo para o pré-processamento, a segmentação textual e a filtragem de documentos fora do escopo. Neste trabalho, foram explorados gêneros recorrentes na *Web*, como *blogs*, notícias, tutoriais e fóruns, com o objetivo de caracterizar seus padrões formais e funcionais. Também foram identificados documentos considerados irregulares — por exemplo, anúncios de livros e apresentações de slides (como os encontrados no *SlideShare*) — que, devido à sua estrutura atípica ou fragmentada, inviabilizam a segmentação coerente de tópicos. Adicionalmente, considerando que a tese se concentra na análise de textos argumentativos, o gênero textual é investigado como um fator potencialmente determinante na forma de construção argumentativa, especialmente no que se refere ao posicionamento discursivo (neutro, pró, contra, totalmente pró ou totalmente contra). Para aprofundar a análise, foram extraídas capturas de tela dos documentos, permitindo a incorporação de informações visuais na avaliação da estrutura textual, caracterizando-se, assim, como um estudo de natureza multimodal.

A tarefa de *Identificação de Gênero Textual* (IGT) é relevante para o PLN (PLN) por sua capacidade de atribuir contexto funcional e estrutural aos textos, favorecendo a eficiência de sistemas computacionais em tarefas como recuperação de informação, sumarização, classificação e extração de dados. Gêneros textuais operam como molduras cognitivas que ativam expectativas sobre a organização e o propósito comunicativo do texto, contribuindo para a redução da carga cognitiva do leitor [Mehler, Sharoff e Santini 2011]. Essa previsibilidade é considerada uma das propriedades fundamentais dos gêneros [Kuzman e Ljubešić 2023] e tem sido explorada para melhorar a navegação e a filtragem de conteúdos em ambientes digitais marcados por alta variabilidade [Mehler, Sharoff e Santini 2011]. A literatura também destaca que o uso de rótulos de gênero pode reduzir ruído em sistemas de busca e facilitar a categorização de documentos em coleções extensas [Kessler, Nunberg e Schutze 1997]. Além disso, a IGT tem se mostrado útil na análise e curadoria de grandes *corpora* utilizados para o treinamento de modelos de linguagem, permitindo maior controle sobre a diversidade textual e

os vieses associados [Kuzman e Ljubešić 2023].

No domínio da argumentação computacional, a consideração do gênero textual é fundamental para a modelagem, anotação e avaliação de argumentos. A organização dos textos, os estilos discursivos e os propósitos comunicativos variam entre gêneros e afetam diretamente a estrutura argumentativa. Wachsmuth et al. [Wachsmuth et al. 2017] destacam que dimensões de qualidade argumentativa — lógica, retórica e dialógica — são sensíveis a essas variações, o que implica a necessidade de abordagens diferenciadas conforme o tipo de texto. Cardoso et al. [Cardoso et al. 2023] reforçam esse ponto ao apontar que a aplicação de modelos argumentativos em tarefas de anotação demanda adequações aos diferentes gêneros, especialmente em contextos como jornalismo, ciência e direito. Pimenov e Salomatina [Pimenov e Salomatina 2024] mostram empiricamente que certos esquemas argumentativos são mais produtivos ou recorrentes em função do gênero, o que afeta a complexidade e a frequência dos padrões identificados. Esses resultados evidenciam que a Identificação de Gênero Textual (IGT) contribui para a interpretação mais precisa de estruturas argumentativas e pode aprimorar tarefas como anotação automática e avaliação da qualidade argumentativa em diferentes contextos discursivos.

O *dataset Argmap* está sendo construído com base na coleta de textos da *Web* por meio de buscas não filtradas, em função do aproveitamento de rótulos do *dataset* UKP Sentential. Esse processo não contempla critérios prévios de viabilidade argumentativa, o que resulta em um *corpus* heterogêneo, com documentos que, em muitos casos, não apresentam estrutura mínima para a formação de uma árvore argumentativa coerente. Essa limitação impacta a qualidade da anotação e pode comprometer a consistência dos dados. Nesse contexto, a Identificação de Gênero Textual (IGT) torna-se uma etapa relevante não apenas para a categorização dos textos e o enriquecimento de análises estatísticas, mas também para a identificação de documentos inviáveis para fins de anotação. Ao atuar como filtro complementar, a IGT contribui para a curadoria do *corpus*, favorecendo a consistência dos dados e a robustez das análises em documentos com qualidade argumentativa mínima aferida.

Conceitos e Definições

A tarefa de *Identificação Automática de Gêneros Textuais (IAGT)* (IAGT) tem sido explorada em diferentes contextos e domínios, com definições que variam conforme os propósitos comunicativos, as convenções formais e os tipos de *corpus* considerados. Neste trabalho, o foco está na aplicação da IAGT a textos argumentativos, os quais apresentam características particulares de organização e estilo. Esta subseção discute os conceitos fundamentais relacionados à tarefa, diferenciando-a de tarefas correlatas, como

a classificação por assunto, e destacando suas especificidades em contextos digitais e dinâmicos, como a *Web*.

Nos trabalhos analisados, a *IAGT* (IAGT) é definida como a tarefa de reconhecer a categoria textual de um documento com base em regularidades formais e funcionais. [Kessler, Nunberg e Schutze 1997] tratam gênero como uma classe distinta de estilo e tópico, associada a padrões estruturais recorrentes. [Lee e Myaeng 2002] o definem como uma categoria percebida pelos leitores, caracterizada por traços formais e funcionais compartilhados. Para [Kuzman e Ljubešić 2023], gênero corresponde a uma função comunicativa atribuída a textos individuais em contextos específicos, como jornalismo ou fóruns online. [Kuzman, Mozetič e Ljubešić 2023] enfatizam sua natureza sociocomunicativa, ligada a padrões estáveis de produção textual. [Mehler, Sharoff e Santini 2011] ampliam essa concepção ao entenderem gênero como uma configuração recorrente de convenções que articula forma, conteúdo e função, permitindo sua identificação em ambientes digitais.

A *IAGT* (IAGT) difere da classificação de textos por assunto ou tópico ao focar não no conteúdo temático, mas na função comunicativa e na organização estrutural do texto. Enquanto a classificação por assunto identifica sobre o que o texto trata, a IAGT busca reconhecer categorias como notícia, resenha ou post de *blog*, independentemente do tema. [Kessler, Nunberg e Schutze 1997] formalizam essa distinção ao separar gênero, tópico e estilo, definindo gênero como uma categoria associada a padrões formais e funcionais recorrentes. Essa diferenciação é essencial, dado que textos com o mesmo assunto podem pertencer a gêneros distintos e vice-versa.

A tarefa de *IAGT* (IAGT) apresenta características distintas em textos gerais e em textos extraídos da *Web*. Em *corpora* tradicionais, os gêneros tendem a seguir convenções estáveis e categorias bem definidas. Na *Web*, por outro lado, os gêneros são mais dinâmicos, híbridos e sujeitos a mudanças frequentes, o que exige esquemas de classificação mais flexíveis. [Mehler, Sharoff e Santini 2011] destacam que os gêneros *web* emergem de configurações recorrentes de convenções comunicativas, mas essas configurações evoluem rapidamente conforme se transformam as plataformas e práticas de interação. A IAGT na *Web*, portanto, demanda abordagens sensíveis à variabilidade e instabilidade dos gêneros.

As definições e distinções apresentadas nesta subseção oferecem a base conceitual necessária para o exame da IAGT em textos argumentativos. A seguir, realiza-se um levantamento de *datasets* utilizados na tarefa, com ênfase tanto em *corpora* de propósito geral quanto em coleções específicas voltadas à análise da estrutura argumentativa.

Datasets

A construção de conjuntos de dados para a tarefa de IAGT (IAGT) envolve desafios conceituais e metodológicos específicos. Diferentemente de tarefas mais consolidadas no PLN, como classificação temática ou análise de sentimentos, a IAGT exige a definição de esquemas de anotação que capturem aspectos funcionais, estruturais e contextuais dos textos. Esses esquemas podem assumir diferentes formas, como taxonomias categóricas, hierárquicas ou dimensionais. A coleta dos dados geralmente é realizada a partir da *web* e envolve estratégias de amostragem e filtragem, seguidas por processos de anotação manual ou via *crowdsourcing*, com validação por especialistas. A heterogeneidade dos gêneros na *web*, a ambiguidade entre classes e a variação entre línguas tornam a curadoria desses recursos particularmente complexa. Entre os esforços recentes, destacam-se os conjuntos CORE, FTD, GINCO e X-GENRE, que combinam qualidade de anotação, diversidade linguística e esquemas complementares, sendo amplamente utilizados em avaliações monolíngues e multilíngues (ver Tabela 7.8).

Nome	**Língua(s)**	**Tam.**	**Esquema de Gêneros**	**Observações**
CORE	EN	48.000	8 principais / 47 sub	Usado em estudos recentes; base para extensões multilíngues. Anotado via *crowdsourcing*.
FTD	EN, RU, AR	1.562 (EN)	10 dimensões funcionais	Multi-rótulo com escala (0–2); disponível para análise funcional em três línguas.
GINCO	SL	1.000	24 categorias	*Corpus* esloveno com anotação manual consistente; usado em avaliações monolíngues e multilíngues.
X-GENRE	EN, SL	2.956	Integra CORE, FTD, GINCO	Criado para experimentos de robustez entre esquemas e idiomas.

Tabela 7.8: *Conjuntos de dados de gêneros textuais disponíveis publicamente com pelo menos 500 instâncias. As siglas de idioma indicam: EN = Inglês, SL = Esloveno, RU = Russo, AR = Árabe. Os conjuntos de dados foram originalmente descritos nos seguintes trabalhos: CORE em [Egbert, Biber e Davies 2015], FTD em [Sharoff 2018], GINCO em [Kuzman, Rupnik e Ljubešić 2022] e X-GENRE em [Kuzman e Ljubešić 2023].*

Os conjuntos de dados utilizados na tarefa de IAGT adotam diferentes esquemas

de anotação, que variam conforme a concepção adotada de gênero textual. No esquema categórico, representado pelos *corpora* CORE e GINCO, cada texto recebe um único rótulo associado a uma taxonomia predefinida e mutuamente exclusiva. O CORE, por exemplo, adota uma taxonomia hierárquica com oito categorias principais e 47 subcategorias. Já o esquema dimensional, utilizado no FTD, atribui múltiplos rótulos a cada texto com base em dimensões funcionais (como instrução, opinião ou propaganda), anotadas em escala ordinal. Essa abordagem busca capturar a natureza multifuncional dos textos na *web*. O *corpus* X-GENRE combina textos dos três recursos anteriores e aplica múltiplos esquemas de anotação em paralelo, permitindo a avaliação da robustez de modelos e da compatibilidade entre taxonomias distintas. Uma visão geral desses conjuntos está apresentada na Tabela 7.8.

Os conjuntos CORE, FTD, GINCO e X-GENRE foram construídos a partir de textos coletados da *web*, com diferentes estratégias de anotação. O CORE adota uma abordagem *bottom-up*, com anotação via *crowdsourcing* baseada em descrições situacionais e posterior indução de categorias. O FTD utiliza um esquema dimensional com múltiplas categorias funcionais anotadas em escala ordinal, aplicado por especialistas a subconjuntos de *corpora web*. O GINCO reúne textos em esloveno anotados manualmente com base em uma taxonomia categórica de 24 gêneros. O X-GENRE integra amostras dos três *corpora* e aplica múltiplos esquemas de anotação em paralelo, com foco na avaliação de robustez entre classificadores e taxonomias.

Os conjuntos de dados analisados refletem diferentes concepções teóricas e estratégias de anotação para a tarefa de IAGT, com categorias que abrangem aspectos informacionais, opinativos, interacionais e funcionais. No entanto, tais recursos não foram concebidos para lidar com textos argumentativos oriundos de coletas abertas e não filtradas, como é o caso do *corpus Argmap*. Nesse contexto, tornou-se necessário construir um novo conjunto de dados de IAGT voltado à caracterização dos textos presentes no *Argmap*, com o objetivo de subsidiar as etapas posteriores de anotação argumentativa. A próxima subseção descreve esse processo de construção, incluindo os critérios de seleção textual e o esquema de anotação adotado.

Processo de Anotação

A tarefa de anotação de gêneros textuais no *corpus Argmap* consistiu na atribuição de um único rótulo a cada um dos 400 documentos, totalizando 1200 decisões distribuídas entre três anotadores especialistas. Por se tratar de uma atividade de menor complexidade em relação às demais tarefas do projeto, optou-se por uma equipe reduzida. Todas as subetapas pertinentes do processo de refinamento descritas na Figura 7.2 foram aplicadas. A anotação foi apoiada por recursos multimodais, incluindo capturas de tela das páginas originais e o fornecimento das respectivas URLs, com o objetivo de preser-

Dataset	Categoria Principal	Subcategorias ou Dimensões
CORE	Informational	Encyclopedia article, Biography, FAQ, Reference work, Scientific article
CORE	Opinionated	**Editorial**, **Blog**, **Review**, **Letter to the editor**, **Personal opinion essay**
CORE	Interactive	**Forum post**, **Comment section**, Live chat, Email thread, Q&A site
CORE	Instructional	How-to guide, Recipe, Troubleshooting, Manual, Tutorial
CORE	News	News report, Police report, Sports news, Weather update, Press release
CORE	Legal	Law, Contract, Terms & conditions, Policy statement, Regulation
CORE	Promotional	Advertisement, Event promotion, Product description, Commercial page, Sponsored content
CORE	Other	Homepage, Portal page, Navigation page, Error message, Uncategorized
FTD	Functional Dimensions	**Opinion**, **Evaluation**, **Persuasion**, Instruction, Narration, Description, Information
GINCO	GINCO Categories	**Opinion**, **Discussion**, **Comment**, News, Instruction, Legal, Promotion
X-GENRE	X-GENRE Categories	**Opinion/Argumentation**, **Forum**, Information/Explanation, Instruction, Legal, News, Promotion, Prose/Lyrical, Other

Tabela 7.9: *Categorias e subcategorias (ou dimensões funcionais) dos principais datasets utilizados na tarefa de IAGT. As categorias com potencial argumentativo estão destacadas em negrito.*

var informações contextuais e visuais relevantes. Como resultado, foram produzidos um *dataset* anotado e um *guideline* de IAGT (IAGT) voltado a textos argumentativos da *Web*.

Em cada ciclo de anotação, dois tópicos foram selecionados, sendo os primeiros "aborto" e "energia nuclear". O *guideline* (Anexo H) foi refinado a cada rodada, resultando no esquema apresentado na Tabela 7.10 e na sistematização das características dos gêneros na Tabela 7.11. Esta última contribuiu para decisões mais precisas, ao apoiar os anotadores na identificação de padrões linguísticos e estruturais recorrentes, fortalecendo a consistência e a confiança na atribuição dos rótulos.

Para garantir maior qualidade e menor viés, o conjunto de dados foi anotado por três anotadores, sendo que, em cada tópico, um deles atuava como adjudicador de forma alternada. A adjudicação consistia, geralmente, na análise dos rótulos discordantes, com a aceitação de uma das anotações como correta ou a confirmação dos rótulos concordantes. Em casos mais raros, o adjudicador podia atribuir um rótulo distinto dos previamente anotados. Esse processo, denominado *curadoria manual*, corresponde a um dos padrões

de processo descritos na Seção 3.1 e contribui diretamente para a qualidade da anotação, especialmente no que diz respeito à sua precisão e imparcialidade.

Categoria	Subcategorias / Descrição
Out of Scope	• **announcement**: anúncio de produtos ou serviços. • **index or aggregator**: índice de links ou resumos. • **presentation**: slides ou apresentações visuais.
Monological	• ***blog* post or editorial**: texto opinativo com uso de primeira pessoa e juízos de valor. • **talk transcription**: transcrição de palestra ou fala sem debate. • **manifesto (opcional)**: apelo opinativo de indivíduos ou coletivos.
Dialogical	• **debate**: troca de argumentos entre dois ou mais autores. • **critical review or technical report**: análise crítica ou parecer técnico sobre outro conteúdo.
Irregular	• **threaded posts**: sequência de postagens (ex.: redes sociais). • **interview**: entrevista com turnos longos de fala. • **clipping**: coletânea de trechos de fontes diversas. • **term**: verbete enciclopédico (ex.: Wikipedia). • **tutorial or guide**: texto didático com argumentos pró/-contra. • **news**: notícia factual e não-opinativa.

Tabela 7.10: *Categorias e subcategorias de gêneros textuais. Os detalhes desse esquema estão no Guideline disponibilizado no Anexo H*

Essa maior clareza na percepção dos padrões linguísticos também evidenciou

#	Critério	*Blog* or Editorial	Clipping	Critical Review	Tutorial or Guide	News	Talk Transcript.	Interview	Debate	Threaded Posts	Manifesto	Term
1	Dialogicidade Predominante	M	I	D	I	I	M	I	D	I	M	I
2	Pronome/Pessoas do Discurso	1P	SP	1P	3P	3P	1P	1P	1P	1P	AP	AP
3	Alto Emprego de Adjetivos	S	S	D (9)	N	D (8)	S	D (10)	S	S	A	N
4	Alto Emprego de Evidências e Fatos	D (10)	A	Alto	D (16,19)	S	A	A	D (10)	D (10)	A	S
5	Alto Emprego de Opinião	S	A	A	A	D (8)	A	A	A	A	A	N
6	Conclusão Principal	S	N	A	N	N	A	N	N	N	S	N
7	Presença de tom didático	N	N	N	S	N	N	N	N	N	S	S
8	Discurso	D	D	I	I	I	I	D	D	D	D	I
9	Uso de ironia ou chacota	D (16)	A	S	N	N	N	A	S	A	A	N
10	Posicionamento Pessoal-Subjetivo	A	A	S	N	N	A	S	S	S	S	N
11	Alternância de turnos de fala	N	N	N	N	N	N	S	S	S	N	N
12	Uso de referências ou citações	N	A	A	S	S	A	N	N	N	N	S
13	Tendência de posicionamento	U	B	U	U,N	N	U	U	B	U,B	U	N
14	Segmentação de Tópicos	S	E,C,S	N	E,C						S	S
15	Reporta fatos temporais	N	N	N	N	S	A	A	S	A	A	S
16	Objetivo de refutação	A	D (13)	S	A	N		A	S	A	S	N
17	Multiautoria	A	S	N	N	N	N	N	S	S	N	N
19	Estilo de escrita	A, I	E	A, I	E	N, E	N, E	A	A	I	I	E, D

Tabela 7.11: *Características linguísticas por gênero textual.* ***Legenda: Geral:*** *S = Sim, N = Não, A = Ambos (S/N), D = Depende de #;* ***Dialogicidade Predominante:*** *M = Monológica, I = Irregular, D = Dialógica;* ***Pronome/Pessoas do Discurso:*** *1P = Primeira Pessoa, 3P = Terceira Pessoa, AP = Apessoal, SP = Sem Predominância;* ***Discurso:*** *D = Direto, I = Indireto Livre;* ***Tendência de posicionamento:*** *U = Unilateral, B = Bilateral, N = Neutro;* ***Segmentação de Tópicos:*** *E = Enumerativa, C = Comparativa, S = Seccionada;* ***Estilo de Escrita:*** *E = Expositiva, A = Argumentativa, I = Injuntiva, N = Narrativa, D = Descritiva;* ***Observações:*** *As categorias* Announcement, Index or Aggregator, Presentation *e* OutOfScope *não possuem estrutura argumentativa viável e ficaram fora dessa tabela.*

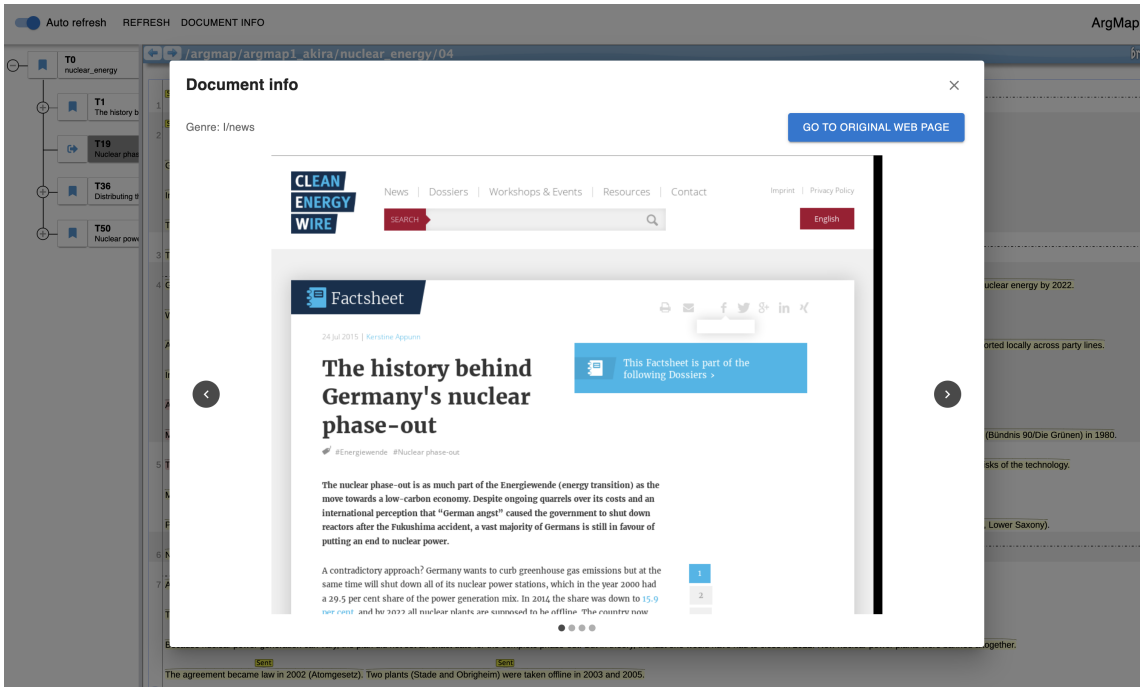


Figura 7.11: *Recurso de visualização da página web original: ao clicar no botão "DOCUMENT INFO", uma página pop-up aparece com a informação do gênero da página, na figura é indicado como I/news (Irregular/-News) e botões de navegação para acessar a captura de tela anterior (<) e a próxima (>). O botão "GO TO ORIGNAL WEB PAGE"abre a página web original deste documento.*

a necessidade de considerar aspectos visuais da página durante o processo de anotação. A identificação do gênero textual em páginas da *Web* exige atenção à multimodalidade dos documentos, uma vez que parte relevante das informações pode ser comprometida durante o processo de raspagem (*scraping*) do conteúdo textual. Elementos como legendas de figuras, cabeçalhos de tabelas, listas hierárquicas e árvores de comentários (como *threaded posts*) frequentemente perdem sua organização visual e semântica, dificultando a análise baseada apenas no texto plano. Para mitigar essa perda, foi incorporado ao processo de anotação o recurso de visualização multimodal descrito na Figura 7.11. A funcionalidade “DOCUMENT INFO” permite o acesso à captura de tela original da página, fornecendo aos anotadores informações visuais complementares — como estrutura, layout e contexto — que auxiliam a reconhecer padrões formais e funcionais próprios de determinados gêneros. Essa estratégia contribuiu significativamente para melhorar a qualidade das anotações, especialmente nos casos em que a estrutura visual era determinante para a identificação do gênero.

O processo de anotação descrito, baseado em ciclos iterativos, refinamento contínuo do *guideline* e suporte multimodal, permitiu maior consistência e segurança

na atribuição dos rótulos de gênero textual. A incorporação de recursos visuais, como o "DOCUMENT INFO", mostrou-se fundamental para recuperar indícios estruturais perdidos na raspagem e apoiar decisões mais informadas. A seguir, são apresentados os resultados quantitativos desse processo, incluindo os níveis de concordância entre anotadores e testes preliminares com modelos de classificação automática, utilizados como *baseline* para a tarefa de IAGT.

Resultados da Anotação

A Tabela 7.12 apresenta os resultados da distribuição dos documentos por classe de gênero textual e as métricas de IAA, representadas por κ (Cohen's Kappa) e F1-score. A análise evidencia um **desbalanceamento expressivo** entre as classes. As três categorias mais frequentes — *Tutorial (Tu)*, *Blog (Bl)* e *News (Ne)* — concentram, juntas, **74,6%** dos exemplos anotados, o que representa uma dominância clara sobre as classes minoritárias. Este cenário pode afetar a qualidade geral da anotação e, principalmente, as métricas de avaliação baseadas em distribuição.

Tópico	**Classes de Gênero (acrônimos)**												**IAA**	
	Tu	Bl	Ne	OS	Cl	Re	Th	Te	Ma	In	Db	Tk	κ	**F1**
• Abortion	7	15	12	3	8	1	0	2	1	1	0	0	0.65	0.72
• Cloning	24	12	5	3	0	4	2	0	0	0	0	0	0.44	0.55
• Death penalty	13	13	11	4	4	3	0	0	1	0	0	1	0.78	0.80
• Gun control	19	16	4	8	1	1	0	0	0	1	0	0	0.33	0.48
• Marijuana legalization	7	12	18	2	2	3	2	2	2	0	0	0	0.55	0.63
• Minimum wage	16	19	12	1	0	0	1	1	0	0	0	0	0.64	0.77
• Nuclear energy	9	13	8	5	5	3	4	2	0	0	1	0	0.67	0.72
• School uniforms	21	9	4	10	2	1	1	1	0	1	0	0	0.37	0.49
Proporção	28.9%	27.2%	18.5%	9.0%	7.5%	4.0%	2.5%	2.0%	1.0%	0.8%	0.2%	0.2%	–	–
Total	116	109	74	36	30	16	10	8	4	3	1	1	0.57	0.65

Tabela 7.12: *Distribuição por tópico e classes de gênero textual com respectivas métricas e proporções.*
***Acrônimos**: Tu = Tutorial, Bl = Blog, Ne = News, OS = Out of Scope, Cl = Clipping, Re = Review, Th = Threaded Posts, Te = Term, Ma = Manifesto, In = Interview, Db = Debate, Tk = Talk.*
***IAA** = Inter-annotator Agreement; κ = métrica de Cohen's Kappa.*

Com o objetivo de compreender melhor as causas das discordâncias na anotação de gêneros textuais, foi realizada uma análise sistemática das matrizes de confusão resultantes do processo de anotação. As matrizes foram utilizadas tanto em sua forma

agregada (ver Figura 7.12) quanto desmembradas por tópico específico (ver Figura 7.13), permitindo identificar padrões recorrentes de confusão entre pares de classes.

Considerando os resultados gerais, observou-se confusão significativa entre dois pares de classes, com erros de classificação que aparecem de forma simétrica na matriz geral. O primeiro par relevante é composto por *blog or editorial* e *tutorial or guide*, cuja confusão mútua também é evidente nas matrizes por tópico, especialmente nos temas *gun_control*, *school_uniforms* e *nuclear_energy*. Uma análise qualitativa dessas instâncias revelou que essa confusão está frequentemente associada a textos híbridos — muitos *blogs* apresentavam objetivos didáticos, com conteúdos explicativos estruturados de forma semelhante a tutoriais, dificultando a distinção clara entre os gêneros com base apenas em critérios formais.

O segundo par que apresentou confusão recorrente foi entre os gêneros *news* e *blog or editorial*. Em diversos casos, os textos exibiam uma combinação de características: autores de *blogs* pessoais adotavam um tom impessoal e objetivo, cobrindo eventos de forma próxima ao estilo noticioso, enquanto alguns portais jornalísticos apresentavam textos com forte carga opinativa, aproximando-se do gênero editorial. Esses casos evidenciam a natureza fluida dos gêneros em contextos digitais e os desafios associados à sua anotação em ambientes onde as fronteiras entre formatos tradicionais de comunicação frequentemente se sobrepõem.

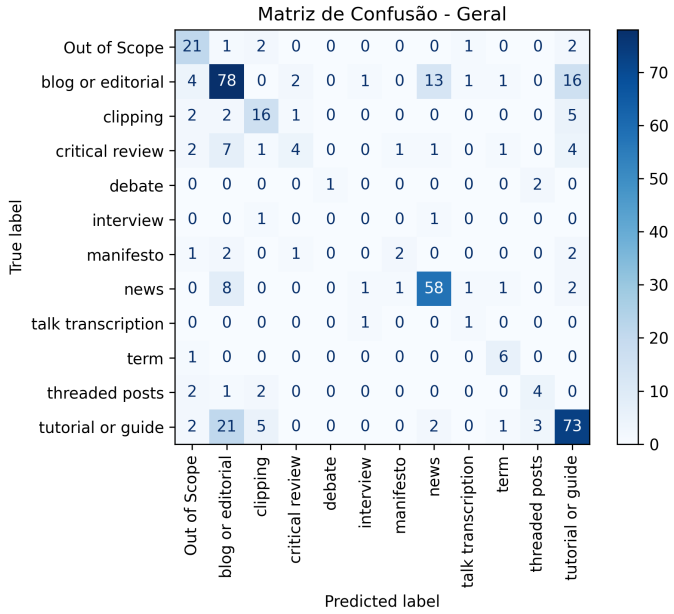


Figura 7.12: *Matriz de confusão geral*

Outro ponto de destaque é a quantidade significativa de documentos **inviáveis para anotação**, representados pela classe *Out of Scope (OS)*, que compreende **9% do total (36 documentos)**. Estes textos não apresentam uma *estrutura tópica ou argumentativa adequada* para o escopo da tarefa de **Segmentação de Tópicos**, o que compro-

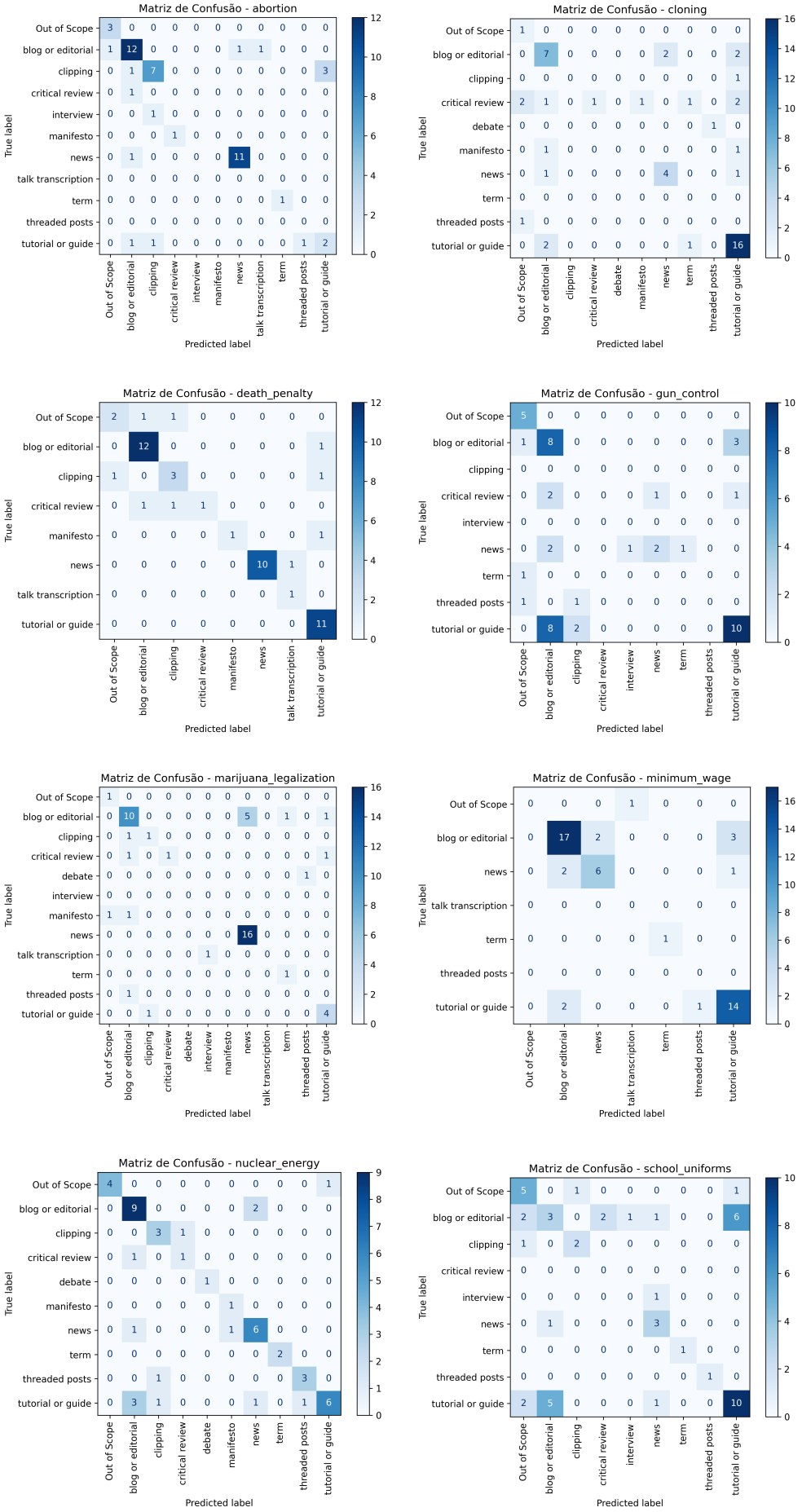


Figura 7.13: *Matrizes de confusão específicas por tópico, organizadas em pares.*

meteria diretamente a aplicação dos resultados em tarefas *downstream*, como extração de argumentos ou sumarização. Além disso, outras classes que merecem atenção por sua presença relevante e **potencial impacto negativo na confiabilidade da anotação das tarefas downstream** são *Clipping (Cl)*, com **7,5%**, e *Threaded posts (Th)*, com **2,5%**. A natureza fragmentada ou não discursiva desses gêneros pode comprometer a aplicabilidade do esquema de anotação e será analisada com maior profundidade em trabalhos futuros.

Em termos gerais, as métricas de concordância são satisfatórias. A **média de Cohen's Kappa foi de** 0,57 e o **F1-score médio alcançou** 0,65, valores que indicam um *nível de acordo considerado bom* entre os anotadores, especialmente diante da diversidade de gêneros e da subjetividade envolvida na tarefa. Ressalta-se que, devido ao desbalanceamento entre as classes, o **F1-score é mais adequado** para refletir a qualidade da anotação em contextos assimétricos, uma vez que pondera precisão e revocação sem ser influenciado diretamente pela distribuição das classes [Saito e Rehmsmeier 2015].

Esses resultados reforçam a validade do esquema de anotação adotado, demonstrando que é possível aplicá-lo com boa confiabilidade em contextos de gêneros diversos, embora ajustes adicionais possam ser necessários para lidar com classes menos representativas ou estruturalmente ambíguas.

Experimentos

Os experimentos realizados nesta etapa tiveram caráter exploratório, com o objetivo de estabelecer linhas de base para a tarefa de IAGT. Não se buscou, neste momento, o desenvolvimento de modelos avançados ou estratégias de otimização. O foco concentrou-se na preparação do conjunto de dados, definição de esquemas de anotação e elaboração de diretrizes (*guidelines*) para anotadores humanos.

Foi utilizado um conjunto balanceado com 400 amostras, contendo 50 textos para cada um dos oito tópicos previamente definidos. A divisão dos dados foi de 70 % para treinamento e 30 % para teste. Um classificador SVM foi ajustado com auxílio da biblioteca *Optuna*, utilizando 100 iterações de busca por hiperparâmetros. A acurácia obtida na tarefa de classificação por tópico foi de **98,67 %**.

O mesmo conjunto foi utilizado na tarefa de IAGT, a fim de permitir comparação direta entre as tarefas. Com o mesmo classificador SVM, a acurácia na tarefa de gênero foi de apenas **51,67 %**. Mesmo com a substituição do modelo por um *Transformer* pré-treinado (BERT), a acurácia atingiu no máximo **71,25 %**. Esse contraste evidencia a maior complexidade envolvida na classificação por gênero em relação à classificação por tópico — um resultado que está em consonância com a literatura, que destaca a natureza multifacetada, híbrida e menos prototípica dos

gêneros textuais na *web* [Kuzman e Ljubešić 2023, Kuzman, Mozetič e Ljubešić 2023, Mehler, Sharoff e Santini 2011].

Para viabilizar os experimentos de IAGT, foi necessário agrupar classes pouco representadas. O conjunto original incluía 12 gêneros, dos quais 9 possuíam menos de 9 % do total de exemplos. Conforme observado em estudos anteriores [Kuzman e Ljubešić 2023, Lee e Myaeng 2002], tal desequilíbrio compromete a estabilidade dos classificadores e dificulta a consistência da anotação. Os gêneros minoritários foram, portanto, agrupados na categoria *Exception*, resultando na seguinte distribuição:

- *tutorial or guide*: 29,0 % (116 textos)
- *blog or editorial*: 27,3 % (109 textos)
- *Exception* (agrupamento de 9 gêneros): 25,3 % (101 textos)
- *news*: 18,5 % (74 textos)

Tabela 7.13: *Acurácia dos modelos na tarefa de classificação de gênero textual*

Modelo	**Acurácia**
BERT	71,25 %
SVM	51,67 %
Regressão Logística	60,00 %

O modelo BERT utilizou a versão *bert-base-uncased*, com tokenizador correspondente. A divisão dos dados foi de 80 % para treinamento e 20 % para teste. O treinamento foi realizado com função de perda *CrossEntropyLoss*, otimizador *Adam*, taxa de aprendizado de 2×10^{-5}, `batch size` de 8 e 10 épocas.

De forma geral, os resultados confirmam que a tarefa de IAGT exige maior capacidade de abstração e sensibilidade a padrões discursivos, sendo mais desafiadora que tarefas temáticas baseadas em conteúdo lexical [Kessler, Nunberg e Schutze 1997, Mehler, Sharoff e Santini 2011].

Discussão

A análise conduzida nesta seção evidencia a complexidade e a relevância da tarefa de Identificação de Gênero Textual (IGT) em contextos digitais, especialmente na *Web*. Diferente de *corpora* tradicionais, nos quais os gêneros tendem a seguir categorias estáveis e bem definidas, os textos da *Web* exibem uma variabilidade estrutural e funcional considerável, exigindo abordagens mais flexíveis e sensíveis ao contexto. Como

discutido por Kessler et al. (1997), gênero não deve ser confundido com tópico ou estilo, mas entendido como uma configuração recorrente de forma e função comunicativa [Kessler, Nunberg e Schutze 1997].

Neste trabalho, a IGT foi aplicada a partir de um esquema de anotação multimodal, alinhado às necessidades específicas da anotação do *corpus* Argmap. A adoção de recursos visuais — como capturas de tela — mostrou-se fundamental para compensar perdas estruturais geradas durante a raspagem dos textos, permitindo uma análise mais precisa da organização dos documentos. O processo de adjudicação, conduzido de forma sistemática, contribuiu para a confiabilidade dos rótulos atribuídos. Os resultados experimentais reforçam a dificuldade da tarefa, mesmo com o uso de modelos pré-treinados como o BERT, o que está de acordo com achados prévios na literatura sobre gêneros na *Web* [Kuzman e Ljubešić 2023].

Mais do que uma etapa preparatória, a IGT demonstrou ser um componente estratégico para melhorar a qualidade de tarefas *downstream*, como a segmentação de tópicos. A remoção de documentos classificados como *Out of Scope*, assim como de outros gêneros estruturalmente inadequados, pode contribuir diretamente para aumentar a consistência e a interpretabilidade das anotações subsequentes. Como discutido na Seção 3.1, [Klie, Castilho e Gurevych 2024] adverte que "escolher textos que raramente ou nunca contêm os fenômenos a serem anotados pode ser ineficaz. Da mesma forma, selecionar textos de baixa qualidade pode ser prejudicial e causar problemas nas etapas posteriores do *pipeline* de aprendizado de máquina". Nesse sentido, a IGT auxilia na seleção de textos mais representativos e adequados, promovendo maior robustez nas análises e nos modelos treinados a partir do *corpus*.

7.3.3 Curadoria de Frases-Chave (CFC)

A tarefa de CFC, introduzida no Capítulo 6, estabeleceu as bases teóricas e práticas para avaliar a qualidade de frases-chave no contexto da anotação de segmentos temáticos. Conforme demonstrado, há uma correlação direta entre a qualidade das frases-chave e a acurácia na tarefa de CT, evidenciando sua importância estratégica na construção de *corpora* anotados. Os detalhes metodológicos sobre o processo de anotação — incluindo a quantidade de anotadores, as diretrizes utilizadas e os resultados de concordância entre anotadores — são apresentados na Seção 6.4.3, no Capítulo 6.

A curadoria foi realizada por três anotadores especialistas, que atuaram de forma independente seguindo diretrizes previamente definidas. A tarefa foi dividida em subtarefas de agrupamento, filtragem e seleção, com o apoio de ferramentas computacionais desenvolvidas especificamente para esse fim. Para avaliar a consistência entre anotadores, foi utilizado o coeficiente Kappa de Fleiss, cujos valores variaram de moderado a

substancial, conforme os critérios de Landis e Koch [Landis e Koch 1977]. Esses resultados, apresentados na Tabela 6.3, indicam que a curadoria manual, ainda que complexa, alcançou níveis satisfatórios de concordância, servindo como referência confiável para a construção do labelset utilizado na tarefa de CT.

Os experimentos revelaram que abordagens baseadas exclusivamente em *prompting* com LLMs não são suficientemente robustas ou consistentes para assegurar a qualidade necessária na curadoria. Nesta seção, a tarefa de CFC é retomada com o objetivo de aprimorar a qualidade da anotação por meio de uma abordagem de *humano no loop* (*human-in-the-loop*), conforme discutido na Seção B.1. Para isso, é apresentada a ferramenta *KPCTool* (ver Apêndice I), que reduz a carga cognitiva dos anotadores ao oferecer *visualizações* interativas e sugestões automáticas baseadas no modelo *Sentence-BERT* [Reimers e Gurevych 2019].

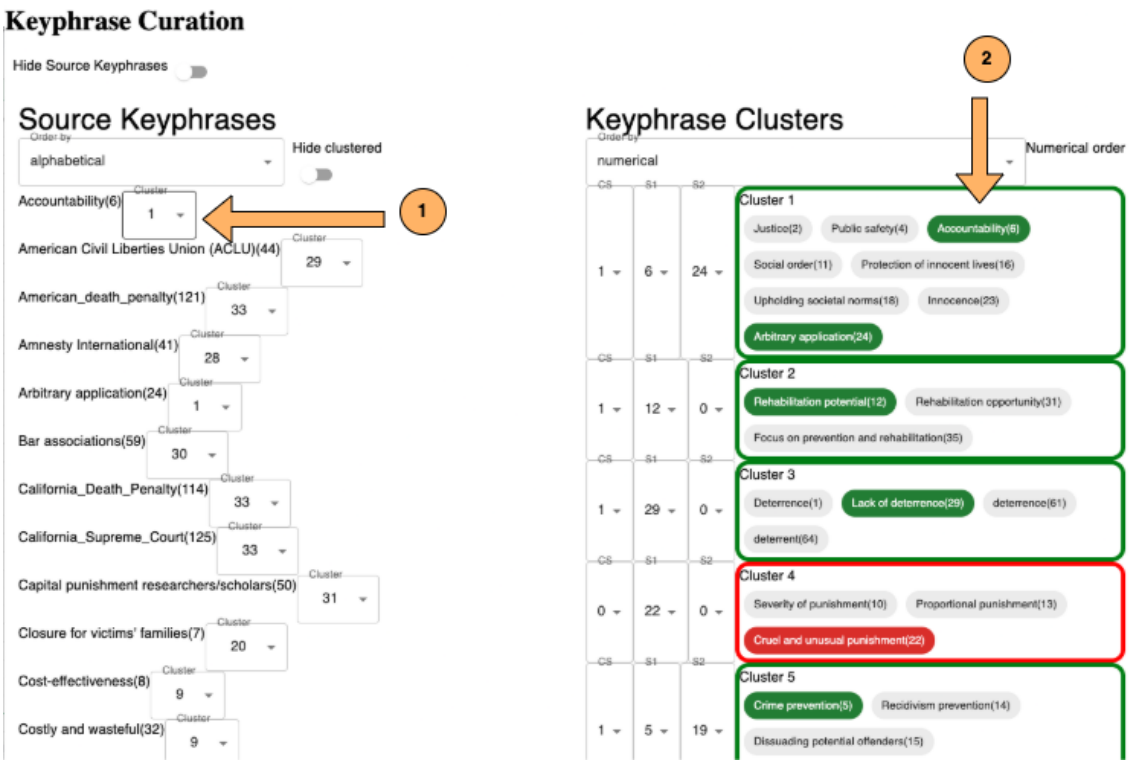


Figura 7.14: *Interface da ferramenta KPCTool na funcionalidade de agrupamento de frases-chave: à esquerda, seleção do identificador de cluster; à direita, inclusão da frase-chave no grupo correspondente (ver Apêndice I).*

A *KPCTool* incorpora o padrão arquitetural Recrutador–Selecionador discutido na Seção 4.5, ao dividir a tarefa de curadoria em subtarefas cognitivamente mais manejáveis, como agrupamento, filtragem e seleção. Nesta subseção, descrevem-se os componentes da ferramenta e os algoritmos que viabilizam essas funcionalidades, com ênfase nas escolhas de *design* voltadas à redução da carga cognitiva dos anotadores. Embora

não se apresentem aqui análises sistemáticas de desempenho ou usabilidade, destaca-se como a decomposição da tarefa, aliada às visualizações interativas, contribui para tornar o processo de curadoria mais estruturado e reprodutível.

Complexidade da tarefa de agrupamento de *keyphrases*

Dentre as subtarefas que compõem a CFC, o agrupamento é a etapa de maior complexidade, tanto do ponto de vista cognitivo quanto computacional. Essa subtarefa exige do anotador a análise comparativa entre múltiplas unidades, a identificação de relações semânticas latentes e a tomada de decisão sobre fronteiras conceituais muitas vezes ambíguas. Ao contrário das etapas de filtragem e seleção, em que a atenção se volta para atributos de cada unidade individualmente, o agrupamento demanda uma visão global do conjunto, obrigando o anotador a lidar com múltiplas possibilidades de organização e sobreposição temática. Além disso, o número de comparações potenciais entre frases cresce quadraticamente com o número de unidades, tornando o processo propenso a inconsistências e sobrecarga cognitiva. Por essas razões, o agrupamento configura-se como a subtarefa central e mais desafiadora da CFC, com impacto direto na qualidade dos agrupamentos temáticos e, por consequência, na consistência da tarefa de classificação de tópicos.

No cenário concreto desta pesquisa, em que se busca agrupar 140 frases-chave em exatamente 33 grupos temáticos, o número de formas distintas de realizar esse agrupamento é dado pelo número de Stirling de segunda espécie, denotado por $S(n,k)$. Esse valor representa a quantidade de partições possíveis de um conjunto de n elementos em exatamente k subconjuntos disjuntos e não vazios, sendo o modelo mais adequado para estimar o espaço de hipóteses em tarefas de agrupamento com cardinalidade fixa [Graham, Knuth e Patashnik 1994, cap. 6]. Por meio de cálculo simbólico, obtém-se que $S(140,33) \approx 2{,}84 \times 10^{175}$, o que evidencia uma explosão combinatória mesmo sob fortes restrições estruturais. A magnitude desse valor torna inviável qualquer abordagem exaustiva, justificando o uso de ferramentas interativas e estratégias supervisionadas que restrinjam o espaço de busca e apoiem o processo de tomada de decisão do anotador.

A opção por uma abordagem supervisionada, baseada em anotação humana, justifica-se pela natureza semântica e interpretativa da tarefa de agrupamento de frases-chave. A decisão sobre quais unidades devem compor um mesmo grupo depende de relações conceituais que envolvem sinonímia, generalização, especificidade ou coocorrência temática, muitas vezes não capturadas por métodos não supervisionados baseados exclusivamente em *embeddings*. Além disso, a ausência de rótulos ou agrupamentos de referência inviabiliza a avaliação sistemática dos agrupamentos gerados automaticamente. A anotação supervisionada, nesse contexto, permite não apenas a construção de um conjunto de dados com agrupamentos semanticamente coerentes, mas também subsidia a indução

de padrões linguísticos reutilizáveis e a avaliação de algoritmos de recomendação. No escopo desta pesquisa, a tarefa de agrupamento supervisionado está diretamente vinculada à melhoria da qualidade da anotação na CFC, impactando de forma direta a tarefa de classificação de tópicos.

Ferramentas para anotação da curadoria de *keyphrases*

Com o objetivo de apoiar o processo de anotação e reduzir a carga cognitiva envolvida na tarefa de agrupamento de frases-chave e outras subtarefas, foram desenvolvidos algoritmos capazes de auxiliar o anotador a tomar decisões mais consistentes e evitar erros comuns. Dentre esses algoritmos, destacam-se a ordenação automática de agrupamentos com base em similaridade semântica, a ordenação de frases-chave segundo sua similaridade par a par (*pairwise similarity*), e a recomendação de fusões entre agrupamentos com base em medidas de coesão interna. Esses recursos foram integrados à ferramenta *KPCTool*, que oferece diferentes modos de visualização e manipulação interativa das frases-chave, como forma de tornar o processo de anotação mais eficiente e confiável. As funcionalidades específicas implementadas na ferramenta são detalhadas no Anexo I.

A base dos algoritmos de agrupamento utilizados nesta pesquisa fundamenta-se em medidas de similaridade aplicadas sobre representações vetoriais densas, geradas pela biblioteca *SentenceTransformers* [Reimers e Gurevych 2019]. Cada frase-chave é representada como um vetor denso em um espaço vetorial de alta dimensionalidade, e a similaridade entre duas frases é estimada por meio da métrica do cosseno. A similaridade cosseno entre dois vetores u e v é definida como:

$$\text{sim}_{\cos}(u,v) = \frac{u \cdot v}{\|u\|\|v\|} = \frac{\sum_{i=1}^{d} u_i v_i}{\sqrt{\sum_{i=1}^{d} u_i^2} \cdot \sqrt{\sum_{i=1}^{d} v_i^2}}$$

onde d é a dimensionalidade dos vetores. Para otimizar o desempenho computacional, todas as similaridades entre frases-chave são pré-computadas e armazenadas em uma matriz de similaridade $S \in \mathbb{R}^{n \times n}$, onde n representa o número total de frases-chave. Cada elemento S_{ij} da matriz corresponde à similaridade cosseno entre as frases i e j:

$$S_{ij} = \text{sim}_{\cos}(x_i, x_j)$$

A matriz S é simétrica ($S_{ij} = S_{ji}$), já que a similaridade cosseno é comutativa, e sua diagonal principal contém exclusivamente valores iguais a 1, pois cada vetor é idêntico a si mesmo ($\text{sim}_{\cos}(x_i, x_i) = 1$). Essa matriz é calculada uma única vez e utilizada como base para os algoritmos de agrupamento, ordenação e recomendação implementados na ferramenta *KPCTool*.

Recomendação de *cluster* para *keyphrase*

A ferramenta *KPCTool* também oferece suporte ao anotador por meio de uma funcionalidade de recomendação de *cluster*, cujo funcionamento detalhado encontra-se na Seção I.4. Essa funcionalidade tem como objetivo sugerir, de forma dinâmica, a inclusão de uma nova frase-chave kp_i em um dos *clusters* já formados. Inicialmente, quando nenhum agrupamento foi construído, nenhuma recomendação é apresentada. No entanto, a partir do momento em que c agrupamentos foram criados, a ferramenta passa a calcular a similaridade média entre kp_i e todas as frases-chave kp_j pertencentes a cada *cluster* C_k. Para cada *cluster* C_k, com $|C_k|$ frases, a similaridade média é computada como:

$$\text{sim}(kp_i, C_k) = \frac{1}{|C_k|} \sum_{j=1}^{|C_k|} \text{sim}_{\cos}(kp_i, kp_j)$$

A recomendação consiste no *cluster* que apresentar maior valor de similaridade média em relação à frase-chave analisada. Essa estratégia visa apoiar o processo de anotação ao oferecer sugestões coerentes com a distribuição semântica atual dos agrupamentos, reduzindo o esforço de comparação manual e contribuindo para maior consistência nos agrupamentos formados.

Ordenação por similaridade entre pares

Uma das funcionalidades centrais da ferramenta *KPCTool* consiste na ordenação por similaridade entre pares, cujo funcionamento está descrito nas Seções I.5 e I.7. O algoritmo responsável por essa funcionalidade, descrito no Algoritmo 7.1, recebe como entrada uma matriz de similaridade entre elementos—sejam eles frases-chave ou agrupamentos (*clusters*)—e produz como saída uma lista ordenada de pares com base em seus respectivos valores de similaridade.

Cada elemento E considerado na ordenação pode representar uma unidade isolada (frase-chave) ou um agrupamento construído pelo anotador. A ordenação é feita de forma iterativa: a cada passo, o par (i, j) com maior similaridade ainda disponível é identificado e adicionado à lista. Em seguida, ambos os elementos são "resetados", ou seja, removidos da matriz de similaridade para que não participem de múltiplos pares, assegurando que cada elemento ocorra no máximo uma vez na lista final. Caso reste apenas um elemento sem pareamento ao final da iteração, este é adicionado à lista com um marcador especial. A diagonal da matriz é previamente ajustada para conter valores nulos, evitando que um elemento seja pareado consigo mesmo. Esse procedimento é detalhado no Algoritmo 7.1.

Essa ordenação serve a dois propósitos distintos no contexto da anotação. No caso de pares de frases-chave, o objetivo é evitar que o anotador deixe de agrupar unidades

semanticamente próximas, garantindo maior coesão temática dentro dos *clusters*. Ao destacar pares altamente similares, a ferramenta contribui para a detecção de redundâncias e a formação de agrupamentos mais consistentes. Por outro lado, quando os pares representam agrupamentos já existentes, a ordenação visa apoiar decisões de fusão ou distinção entre *clusters* semanticamente semelhantes. Nesse contexto, a recomendação permite ao anotador identificar rapidamente grupos que poderiam ser consolidados ou, alternativamente, optar por manter apenas um deles, evitando confusões conceituais. Essa estratégia contribui para a minimalidade e clareza dos agrupamentos finais, refletindo diretamente na qualidade e na reusabilidade dos dados anotados.

Algoritmo 7.1: Geração de Lista Ordenada por Similaridade entre Pares

Entrada: $S \in \mathbb{R}^{n \times n}$ – Matriz de similaridade entre elementos.

Saída: L – Lista ordenada de pares (i, j, sim_{ij}), onde $\text{sim}_{ij} = S[i][j]$

```
1  Função GerarListaDePares(S):
2      M ← CopiarMatriz(S);
3      SubstituirDiagonal(M, −∞);
4      L ← [ ] ;                                  // Lista de pares ordenada
5      enquanto existe valor > −∞ em M faça
6          (i, j) ← ObterParMaisSimilar(M);
7          se (i, j) ≠ nulo então
8              L ← L ∪ {(i, j, S[i][j])};
9              M ← ResetarPar(i, j, M);
10         senão
11             pare;
12     se existe apenas um elemento x não pareado então
13         L ← L ∪ {(x, −1, 0)} ;                 // Elemento sem par
14     retorne L;
```

Ordenação por similaridade com o centróide

Além da análise de similaridade entre pares ou entre agrupamentos, a ferramenta *KPCTool* implementa uma funcionalidade de ordenação baseada na centralidade semântica das frases-chave dentro de um agrupamento. Essa abordagem tem como objetivo destacar quais frases-chave ocupam posições mais centrais em termos semânticos, oferecendo suporte adicional para decisões de filtragem, seleção ou reagrupamento. A noção de centralidade adotada considera tanto a conectividade estrutural entre frases quanto a sua proximidade direta em relação à frase mais central — tratada como um centróide semântico. Os detalhes da implementação dessa funcionalidade na ferramenta encontram-se

descritos na Seção I.5.

O algoritmo responsável por esse cálculo está descrito no Algoritmo 7.2. A partir da matriz de similaridade entre as frases-chave, o algoritmo constrói um grafo não direcionado ponderado, em que os nós representam frases e as arestas representam a similaridade entre elas. Em seguida, aplica-se o algoritmo de *PageRank* [Page et al. 1999] para estimar a importância relativa de cada frase na estrutura global do grafo. A frase com maior pontuação é considerada a mais central. Para cada frase-chave, calcula-se ainda sua similaridade direta com essa frase central, fornecendo dois indicadores: um baseado em conectividade global e outro baseado em similaridade direta. Essa ordenação permite ao anotador priorizar frases semanticamente mais representativas ou identificar aquelas mais periféricas ao tema central do agrupamento.

Essa funcionalidade é especialmente útil para apoiar o processo de seleção e revisão de agrupamentos durante a anotação. A frase-chave mais central, identificada pelo algoritmo, é frequentemente uma boa candidata a representar semanticamente o agrupamento e pode ser considerada para seleção final dentro do *cluster*. Além disso, a análise de similaridade das demais frases em relação à frase central permite identificar quais unidades estão semanticamente mais próximas do núcleo conceitual do agrupamento e quais se distanciam. Frases-chave com baixa similaridade em relação ao centróide semântico podem indicar possíveis ruídos ou desvios temáticos, sendo, portanto, candidatas a serem reposicionadas em outros agrupamentos. Essa abordagem orientada por centralidade contribui para aumentar a coerência interna dos *clusters* e aprimorar a qualidade da anotação.

Algoritmo 7.2: Cálculo de Centralidade Semântica por PageRank

Entrada: $S \in \mathbb{R}^{n \times n}$ – Matriz de similaridade entre frases-chave

IDs $= [id_1, \ldots, id_n]$ – Lista ordenada de identificadores

Saída: R – Dicionário: $id_i \mapsto (\text{centralidade}, \text{similaridade com a mais central})$

se $n = 0$ **então**
  Retorne dicionário vazio;
se $n = 1$ **então**
  Retorne $\{id_1 \mapsto (1,1)\}$;
se $n = 2$ **então**
  Retorne $\{id_1 \mapsto (0.5, S[0][1]),\ id_2 \mapsto (0.5, S[1][0])\}$;
$G \leftarrow$ grafo não-direcionado vazio;
para $i \leftarrow 0$ **to** $n-1$ **faça**
  para $j \leftarrow 0$ **to** $n-1$ **faça**
    se $S[i][j] > 0$ **então**
      Adicionar aresta (i, j) com peso $S[i][j]$ em G;
$C \leftarrow \text{PageRank}(G)$; `// centralidade de cada nó`
$i^* \leftarrow \arg\max C[i]$; `// índice da mais central`
$id^* \leftarrow IDs[i^*]$;
$kp^* \leftarrow$ frase-chave com id^*;
$R \leftarrow \{\}$; `// dicionário de resultados`
para $i \leftarrow 0$ **to** $n-1$ **faça**
  $s \leftarrow \text{similaridade}(kp_i, kp^*)$;
  $R[IDs[i]] \leftarrow (C[i], s)$;
Retorne R;

Figura 7.15: *Algoritmo de cálculo de centralidade semântica baseado em PageRank. A função* `PageRank` *calcula a importância relativa dos nós com base na estrutura de conexões ponderadas do grafo. A função* `similaridade` *retorna o valor direto de similaridade entre uma frase e a mais central. A saída associa cada frase-chave a um par de valores representando sua importância global e sua proximidade direta ao centróide semântico do agrupamento.*

Ordenação por coesão do cluster

A ferramenta *KPCTool* também oferece suporte ao anotador por meio da ordenação dos agrupamentos com base em sua coesão interna. Essa funcionalidade visa destacar, entre os *clusters* construídos, aqueles que apresentam maior ou menor homogeneidade semântica entre suas frases-chave. O cálculo da coesão de um *cluster* é realizado por meio da média da similaridade entre todos os pares únicos de frases que o compõem. Quanto maior essa média, maior a coesão do agrupamento e, portanto, maior a evidência de que as unidades agrupadas compartilham um núcleo temático comum. A métrica também permite identificar agrupamentos com baixa coesão, os quais podem conter ruídos, frases desviantes ou heterogeneidade temática. Esses casos são úteis para orientar o anotador a revisar, dividir ou reorganizar agrupamentos. O funcionamento detalhado dessa funcionalidade está descrito na Seção I.6. O Algoritmo 7.3 apresenta a lógica usada para o cálculo da coesão média de um agrupamento.

Algoritmo 7.3: Cálculo da Coesão Interna de um Cluster

Entrada: C – *Cluster* contendo um conjunto de frases-chave $\{kp_1, \ldots, kp_n\}$
Saída: cohesion $\in [0,1]$ – Média de similaridade entre pares do *cluster*

```
cohesion ← 0;
combinações ← (n 2) pares únicos entre frases do cluster;
se n = 0 então
    Retorne 0;
se n = 1 então
    Retorne 1;
para cada (kp_i, kp_j) ∈ combinações faça
    sim ← similaridade(kp_i, kp_j);
    cohesion ← cohesion + sim;
cohesion ← cohesion/|combinações|;
Retorne cohesion;
```

Métrica de IAA para Agrupamento de Frases-Chave

A avaliação da consistência entre anotadores na tarefa de agrupamento de frases-chave exige métricas específicas que capturem a natureza subjetiva e estrutural dessa atividade. Para esse fim, esta pesquisa propõe uma métrica baseada no pareamento ótimo entre *clusters*, denominada **Best Matching Jaccard (BMJ)**. Essa métrica utiliza o índice de Jaccard para mensurar a interseção relativa entre pares de *clusters* provenientes de dois agrupamentos distintos, realizando o pareamento ótimo entre eles por meio do *Modified*

Jonker–Volgenant Algorithm[8].

O **índice de Jaccard** entre dois conjuntos A e B é definido como:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Com base nisso, define-se o **IAA-BMJ** como a média dos índices de Jaccard entre os pares de *clusters* com o maior grau de sobreposição possível entre dois agrupamentos $\mathcal{C}_1$ e $\mathcal{C}_2$, ou seja:

$$\text{IAA-BMJ}(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{n} \sum_{i=1}^{n} J(C_i^1, C_{\pi(i)}^2)$$

em que π representa uma permutação ótima de pareamento entre os *clusters* dos dois anotadores, tal que a soma total dos índices de Jaccard seja maximizada. O processo computacional que realiza esse pareamento é descrito no Algoritmo 7.4, o qual aplica o *Modified Jonker–Volgenant Algorithm* sobre a matriz de similaridade entre os agrupamentos para encontrar a melhor correspondência possível entre os *clusters*.

[8] Implementado na função `linear_sum_assignment` da biblioteca `scipy.optimize`. Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

Algoritmo 7.4: Pareamento ótimo entre conjuntos de *clusters* (baseado em Jaccard)

Entrada: $\mathcal{C}_1, \mathcal{C}_2$ – Dois conjuntos de *clusters* de frases-chave

Saída: Lista de pares $((i, j), J_{ij})$, onde J_{ij} é o índice de Jaccard entre os *clusters* pareados

$k \leftarrow |\mathcal{C}_1| - 1$; `// Último cluster (ignorados)`

Obter $K_1 \leftarrow$ conjuntos de frases de $\mathcal{C}_1$;

Obter $K_2 \leftarrow$ conjuntos de frases de $\mathcal{C}_2$;

$K_1' \leftarrow$ primeiros k *clusters* de K_1;

$K_2' \leftarrow$ primeiros k clusters de K_2;

$M \leftarrow$ matriz de Jaccard entre K_1' e K_2';

Aplicar *Modified Jonker–Volgenant Algorithm* para maximizar M:

  pareamento $\leftarrow$ `linear_sum_assignment`$(M$,`maximize=True`$)$;

para cada $(i, j) \in$ *pareamento* **faça**

  $J_{ij} \leftarrow M[i][j]$;

  Adicionar par $((i+1, j+1), J_{ij})$ à lista P;

Calcular $J_{kk} \leftarrow \text{Jaccard}(K_1[k+1], K_2[k+1])$;

Adicionar par $((k+1, k+1), J_{kk})$ à lista P;

Ordenar P por J_{ij} em ordem decrescente;

Retorne lista ordenada P;

Diferentemente de métricas tradicionais como o *Adjusted Rand Index* (ARI), o *Normalized Mutual Information* (NMI) e o *Fowlkes-Mallows Score* (FMS), a métrica BMJ foi concebida para oferecer maior interpretabilidade no contexto da anotação humana, privilegiando o conteúdo semântico dos agrupamentos. Enquanto ARI e NMI dependem fortemente da estrutura matemática de partições, sendo mais apropriadas para avaliação de algoritmos de *clustering*, o BMJ busca refletir diretamente o grau de sobreposição semântica entre agrupamentos produzidos por humanos — isto é, o quanto de conteúdo os anotadores de fato compartilharam entre pares de *clusters* pareados.

A Figura 7.16 apresenta um exemplo ilustrativo do cálculo da métrica BMJ a partir do pareamento entre *clusters*. Cada linha representa um par de *clusters*, e o valor de Jaccard calculado reflete o grau de sobreposição de frases-chave entre os agrupamentos correspondentes. O valor médio desses índices, neste caso **IAA-BMJ = 0,41**, oferece uma medida intuitiva e interpretável da concordância entre os agrupamentos anotados.

Cluster	Annotator 1	Annotator 2	Jaccard
1	{'social equity'}	{'social equity'}	1,0
2	{'law enforcement agencies', 'states drug enforcement', 'drug enforcement administration'}	{'law enforcement agencies', 'states drug enforcement', 'drug enforcement administration'}	1,0
3	{'tetrahydrocannabinol', 'endocannabinoid', 'cannabinoids', 'endocannabinoids', 'cannabinoid'}	{'endocannabinoids', 'cannabinoids', 'cannabinoid', 'endocannabinoid'}	0,8
4	{'drug policy alliance', 'marijuana policy project', 'drug policy foundation', 'advocacy groups (e,g,, n	{'marijuana policy project', 'drug policy alliance', 'drug policy research', 'advocacy groups (e,g,, norm	0,8
5	{'individual freedom', 'marijuanaright', 'consumer choice', 'marijuanatopia'}	{'individual freedom', 'marijuanaright', 'consumer choice'}	0,8
6	{'drug free australia', 'drug free'}	{'drug free australia', 'sobriety', 'drug free'}	0,7
7	{'overdose', 'drugabuse', 'intoxication', 'drug abuse', 'increased drug abuse', 'intoxicants', 'intoxicat	{'drugged', 'drugabuse', 'intoxication', 'drug abuse', 'increased drug abuse', 'intoxicated'}	0,6
8	{'national marijuana policy', 'swedish drug policy', 'national drug', 'national drug policy', 'national dr	{'national marijuana policy', 'national drug', 'federal government', 'national drug policy', 'united states'	0,6
9	{'social and moral concerns', 'higher dropout rates', 'negative effects on youth', 'negative impact o	{'social and moral concerns', 'negative impact on communities', 'negative effects on youth', 'workplac	0,6
10	{'prescriptions', 'prophylactic', 'medical research opportunities', 'prescribing', 'prescribed', 'prescrip	{'prescriptions', 'medicinal', 'prescribing', 'medication', 'prescribed', 'prescription', 'medical profession	0,6
11	{'drug control policy', 'drugpolicy', 'drug policy', 'drug control', 'marijuana policy'}	{'drug control policy', 'legalizing drugs', 'drugpolicy', 'drug policy', 'drug control', 'drug legalization', 'dr	0,5
12	{'alcohol and tobacco industries'}	{'nicotine', 'alcohol and tobacco industries'}	0,5
13	{'criminal justice reform', 'drug law reform', 'drug policy reform', 'marijuana laws', 'drug laws'}	{'transform drug policy', 'criminal justice reform', 'drug law reform', 'drug policy reform'}	0,5
14	{'tourism and industry boost', 'economic growth', 'job creation', 'workplace productivity decline'}	{'tourism and industry boost', 'marijuanatopia', 'economic growth', 'tax revenue', 'job creation'}	0,5
15	{'netherlands', 'canada', 'washington', 'oregon', 'portugal', 'california', 'united states', 'colorado', 'uru	{'netherlands', 'canada', 'swedish drug policy', 'portugal', 'uruguay', 'mexico'}	0,5
16	{'legalization', 'legalizing drugs', 'marijuana', 'drugs', 'cannabis', 'legalizing marijuana', 'weed', 'mari	{'marijuana', 'drugs', 'drugs make marijuana', 'cannabis', 'weed', 'marijuanainside', 'marihuana', 'smok	0,4
17	{'methadone', 'opiate', 'reduction of opioid use', 'narcotic', 'illicit drug', 'drugged', 'illegal drugs', 'opi	{'narcotic', 'illicit drug', 'illegal drugs', 'narcotics', 'intoxicants'}	0,4
18	{'addiction potential', 'psychoactive', 'health risks', 'mental health risks'}	{'addiction potential', 'health risks', 'overdose'}	0,4
19	{'taxpayer burden for regulation and enforcement', 'cost savings in law enforcement', 'increased e	{'cost savings in law enforcement', 'taxpayer burden for regulation and enforcement'}	0,4
20	{'reducing drug-related violence'}	{'black market persistence', 'drug war', 'reducing drug-related violence'}	0,3
21	{'safety and quality control', 'regulation and control', 'marijuanaproducts', 'marijuanainside'}	{'safety and quality control', 'harm reduction', 'marijuana users', 'regulation and control'}	0,3
22	{'teens marijuana brochure', 'decreased academic performance', 'harm reduction', 'decreased mot	{'teens marijuana brochure', 'marijuana files', 'drugeducation'}	0,3
23	{'medicinal benefits', 'medical marijuana states', 'medical marijuana', 'medicinal', 'medication', 'alle	{'medical marijuana', 'prophylactic', 'medical research opportunities', 'alleviation of chronic pain', 'trea	0,3
24	{'marijuana dispensaries', 'drugs make marijuana', 'marijuana growers', 'marijuana users', 'smokin	{'marijuana growers', 'pharmaceutical companies', 'marijuana dispensaries', 'marijuanaproducts'}	0,3
25	{'gateway drug'}	{'gateway drug', 'public safety concerns', 'increased emergency room visits', 'secondhand smoke exp	0,2
26	{'state governments', 'federal government'}	{'state governments', 'medical marijuana states', 'washington', 'oregon', 'california', 'colorado'}	0,1
27	{'sobriety', 'impaired cognitive function', 'impaired driving'}	{'decreased academic performance', 'impaired cognitive function', 'decreased motivation and ambitic	0,1
28	{'drug war'}	{'legalization', 'marijuana policy', 'marijuana laws', 'legalizing marijuana', 'marijuana legalization'}	0,0
29	{'pharmaceutical companies'}	{'methadone', 'opiates', 'opiate'}	0,0
30	{'nicotine', 'secondhand smoke exposure'}	{'tetrahydrocannabinol', 'psychoactive'}	0,0
31	{'marijuana files', 'transform drug policy', 'drug policy research'}	{'less burden on the judicial system'}	0,0
32	{'conflict with federal law'}	{'reduction of opioid use'}	0,0
33	{'black market persistence', 'public safety concerns'}	{'medicinal benefits', 'alternative medicine option'}	0,0
		TOTAL	13,5

Figura 7.16: *Pareamento entre clusters de dois anotadores com cálculo do índice de Jaccard (BMJ).*

Como forma de comparação, aplicaram-se também as métricas tradicionais de avaliação de agrupamento para os mesmos dados. Os valores obtidos foram os seguintes:

- **IAA-BMJ**: 0,41
- **Adjusted Rand Index (ARI)**: 0,386
- **Normalized Mutual Information (NMI)**: 0,778
- **Fowlkes-Mallows Score (FMS)**: 0,406

Esses resultados mostraram-se consistentes com os valores obtidos em experimentos anteriores realizados sem o uso da ferramenta KPCTool, conforme documentado na Tabela 6.3. No experimento realizado com o tópico *Marijuana legalization*, os valores observados foram: ARI = 0,39, FMS = 0,41 e NMI = 0,51. A estabilidade entre os dois cenários sugere que, embora a ferramenta auxilie a anotação e forneça recomendações úteis, a subjetividade inerente à tarefa de agrupamento permanece significativa, exigindo critérios bem definidos e validações cruzadas.

Tópico	IAA-BMJ	ARI	NMI	FMS
marijuana_legalization	0.41	0.39	0.78	0.41
death_penalty	0.43	0.33	0.76	0.35
cloning	0.57	0.49	0.82	0.52
school_uniforms	0.51	0.46	0.80	0.49
gun_control	0.42	0.44	0.77	0.47
minimum_wage	0.45	0.53	0.80	0.55

Tabela 7.14: *Métricas de concordância entre anotadores para seis tópicos anotados com o apoio da ferramenta KPCTool*

Os resultados consolidados na Tabela 7.14 indicam um padrão estável de concordância entre anotadores ao longo dos seis tópicos analisados com o uso da ferramenta KPCTool. Em todos os casos, as métricas de *Normalized Mutual Information* (NMI) mantiveram-se acima de 0,75, sinalizando uma forte sobreposição global entre os agrupamentos. A métrica *IAA-BMJ*, mais sensível à estrutura semântica dos agrupamentos pareados, variou de 0,41 a 0,57, com destaque para o tópico *cloning*, que apresentou a maior média de similaridade Jaccard entre os *clusters*. As métricas ARI e FMS, que refletem diretamente a granularidade estrutural, apresentaram valores entre 0,33 e 0,55, evidenciando variações sutis nas decisões de agrupamento entre anotadores. De forma geral, os dados sugerem uma consistência satisfatória, especialmente nos tópicos *cloning*, *minimum wage* e *school uniforms*, que apresentaram altos valores em todas as métricas. Isso indica que a ferramenta contribuiu para um alinhamento semântico mais robusto e favoreceu decisões de agrupamento mais uniformes.

Por fim, essa abordagem também permite uma análise qualitativa do pareamento, destacando *clusters* com alta ou baixa sobreposição, oferecendo pistas valiosas sobre divergências ou ambiguidades nas decisões dos anotadores. Dessa forma, a métrica BMJ não apenas quantifica a concordância, mas também se mostra uma ferramenta útil para diagnóstico e aprimoramento de diretrizes em processos de curadoria colaborativa de frases-chave, como na tarefa de *keyphrase clustering*.

Os resultados apresentados nesta seção evidenciam que a tarefa de CFC pode ser decomposta em subtarefas bem definidas, como filtragem de agrupamentos e seleção de frases representativas, cada uma com características próprias e desafios distintos. A aplicação de métricas apropriadas para cada subtarefa demonstrou níveis satisfatórios de concordância entre anotadores, especialmente quando apoiados por funcionalidades interativas da ferramenta KPCTool. Tais evidências reforçam o potencial dessa abordagem com *humano no circuito* para promover maior consistência e qualidade na anotação semântica. A seguir, será abordada a tarefa de CT, que se beneficia diretamente dos agrupamentos validados na etapa de CFC, servindo como base para a anotação de categorias temáticas e a estruturação hierárquica do conteúdo textual.

7.3.4 Classificação de Tópico (CT)

A tarefa de *CT* consiste na atribuição de rótulos temáticos a segmentos previamente definidos do texto. Por se tratar da possibilidade de múltiplos tópicos por segmento, ela é tratada como uma tarefa de classificação *multirrótulo*, sendo permitida a atribuição de 1 a 3 rótulos por segmento, vide Figura 6.2. Esses rótulos são obtidos a partir da subtarefa de *CFC*, a qual fornece um conjunto de expressões candidatas representativas dos principais tópicos identificados no *corpus*. Assim, as tarefas de segmentação e classificação são tratadas de forma integrada nesta tese, compondo uma camada anotativa única. A segmentação tem como objetivo a delimitação de unidades textuais topicamente coesas, enquanto a classificação busca explicitar os temas centrais presentes em cada uma dessas unidades. A integração entre essas tarefas tem-se mostrado fundamental tanto para a estabilidade da anotação (com impacto positivo na concordância entre anotadores) quanto para a construção de *corpora* semanticamente enriquecidos, adequados ao treinamento de modelos em tarefas *downstream* de PLN.

A classificação de tópicos é motivada por sua utilidade em fornecer o contexto necessário para subtarefas que exigem interpretação dependente de segmento, como no caso da DCDC (Detecção de Conclusão Dependente de Contexto) e da DEDC (Detecção de Evidência Dependente de Contexto), descritas nas Seções 7.2.4 e 7.2.5, respectivamente. Nestas tarefas, a compreensão de uma unidade argumentativa (UA) depende do entendimento temático do segmento onde está inserida. Para suprir esse contexto, propõe-

se a utilização de um conjunto de tópicos, representados por frases-chave, associados a trechos específicos do texto. Essas frases-chave funcionam como índices conceituais que apontam para os argumentos relevantes presentes em seus respectivos segmentos. Tal estrutura não apenas viabiliza a interpretação contextual durante a anotação, mas também contribui para a recuperação futura de informações com base em tópicos, potencializando aplicações como sumarização, busca semântica e construção de mapas de argumentos.

A classificação de texto é uma tarefa ampla em processamento de linguagem natural que envolve a categorização de documentos com base em critérios variados, como sentimentos, intenções ou características específicas, enquanto a classificação de tópicos é uma subtarefa mais específica, focada exclusivamente na identificação de temas ou assuntos predominantes em um texto. Por exemplo, a classificação de texto pode incluir a detecção de spam, onde e-mails são rotulados como "spam"ou "não spam"com base em padrões de conteúdo, sem relação direta com temas. Já a classificação de tópicos, por sua vez, atribuiria rótulos como "esportes"ou "política"a notícias, visando capturar o conteúdo temático, destacando assim sua natureza distinta dentro do espectro mais geral da classificação de texto.

A tarefa de CT apresenta similaridades estruturais com a Classificação Multirrótulo de Textos (MLTC), especialmente por envolver a atribuição de múltiplos rótulos a uma mesma unidade textual. Embora partam de motivações distintas, ambas compartilham desafios comuns, como ambiguidade semântica e correlação entre rótulos. Nesse sentido, os avanços metodológicos e os conhecimentos consolidados em MLTC — incluindo modelos, estratégias de modelagem e métricas de avaliação — podem ser aproveitados ou adaptados para aprimorar a abordagem em CT.

A tarefa de CT, conforme abordada nesta subseção, é intrinsecamente supervisionada, exigindo um conjunto pré-definido de rótulos temáticos obtidos a partir da curadoria de frases-chave, diferindo fundamentalmente da modelagem de tópicos discutida na Seção B.2.5. Enquanto a modelagem de tópicos emprega métodos não supervisionados, como LDA, para descobrir temas latentes em um *corpus* sem rótulos prévios, a CT multirrótulo proposta aqui atribui até três rótulos explícitos por segmento, com base em anotações manuais guiadas por frases-chave. Essa abordagem supervisionada garante maior controle sobre os temas identificados, alinhando-se às necessidades de tarefas *downstream*, como DCDC e DEDC, e distinguindo-se pela sua capacidade de integrar segmentação e classificação em uma camada anotativa coesa, voltada para a construção de *corpora* semanticamente enriquecidos.

Classificação de Textos Multirrótulo (MLTC)

A classificação de textos multirrótulo (*Multi-Label Text Classification* – MLTC) é uma tarefa supervisionada que associa múltiplos rótulos a um mesmo texto. Trata-

se da tarefa mais próxima da CT discutida nesta tese, com diferenças pontuais, como o uso de rótulos não necessariamente tópicos, documentos mais longos e, em alguns casos, hierarquias entre classes. Por essa proximidade, os desafios, soluções e métricas consolidadas na literatura de MLTC oferecem subsídios relevantes para a abordagem adotada neste trabalho.

A MLTC apresenta desafios específicos, como a modelagem de dependências entre rótulos, já que eles não são mutuamente exclusivos, e o desbalanceamento das classes, com rótulos de baixa frequência afetando o desempenho dos modelos. Há também dificuldades em definir o número ideal de rótulos por instância e em escalar os modelos para cenários com muitos rótulos ou documentos. Esses pontos são amplamente discutidos por [Ganda e Buch 2018] e [Han et al. 2023], que destacam ainda a complexidade associada à ausência de estrutura explícita entre os rótulos e à variabilidade nos formatos e tamanhos dos textos. [Wang et al. 2021] chamam atenção para o impacto da ordem dos rótulos no treinamento de modelos baseados em cadeias ou sequências, propondo uma arquitetura iterativa que evita essa limitação. Já [Ishita et al. 2010] apontam os limites da anotação humana e da variação de granularidade nos rótulos como fatores que afetam a qualidade do treinamento supervisionado.

Para lidar com esses desafios, diferentes estratégias têm sido propostas na literatura. Os métodos de transformação, como *Binary Relevance* (BR) e *Label Powerset* (LP), são os mais tradicionais. BR treina um classificador binário independente para cada rótulo, ignorando correlações. LP trata cada combinação de rótulos como uma classe única, capturando dependências, mas com baixa escalabilidade em domínios com muitos rótulos [Ganda e Buch 2018]. Os métodos de adaptação estendem algoritmos como SVM, kNN e redes neurais para lidar diretamente com múltiplos rótulos, permitindo explorar relações entre classes e ajustar-se a dados desbalanceados [Han et al. 2023]. Abordagens mais recentes buscam modelar dependências de forma mais robusta, como o uso de atenção e grafos [Pal, Selvakumar e Sankarasubbu 2020] ou mecanismos iterativos que evitam a sensibilidade à ordem dos rótulos, como no ML-Reasoner [Wang et al. 2021]. Também têm sido propostas representações específicas por rótulo, com bons resultados em rótulos de baixa frequência [Xiao et al. 2019]. Em cenários com limitação de anotações, técnicas semi-supervisionadas têm se mostrado promissoras [Han et al. 2023].

Diversos *datasets* têm sido utilizados na avaliação de modelos de MLTC, variando em número de documentos, tamanho dos textos e cardinalidade média de rótulos por instância. Entre os mais comuns estão o RCV1-V2, AAPD, Reuters-21578 e Slashdot, com textos oriundos de domínios como jornalismo, ciência e comentários online [Pal, Selvakumar e Sankarasubbu 2020, Wang et al. 2021]. Em termos de avaliação, métricas tradicionais de acurácia não são adequadas para cenários multirrótulo. Por isso, a literatura adota métricas específicas como micro e macro F1, Hamming Loss, Exact Match

e Subset Accuracy [Ganda e Buch 2018, Han et al. 2023]. O uso combinado dessas métricas busca capturar tanto o desempenho global quanto os efeitos do desbalanceamento e das classes raras. Algumas abordagens também utilizam métricas baseadas em ranking ou cobertura, especialmente em contextos com muitos rótulos ou estruturas hierárquicas [Han et al. 2023].

Muitos dos problemas observados em MLTC têm origem na baixa qualidade do conjunto de rótulos utilizados, frequentemente redundantes, ambíguos ou pouco representativos. Neste trabalho, esse aspecto é tratado pela tarefa de CFC (CFC), que define um conjunto de rótulos temáticos mais consistente, específico por tópico principal e semanticamente controlado. O labelset gerado pela CFC resolve boa parte dos desafios citados — reduz ambiguidade, melhora a cobertura e fornece uma base mais estável para a classificação supervisionada. Com isso, a tarefa de CT se beneficia diretamente, tanto em termos de desempenho quanto de interpretabilidade dos resultados.

Definição Matemática

No contexto da anotação de *corpus* para tarefas de SCT, a classificação multirrótulo de subtópicos em segmentos textuais é formalizada como um problema de aprendizado supervisionado. Considerando documentos com tópicos principais previamente definidos, define-se um conjunto de dados $\mathcal{D} = \{(\mathbf{s}_i, t_i, \mathbf{y}_i)\}_{i=1}^{N}$, onde $\mathbf{s}_i \in \mathcal{S}$ representa um segmento de texto (por exemplo, uma sentença ou parágrafo), $t_i \in \mathcal{T}$ é o tópico principal do documento (por exemplo, "educação"), e $\mathbf{y}_i \in \{0,1\}^{L(t_i)}$ é um vetor binário que indica a presença ($y_{i,j} = 1$) ou ausência ($y_{i,j} = 0$) de cada um dos $L(t_i)$ subtópicos relevantes associados a t_i (como "ensino remoto" ou "inclusão").

O objetivo é treinar um modelo $f : \mathcal{S} \times \mathcal{T} \to [0,1]^{L(t_i)}$, parametrizado por θ, que minimize a perda agregada:

$$\mathcal{L}_{\text{total}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{y}_i, f(\mathbf{s}_i, t_i; \theta)), \tag{7-1}$$

de forma a prever os rótulos $\hat{\mathbf{y}}_i$ correspondentes aos subtópicos discutidos em cada segmento, condicionados ao tópico principal do documento.

7.3.5 Processo Geral de Anotação

A tarefa de CT adotou uma estratégia multirrótulo restrita, permitindo a seleção de 1 a 3 rótulos por segmento textual, a partir de um conjunto fixo de 16 frases-chave curadas previamente. Essa decisão baseia-se na observação empírica de que a maioria dos segmentos aborda de forma sucinta um número limitado de subtópicos. Trabalhos anteriores em tarefas análogas, como [Ishita et al. 2010], demonstram a viabilidade desse tipo de

anotação, com média próxima a dois rótulos por instância. A definição do conjunto de 16 rótulos resultou de um processo de curadoria das frases mais representativas do *corpus*, com o objetivo de garantir cobertura semântica adequada sem comprometer a consistência da anotação. Além disso, a literatura em classificação multirrótulo recomenda conjuntos de rótulos com granularidade controlada, a fim de mitigar a esparsidade e facilitar tanto a anotação quanto o aprendizado supervisionado [Han et al. 2023, Ganda e Buch 2018].

O aperfeiçoamento da compreensão da tarefa de SCT foi alcançado por meio de sucessivos ciclos de atividades descritas na Figura 7.2, os quais resultaram na geração iterativa de revisões do manual de diretrizes (vide Anexo J) e na criação de um dicionário de frases-chave[9]. Esse dicionário apresenta a definição de cada rótulo de tópico, acompanhada de um conjunto de frases-chave semanticamente relacionadas, obtidas a partir da atividade de agrupamento realizada na CFC. Esses agrupamentos funcionam como conjuntos de sinônimos ou variações derivadas, oferecendo suporte semântico à tarefa de classificação. Como resultado desse processo de refinamento, observou-se uma estabilidade na tarefa de segmentação textual (ST), com coeficiente de concordância entre os anotadores especialistas atingindo 0,55 na métrica de Kappa de Krippendorff.

Apesar da estabilização da tarefa de segmentação textual (ST), com níveis de concordância semelhantes entre especialistas e anotadores treinados, a tarefa de CT (CT) não apresentou a mesma estabilidade. Além do dicionário de frases-chave, algumas medidas contribuíram para uma leve melhora na concordância da tarefa, como o uso de um *outliner* automático em tempo real na ferramenta (Figura 7.17), a identificação de segmentos digressivos marcados como *Ignored* e a remoção de documentos fora de escopo (*OutOfScope*) a partir da Identificação de Gênero Textual. No entanto, tais medidas mostraram-se insuficientes. Diante disso, optou-se por suspender a tarefa de CT e considerar, como trabalho futuro, a implementação de um conjunto de medidas adicionais, tais como a integração de recomendadores automáticos para apoiar a anotação nessa tarefa.

Escalonamento da Anotação com Anotadores Treinados

Após a estabilização da concordância entre os anotadores especialistas, alcançada por meio de diversos ciclos de ensaio e adjudicação coletiva, foi possível estruturar um processo de escalonamento da tarefa de anotação com o apoio de novos anotadores treinados. O fluxo completo de atividades que viabilizou essa transição está ilustrado na Figura 7.2.

Como etapa preparatória, conduziu-se um treinamento sistemático dos novos anotadores, seguido de um processo seletivo que garantiu a familiaridade com as diretrizes

[9]Disponível em: https://argmap.inf.ufg.br/guideline/appendix5/

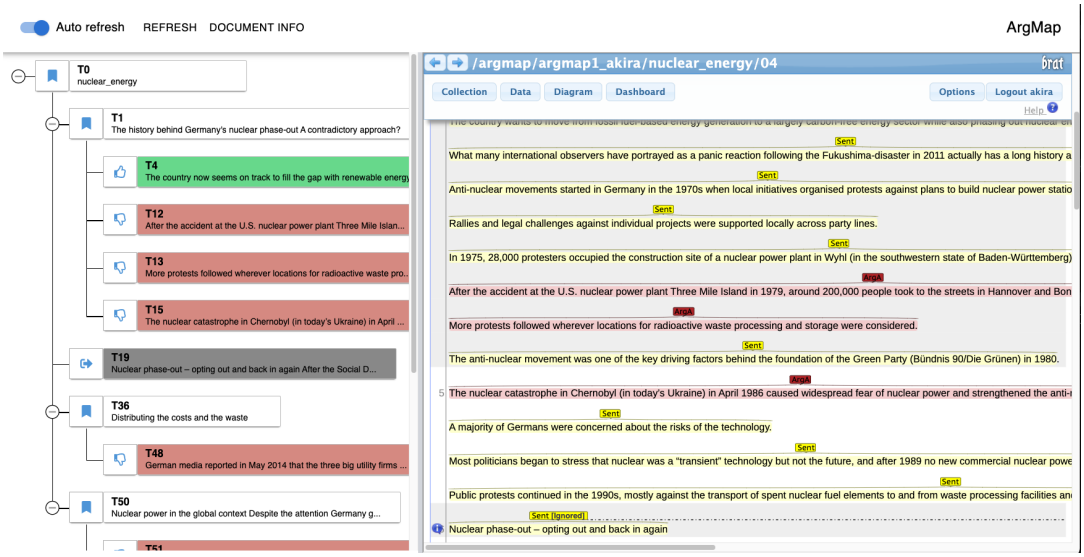


Figura 7.17: *Outliner automático: a ferramenta Argmap gera, em tempo real, uma árvore de tópicos com base na segmentação realizada na janela de anotação. Na figura, é exibido um nó identificado como T19, rotulado como* `Ignored`*, correspondente a um segmento digressivo que deve ser ignorado por não conter argumentos relacionados ao tópico principal.*

de anotação, bem como a capacidade de julgamento linguístico adequada. A partir dessa etapa, os documentos passaram a ser organizados em lotes balanceados de anotação, com o objetivo de assegurar a qualidade e a representatividade dos dados anotados — conforme diretrizes discutidas na Seção 3.1.

Cada lote foi composto por 10 documentos, sendo 5 documentos de um tópico (por exemplo, aborto) e 5 documentos de outro tópico (por exemplo, energia nuclear). Para cada lote, também foi garantida uma distribuição equilibrada quanto à densidade argumentativa dos documentos, com base na quantidade de argumentos automaticamente detectados pela tarefa de Detecção de Argumentos. Assim, cada lote continha: 2 documentos pequenos (menos de 15 argumentos), 2 documentos médios (entre 15 e 35 argumentos) e 1 documento grande (mais de 35 argumentos).

Para cada lote, foram atribuídos dois anotadores treinados e, posteriormente, um anotador especialista realizou a adjudicação das anotações divergentes, produzindo assim a versão final do padrão ouro. A concordância entre os dois anotadores treinados era calculada individualmente para cada documento e também como média para o lote, com o objetivo de monitorar a qualidade da anotação e detectar possíveis desvios ou inconsistências ao longo do processo.

Ao todo, 40 lotes de 10 documentos estavam disponíveis no conjunto de dados original, mas nem todos foram aproveitados. Quatro desses lotes (10%) foram atribuídos exclusivamente aos anotadores especialistas e utilizados como base para o treinamento e a

seleção dos novos anotadores, bem como para discussões de adjudicação coletiva que contribuíram para a construção e o refinamento das diretrizes de anotação (*guideline*). Além disso, 34 documentos foram descartados por não apresentarem argumentos detectáveis, e outros 6 foram utilizados em anotações não independentes realizadas pela equipe durante a fase de pré-estudos. Dos 36 lotes restantes, apenas 32 foram destinados à anotação independente, totalizando 320 documentos. Esses documentos foram, então, distribuídos entre os anotadores treinados, que os anotaram de forma autônoma, porém sob supervisão.

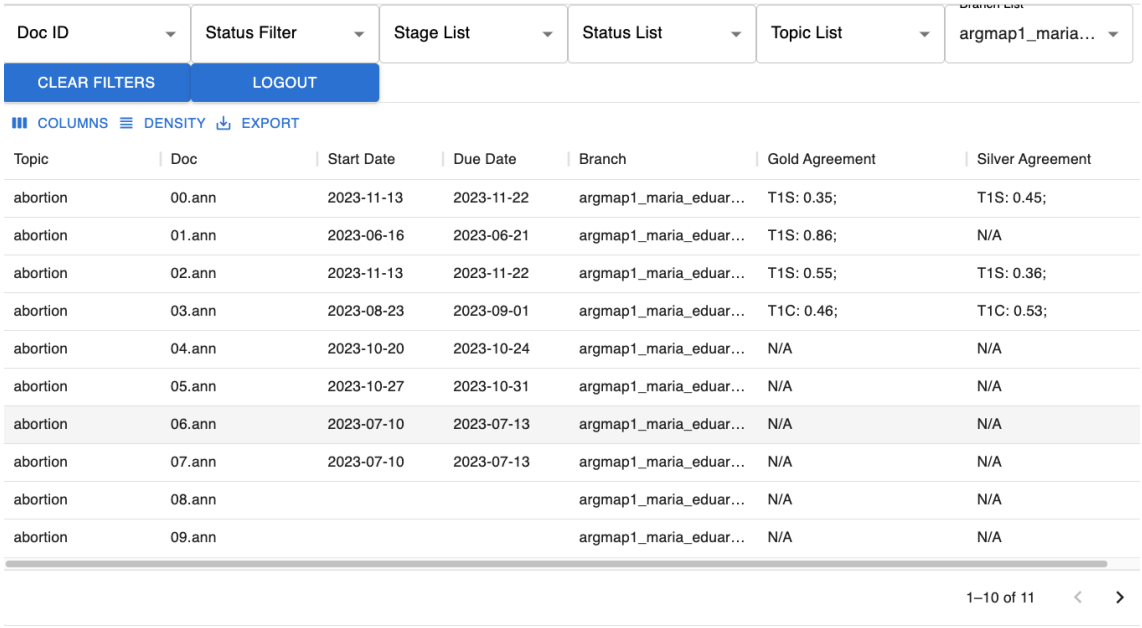
Controle de Qualidade por Lote

Cada lote era atribuído a dois anotadores treinados, com prazo padrão de conclusão de até uma semana. À medida que um anotador finalizava a anotação de um documento, a ferramenta verificava automaticamente se o outro anotador já havia concluído o mesmo documento; em caso positivo, exibia imediatamente o valor da concordância entre as duas anotações, denominado *silver agreement* (SA). Esse mecanismo de feedback em tempo real mostrou-se eficaz, pois fornecia ao anotador uma estimativa instantânea sobre sua aderência às diretrizes estabelecidas.

Após a finalização do lote pelos dois anotadores, uma notificação automática era enviada ao adjudicador responsável, que realizava a adjudicação das anotações, produzindo o rótulo de referência *gold*. A concordância entre o adjudicador e cada anotador era então calculada, sendo denominada *gold agreement* (GA). Em casos de GA muito baixos, o adjudicador realizava uma análise qualitativa para diagnosticar os fatores que contribuíram para a baixa concordância, entrando em contato com o anotador para fornecer feedback individualizado com base nas inconsistências observadas e reforçar pontos críticos do *guideline*.

Tanto os valores de SA quanto de GA eram atualizados em tempo real na interface de *dashboard* da ferramenta, conforme ilustrado na Figura 7.18. Essa visualização permitia que o gerente de anotação acompanhasse o desempenho por lote da equipe composta por dois anotadores e um adjudicador. Quando valores de GA consistentemente baixos eram observados para um mesmo anotador, o gerente iniciava uma análise em conjunto com o adjudicador e, se necessário, tomava medidas corretivas, incluindo o afastamento ou substituição do anotador por baixo desempenho.

Outro fator importante no processo era o estímulo à produtividade. Quando um anotador finalizava um lote antes do prazo de uma semana, ele recebia outro lote para anotação, o que possibilitava ganhos de produtividade sem prejuízo à qualidade, já que o processo continha mecanismos suficientes de controle e validação. Para que esse ritmo acelerado fosse sustentável, era necessário manter um *pool* ativo de anotadores treinados. No entanto, apesar dos esforços de formação de turmas com até 20 candidatos, poucos atingiam o desempenho esperado após o treinamento. Com isso, o número de



Topic	Doc	Start Date	Due Date	Branch	Gold Agreement	Silver Agreement
abortion	00.ann	2023-11-13	2023-11-22	argmap1_maria_eduar...	T1S: 0.35;	T1S: 0.45;
abortion	01.ann	2023-06-16	2023-06-21	argmap1_maria_eduar...	T1S: 0.86;	N/A
abortion	02.ann	2023-11-13	2023-11-22	argmap1_maria_eduar...	T1S: 0.55;	T1S: 0.36;
abortion	03.ann	2023-08-23	2023-09-01	argmap1_maria_eduar...	T1C: 0.46;	T1C: 0.53;
abortion	04.ann	2023-10-20	2023-10-24	argmap1_maria_eduar...	N/A	N/A
abortion	05.ann	2023-10-27	2023-10-31	argmap1_maria_eduar...	N/A	N/A
abortion	06.ann	2023-07-10	2023-07-13	argmap1_maria_eduar...	N/A	N/A
abortion	07.ann	2023-07-10	2023-07-13	argmap1_maria_eduar...	N/A	N/A
abortion	08.ann			argmap1_maria_eduar...	N/A	N/A
abortion	09.ann			argmap1_maria_eduar...	N/A	N/A

Figura 7.18: *Módulo de dashboard da ferramenta Argmap: as últimas colunas — Gold Agreement (GA) e Silver Agreement (SA) — informam as métricas de concordância entre o anotador com o adjudicador e com o outro anotador, respectivamente. Por exemplo, na primeira linha, um documento do tópico abortion, identificado pelo nome de arquivo* `00.ann`*, foi atribuído em 13/11/2023 e finalizado em 22/11/2023 no ramo* `argmap1_maria_eduarda`*. Os valores obtidos de GA e SA foram 0,35 e 0,45, respectivamente, na tarefa de Segmentação de Tópicos (T1S).*

anotadores realmente capacitados permanecia limitado, raramente ultrapassando três a quatro anotadores treinados.

Todos esses procedimentos de controle de qualidade estão alinhados com as boas práticas de anotação de *corpus* discutidas na Seção 3.1.

7.3.6 Resultados da Segmentação de Tópicos

Esta seção apresenta uma análise quantitativa dos resultados obtidos na tarefa de Segmentação de Tópicos, considerando três parâmetros principais dos documentos anotados: a quantidade de argumentos detectados (oriundos da tarefa de detecção de argumentos), a quantidade total de documentos e os gêneros textuais predominantes. A partir desses parâmetros, investigamos a correlação entre essas variáveis e os índices de concordância interanotadores (*silver agreement*), conforme discutido na Seção 7.3.5. A concordância foi calculada por meio de três métricas complementares: Cohen's Kappa (modelando a segmentação como uma tarefa binária), WindowDiff e Pk, ambas reconhecidas como métricas robustas para avaliação de segmentação textual. Ressalta-se que,

conforme descrito na Seção 7.3.5, a tarefa de CT será tratada em trabalhos futuros e não é contemplada nesta análise.

Métricas de Concordância Entre Anotadores

A avaliação da tarefa de segmentação de tópicos foi realizada por meio da métrica tradicionalmente utilizada na literatura: *Pk*. Proposta por Beeferman et al. [Beeferman, Berger e Lafferty 1999], essa métrica calcula a probabilidade de que dois pontos arbitrários do texto estejam incorretamente classificados como pertencentes ao mesmo segmento (ou a segmentos distintos), com base em uma janela deslizante. Apesar de amplamente utilizada, ela apresenta limitações conhecidas, como a penalização excessiva de pequenos deslocamentos e uma sensibilidade desigual a falsos positivos e negativos [Pevzner e Hearst 2002].

Além do uso da métrica *Pk*, adotou-se também a métrica *Cohen's Kappa* como medida principal de concordância IAA. Conforme discutido na Seção 7.3.5, essa escolha foi motivada por análises empíricas nas etapas iniciais da anotação, nas quais a métrica *Pk* demonstrou baixa sensibilidade para identificar segmentações de baixa qualidade, dificultando o controle de consistência entre lotes. Considerando a tarefa como uma classificação binária (presença ou ausência de limite entre sentenças), a métrica *Kappa* demonstrou maior aderência ao objetivo de monitoramento da confiabilidade das anotações. Seus valores se mostraram mais coerentes e informativos, contribuindo para decisões sobre a estabilidade da tarefa e validação por adjudicação.

Para contextualizar os resultados obtidos neste trabalho, a Tabela 7.15 apresenta métricas de segmentação de tópicos reportadas na literatura para *datasets* amplamente utilizados. No trabalho de Arnold et al. [Arnold et al. 2019], os valores de *Pk* para o *dataset WikiSection*, que contém artigos da Wikipédia sobre cidades e doenças com seções editoriais explícitas, variam entre 0,098 e 0,219 para cidades (inglês) e entre 0,225 e 0,374 para doenças (inglês), dependendo da configuração do modelo SECTOR. Koshorek et al. [Koshorek et al. 2018] avaliam o *corpus WIKI-727K*, com segmentações derivadas automaticamente da estrutura da Wikipédia, alcançando *Pk* de 0,2213, e o subconjunto *WIKI-50*, com anotações manuais, com *Pk* de 0,1824. Glavaš e Somasundaran [Glavaš e Somasundaran 2020] propuseram o modelo CATS, que combina Transformers hierárquicos e modelagem auxiliar de coerência, atingindo *Pk* de 0,1595 para o *WIKI-727K*, 0,1653 para o *WIKI-50*, e 0,1685 para o subconjunto *CITIES*.

Dataset	Domínio	Pk
WikiSection (SECTOR)	Cidades (EN)	0,127–0,245
WikiSection (SECTOR)	Doenças (EN)	0,263–0,301
WIKI-727K (Koshorek)	Wikipédia Geral (EN)	0,2213
WIKI-50 (Koshorek)	Wikipédia Geral (EN)	0,1824
WIKI-727K (CATS)	Wikipédia Geral (EN)	0,1595
WIKI-50 (CATS)	Wikipédia Geral (EN)	0,1653

Tabela 7.15: *Comparação de métricas Pk em benchmarks da literatura*

Os resultados apresentados na Tabela 7.15 sugerem que valores de *Pk* abaixo de 0,25 são indicativos de desempenho competitivo em *corpora* bem estruturados, como os artigos da Wikipédia, que possuem seções explícitas e transições de tópicos relativamente claras [Beeferman, Berger e Lafferty 1999, Pevzner e Hearst 2002, Glavaš e Somasundaran 2020]. No entanto, os textos segmentados neste trabalho são predominantemente argumentativos, caracterizados por maior variação discursiva, ausência de seções explícitas e transições entre tópicos mais ambíguas. Essas características aumentam significativamente a complexidade da tarefa de segmentação, tornando os resultados da literatura referências úteis apenas como diretriz aproximada. Assim, a análise dos resultados obtidos no *corpus* desta tese deve considerar esse contexto distinto, que impõe desafios adicionais à identificação de fronteiras entre tópicos.

Análise Exploratória de Dados

Durante a condução da tarefa de segmentação de tópicos, tornou-se necessário definir um parâmetro quantitativo representativo do tamanho dos documentos. Inicialmente, considerou-se a quantidade de sentenças como métrica principal. No entanto, análises exploratórias revelaram que muitos documentos apresentavam trechos extensos com conteúdos digressivos ou fora do escopo temático, o que inflacionava artificialmente essa medida. Diante disso, optou-se por utilizar a quantidade de argumentos identificados como indicador de tamanho. Essa decisão não apenas refletia melhor o objetivo argumentativo da análise, como também foi útil na divisão equitativa da carga de trabalho entre os anotadores.

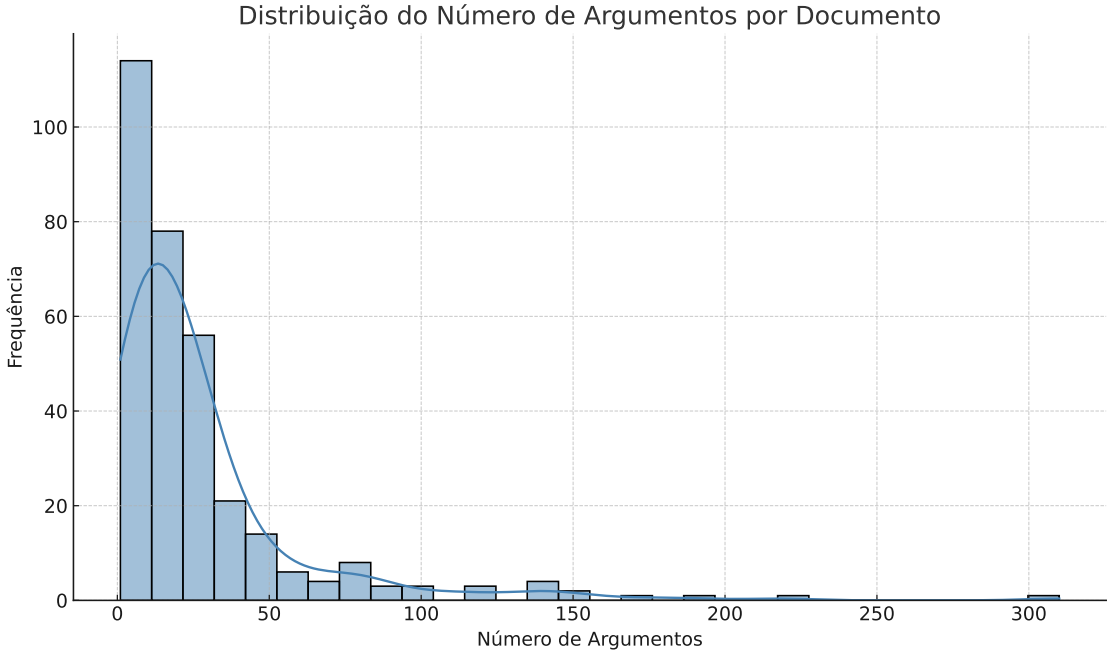


Figura 7.19: *Distribuição do número de argumentos por documento no corpus. A densidade da distribuição apresenta concentração entre 10 e 30 argumentos, com cauda longa à direita, indicando a presença de documentos com número elevado de argumentos.*

A Figura 7.19 ilustra a distribuição da quantidade de argumentos por documento no *corpus*. A média geral é de 27,05 argumentos por documento, com mediana de 17,5, indicando uma assimetria à direita (assimetria = 2,41). O desvio padrão é 35,33, refletindo uma dispersão elevada em relação à média. A maioria dos documentos concentra-se entre 10 e 30 argumentos, como indicado pelos quartis Q1 = 10 e Q3 = 30. No entanto, observa-se uma cauda longa à direita, com documentos que chegam a ultrapassar 300 argumentos. Essa configuração é reforçada pela curtose de 10,62, evidenciando uma distribuição com pico mais acentuado e presença de valores extremos. Tais características estatísticas justificam a adoção da quantidade de argumentos como métrica de tamanho, pois ela oferece uma estimativa mais robusta e sensível ao conteúdo argumentativo efetivo dos textos. Esses dados também fundamentam a definição de faixas utilizadas nas análises estratificadas por tamanho e por tópico.

Análise da Concordância por Tamanho de Documentos

A Tabela 7.17 apresenta a análise da concordância entre anotadores estratificada por faixas de número de argumentos. Os documentos foram organizados em cinco faixas com distribuição próxima à normal, conforme ilustrado na Figura 7.19. As faixas foram definidas da seguinte forma: *1–3*, *4–9*, *10–27*, *28–62* e *63+* argumentos, correspondendo aproximadamente a 10%, 20%, 40%, 20% e 10% da amostra, respectivamente.

Num. de Args.	Qtd de Docs (%)	Gêneros Predominantes (%)			Pk↓	Cohen's Kappa↑
		1º	2º	3º		
1–3	40 (12,50%)	Ne (42,5)	OS (22,5)	Tu (17,5)	0,30	0,57
4–9	59 (18,44%)	Ne (33,9)	Tu (23,73)	Bl (13,56)	0,33	0,53
10–27	128 (40,00%)	Bl (42,97)	Tu (30,47)	Ne (14,06)	0,30	0,56
28–62	62 (19,38%)	Tu (43,55)	Bl (30,65)	Cl (6,45)	0,23	0,61
63+	31 (9,69%)	Tu (38,71)	Bl (16,13)	Re (12,9)	0,23	0,47

Notas: Tu = Tutorial, Bl = *Blog*, Ne = News, OS = Out of Scope, Cl = Clipping, Re = Review.
↓: quanto menor, melhor a concordância. ↑: quanto maior, melhor a concordância.

Tabela 7.16: *Métricas de concordância e gêneros predominantes por faixa de número de argumentos*

A análise revela um comportamento não linear entre o tamanho dos documentos e a concordância entre anotadores. Documentos muito curtos (1–3 argumentos) apresentaram valores relativamente altos de *Pk* (0,30), sugerindo maior imprecisão na delimitação de segmentos, apesar de um valor razoável de *Kappa* (0,57). Isso pode ser explicado pela baixa densidade de fronteiras, o que aumenta o impacto de decisões divergentes em poucas sentenças.

Por outro lado, documentos muito longos (63+ argumentos) apresentaram os melhores resultados segundo a métrica *Pk* (0,23). No entanto, a métrica de *Cohen's Kappa* foi a mais baixa entre todas as faixas (0,47). Esse resultado corrobora observações qualitativas realizadas durante o processo de anotação: embora as fronteiras tenham sido marcadas em posições semelhantes entre os anotadores, o alto volume de sentenças sem fronteira resultou em um forte desbalanceamento entre classes (fronteira vs. não-fronteira), reduzindo artificialmente o valor de *Kappa*. Esse comportamento evidencia a sensibilidade da métrica *Pk* a falsos positivos e negativos localizados, enquanto o *Kappa* tende a ser mais afetado por desbalanceamentos de classe em documentos extensos.

De forma geral, os documentos da faixa intermediária (10–27 argumentos), que representam a maior parte da amostra (40%), apresentaram métricas de concordância estáveis e consistentes, com destaque para o *Kappa* (0,56), sugerindo que essa faixa reflete de maneira mais confiável os padrões de segmentação adotados durante a anotação.

Tópico	Gêneros Predominantes (%) 1º	2º	3º	Pk↓	Cohen's Kappa↑
death penalty	Tu (29)	Bl (24)	Ne (21)	0,21	0,68
cloning	Tu (48)	Bl (23)	Re (9)	0,23	0,64
marijuana legalization	Ne (37)	Bl (26)	Tu (12)	0,27	0,56
nuclear energy	Bl (23)	Tu (20)	Cl (14)	0,29	0,54
abortion	Ne (28)	Bl (26)	Tu (18)	0,32	0,46
school uniforms	Tu (50)	Bl (22)	Ne (8)	0,31	0,55
gun control	Tu (39)	Bl (32)	Ne (10)	0,32	0,56
minimum wage	Bl (43)	Tu (32)	Ne (18)	0,35	0,47
Média				**0,29**	**0,56**

Acrônimos: Tu = Tutorial, Bl = *Blog*, Ne = News, Cl = Clipping, Re = Review.
↓: quanto menor, melhor a concordância. ↑: quanto maior, melhor a concordância.

Tabela 7.17: *Métricas de segmentação e gêneros predominantes por tópico*

Análise da Concordância por Tópico

A Tabela 7.17 apresenta a análise da concordância entre anotadores estratificada por tópico, com destaque para os gêneros textuais predominantes em cada conjunto. Em linha com a análise discutida na Seção 7.3.2, observa-se que os documentos estão concentrados majoritariamente em três gêneros recorrentes — *tutorial* (Tu), *blog* (Bl) e *news* (Ne). Em análise qualitativa, foi identificado que os gêneros *tutorial* e *blog* são frequentemente compostos por textos com função instrucional, voltados a estudantes ou membros de comunidades específicas, o que contribui para uma estrutura discursiva mais regular e, por consequência, uma segmentação mais estável.

Duas exceções relevantes surgem nos tópicos *cloning* e *nuclear energy*, nos quais aparecem os gêneros *review* (Re) e *clipping* (Cl), respectivamente. Esses gêneros não são predominantes nos demais tópicos e refletem particularidades editoriais desses temas: apesar de polêmicos, são menos frequentes na mídia jornalística convencional, o que resulta em uma presença mais expressiva de textos opinativos, científicos ou curados a partir de fontes diversas. Tal diversidade pode introduzir variações estruturais que impactam negativamente a qualidade da segmentação, como evidenciado pelos valores relativamente mais altos de *Pk* (0,23 e 0,29) nesses dois tópicos.

A média geral de *Pk* foi de 0,29, dentro do intervalo típico de desempenho competitivo observado em *corpora* com estrutura explícita, como a Wikipedia. No entanto, é importante destacar que o *corpus* utilizado nesta tarefa é significativamente mais heterogêneo, tanto em termos de estrutura discursiva quanto de gêneros textuais, o que natural-

mente impõe maiores desafios à tarefa de segmentação. Ainda assim, os valores obtidos para *Pk* e *Cohen's Kappa* (média de 0,56) indicam concordância moderada a substancial entre os anotadores, com destaque positivo para o tópico *death penalty*, que obteve os melhores índices de concordância em ambas as métricas.

Análise da Concordância por Gênero Textual

Gênero Textual	**Qtd. de Documentos (%)**	**Pk$^{\downarrow}$**	**Cohen's Kappa$^{\uparrow}$**
term	7 (2,19%)	0,19	0,52
interview	3 (0,94%)	0,23	0,70
critical review	15 (4,69%)	0,25	0,55
threaded posts	8 (2,50%)	0,25	0,49
OutOfScope	19 (5,94%)	0,33	0,46
clipping	19 (5,94%)	0,27	0,50
blog or editorial	88 (27,50%)	0,28	0,57
tutorial or guide	99 (30,94%)	0,27	0,62
talk transcription	1 (0,31%)	0,41	0,58
manifesto	2 (0,62%)	0,28	0,65
news	59 (18,44%)	0,34	0,49
Média Ponderada	**320***	**0,29**	**0,55**
Média Ponderada (sem OutOfScope)	**301****	**0,28**	**0,56**

Tabela 7.18: *Métricas de qualidade por gênero textual.*
** Valor representa o total de documentos utilizados no cálculo da média ponderada geral.*
*** Valor representa o total de documentos utilizados, desconsiderando o gênero OutOfScope.*

A Tabela 7.18 apresenta a análise da concordância entre anotadores por gênero textual, com valores ponderados pela quantidade de documentos em cada grupo, a fim de compensar o desbalanceamento da distribuição.

O gênero *term*, composto por definições enciclopédicas da Wikipedia, apresentou os melhores resultados entre todos os gêneros analisados, com *Pk* de 0,19, valor inferior à faixa de referência observada em *corpora* homogêneos. Esse resultado reforça a influência positiva da homogeneidade textual na tarefa de segmentação, dado o estilo expositivo padronizado e bem delimitado desse tipo de conteúdo.

Por outro lado, o gênero *OutOfScope*, que agrupa documentos sem estrutura argumentativa aproveitável para a tarefa, apresentou o pior valor de *Cohen's Kappa*

(0,46), indicando maior divergência entre os anotadores quanto à presença ou ausência de fronteiras. No entanto, esse comportamento foi menos evidente na métrica *Pk* (0,33), sugerindo que, apesar das divergências locais, a distribuição geral das fronteiras não se distanciou drasticamente entre os anotadores. Ao remover os documentos desse gênero, observou-se uma leve melhora nos valores de *Kappa* (de 0,55 para 0,56) e *Pk* (de 0,29 para 0,28), o que corrobora sua influência negativa.

Entre os demais gêneros, destacam-se *tutorial or guide*, *blog or editorial* e *critical review*, que apresentaram bons níveis de concordância (*Kappa* entre 0,55 e 0,62), compatíveis com sua estrutura mais organizada e segmentável. Gêneros mais variáveis, como *news* e *clipping*, apresentaram métricas ligeiramente inferiores, possivelmente devido à diversidade de formatos e à ausência de divisões temáticas explícitas.

Esses resultados reforçam a importância de considerar o gênero textual como um fator que impacta diretamente na qualidade da segmentação, especialmente em *corpora* heterogêneos.

Considerações Finais da Anotação de Segmentação de Tópicos

A tarefa de segmentação de tópicos constituiu uma etapa essencial na estruturação do *corpus* anotado, fornecendo a base sobre a qual se organizam as anotações semânticas subsequentes, como a classificação de tópicos e a anotação argumentativa. Os resultados obtidos revelam padrões relevantes de concordância entre anotadores que refletem tanto a natureza do conteúdo anotado quanto as propriedades discursivas dos textos.

A métrica *Pk*, adotada como principal indicador de qualidade, apresentou valores médios compatíveis com *benchmarks* reportados na literatura para *corpora* estruturados, apesar da maior heterogeneidade do material aqui analisado. A análise estratificada por tamanho de documento, tópico e gênero textual evidenciou variações esperadas: documentos muito curtos apresentaram maior instabilidade na marcação de fronteiras, enquanto textos longos, embora com boa cobertura de segmentos, apresentaram queda nos valores de *Kappa* devido ao desbalanceamento de classes. Faixas intermediárias de tamanho, tópicos com maior regularidade discursiva e gêneros como *tutorial* e *blog* concentraram os melhores índices de concordância.

A presença de documentos classificados como *OutOfScope* indicou a limitação de métricas como *Kappa* diante de estruturas textuais ruidosas ou não segmentáveis, e reforçou a importância de processos de adjudicação e análise qualitativa complementar. A exclusão desses casos da análise final resultou em leve melhora nas métricas agregadas, sem comprometer a representatividade geral do *corpus*.

De modo geral, os resultados obtidos demonstram que o processo de segmentação atingiu níveis satisfatórios de estabilidade e coerência, especialmente nas regiões centrais da distribuição amostral. Essa camada de anotação cumpre papel central na de-

limitação dos segmentos que servirão de entrada para tarefas posteriores, como a classificação de tópicos e a identificação de unidades argumentativas. Além disso, as análises realizadas contribuíram para consolidar diretrizes práticas para anotação em cenários com variabilidade discursiva e temática significativa.

7.4 Mapeamento de Argumentos Multidocumento (MAMD)

Apesar do foco principal da tese ter sido redirecionado para o mapeamento monodocumento (MAM1), a tarefa original previa o MAMD, com destaque para a mesclagem incremental de árvores argumentativas provenientes de diferentes fontes. Esta tarefa exige não apenas a identificação de unidades argumentativas dentro de cada documento, mas também a integração de suas estruturas em um grafo coeso e não redundante.

A Figura 7.20 ilustra a interface desenvolvida na plataforma *Argmap* para apoiar a anotação dessa etapa. Nela, o anotador pode percorrer argumentos semelhantes (botões `previous` e `next`) e decidir por mesclá-los (botão `merge`) ou rejeitar a sugestão (botão `next source`). O processo ocorre de forma iterativa, com base na similaridade semântica entre subárvores.

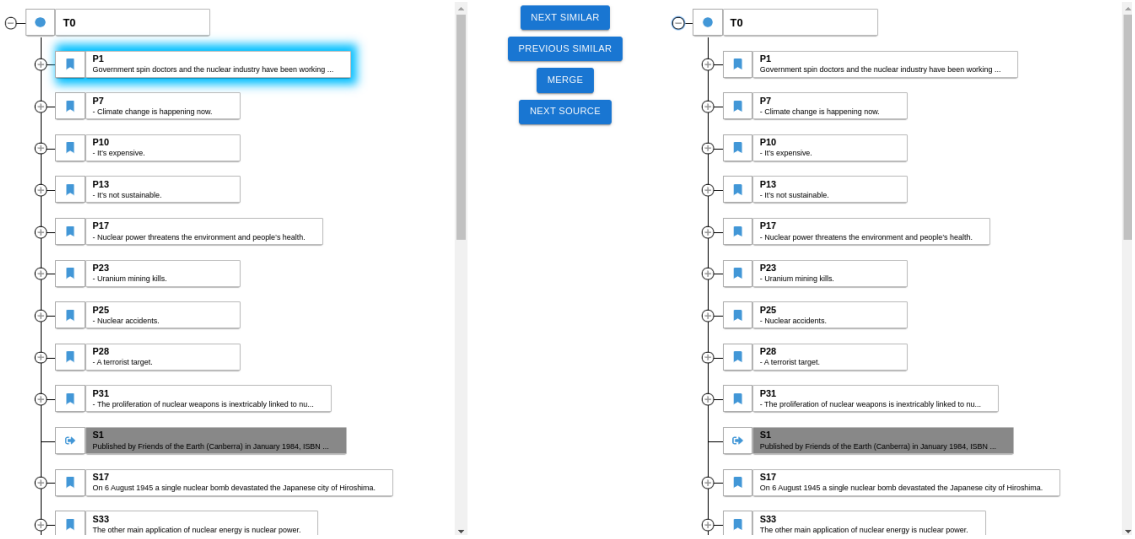


Figura 7.20: *Interface de anotação para mesclagem de árvores de argumentos*

O algoritmo de mesclagem previsto seguia três princípios principais:

- **Agrupamento por tópicos e frases-chave:** as árvores argumentativas seriam inicialmente agrupadas conforme a convergência temática, com base nos rótulos de tópicos e frases-chave associadas.

- **Podagem de redundâncias:** ao percorrer as árvores mescladas, o sistema sugeriria a remoção de subestruturas semanticamente redundantes, mantendo os argumentos mais representativos.
- **Mesclagem semântica assistida:** a fusão dos argumentos seria guiada por *embeddings* de sentenças e medidas de similaridade textual, com validação final por anotadores humanos.

Embora o desenvolvimento completo dessa tarefa tenha sido adiado para trabalhos futuros (ver Seção 8.7), a interface ilustrada acima e a concepção do processo de mesclagem fundamentam uma contribuição promissora para a organização argumentativa em cenários multidocumento.

7.5 Comparação com Abordagens Convencionais de Anotação

A abordagem de anotação proposta nesta tese, fundamentada na decomposição hierárquica de tarefas e na aplicação de padrões reutilizáveis, difere significativamente das práticas convencionais observadas em projetos de anotação de *corpus*. Nesta subseção, apresenta-se uma comparação entre as características metodológicas da abordagem aqui adotada e aquelas tradicionalmente descritas na literatura, com base em trabalhos como [Pustejovsky e Stubbs 2012, Klie, Castilho e Gurevych 2024].

As práticas tradicionais de anotação, conforme sistematizadas por [Pustejovsky e Stubbs 2012], geralmente adotam esquemas fixos, orientados por *guidelines* pouco flexíveis, aplicadas de forma direta sobre o *corpus*-alvo. Em tais abordagens, a avaliação da qualidade costuma ser feita a posteriori, com base em métricas de concordância entre anotadores, como o Kappa de Cohen ou o Alfa de Krippendorff, e os ajustes são realizados manualmente, após ciclos completos de anotação.

Em contraste, a metodologia proposta nesta tese adota um processo iterativo de anotação com controle dinâmico de qualidade, estruturado por um algoritmo de decomposição hierárquica (ver Seção 4.6.1). Essa abordagem viabiliza a subdivisão da tarefa original em subtarefas cognitivamente mais leves (tais como CFC, CT, SCT), cuja confiabilidade pode ser avaliada de forma isolada. Além disso, o processo é assistido por ferramentas especializadas como a *Argmap* e a *KPCTool*, que operacionalizam o padrão arquitetural Recrutador–Selecionador (Seção 4.5) e oferecem mecanismos para adjudicação em tempo real e controle de estabilidade anotativa por lote. A Tabela 7.19 resume as principais diferenças entre essas abordagens.

Conclui-se que, ao adotar uma abordagem baseada em padrões reutilizáveis e estratégias de decomposição, a presente proposta não apenas favorece a reprodutibilidade

Critério	Abordagens Convencionais	Abordagem Proposta
Modelagem da Tarefa	Aplicação direta de esquemas complexos	Decomposição em subtarefas cognitivamente simples
Diretrizes de Anotação	Extensas e estáticas	Iterativas, refinadas por ciclo de estabilidade
Ferramentas de Apoio	Ferramentas genéricas (e.g., BRAT, WebAnno)	Ferramentas próprias com suporte a adjudicação e decomposição (Argmap, KPCTool)
Controle de Qualidade	Avaliação posterior via IAA global	Controle por lote e critérios de estabilidade (Seção 4.6.1)
Automação Parcial	Limitada a pré-rotuladores	Integrada por LLMs, *embeddings* e heurísticas baseadas em padrões linguísticos

Tabela 7.19: *Comparação entre abordagens convencionais e a abordagem proposta para anotação de corpus*

e a escalabilidade do processo de anotação, como também promove ganhos mensuráveis em termos de confiabilidade interanotadores, clareza diretiva e viabilidade de automação. Esta comparação justifica a adoção da abordagem como alternativa metodológica robusta para tarefas complexas de anotação em PLN.

7.6 Análise dos Resultados

A decomposição da tarefa de MAMD revelou-se fundamental para viabilizar sua execução, tanto do ponto de vista técnico quanto operacional. A aplicação do algoritmo de decomposição hierárquica permitiu identificar com precisão os pontos críticos da tarefa original, segmentando-os em subtarefas mais objetivas, com escopo reduzido e maior possibilidade de controle e validação.

Dentre os principais resultados observados, destacam-se:

- **Redução da complexidade cognitiva:** a tarefa original de MAMD, quando abordada diretamente, apresentou baixa reprodutibilidade e exigência elevada de julgamento interpretativo. A divisão em subtarefas, como Detecção de Argumentos (DA), Detecção de Conclusão Dependente de Contexto (DCDC) e CFC, permitiu isolar camadas de decisão, facilitando a orientação por meio de diretrizes mais específicas.

- **Aproveitamento de dados previamente anotados:** ao selecionar o *corpus* UKP Sentential, foi possível reutilizar parcialmente anotações existentes (como rótulos pró/contra), o que diminuiu significativamente o custo de anotação da subtarefa DA. Essa reutilização foi potencializada pela recuperação do contexto original das sentenças, transformando anotações descontextualizadas em dados contextualizados.
- **Geração de dados derivados e adaptação de *guidelines*:** a necessidade de mapear sentenças isoladas para seus respectivos contextos exigiu a criação de novas ferramentas e diretrizes, ampliando a base de conhecimento sobre anotações dependentes de contexto. Essa etapa foi essencial para o refinamento da tarefa DCDC e evidenciou a importância de *guidelines* dinâmicas, capazes de evoluir com base na evidência observacional.
- **Identificação de padrões argumentativos via indicadores discursivos:** a extração de indicadores para conclusões e premissas, conforme descrito nas Tabelas 7.4 a C.1, possibilitou a construção de heurísticas para apoio à anotação. Esses padrões funcionaram como base para uma abordagem mais sistemática da identificação de componentes argumentativos, mitigando subjetividades.
- **Viabilidade incremental da tarefa original:** a adoção de uma estratégia de busca em profundidade (DFS) permitiu o desenvolvimento progressivo de subtarefas com menor dependência entre si. Esse encadeamento lógico facilitou o avanço controlado da tarefa, respeitando as interdependências da árvore de decomposição.

De maneira geral, os resultados indicam que a abordagem de decomposição baseada em evidências e dependências funcionais contribui significativamente para a melhoria da qualidade e viabilidade de anotações complexas em PLN. A geração iterativa de dados, orientada por heurísticas e pela reavaliação contínua das diretrizes, consolidou um ciclo virtuoso de refinamento anotativo.

7.7 Conclusão

A aplicação do algoritmo de decomposição hierárquica de tarefas à anotação de *corpus* demonstrou-se uma estratégia eficaz para lidar com a complexidade inerente à tarefa de MAMD. Ao segmentar a tarefa em unidades menores, orientadas por dependências funcionais e evidências empíricas, foi possível reduzir a carga cognitiva dos anotadores, aumentar a clareza das diretrizes e promover maior estabilidade nas anotações.

A estratégia de decomposição permitiu também a identificação de padrões de projeto recorrentes no processo anotativo, como o uso de *corpora* parcialmente anotados, a curadoria de sentenças com base em indicadores discursivos e a priorização de tarefas mais objetivas como forma de mitigar ambiguidade interpretativa. Tais padrões não

apenas facilitaram a execução prática das subtarefas, como também forneceram insumos valiosos para o desenvolvimento de *guidelines* reutilizáveis em futuros projetos de anotação.

Outro aspecto relevante foi a adoção de uma abordagem iterativa, inspirada na arquitetura da complexidade proposta por Simon, em que decisões são tomadas com base em dados coletados ao longo do próprio processo. Essa postura permitiu adaptar a tarefa às limitações práticas (como custo e disponibilidade de especialistas) sem comprometer sua validade científica.

Por fim, conclui-se que a decomposição de tarefas de anotação, quando guiada por critérios empíricos e teóricos bem definidos, configura-se como um poderoso recurso metodológico para pesquisas em PLN. Ela não apenas torna viáveis tarefas inicialmente inviáveis, como também contribui para a construção de *corpora* mais confiáveis, escaláveis e úteis à comunidade científica.

CAPÍTULO 8

Considerações Finais

Este capítulo apresenta uma síntese dos principais resultados da pesquisa, articulando as respostas às questões de pesquisa, a avaliação dos objetivos definidos, as limitações encontradas, as perspectivas para trabalhos futuros e uma conclusão geral. A organização das seções visa também esclarecer os problemas abordados, os padrões identificados e os resultados empíricos obtidos ao longo dos estudos de caso.

A tese foi construída com base em quatro conjecturas fundamentais, apresentadas no Capítulo 1, que fornecem a sustentação teórica da abordagem adotada. A primeira conjectura afirma que sistemas complexos hierárquicos podem ser decompostos, até certo limite, em partes compreensíveis por seres humanos. A segunda propõe que a nomeação de padrões recorrentes constitui uma linguagem operacional que potencializa a abstração e a sistematização de processos. A terceira defende que a anotação de qualidade funciona como forma de modelagem conceitual da tarefa, sendo essencial para sua posterior automatização. Por fim, a quarta conjectura estabelece que o ponto de parada da decomposição ocorre quando as subtarefas atingem um nível de simplicidade suficiente para garantir anotações confiáveis entre diferentes agentes humanos.

Essas conjecturas fundamentaram o desenvolvimento de uma metodologia de decomposição de tarefas voltada à anotação de *corpus* em PLN, com ênfase na estruturação hierárquica, na identificação de padrões reutilizáveis e no controle sistemático da qualidade. A aplicação dessa abordagem em três estudos de caso — segmentação de *hashtags*, curadoria de frases-chave e anotação de estruturas argumentativas — permitiu avaliar seus efeitos práticos e validar suas premissas conceituais.

Como contribuição central, a tese propõe uma teoria de padrões aplicáveis à decomposição de tarefas de anotação, acompanhada de um algoritmo para modelagem hierárquica de fluxos de anotação e de artefatos empíricos desenvolvidos e avaliados em contextos reais. Os resultados demonstraram ganhos em clareza conceitual, replicabilidade, controle de qualidade e viabilidade de automação das tarefas-alvo.

As seções a seguir aprofundam essas reflexões, estruturadas da seguinte forma: (i) uma síntese dos problemas enfrentados e das contribuições propostas; (ii) a revisão das questões de pesquisa e suas respectivas respostas; (iii) a avaliação dos objetivos

específicos e dos resultados empíricos alcançados; (iv) a sistematização dos padrões identificados e das métricas aplicadas; (v) a discussão das limitações do estudo; (vi) as direções para trabalhos futuros; e (vii) uma conclusão geral da tese.

8.1 Síntese dos Problemas Abordados e Contribuições

A presente tese partiu do reconhecimento de que muitas tarefas fundamentais do PLN dependem da disponibilidade de *corpora* anotados com alta qualidade. No entanto, a anotação de tarefas complexas apresenta diversos entraves recorrentes, entre os quais se destacam: a elevada carga cognitiva imposta aos anotadores, a dificuldade de formular diretrizes compreensíveis e reproduzíveis, a baixa consistência entre agentes humanos e os altos custos para obtenção de dados confiáveis em larga escala.

Esses desafios se agravam em contextos onde as tarefas exigem raciocínio estruturado, como nas tarefas de segmentação temática, curadoria semântica e mapeamento de estruturas argumentativas. Nesses casos, a ausência de ferramentas especializadas, a rigidez dos esquemas de anotação e a falta de controle sobre o processo dificultam tanto a produção de dados quanto sua posterior utilização para fins de avaliação e automação.

Para enfrentar esse conjunto de problemas, a tese propõe uma abordagem centrada na decomposição de tarefas, fundamentada teoricamente por uma teoria de padrões aplicáveis ao processo de anotação. Essa abordagem orienta-se por três eixos principais: (i) a estruturação hierárquica das tarefas a partir de padrões cognitivos e computacionais; (ii) a modelagem de subtarefas cognitivamente mais simples, com diretrizes mais localizadas e menos ambíguas; e (iii) o controle da qualidade por meio de ciclos iterativos de anotação e validação, com métricas específicas para cada subtarefa.

A aplicação dessa abordagem em três estudos de caso permitiu validar suas premissas em contextos distintos: na tarefa de segmentação de *hashtags*, foi possível atingir o estado da arte com modelos *zero-shot*; na curadoria de frases-chave, obteve-se maior cobertura semântica e estabilidade entre anotadores; e na anotação de estruturas argumentativas, demonstrou-se a viabilidade de modular fluxos complexos de decisão anotativa com o suporte de ferramentas desenvolvidas especificamente para tal fim.

Como contribuição, a tese entrega não apenas uma proposta metodológica de decomposição, mas também uma teoria de padrões reutilizáveis para modelagem de tarefas de anotação, um algoritmo formal para decomposição hierárquica, ferramentas de apoio (KPCTool e Argmap) e artefatos empíricos avaliados com métricas apropriadas a cada subtarefa. Tais contribuições visam fortalecer a base teórica e prática para o desenvolvimento de *corpora* mais confiáveis, reprodutíveis e adequados à avaliação de modelos em PLN.

8.2 Revisitação das Questões de Pesquisa

A pesquisa foi orientada por três questões centrais, apresentadas no Capítulo 1, cuja investigação guiou a construção teórica, o delineamento metodológico e a avaliação empírica dos estudos de caso. A seguir, revisitam-se essas questões à luz dos resultados obtidos.

Q1. Como decompor tarefas complexas de anotação em subtarefas cognitivamente mais simples e passíveis de automação?

A tese propôs uma metodologia de decomposição baseada na identificação de padrões de projeto recorrentes, estruturados em um grafo acíclico dirigido de subtarefas. Cada subtarefa é modelada como uma função parcial, cuja composição permite a reconstituição da tarefa original. Essa decomposição pode assumir formas sequenciais, paralelas ou hierárquicas, dependendo da natureza do fluxo anotativo.

A abordagem foi aplicada com sucesso nos três estudos de caso, permitindo transformar tarefas originalmente complexas em fluxos mais controláveis, tanto do ponto de vista cognitivo quanto computacional. Na curadoria de frases-chave, por exemplo, a divisão em subtarefas de agrupamento, filtragem e seleção revelou-se particularmente eficaz para isolar critérios linguísticos e semânticos distintos, facilitando tanto a anotação quanto a avaliação automática.

Q2. Quais padrões são úteis para organizar o fluxo de anotação e promover qualidade, reprodutibilidade e automação?

A pesquisa identificou e formalizou diferentes tipos de padrões aplicáveis à decomposição de tarefas: padrões arquiteturais (como o Recrutador–Selecionador), padrões linguísticos (como as construções argumentativas recorrentes) e padrões operacionais (como ciclos de refinamento por estabilidade).

Esses padrões foram abstraídos a partir da análise de tarefas reais e validados empiricamente nos estudos de caso. A operacionalização dos padrões em ferramentas específicas (KPCTool e Argmap) permitiu não apenas modular o fluxo de anotação, mas também incorporar mecanismos internos de controle de qualidade (como adjudicação, avaliação parcial e ciclos de validação), promovendo maior consistência e reprodutibilidade entre agentes anotadores.

Q3. Como avaliar os efeitos da decomposição no processo anotativo e na qualidade dos dados resultantes?

A avaliação dos efeitos da decomposição foi conduzida a partir de múltiplas métricas, específicas para cada tipo de subtarefa: - para segmentação, foram utilizadas métricas baseadas em fronteiras como *WindowDiff* e P_k; - para tarefas de classificação multirrótulo, foi empregado o alfa de Krippendorff com distância de Jaccard; - para etapas de seleção categórica, utilizou-se o Kappa de Fleiss.

Os resultados indicam que a decomposição promove maior clareza nas diretrizes, maior estabilidade entre anotadores e melhor suporte à automação supervisionada. Adicionalmente, observou-se que a decomposição favorece o uso de métricas mais sensíveis às propriedades locais das decisões de anotação, o que permite diagnósticos mais precisos e intervenções mais eficazes no processo de anotação.

Essas evidências empíricas reforçam a tese de que a decomposição orientada por padrões é uma estratégia promissora para enfrentar os principais obstáculos na construção de *corpora* anotados com qualidade em PLN.

8.3 Avaliação dos Objetivos e Resultados Empíricos

Esta seção avalia o grau de atendimento aos objetivos estabelecidos na tese, conforme apresentados no Capítulo 1, relacionando-os aos resultados obtidos nos estudos de caso. O objetivo geral desta pesquisa consistiu em investigar como a decomposição de tarefas pode contribuir para a anotação de *corpus* em PLN, promovendo maior clareza conceitual, qualidade de anotação, reprodutibilidade e viabilidade de automação.

Para alcançar esse objetivo central, foram definidos cinco objetivos específicos, cujas realizações são detalhadas a seguir. Para cada objetivo, identifica-se o estudo de caso mais diretamente associado, os artefatos produzidos e as evidências empíricas que sustentam sua validação.

Objetivo Específico	**Estudo de Caso**	**Resultados Empíricos**
Investigar a viabilidade da decomposição de tarefas de anotação em subtarefas mais simples	CFC (Cap. 6)	Definição de três subtarefas (agrupamento, filtragem, seleção) com diretrizes específicas; aumento da consistência entre anotadores; avaliação por Kappa de Fleiss
Identificar padrões de projeto úteis à organização de tarefas anotativas	Três estudos de caso	Formalização dos padrões Recrutador–Selecionador, Ciclo de Estabilidade e Segmentador Iterativo; abstração de padrões linguísticos e arquiteturais
Avaliar os efeitos da decomposição sobre a qualidade da anotação e a reprodutibilidade	Segmentação de *Hashtags* e Mapeamento de Argumentos (Cap. 5, 7)	Redução da variância inter-anotador; aplicação de métricas específicas (WindowDiff, P_k, Krippendorff-α com Jaccard); maior replicabilidade dos resultados
Desenvolver ferramentas para apoiar fluxos de anotação baseados em decomposição	KPCTool e Argmap	Ferramentas com suporte a adjudicação, decomposição modular, visualização e controle por lote; publicadas em repositórios públicos
Gerar e disponibilizar *corpora* anotados com qualidade verificável	Três estudos de caso	Lançamento de *corpora* com diferentes granularidades (segmentos, frases-chave, estruturas argumentativas), avaliados por múltiplos anotadores com critérios de estabilidade

Tabela 8.1: *Relação entre objetivos específicos, estudos de caso e resultados alcançados*

A tabela 8.1 demonstra que todos os objetivos específicos foram efetivamente endereçados por meio de estudos empíricos e artefatos documentados. Além disso, os resultados corroboram as conjecturas teóricas propostas e evidenciam os benefícios da abordagem adotada, tanto no plano metodológico quanto prático.

Em particular, destaca-se que a decomposição não apenas viabilizou a execução das tarefas por anotadores humanos, como também promoveu maior controle sobre a qualidade dos dados, facilitando a aplicação de modelos automáticos e a reusabilidade dos *corpora* gerados.

8.4 Padrões Identificados e Métricas Aplicadas

Ao longo dos estudos de caso, foram identificados diversos padrões recorrentes no processo de decomposição e anotação, os quais foram formalizados, testados empiricamente e utilizados para organizar fluxos de trabalho, orientar diretrizes e estruturar esquemas de anotação. Esses padrões são consistentes com a abordagem teórica adotada nesta tese, especialmente a perspectiva da teoria baseada em uso, segundo a qual categorias e estruturas funcionais emergem e se estabilizam progressivamente a partir de práticas recorrentes e intersubjetivamente validadas.

Os padrões identificados podem ser agrupados em três categorias principais:

- **Padrões arquiteturais:** dizem respeito à forma de organização do fluxo anotativo. O principal exemplo é o padrão *Recrutador–Selecionador*, aplicado na Segmentação de *Hashtags* por meio do algoritmo HSBS, que adota uma etapa inicial heurística de geração de candidatos e uma etapa posterior de reclassificação com base em verossimilhança. Essa separação de responsabilidades torna o fluxo mais controlável e interpretável.
- **Padrões linguísticos:** emergem de regularidades na estrutura textual das unidades anotadas. No caso da anotação argumentativa, foram identificadas construções linguísticas típicas com base em operadores discursivos (e.g., “porque”, “embora”, “por exemplo”), formalizadas em estruturas como PIC, CIP, IPC e ICP, descritas no Apêndice C. Essas construções funcionam como padrões linguísticos recorrentes úteis para segmentação e classificação.
- **Padrões operacionais:** dizem respeito à condução prática do processo de anotação e validação. Três padrões merecem destaque:
  - O *Ciclo de Estabilidade*, utilizado no mapeamento argumentativo, no qual lotes de anotação sucessivos são produzidos até que se atinja uma estabilidade aceitável entre anotadores;
  - A formulação de *guidelines estáveis*, cuja reusabilidade e robustez são aferidas pela consistência entre anotações independentes;
  - A estruturação de *esquemas de anotação* como padrões emergentes, validados empiricamente por meio de exemplos anotados e refinados ao longo do processo, em conformidade com a teoria baseada em uso.

A Tabela 8.2 sintetiza os principais padrões utilizados e as métricas associadas à sua avaliação nos diferentes estudos de caso:

Padrão	**Estudo de Caso**	**Métrica(s) Utilizada(s)**
Recrutador–Selecionador	Segmentação de *Hashtags*	Método heurístico baseado em verossimilhança (*likelihood*) de padrões tokenizados no algoritmo HSBS
Ciclo de Estabilidade	Mapeamento Argumentativo	Critério de convergência por estabilidade interanotador; ciclos iterativos com validação qualitativa
Construções Argumentativas por Indicadores Discursivos (PIC, CIP, etc.)	Mapeamento de Argumentos (Apêndice C)	Extração fundamentada na teoria baseada em uso; categorização funcional de unidades por operadores linguísticos
Guidelines Estáveis como Padrões Operacionais	Todos os estudos	Concordância interanotador (Kappa de Fleiss, Krippendorff-α); ciclos de adjudicação e validação independente
Esquemas de Anotação como Padrões Emergentes	Todos os estudos	Validação empírica a partir de exemplos anotados; refinamento iterativo com base na Teoria Baseada no Uso; estabilidade estrutural nos *datasets* resultantes

Tabela 8.2: *Padrões identificados e métricas aplicadas nos estudos de caso*

A análise dos resultados indica que a adoção explícita desses padrões teve impacto direto na clareza das diretrizes, na organização do processo anotativo, na consistência entre anotadores e na qualidade dos *corpora* produzidos. Em especial, os padrões operacionais — tanto os esquemas quanto os *guidelines* — revelaram-se essenciais para promover reprodutibilidade e confiabilidade, ao passo que os padrões linguísticos forneceram base funcional para delimitar unidades e categorizar argumentos. Já os padrões arquiteturais possibilitaram modular fluxos complexos em etapas interpretáveis e automa-

tizáveis, demonstrando a força da decomposição por padrões como princípio organizador da anotação em PLN.

Em especial, destaca-se que a combinação entre decomposição estruturada e uso de padrões linguísticos fundamentados na teoria baseada em uso fortaleceu a interpretabilidade dos dados anotados, contribuindo para uma modelagem conceitual mais transparente das unidades argumentativas.

8.5 Revisitação das Conjecturas Fundamentais

As quatro conjecturas formuladas no Capítulo 1 ofereceram a base conceitual sobre a qual esta tese foi construída. A seguir, revisitam-se essas conjecturas à luz das evidências empíricas e teóricas acumuladas ao longo dos estudos de caso.

Conjectura 1 (Compreensão Hierárquica). Todo sistema complexo hierárquico é, até certo limite, decomponível em partes compreensíveis por seres humanos. Os estudos de caso demonstraram que a decomposição de tarefas de anotação em subtarefas hierarquicamente organizadas — como segmentar, agrupar, filtrar, classificar — permite uma distribuição mais equilibrada da carga cognitiva, resultando em maior clareza para os anotadores e melhor controle para os supervisores. A formalização matemática da decomposição em termos de funções parciais e DAGs contribuiu para evidenciar a relação entre estrutura e esforço cognitivo.

Conjectura 2 (Linguagem de Padrões). A compreensão humana de tarefas complexas depende de representações simbólicas. Nomear padrões recorrentes nessas tarefas alavanca a capacidade de raciocínio, abstração e comunicação, constituindo uma linguagem operacional que facilita a estruturação, o reuso e a automação de processos. Os padrões identificados e documentados nesta tese — arquiteturais, linguísticos e operacionais — ilustram como a nomeação, exemplificação e formalização de boas práticas recorrentes permitiram sistematizar processos complexos, apoiando tanto a anotação manual quanto o *design* de ferramentas.

Conjectura 3 (Anotação como Modelagem). Dado um processo de anotação que produza exemplos consistentes e representativos, é possível treinar modelos de aprendizado de máquina capazes de executar a tarefa-alvo com desempenho satisfatório. Assim, a anotação opera como forma de modelagem conceitual da tarefa, sendo tanto meio de formalização quanto ponte para sua automatização. Essa conjectura foi corroborada especialmente nas tarefas de segmentação e classificação, nas quais os dados anotados com base em esquemas bem definidos foram reutilizados para avaliar modelos automáticos ou orientar abordagens supervisionadas.

Conjectura 4 (Primitivas de Anotação). A decomposição de uma tarefa atinge um ponto de parada quando as subtarefas resultantes se tornam suficientemente simples

para serem executadas com qualidade consistente entre diferentes anotadores, baseadas em confiança mútua e entendimento compartilhado. Essas unidades mínimas são chamadas de primitivas de anotação e representam o limite funcional da decomposição, além do qual há risco de perda de significado ou aumento desnecessário da fragmentação. A estratégia de ciclos de estabilidade, aplicada nos estudos, permitiu identificar empiricamente esse ponto de parada, com base em critérios como concordância entre anotadores e redução de ambiguidade interpretativa.

Ao serem testadas em contextos distintos, essas conjecturas não apenas orientaram a metodologia adotada, mas também foram operacionalizadas por meio de algoritmos, ferramentas e esquemas que reforçam sua validade como princípios estruturantes para a organização de tarefas de anotação no PLN.

De modo geral, os estudos de caso serviram como prova de conceito para essas quatro conjecturas, demonstrando que elas não apenas fundamentam teoricamente a abordagem, mas também oferecem critérios práticos para seu planejamento, aplicação e avaliação.

8.6 Limitações e Ameaças à Validade

Embora os resultados obtidos nesta tese ofereçam evidências favoráveis à decomposição de tarefas orientada por padrões como estratégia metodológica para anotação de *corpus* em PLN, é importante reconhecer as limitações e ameaças à validade que permeiam o estudo.

Cobertura limitada de tarefas e domínios

Os estudos de caso selecionados abordam tarefas distintas — segmentação de *hashtags*, curadoria de frases-chave e mapeamento de estruturas argumentativas —, mas ainda representam um subconjunto das múltiplas tarefas anotativas possíveis no PLN. Além disso, os domínios dos textos utilizados são predominantemente opinativos e jornalísticos, o que pode limitar a generalização dos padrões identificados a outros contextos, como textos científicos, jurídicos ou conversacionais.

Dependência de ferramentas em desenvolvimento

Parte dos experimentos dependeu de ferramentas desenvolvidas ao longo da própria tese (como a KPCTool e o Argmap), o que trouxe ganhos de alinhamento com os objetivos da pesquisa, mas também impôs limitações práticas relacionadas à maturidade das interfaces, suporte a usuários externos e replicação independente dos experimentos.

Embora os códigos e *datasets* estejam disponíveis, a reprodutibilidade completa pode demandar suporte técnico adicional.

Uso parcial de LLMs e limitações de avaliação automática

Apesar de LLMs terem sido explorados pontualmente em algumas tarefas (como segmentação zero-shot de *hashtags*), sua aplicação não foi sistemática ao longo de todos os estudos de caso. Isso reflete tanto o escopo metodológico definido (centrado na ação de agentes humanos), quanto as limitações práticas e orçamentárias da época da pesquisa. Ainda assim, isso limita a avaliação do potencial da abordagem proposta em cenários híbridos humano-LLM.

Além disso, as métricas aplicadas — embora específicas e bem justificadas — possuem limitações inerentes. Por exemplo, métricas como *WindowDiff* e P_k são sensíveis à granularidade da segmentação; o Kappa de Fleiss assume independência entre rótulos; e o alfa de Krippendorff depende da definição adequada de distância semântica. Tais fatores podem introduzir ruído nas análises se não forem cuidadosamente controlados.

Adoção limitada dos padrões por outros anotadores

Embora os padrões propostos tenham-se mostrado úteis nos contextos avaliados, sua adoção por outros grupos de anotadores ainda não foi avaliada de forma sistemática. A nomeação de padrões como linguagem operacional e a decomposição por meio de grafos acíclicos são propostas inovadoras, mas ainda carecem de validação ampla em diferentes equipes, idiomas e tarefas. Essa lacuna impõe restrições à robustez externa da teoria desenvolvida.

Possíveis vieses na interpretação e na anotação

Como em qualquer tarefa anotativa, há riscos de viés cognitivo, cultural ou interpretativo, tanto na formulação das diretrizes quanto na aplicação pelos anotadores. A tese buscou mitigar esse problema por meio de ciclos de validação e adjudicação, mas o viés pode persistir, especialmente em tarefas como a anotação argumentativa, nas quais o julgamento pragmático desempenha papel central.

Reconhecer essas limitações é fundamental para delinear com maior precisão o escopo dos resultados e orientar pesquisas futuras voltadas à expansão, validação cruzada e consolidação da abordagem aqui proposta.

8.7 Trabalhos Futuros

A pesquisa desenvolvida nesta tese abre um conjunto de possibilidades práticas e teóricas que podem ser exploradas em trabalhos futuros, tanto no aprofundamento de estudos em andamento quanto na aplicação da abordagem de decomposição de tarefas a novas áreas.

- **Completar a anotação do *dataset* de Mapeamento de Argumentos.** A conclusão do *corpus* de anotação argumentativa ainda demanda viabilização financeira, o que pode ser alcançado pela redução dos custos por meio de recomendadores automáticos, interfaces mais produtivas e estratégias de ancoragem cognitiva que aumentem a eficiência dos anotadores humanos.
- **Automatização da tarefa de Mapeamento de Argumentos.** Uma vez concluído o *dataset*, será possível compor as subtarefas anotadas em um *pipeline* completo, treinando modelos supervisionados para executar a tarefa automaticamente e avaliando o desempenho obtido em cada etapa e no sistema integrado.
- **Aplicação do Mapeamento de Argumentos em diferentes domínios.** O framework desenvolvido pode ser adaptado para áreas como: jurídico (análise de defesas e acusações), político (comentários sobre projetos de lei), científico (argumentos sobre hipóteses concorrentes), redes sociais (análise de controvérsias públicas) e comércio (motivações de compra, reclamações ou avaliações).
- **Inteligência Artificial Explicável (xIA).** A decomposição de tarefas pode contribuir com a auditoria e compreensão de decisões automatizadas em sistemas críticos, como os aplicados nas áreas médica, jurídica e financeira. A modularidade da tarefa permite maior rastreabilidade de inferências e facilita o diagnóstico de falhas.
- **Inteligência Artificial Geral (AGI).** A metodologia proposta pode ser explorada como ferramenta para analisar a complexidade estrutural de tarefas, identificar encadeamentos de raciocínio e avaliar a capacidade de manipulação simbólica de agentes inteligentes, contribuindo para o estudo da emergência de habilidades cognitivas em sistemas generalistas.
- **Engenharia de *Software* baseada em LLMs.** A taxonomia de padrões proposta pode apoiar a modularização de tarefas e a composição de arquiteturas de inferência integradas a processos de negócio. Isso inclui a integração com *pipelines* empresariais assistidos por LLMs, com potencial para organização de fluxos de decisão e controle de qualidade automatizado.
- **Automatização da anotação e da decomposição de tarefas.** Uma frente promissora envolve o desenvolvimento de agentes capazes de propor decomposições automaticamente, gerar exemplos anotados com supervisão mínima e apoiar a criação

de *guidelines* modulares. Esses sistemas híbridos podem reduzir significativamente a carga cognitiva e o custo humano da anotação de *corpus*.

8.8 Conclusão Geral

Esta tese investigou como a decomposição de tarefas pode contribuir para a anotação de *corpus* em PLN, promovendo clareza conceitual, qualidade de anotação, reprodutibilidade e viabilidade de automação. O trabalho partiu de quatro conjecturas teóricas fundamentais que sustentaram a abordagem adotada: a decomponibilidade hierárquica de sistemas complexos, a nomeação de padrões como linguagem operacional, a anotação como forma de modelagem conceitual e a existência de primitivas de anotação como ponto de parada da decomposição.

A partir dessas bases, foi desenvolvida uma metodologia de decomposição orientada por padrões, operacionalizada por meio de três estudos de caso complementares: segmentação de *hashtags*, curadoria de frases-chave e mapeamento de estruturas argumentativas. Cada estudo explorou diferentes desafios cognitivos, linguísticos e operacionais, permitindo validar as hipóteses teóricas e testar os padrões em contextos distintos.

Como resultado, a tese consolidou uma teoria prática de padrões reutilizáveis aplicáveis à anotação de *corpus*, estruturada em três classes principais:

- **Padrões arquiteturais**, como o Recrutador–Selecionador, que modularizam fluxos complexos em subtarefas especializadas;
- **Padrões linguísticos**, como as construções baseadas em indicadores discursivos, que guiam a segmentação e classificação de unidades argumentativas;
- **Padrões operacionais**, como esquemas de anotação e *guidelines* estáveis, cuja robustez é aferida por métricas de concordância entre anotadores.

A formalização da decomposição em termos de grafos acíclicos de subtarefas, a implementação de ferramentas como KPCTool e Argmap, e a geração de *corpora* anotados com critérios rigorosos de qualidade configuram um conjunto coerente de contribuições teóricas e empíricas. Esses artefatos não apenas demonstram a viabilidade da abordagem, mas também oferecem caminhos replicáveis para futuras tarefas de anotação em PLN.

A tese contribui, portanto, para o avanço metodológico na construção de dados anotados, fornecendo uma estrutura conceitual robusta, instrumentos de apoio à prática de anotação de dados e evidências de que a decomposição orientada por padrões é uma estratégia eficaz para enfrentar os desafios inerentes à anotação de tarefas complexas.

Índice Remissivo

Referências

[Abdelrazek et al. 2023]ABDELRAZEK, A. et al. Topic modeling algorithms and applications: A survey. *Information Systems*, v. 112, p. 102131, 2023. ISSN 0306-4379. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0306437922001090>.

[Adebayo, Caro e Boella 2016]ADEBAYO, K. J.; CARO, L. D.; BOELLA, G. Text segmentation with topic modeling and entity coherence. In: IEEE. *Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS)*. [S.l.], 2016. p. 130–137.

[Alammar 2025]ALAMMAR, J. *The Illustrated Transformer*. 2025. Licenciado sob CC BY-SA 4.0 Internacional. https://creativecommons.org/licenses/by-sa/4.0/deed.en. Disponível em: <https://jalammar.github.io/illustrated-transformer/>.

[Alexander 1979]ALEXANDER, C. *The Timeless Way of Building*. [S.l.]: Oxford University Press, 1979. ISBN 0-19-502402-8 978-0-19-502402-9.

[Alexander, Ishikawa e Silverstein 1977]ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M. *A Pattern Language: Towns, Buildings, Construction (Cess Center for Environmental)*. [S.l.]: Oxford University Press, 1977. ISBN 0-19-501919-9 978-0-19-501919-3.

[Ali e Pazzani 1996]ALI, K. M.; PAZZANI, M. J. Error reduction through learning multiple descriptions. *Machine Learning*, v. 24, n. 3, p. 173–202, set. 1996. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00058611>.

[Anand et al. 1995]ANAND, R. et al. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, v. 6, n. 1, p. 117–124, 1995.

[Andersen e Maalej 2023]ANDERSEN, J. S.; MAALEJ, W. *Design Patterns for Machine Learning Based Systems with Human-in-the-Loop*. arXiv, 2023. ArXiv:2312.00582 [cs] version: 1. Disponível em: <http://arxiv.org/abs/2312.00582>.

[Arksey e O'Malley 2005]ARKSEY, H.; O'MALLEY, L. Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology*, v. 8, n. 1, p. 19–32, fev. 2005. ISSN 1364-5579. Publisher: Rou-

tledge _eprint: https://doi.org/10.1080/1364557032000119616. Disponível em: <https://doi.org/10.1080/1364557032000119616>.

[Arnold et al. 2019]ARNOLD, S. et al. SECTOR: A Neural Model for Coherent Topic Segmentation and Classification. *Transactions of the Association for Computational Linguistics*, v. 7, p. 169–184, 2019. Place: Cambridge, MA Publisher: MIT Press. Disponível em: <https://aclanthology.org/Q19-1011/>.

[Artstein e Poesio 2008]ARTSTEIN, R.; POESIO, M. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, v. 34, n. 4, p. 555–596, 2008.

[Asher et al. 2023]ASHER, N. et al. Limits for learning with language models. In: PALMER, A.; CAMACHO-COLLADOS, J. (Ed.). *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*. Toronto, Canada: Association for Computational Linguistics, 2023. p. 236–248. Disponível em: <https://aclanthology.org/2023.starsem-1.22/>.

[Atkinson 1968]ATKINSON, R. C. Human memory: A proposed system and its control processes. *The psychology of learning and motivation*, v. 2, 1968.

[Austin et al. 2021]AUSTIN, J. et al. Program synthesis with large language models. *NeurIPS*, 2021.

[Bay 1999]BAY, S. D. Nearest neighbor classification from multiple feature subsets. *Intelligent Data Analysis*, v. 3, n. 3, p. 191–209, 1999. ISSN 1088-467X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1088467X99000189>.

[Beck 1999]BECK, K. *Extreme Programming Explained: Embrace Change*. 1. ed. Boston, MA: Addison-Wesley, 1999. ISBN 978-0201616415.

[Beeferman, Berger e Lafferty 1999]BEEFERMAN, D.; BERGER, A.; LAFFERTY, J. Statistical models for text segmentation. *Machine Learning*, Springer, v. 34, n. 1, p. 177–210, 1999.

[Bengio, Courville e Vincent 2013]BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 35, n. 8, p. 1798–1828, 2013.

[Bengio, Simard e Frasconi 1994]BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, IEEE, v. 5, n. 2, p. 157–166, 1994.

[Bertalanffy 1969]BERTALANFFY, L. V. *General System Theory: Foundations, Development, Applications*. 1. ed. [S.l.]: George Braziller Inc., 1969. (Penguin University Books).

[Bhargava 1999]BHARGAVA, H. K. Data Mining by Decomposition: Adaptive Search for Hypothesis Generation. *INFORMS Journal on Computing*, ago. 1999. Publisher: INFORMS. Disponível em: <https://pubsonline.informs.org/doi/abs/10.1287/ijoc.11.3.239>.

[Bishop e Nasrabadi 2006]BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4.

[Blei, Carin e Dunson 2010]BLEI, D.; CARIN, L.; DUNSON, D. Probabilistic Topic Models. *IEEE Signal Processing Magazine*, v. 27, n. 6, p. 55–65, nov. 2010. ISSN 1558-0792. Conference Name: IEEE Signal Processing Magazine.

[Blei 2012]BLEI, D. M. Probabilistic topic models. *Communications of the ACM*, v. 55, n. 4, p. 77–84, abr. 2012. ISSN 0001-0782, 1557-7317. Disponível em: <https://dl.acm.org/doi/10.1145/2133806.2133826>.

[Blei, Ng e Jordan 2003]BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003.

[Box e Jenkins 1990]BOX, G. E. P.; JENKINS, G. *Time Series Analysis, Forecasting and Control*. USA: Holden-Day, Inc., 1990. ISBN 0-8162-1104-3.

[Breiman 2017]BREIMAN, L. *Classification and regression trees*. [S.l.]: Routledge, 2017.

[Brown et al. 2020]BROWN, T. B. et al. *Language Models are Few-Shot Learners*. arXiv, 2020. ArXiv:2005.14165 [cs]. Disponível em: <http://arxiv.org/abs/2005.14165>.

[Buntine 2000]BUNTINE, W. Graphical Models for Discovering Knowledge. nov. 2000.

[Cambria et al. 2023]CAMBRIA, E. et al. Seven Pillars for the Future of Artificial Intelligence. *IEEE Intelligent Systems*, v. 38, n. 6, p. 62–69, nov. 2023. ISSN 1941-1294. Conference Name: IEEE Intelligent Systems. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10352155>.

[Cambria et al. 2017]CAMBRIA, E. et al. Sentiment Analysis Is a Big Suitcase. *IEEE Intelligent Systems*, v. 32, n. 6, p. 74–80, nov. 2017. ISSN 1941-1294. Conference Name: IEEE Intelligent Systems. Disponível em: <https://ieeexplore.ieee.org/document/8267597>.

[Cardoso et al. 2023]CARDOSO, H. L. et al. Argumentation models and their use in corpus annotation: Practice, prospects, and challenges. *Natural Language Engineering*, v. 29, n. 4, p. 1150–1187, 2023.

[Caseli e Nunes 2024]CASELI, H. M.; NUNES, M. G. V. (Ed.). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. 2. ed. BPLN, 2024. ISBN 978-65-00-95750-1. Disponível em: <https://brasileiraspln.com/livro-pln/2a-edicao/>.

[Celebi e Özgür 2016]CELEBI, A.; ÖZGÜR, A. Segmenting hashtags using automatically created training data. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. [S.l.: s.n.], 2016. p. 2981–2985.

[Celebi e Özgür 2018]CELEBI, A.; ÖZGÜR, A. Segmenting hashtags and analyzing their grammatical structure. *Journal of the Association for Information Science and Technology*, Wiley Online Library, v. 69, n. 5, p. 675–686, 2018.

[Charmaz 2009]CHARMAZ, K. *A construção da teoria fundamentada: guia prático para análise qualitativa*. [S.l.]: Bookman Editora, 2009.

[Chen et al. 2009]CHEN, H. et al. Global Models of Document Structure using Latent Permutations. In: OSTENDORF, M. et al. (Ed.). *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado: Association for Computational Linguistics, 2009. p. 371–379. Disponível em: <https://aclanthology.org/N09-1042/>.

[Chen, Tworek e Jun Heewoo 2021]CHEN, M.; TWOREK, J.; JUN HEEWOO, e. a. Evaluating large language models trained on code. *arXiv preprint*, 2021.

[Chen et al. 2024]CHEN, Y. et al. On the design and analysis of llm-based algorithms. *arXiv preprint*, 2024.

[Cho et al. 2014]CHO, K. et al. *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. 2014.

[Choi 2000]CHOI, F. Y. Y. Advances in domain independent linear text segmentation. In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*. [s.n.], 2000. Disponível em: <https://aclanthology.org/A00-2004/>.

[Chomsky 1965]CHOMSKY, N. *Aspects of the Theory of Syntax*. 50. ed. The MIT Press, 1965. ISBN 978-0-262-52740-8. Disponível em: <http://www.jstor.org/stable/j.ctt17kk81z>.

[Chomsky 2002]CHOMSKY, N. *Syntactic Structures*. [S.l.]: Mouton de Gruyter, 2002. (A Mouton classic). ISBN 978-3-11-017279-9.

[Chu et al. 2024]CHU, Z. et al. Navigate through enigmatic labyrinth: A survey of chain of thought reasoning. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

[Clark et al. 2019]CLARK, K. et al. What does bert look at? an analysis of bert's attention. *Proceedings of ACL*, p. 3504–3519, 2019. Disponível em: <https://aclanthology.org/P19-1357/>.

[Courtois 1977]COURTOIS, P. J. *Decomposability: Queueing and Computer System Applications*. [S.l.]: Association for Computing Machinery Inc., 1977. v. 121.

[Croft 2001]CROFT, W. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. 1. ed. Oxford University PressOxford, 2001. ISBN 978-0-19-829955-4 978-0-19-170809-1. Disponível em: <https://academic.oup.com/book/32815>.

[Dearden e Finlay 2006]DEARDEN, A.; FINLAY, J. Pattern languages in hci: A critical review. *Human–computer interaction*, Taylor & Francis, v. 21, n. 1, p. 49–102, 2006.

[Devlin et al. 2018]DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Devlin et al. 2019]DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv, 2019. ArXiv:1810.04805 [cs]. Disponível em: <http://arxiv.org/abs/1810.04805>.

[Dietterich e Bakiri 1994]DIETTERICH, T. G.; BAKIRI, G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, v. 2, p. 263–286, 1994. ISSN 1076-9757. Disponível em: <https://www.jair.org/index.php/jair/article/view/10127>.

[Domingos 1996]DOMINGOS, P. Using partitioning to speed up specific-to-general rule induction. In: AAAI PRESS. *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models*. [S.l.], 1996. p. 29–34.

[Doumen, Beuls e Eecke 2024]DOUMEN, J.; BEULS, K.; EECKE, P. V. Modelling constructivist language acquisition through syntactico-semantic pattern finding. *Royal Society Open Science*, v. 11, n. 7, p. 231998, jul. 2024. Publisher: Royal Society. Disponível em: <https://royalsocietypublishing.org/doi/10.1098/rsos.231998>.

[Doval e Gómez-Rodríguez 2019]DOVAL, Y.; GÓMEZ-RODRÍGUEZ, C. Comparing neural-and n-gram-based language models for word segmentation. *Journal of the Association for Information Science and Technology*, Wiley Online Library, v. 70, n. 2, p. 187–197, 2019.

[Dresch Daniel Pacheco Lacerda 2015]DRESCH DANIEL PACHECO LACERDA, J. A. V. A. J. a. A. *Design Science Research: A Method for Science and Technology Advancement*. 1. ed. [S.l.]: Springer International Publishing, 2015. ISBN 978-3-319-07373-6 978-3-319-07374-3.

[Dziri et al. 2023]DZIRI, N. et al. *Faith and Fate: Limits of Transformers on Compositionality*. arXiv, 2023. ArXiv:2305.18654. Disponível em: <http://arxiv.org/abs/2305.18654>.

[Egbert, Biber e Davies 2015]EGBERT, J.; BIBER, D.; DAVIES, M. Developing a bottom-up, user-based method of web register classification. *Journal of the Association for Information Science and Technology*, v. 66, n. 9, p. 1817–1831, 2015. ISSN 2330-1643. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23308. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23308>.

[Ellis, Pu e Solar-Lezama Armando 2021]ELLIS, K.; PU, Y.; SOLAR-LEZAMA ARMANDO, e. a. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 2021.

[Fielding e Taylor 2002]FIELDING, R. T.; TAYLOR, R. N. Principled design of the modern Web architecture. *ACM Trans. Internet Technol.*, v. 2, n. 2, p. 115–150, maio 2002. ISSN 1533-5399. Disponível em: <https://doi.org/10.1145/514183.514185>.

[Firoozeh et al. 2020]FIROOZEH, N. et al. Keyword extraction: Issues and methods. *Natural Language Engineering*, v. 26, n. 3, p. 259–291, May 2020. ISSN 1351-3249, 1469-8110.

[Fleiss 1971]FLEISS, J. Measuring nominal scale agreement among many raters. Psychological Bulletin, 1971. Disponível em: <https://psycnet.apa.org/record/1972-05083-001>.

[Fowler 2013]FOWLER, M. *Patterns of enterprise application architecture*. Nineteenth printing. Boston San Francisco New York Toronto Montreal London Munich Paris Madrid Capetown: Addison-Wesley, 2013. (The Addison-Wesley Signature Series). ISBN 978-0-321-12742-6.

[Gamma et al. 1995]GAMMA, E. et al. *Design patterns: elements of reusable object-oriented software*. USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN 0201633612.

[Gams 1989]GAMS, M. New measurements highlight the importance of redundant knowledge. In: *Proceedings of the fourth european working session on learning*. [S.l.]: Pitman Montpellier, France, 1989. p. 71–79.

[Ganda e Buch 2018]GANDA, D.; BUCH, R. A Survey on Multi Label Classification. *Multi Label Classification*, 2018.

[Gao et al. 2022]GAO, Y. et al. Do Discourse Indicators Reflect the Main Arguments in Scientific Papers? In: LAPESA, G. et al. (Ed.). *Proceedings of the 9th*

Workshop on Argument Mining. Online and in Gyeongju, Republic of Korea: International Conference on Computational Linguistics, 2022. p. 34–50. Disponível em: <https://aclanthology.org/2022.argmining-1.3/>.

[Gao et al. 2024]GAO, Y. et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv, 2024. ArXiv:2312.10997 [cs]. Disponível em: <http://arxiv.org/abs/2312.10997>.

[Ghallab, Nau e Traverso 2004]GHALLAB, M.; NAU, D. S.; TRAVERSO, P. *Automated planning: theory and practice*. Amsterdam Boston: Elsevier/Morgan Kaufmann, 2004. ISBN 978-1-55860-856-6.

[Glaser e Strauss 2017]GLASER, B.; STRAUSS, A. *Discovery of grounded theory: Strategies for qualitative research*. [S.l.]: Routledge, 2017.

[Glavaš, Nanni e Ponzetto 2016]GLAVAš, G.; NANNI, F.; PONZETTO, S. P. Unsupervised Text Segmentation Using Semantic Relatedness Graphs. In: GARDENT, C.; BERNARDI, R.; TITOV, I. (Ed.). *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. Berlin, Germany: Association for Computational Linguistics, 2016. p. 125–130. Disponível em: <https://aclanthology.org/S16-2016/>.

[Glavaš e Somasundaran 2020]GLAVAš, G.; SOMASUNDARAN, S. Two-Level Transformer and Auxiliary Coherence Modeling for Improved Text Segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 34, n. 05, p. 7797–7804, abr. 2020. ISSN 2374-3468. Number: 05. Disponível em: <https://ojs.aaai.org/index.php/AAAI/article/view/6284>.

[Glorot e Bengio 2010]GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. [S.l.], 2010. p. 249–256.

[Graham, Knuth e Patashnik 1994]GRAHAM, R. L.; KNUTH, D. E.; PATASHNIK, O. *Concrete Mathematics: A Foundation for Computer Science*. 2nd. ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1994. ISBN 0201558025.

[Hadi et al. 2023]HADI, M. U. et al. Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects. dez. 2023.

[Han et al. 2023]HAN, M. et al. A survey of multi-label classification based on supervised and semi-supervised learning. *International Journal of Machine Learning and Cybernetics*, v. 14, n. 3, p. 697–724, mar. 2023. ISSN 1868-8071, 1868-808X. Disponível em: <https://link.springer.com/10.1007/s13042-022-01658-9>.

[Han Micheline Kamber 2006]HAN MICHELINE KAMBER, J. P. J. *Data Mining: Concepts and Techniques*. 2. ed. [S.l.]: Morgan Kaufmann, 2006. (The Morgan Kaufmann Series in Data Management Systems). ISBN 978-1-55860-901-3 1-55860-901-6.

[Hasan e Ng 2014]HASAN, K. S.; NG, V. Automatic Keyphrase Extraction: A Survey of the State of the Art. In: TOUTANOVA, K.; WU, H. (Ed.). *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, 2014. p. 1262–1273. Disponível em: <https://aclanthology.org/P14-1119>.

[Hewitt e Manning 2019]HEWITT, J.; MANNING, C. D. A structural probe for finding syntax in word representations. *Proceedings of NAACL*, p. 4129–4138, 2019. Disponível em: <https://aclanthology.org/N19-1419/>.

[Hochreiter 1991]HOCHREITER, S. *Untersuchungen zu dynamischen neuronalen Netzen*. Tese (Doutorado) — TU München, 1991.

[Hochreiter e Schmidhuber 1997]HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

[Hoda 2022]HODA, R. Socio-Technical Grounded Theory for Software Engineering. *IEEE Transactions on Software Engineering*, v. 48, n. 10, p. 3808–3832, out. 2022. ISSN 1939-3520. Conference Name: IEEE Transactions on Software Engineering.

[Hu e Lu 2024]HU, Y.; LU, Y. *RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing*. arXiv, 2024. ArXiv:2404.19543 [cs]. Disponível em: <http://arxiv.org/abs/2404.19543>.

[Huang et al. 2024]HUANG, X. et al. Understanding the planning of llm agents: A survey. *arXiv preprint*, 2024.

[Hupkes et al. 2023]HUPKES, D. et al. A taxonomy and review of generalization research in NLP. *Nature Machine Intelligence*, v. 5, n. 10, p. 1161–1174, out. 2023. ISSN 2522-5839. Publisher: Nature Publishing Group. Disponível em: <https://www.nature.com/articles/s42256-023-00729-y>.

[Ibbotson 2013]IBBOTSON, P. The Scope of Usage-Based Theory. *Frontiers in Psychology*, v. 4, maio 2013. ISSN 1664-1078. Publisher: Frontiers. Disponível em: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2013.00255/full>.

[Iivari 1992]IIVARI, J. Relationships, aggregations and complex objects. *Information Modelling and Knowledge Bases*, v. 3, p. 141–159, 1992.

[Inuzuka et al. 2020]INUZUKA, M. et al. Doclass: open-source software to support document labeling and classification. In: *Anais do Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*. SBC, 2020. p. 105–112. Disponível em: <https://sol.sbc.org.br/index.php/kdmile/article/view/11965>.

[Inuzuka, Rocha e Nascimento 2020]INUZUKA, M. A.; ROCHA, A. S.; NASCIMENTO, H. A. D. Segmentation of Words Written in the Latin Alphabet: A Systematic Review. In: QUARESMA, P. et al. (Ed.). *Computational Processing of the Portuguese Language*. Cham: Springer International Publishing, 2020. v. 12037, p. 291–302. ISBN 978-3-030-41504-4 978-3-030-41505-1. Disponível em: <http://link.springer.com/10.1007/978-3-030-41505-1$_2$8>.

[Ishita et al. 2010]ISHITA, E. et al. Investigating multi-label classification for human values. *Proceedings of the American Society for Information Science and Technology*, v. 47, n. 1, p. 1–4, 2010. ISSN 1550-8390.

[Jawahar, Sagot e Seddah 2019]JAWAHAR, G.; SAGOT, B.; SEDDAH, D. What does bert learn about the structure of language? *Proceedings of ACL*, p. 3651–3657, 2019. Disponível em: <https://aclanthology.org/P19-1356/>.

[Jiang e Matsubara 2014]JIANG, H.; MATSUBARA, S. Efficient task decomposition in crowdsourcing. In: DAM, H. K. et al. (Ed.). *PRIMA 2014: Principles and Practice of Multi-Agent Systems*. Cham: Springer International Publishing, 2014. p. 65–73. ISBN 978-3-319-13191-7.

[Kay e Fillmore 1999]KAY, P.; FILLMORE, C. J. Grammatical constructions and linguistic generalizations: The what's x doing y? construction. *Language*, Linguistic Society of America, v. 75, n. 1, p. 1–33, 1999. ISSN 00978507, 15350665. Disponível em: <http://www.jstor.org/stable/417472>.

[Kessler, Nunberg e Schutze 1997]KESSLER, B.; NUNBERG, G.; SCHUTZE, H. Automatic Detection of Text Genre. In: *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*. Madrid, Spain: Association for Computational Linguistics, 1997. p. 32–38. Disponível em: <https://aclanthology.org/P97-1005>.

[Kitchenham 2004]KITCHENHAM, B. *Procedures for Performing Systematic Reviews*. [S.l.], 2004.

[Klie, Castilho e Gurevych 2024]KLIE, J.-C.; CASTILHO, R. E. de; GUREVYCH, I. Analyzing Dataset Annotation Quality Management in the Wild. *Computational Linguistics*, p. 1–50, jul. 2024. ISSN 0891-2017. Disponível em: <https://doi.org/10.1162/coli$_{a0}$0516>.

[Kodali et al. 2022]KODALI, P. et al. HashSet - A Dataset For Hashtag Segmentation. In: CALZOLARI, N. et al. (Ed.). *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2022. p. 7215–7219. Disponível em: <https://aclanthology.org/2022.lrec-1.782>.

[Kojima et al. 2023]KOJIMA, T. et al. *Large Language Models are Zero-Shot Reasoners*. [S.l.]: arXiv, 2023. ArXiv:2205.11916 [cs].

[Koshorek et al. 2018]KOSHOREK, O. et al. Text Segmentation as a Supervised Learning Task. In: WALKER, M.; JI, H.; STENT, A. (Ed.). *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 469–473. Disponível em: <https://aclanthology.org/N18-2075/>.

[Krippendorff 2004]KRIPPENDORFF, K. Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research*, v. 30, n. 3, p. 411–433, jul. 2004. ISSN 0360-3989, 1468-2958. Disponível em: <http://doi.wiley.com/10.1093/hcr/30.3.411>.

[Krippendorff 2011]KRIPPENDORFF, K. Computing krippendorff's alpha-reliability. In: . [s.n.], 2011. Disponível em: <https://api.semanticscholar.org/CorpusID:59901023>.

[Kryscinski et al. 2019]KRYSCINSKI, W. et al. Neural Text Summarization: A Critical Evaluation. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 540–551. Disponível em: <https://aclanthology.org/D19-1051>.

[Kumar 2024]KUMAR, P. Large language models (LLMs): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, v. 57, n. 10, p. 260, ago. 2024. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-024-10888-y>.

[Kusiak 2000]KUSIAK, A. Decomposition in data mining: an industrial case study. *IEEE Transactions on Electronics Packaging Manufacturing*, v. 23, n. 4, p. 345–353, 2000.

[Kuzman e Ljubešić 2023]KUZMAN, T.; LJUBEšIć, N. Automatic genre identification: a survey. *Language Resources and Evaluation*, nov. 2023. ISSN 1574-0218. Disponível em: <https://doi.org/10.1007/s10579-023-09695-8>.

[Kuzman, Mozetič e Ljubešić 2023]KUZMAN, T.; MOZETIč, I.; LJUBEšIć, N. Automatic Genre Identification for Robust Enrichment of Massive Text Collections: Investiga-

tion of Classification Methods in the Era of Large Language Models. *Machine Learning and Knowledge Extraction*, v. 5, n. 3, p. 1149–1175, set. 2023. ISSN 2504-4990. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute. Disponível em: <https://www.mdpi.com/2504-4990/5/3/59>.

[Kuzman, Rupnik e Ljubešić 2022]KUZMAN, T.; RUPNIK, P.; LJUBEšIć, N. The GINCO Training Dataset for Web Genre Identification of Documents Out in the Wild. In: CALZOLARI, N. et al. (Ed.). *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2022. p. 1584–1594. Disponível em: <https://aclanthology.org/2022.lrec-1.170/>.

[Lacerda 2017]LACERDA, C. GRAMÁTICA DE CONSTRUÇÕES: PRINCÍPIOS BÁSICOS E CONTRIBUIÇÕES. *Funcionalismo linguístico: diálogos e vertentes. 1. ed. Niterói: Eduff*, 2017.

[Landis e Koch 1977]LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. *Biometrics*, v. 33, n. 1, p. 159–174, mar. 1977. ISSN 0006-341X.

[Lauscher, Glavaš e Ponzetto 2018]LAUSCHER, A.; GLAVAš, G.; PONZETTO, S. P. An Argument-Annotated Corpus of Scientific Publications. In: *Proceedings of the 5th Workshop on Argument Mining*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 40–46. Disponível em: <https://aclanthology.org/W18-5206>.

[LeCun et al. 1998]LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998.

[Lee e Myaeng 2002]LEE, Y.-B.; MYAENG, S. H. Text genre classification with genre-revealing and subject-revealing features. In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: Association for Computing Machinery, 2002. (SIGIR '02), p. 145–150. ISBN 978-1-58113-561-9. Disponível em: <https://doi.org/10.1145/564376.564403>.

[Levy et al. 2014]LEVY, R. et al. Context Dependent Claim Detection. In: TSUJII, J.; HAJIC, J. (Ed.). *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, 2014. p. 1489–1500. Disponível em: <https://aclanthology.org/C14-1141/>.

[Lewis et al. 2021]LEWIS, P. et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv, 2021. ArXiv:2005.11401 [cs]. Disponível em: <http://arxiv.org/abs/2005.11401>.

[Li, Wang e Wu Jian 2022]LI, Y.; WANG, Y.; WU JIAN, e. a. Competition-level code generation with alphacode. *DeepMind Blog*, 2022.

[Loper e Bird 2002]LOPER, E.; BIRD, S. *NLTK: The Natural Language Toolkit*. [S.l.]: arXiv, 2002.

[Ma, Ganchev e Weiss 2018]MA, J.; GANCHEV, K.; WEISS, D. State-of-the-art Chinese word segmentation with Bi-LSTMs. In: RILOFF, E. et al. (Ed.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 4902–4908. Disponível em: <https://aclanthology.org/D18-1529/>.

[MacQueen 1967]MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. University of California Press, 1967. v. 5.1, p. 281–298. Disponível em: <https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992>.

[Maddela, Xu e Preoţiuc-Pietro 2019]MADDELA, M.; XU, W.; PREOŢIUC-PIETRO, D. Multi-task pairwise neural ranking for hashtag segmentation. *arXiv preprint arXiv:1906.00790*, 2019.

[Manning e Schütze 1999]MANNING, C. D.; SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999. Disponível em: <http://nlp.stanford.edu/fsnlp/>.

[Marcus, Marcinkiewicz e Santorini 1993]MARCUS, M. P.; MARCINKIEWICZ, M. A.; SANTORINI, B. Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.*, v. 19, n. 2, p. 313–330, jun. 1993. ISSN 0891-2017.

[Marneffe et al. 2021]MARNEFFE, M.-C. de et al. Universal Dependencies. *Computational Linguistics*, v. 47, n. 2, p. 255–308, jul. 2021. ISSN 0891-2017. Disponível em: <https://doi.org/10.1162/coli$_{a0}$0402>.

[Mehler, Sharoff e Santini 2011]MEHLER, A.; SHAROFF, S.; SANTINI, M. (Ed.). *Genres on the Web*. Dordrecht: Springer Netherlands, 2011. v. 42. (Text, Speech and Language Technology, v. 42). ISBN 978-90-481-9177-2 978-90-481-9178-9. Disponível em: <http://link.springer.com/10.1007/978-90-481-9178-9>.

[Meilă 2007]MEILă, M. Comparing clusterings-an information based distance. *Journal of Multivariate Analysis*, 2007.

[Michie 1995]MICHIE, D. Problem decomposition and the learning of skills. In: *Proceedings of the 8th European Conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 1995. (ECML'95), p. 17–31. ISBN 3-540-59286-5. Event-place: Heraclion, Crete, Greece. Disponível em: <https://doi.org/10.1007/3-540-59286-5$_4$6>.

[Mikolov et al. 2013]MIKOLOV, T. et al. Efficient estimation of word representations in vector space. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2013.

[Miller 1956]MILLER, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, v. 63, n. 2, p. 81, 1956. Publisher: American Psychological Association.

[Mitchell 1997]MITCHELL, T. M. *Machine Learning*. 1. ed. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill Series in Computer Science). ISBN 978-0-07-042807-2 0-07-042807-7.

[Moraes et al. 2024]MORAES, L. d. C. et al. *Análise de ambiguidade linguística em modelos de linguagem de grande escala (LLMs)*. arXiv, 2024. ArXiv:2404.16653 [cs]. Disponível em: <http://arxiv.org/abs/2404.16653>.

[Mosqueira-Rey et al. 2023]MOSQUEIRA-REY, E. et al. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, v. 56, n. 4, p. 3005–3054, abr. 2023. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-022-10246-w>.

[Nair e Hinton 2010]NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814.

[Newell 1994]NEWELL, A. *Unified Theories of Cognition (The William James Lectures)*. [S.l.]: Harvard University Press, 1994. ISBN 978-0-674-92101-6 978-0-674-92099-6 0-674-92101-1 0-674-92099-6.

[Nogueira, Jiang e Lin 2021]NOGUEIRA, R.; JIANG, Z.; LIN, J. *Investigating the Limitations of Transformers with Simple Arithmetic Tasks*. arXiv, 2021. ArXiv:2102.13019 [cs]. Disponível em: <http://arxiv.org/abs/2102.13019>.

[Numiri 2023]Numiri. *Calculation flow through a single attention head*. 2023. Licenciado sob CC BY-SA 4.0 Internacional. https://creativecommons.org/licenses/by-sa/4.0/deed.en. Disponível em: <https://commons.wikimedia.org/wiki/File:Attention-qkv.png>.

[Nye et al. 2021]NYE, M. et al. *Show Your Work: Scratchpads for Intermediate Computation with Language Models*. arXiv, 2021. ArXiv:2112.00114 [cs]. Disponível em: <http://arxiv.org/abs/2112.00114>.

[Ortiz 2007]ORTIZ, C. M. A. *Does philosophy improve critical thinking skills?* Tese (Doutorado) — Melbourne University, 2007. Disponível em: <https://web.archive.org/web/20120708000509/http://images.austhink.com/pdf/Claudia-Alvarez-thesis.pdf>.

[Ouyang et al. 2022]OUYANG, L. et al. *Training language models to follow instructions with human feedback*. arXiv, 2022. ArXiv:2203.02155 [cs]. Disponível em: <http://arxiv.org/abs/2203.02155>.

[Page et al. 1999]PAGE, L. et al. *The PageRank citation ranking: Bringing order to the web.* [S.l.], 1999.

[Pal, Selvakumar e Sankarasubbu 2020]PAL, A.; SELVAKUMAR, M.; SANKARASUBBU, M. Multi-Label Text Classification using Attention-based Graph Neural Network. In: *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*. [s.n.], 2020. p. 494–505. ArXiv:2003.11644 [cs]. Disponível em: <http://arxiv.org/abs/2003.11644>.

[Palomaki, Rhinehart e Tseng 2018]PALOMAKI, J.; RHINEHART, O.; TSENG, M. A case for a range of acceptable annotations. In: *SAD/CrowdBias@ HCOMP*. [S.l.: s.n.], 2018. p. 19–31.

[Pedregosa et al. 2011]PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[Peffers et al. 2007]PEFFERS, K. et al. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, v. 24, n. 3, p. 45–77, dez. 2007. ISSN 0742-1222, 1557-928X. Disponível em: <https://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302>.

[Pevzner e Hearst 2002]PEVZNER, L.; HEARST, M. A. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, MIT Press, v. 28, n. 1, p. 19–36, 2002. Disponível em: <https://aclanthology.org/J02-1002>.

[Pimenov e Salomatina 2024]PIMENOV, I. S.; SALOMATINA, N. V. Productivity-Based Analysis of Argumentation Patterns across Texts of Different Genres. In: *2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM)*. [s.n.], 2024. p. 2270–2275. ISSN: 2325-419X. Disponível em: <https://ieeexplore.ieee.org/abstract/document/10615197>.

[Pustejovsky e Stubbs 2012]PUSTEJOVSKY, J.; STUBBS, A. *Natural language annotation for machine learning: a guide to corpus-building for applications*. 1. ed. ed. Beijing: O'Reilly, 2012. 00156 OCLC: 930811574. ISBN 978-1-4493-0666-3.

[Qian et al. 2022]QIAN, J. et al. *Limitations of Language Models in Arithmetic and Symbolic Induction*. arXiv, 2022. ArXiv:2208.05051 [cs]. Disponível em: <http://arxiv.org/abs/2208.05051>.

[Quinlan 1986]QUINLAN, J. R. Induction of Decision Trees. *Mach. Learn.*, v. 1, n. 1, p. 81–106, mar. 1986. ISSN 0885-6125. Place: USA Publisher: Kluwer Academic Publishers. Disponível em: <https://doi.org/10.1023/A:1022643204877>.

[Quinlan 2014]QUINLAN, J. R. *C4. 5: programs for machine learning*. [S.l.]: Elsevier, 2014.

[Radford et al. 2018]RADFORD, A. et al. Improving Language Understanding by Generative Pre-Training. 2018. Disponível em: <https://cdn.openai.com/research-covers/language-unsupervised/language$_u nderstanding_p aper.pdf$>.

[Radford et al. 2019]RADFORD, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog*, v. 1, n. 8, p. 9, 2019.

[Rajkovic 1989]RAJKOVIC, V. *Nursery*. 1989. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5P88W.

[Ramamurti e Ghosh 1999]RAMAMURTI, V.; GHOSH, J. Structurally adaptive modular networks for nonstationary environments. *IEEE Transactions on Neural Networks*, v. 10, n. 1, p. 152–160, 1999.

[Rana e Cheah 2016]RANA, T. A.; CHEAH, Y.-N. Aspect extraction in sentiment analysis: comparative analysis and survey. *Artificial Intelligence Review*, v. 46, n. 4, p. 459–483, dez. 2016. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-016-9472-z>.

[Reidsma e Akker 2008]REIDSMA, D.; AKKER, R. op den. Exploiting 'Subjective' Annotations. In: ARTSTEIN, R. et al. (Ed.). *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*. Manchester, UK: Coling 2008 Organizing Committee, 2008. p. 8–16. Disponível em: <https://aclanthology.org/W08-1203>.

[Reimers e Gurevych 2019]REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. Disponível em: <https://arxiv.org/abs/1908.10084>.

[Resplande et al. 2020]RESPLANDE, J. et al. Construção de Datasets para Segmentação Automática de Hashtags. In: *Anais do Encontro Anual de Computação de 2020*. [S.l.: s.n.], 2020.

[Rinott et al. 2015]RINOTT, R. et al. Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection. In: MàRQUEZ, L.; CALLISON-BURCH, C.; SU, J. (Ed.). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, 2015. p. 440–450. Disponível em: <https://aclanthology.org/D15-1050/>.

[Rodrigues et al. 2021]RODRIGUES, R. C. et al. Zero-shot hashtag segmentation for multilingual sentiment analysis. *arXiv:2112.03213 [cs]*, dez. 2021. Disponível em: <http://arxiv.org/abs/2112.03213>.

[Rodrigues et al. 2020]RODRIGUES, R. C. et al. Domain Adaptation of Transformers for English Word Segmentation. In: *Intelligent Systems*. Springer, Cham, 2020. p. 483–496. Disponível em: <https://link.springer.com/chapter/10.1007

[Rokach 2006]ROKACH, L. Decomposition methodology for classification tasks: a meta decomposer framework. *Pattern Analysis and Applications*, v. 9, n. 2, p. 257–271, out. 2006. ISSN 1433-755X. Disponível em: <https://doi.org/10.1007/s10044-006-0041-y>.

[Rokach, Maimon e Arad 2005]ROKACH, L.; MAIMON, O.; ARAD, O. Improving supervised learning by sample decomposition. *International Journal of Computational Intelligence and Applications*, v. 05, n. 01, p. 37–53, mar. 2005. ISSN 1469-0268. Publisher: World Scientific Publishing Co. Disponível em: <https://www.worldscientific.com/doi/abs/10.1142/S146902680500143X>.

[Rosenblatt 1958]ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.

[Roush e Balaji 2020]ROUSH, A.; BALAJI, A. DebateSum: A large-scale argument mining and summarization dataset. In: *Proceedings of the 7th Workshop on Argument Mining*. Online: Association for Computational Linguistics, 2020. p. 1–7. Disponível em: <https://aclanthology.org/2020.argmining-1.1>.

[Rumbaugh et al. 1991]RUMBAUGH, J. et al. *Object-oriented modeling and design*. USA: Prentice-Hall, Inc., 1991. ISBN 0-13-629841-9.

[Rumelhart, Hinton e Williams 1986]RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, Springer, v. 323, n. 6088, p. 533–536, 1986.

[Saenger 1997]SAENGER, P. *Space between words: The origins of silent reading*. [S.l.]: Stanford University Press, 1997.

[Saito e Rehmsmeier 2015]SAITO, T.; REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, Public Library of Science, v. 10, n. 3, p. e0118432, 2015.

[Salazar et al. 2020]SALAZAR, J. et al. Masked language model scoring. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020. Disponível em: <http://dx.doi.org/10.18653/v1/2020.acl-main.240>.

[Seibel 1963]SEIBEL, R. Discrimination reaction time for a 1,023-alternative task. *Journal of Experimental Psychology*, v. 66, n. 3, p. 215–226, 1963. ISSN 0022-1015. Disponível em: <https://doi.apa.org/doi/10.1037/h0048914>.

[Shannon 1951]SHANNON, C. E. Prediction and entropy of printed English. *The Bell System Technical Journal*, v. 30, n. 1, p. 50–64, 1951.

[Sharoff 2018]SHAROFF, S. Functional Text Dimensions for the annotation of web corpora. *Corpora*, v. 13, n. 1, p. 65–95, abr. 2018. ISSN 1749-5032, 1755-1676. Disponível em: <https://www.euppublishing.com/doi/10.3366/cor.2018.0136>.

[Sharp e Eckstein 2003]SHARP, M. L. M. H.; ECKSTEIN, J. Evolving pedagogical patterns: The work of the pedagogical patterns project. *Computer Science Education*, Routledge, v. 13, n. 4, p. 315–330, 2003. Disponível em: <https://doi.org/10.1076/csed.13.4.315.17493>.

[Shaw e Garlan 1996]SHAW, M.; GARLAN, D. *Software architecture: perspectives on an emerging discipline*. USA: Prentice-Hall, Inc., 1996. ISBN 0-13-182957-2.

[Simon 1962]SIMON, H. A. The Architecture of Complexity. *Proceedings of the American Philosophical Society*, v. 106, n. 6, p. 467–482, 1962. ISSN 0003-049X. Publisher: American Philosophical Society. Disponível em: <https://www.jstor.org/stable/985254>.

[Simon 2019]SIMON, H. A. *The Sciences of the Artificial*. The MIT Press, 2019. ISBN 978-0-262-35474-5. Disponível em: <https://direct.mit.edu/books/monograph/4551/The-Sciences-of-the-Artificial>.

[Simon e Newell 1971]SIMON, H. A.; NEWELL, A. Human problem solving: The state of the theory in 1970. *American Psychologist*, v. 26, n. 2, p. 145–159, fev. 1971. ISSN 1935-990X, 0003-066X. Disponível em: <https://doi.apa.org/doi/10.1037/h0030806>.

[Sinha, Premsri e Kordjamshidi 2024]SINHA, S.; PREMSRI, T.; KORDJAMSHIDI, P. *A Survey on Compositional Learning of AI Models: Theoretical and Experimental Practices*. arXiv, 2024. ArXiv:2406.08787 [cs]. Disponível em: <http://arxiv.org/abs/2406.08787>.

[Smith e Smith 1977]SMITH, J. M.; SMITH, D. C. P. Database abstractions: aggregation. *Commun. ACM*, v. 20, n. 6, p. 405–413, jun. 1977. ISSN 0001-0782. Disponível em: <https://dl.acm.org/doi/10.1145/359605.359620>.

[Smith 1999]SMITH, N. *Chomsky: Ideas and Ideals*. Cambridge University Press, 1999. ISBN 978-0-521-47570-9. Disponível em: <https://books.google.com.br/books?id=fVj41BDjDeYC>.

[Sommerfield 1997]SOMMERFIELD, R. K. B. B. D. Improving Simple Bayes. In: *Proceedings of the ninth European conference on machine learning, Prague, http://robotics. stanford. edu/ ronnyk/impSBC. ps. Z*. [S.l.: s.n.], 1997.

[Stab e Gurevych 2014]STAB, C.; GUREVYCH, I. Annotating argument components and relations in persuasive essays. p. 1–10, 2014.

[Stab e Gurevych 2014]STAB, C.; GUREVYCH, I. *Argument Annotated Essays*. Technical University of Darmstadt, 2014. Disponível em: <https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2421>.

[Stab et al. 2018]STAB, C. et al. *UKP Sentential Argument Mining Corpus*. Technical University of Darmstadt, 2018. Disponível em: <https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2345>.

[Stab et al. 2018]STAB, C. et al. Cross-topic Argument Mining from Heterogeneous Sources. In: RILOFF, E. et al. (Ed.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 3664–3674. Disponível em: <https://aclanthology.org/D18-1402/>.

[Stake 1995]STAKE, R. E. *The Art of Case Study Research*. Thousand Oaks, CA: SAGE Publications, 1995. ISBN 978-0803957671.

[Stol, Ralph e Fitzgerald 2016]STOL, K.-J.; RALPH, P.; FITZGERALD, B. Grounded Theory in Software Engineering Research: A Critical Review and Guidelines. In: *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. [s.n.], 2016. p. 120–131. ISSN: 1558-1225. Disponível em: <https://ieeexplore.ieee.org/document/7886897>.

[Sutskever, Vinyals e Le 2014]SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2014. p. 3104–3112.

[Tomasello 2003]TOMASELLO, M. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, 2003. ISBN 978-0-674-01030-7. Disponível em: <https://www.jstor.org/stable/j.ctv26070v8>.

[Vaswani et al. 2017]VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.

[Verma e Om 2019]VERMA, P.; OM, H. MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization. *Expert Systems with Applications*, v. 120, p. 43–56, abr. 2019. ISSN 09574174. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0957417418307425>.

[Wachsmuth et al. 2017]WACHSMUTH, H. et al. Computational Argumentation Quality Assessment in Natural Language. In: LAPATA, M.; BLUNSOM, P.; KOLLER, A. (Ed.). *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, 2017. p. 176–187. Disponível em: <https://aclanthology.org/E17-1017/>.

[Wang et al. 2017]WANG, L. et al. Learning to rank semantic coherence for topic segmentation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017. p. 1340–1344. Disponível em: <https://aclanthology.org/D17-1139/>.

[Wang et al. 2021]WANG, R. et al. A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, v. 58, n. 2, p. 102441, mar. 2021. ISSN 03064573. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0306457320309341>.

[Warstadt et al. 2023]WARSTADT, A. et al. *BLiMP: The Benchmark of Linguistic Minimal Pairs for English*. arXiv, 2023. ArXiv:1912.00582 [cs]. Disponível em: <http://arxiv.org/abs/1912.00582>.

[Washizaki et al. 2022]WASHIZAKI, H. et al. Software-engineering design patterns for machine learning applications. *Computer*, IEEE Computer Society, v. 55, n. 3, p. 30–39, 2022. Disponível em: <https://doi.org/10.1109/MC.2021.3137227>.

[Weaver 1948]WEAVER, W. Science and Complexity. *American Scientist*, v. 36, n. 4, p. 536–544, 1948. ISSN 0003-0996. Publisher: Sigma Xi, The Scientific Research Society. Disponível em: <https://www.jstor.org/stable/27826254>.

[Wei et al. 2023]WEI, J. et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv, 2023. ArXiv:2201.11903 [cs]. Disponível em: <http://arxiv.org/abs/2201.11903>.

[Wikipedia 2017]WIKIPEDIA. *Scriptio continua*. 2017. Page Version ID: 49004554. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Scriptio$_continuaoldid$ = 49004554>.

[Wu, Nguyen e Luu 2024]WU, X.; NGUYEN, T.; LUU, A. T. A Survey on Neural Topic Models: Methods, Applications, and Challenges. *Artificial Intelligence Review*, v. 57, n. 2, p. 18, jan. 2024. ISSN 1573-7462. ArXiv:2401.15351 [cs]. Disponível em: <http://arxiv.org/abs/2401.15351>.

[Wu et al. 2022]WU, X. et al. A Survey of Human-in-the-loop for Machine Learning. *Future Generation Computer Systems*, v. 135, p. 364–381, out. 2022. ISSN 0167739X. ArXiv:2108.00941 [cs]. Disponível em: <http://arxiv.org/abs/2108.00941>.

[Xia et al. 2024]XIA, Y. et al. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms. *arXiv preprint arXiv:2404.15676*, 2024.

[Xiao et al. 2019]XIAO, L. et al. Label-Specific Document Representation for Multi-Label Text Classification. In: INUI, K. et al. (Ed.). *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 466–475. Disponível em: <https://aclanthology.org/D19-1044/>.

[Yin 2015]YIN, R. K. *Estudo de Caso-: Planejamento e métodos*. [S.l.]: Bookman editora, 2015.

[Yu et al. 2023]YU, H. et al. Improving long document topic segmentation models with enhanced coherence modeling. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023. p. 5592–5605. Disponível em: <https://aclanthology.org/2023.emnlp-main.341/>.

[Yu et al. 2024]YU, H. et al. *Evaluation of Retrieval-Augmented Generation: A Survey*. arXiv, 2024. ArXiv:2405.07437 [cs]. Disponível em: <http://arxiv.org/abs/2405.07437>.

[Yu et al. 2023]YU, Z. et al. Towards better chain-of-thought prompting strategies: A survey. *arXiv preprint arXiv:2310.04959*, 2023.

[Zhang e Clark 2011]ZHANG, Y.; CLARK, S. Syntactic processing using the generalized perceptron and beam search. *Computational linguistics*, MIT Press, v. 37, n. 1, p. 105–151, 2011.

[Zhao et al. 2024]ZHAO, P. et al. *Retrieval-Augmented Generation for AI-Generated Content: A Survey*. arXiv, 2024. ArXiv:2402.19473 [cs]. Disponível em: <http://arxiv.org/abs/2402.19473>.

[Zupan et al. 1997]ZUPAN, B. et al. Machine learning by function decomposition. In: *ICML*. [S.l.]: Citeseer, 1997. p. 421–429.

[Zupan et al. 1999]ZUPAN, B. et al. Learning by discovering concept hierarchies. *Artificial Intelligence*, v. 109, n. 1, p. 211–242, jun. 1999. ISSN 0004-3702. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370299000089>.

[Åström e McAvoy 1992]ÅSTRöM, K.; MCAVOY, T. J. Intelligent control. *Journal of Process Control*, v. 2, n. 3, p. 115–127, 1992. ISSN 0959-1524. Disponível em: <https://www.sciencedirect.com/science/article/pii/095915249285001D>.

APÊNDICE A

Trajetória de Estudos

A fim de contextualizar a trajetória da pesquisa de doutorado, segue uma descrição resumida dos estudos divididos em três fases: pré-estudos, desenvolvimento de estudos e conclusão de estudos.

A.1 Fase de Pré-Estudos

Nesta fase inicial da pesquisa, foi coletado um corpus com aproximadamente 5.000 documentos da área do Direito, especificamente atas de audiência de custódia, fornecidas pela empresa Ultimatum[1]. A partir desse material, foram conduzidas investigações em duas frentes principais:

- **Aprendizado Ativo (Active Learning)** — realizou-se um estudo exploratório em Direito Penal, com o objetivo de classificar os documentos e entrevistar especialistas jurídicos. Como resultado, foram obtidos dois produtos: (i) um *dataset* com 500 documentos classificados como atas de audiência de custódia ou não; e (ii) uma ferramenta interativa de classificação com suporte a *active learning*, denominada *Doclass* [Inuzuka et al. 2020].
- **Segmentação de Palavras (Word Segmentation)** — observou-se que diversos documentos apresentavam erros de segmentação de palavras, decorrentes de falhas na conversão de arquivos PDF para texto puro. Um exemplo recorrente era a junção de palavras como *'decisãoanteriorjáservecomomandadodeprisão'*, que deveria ser segmentada como *'decisão anterior já serve como mandado de prisão'*. A análise desse problema resultou em quatro publicações científicas [Inuzuka, Rocha e Nascimento 2020, Resplande et al. 2020, Rodrigues et al. 2020, Rodrigues et al. 2021].

Essa fase preliminar proporcionou aprendizados importantes que fundamentaram as etapas seguintes da pesquisa. Dentre os principais insights, destacam-se:

[1] https://ultimatum.com.br

- **Viabilidade da anotação interativa** — a experiência com a ferramenta *Doclass* demonstrou que a anotação interativa, guiada por aprendizado ativo e com treinamento em tempo real, é viável para modelos leves, como o SVM.
- **Alto custo de anotação especializada** — a anotação do *dataset* jurídico evidenciou a complexidade dos textos e o elevado tempo necessário para análise especializada, tornando inviável, do ponto de vista prático e financeiro, a construção de um corpus argumentativo nessa área.
- **Domínio prático de modelos de aprendizado de máquina** — embora a tarefa de segmentação de palavras não esteja diretamente vinculada ao objetivo central da tese, ela proporcionou um domínio técnico relevante sobre treinamento e avaliação de modelos. A abundância de dados disponíveis permitiu o desenvolvimento ágil da ferramenta *Hashformers*[2], atualmente considerada estado da arte na tarefa, conforme apontado por [Kodali et al. 2022].

A.2 Fase de Desenvolvimento de Estudos

Esta fase corresponde ao período de qualificação da tese, momento em que foi consolidado o objetivo de desenvolver artefatos voltados ao mapeamento automático ou semi-automático de argumentos. Para aprofundar os conhecimentos sobre anotação de corpus, foi realizado um estágio doutoral (modalidade sanduíche) junto ao Grupo de Pesquisa NLX da Faculdade de Ciências da Universidade de Lisboa (FCUL)[3]. Nesse contexto, foram conduzidos três estudos principais:

- **Preparação do *dataset* Argmap** — durante o estágio, foi realizada uma busca por um *dataset* compatível com a tarefa de mapeamento de argumentos. Apesar da avaliação de múltiplos recursos, nenhum se mostrou plenamente adequado. O mais próximo da tarefa-alvo foi o *dataset* UKP Sentential [Stab et al. 2018], que contém sentenças anotadas com posicionamentos favoráveis, contrários ou neutros frente a determinados tópicos. Contudo, o recurso original não disponibiliza os textos completos dos documentos, o que exigiu a realização de raspagem de dados (*web scraping*) para reconstrução do corpus e sua posterior adaptação à ferramenta de anotação Brat[4].
- **Ensaios de anotação** — com o *dataset* reconstruído e a ferramenta Brat configurada, foram conduzidas diversas sessões de anotação independente com uma equipe de anotadores especialistas. Nessas sessões, a concordância entre anotadores (*Inter-*

[2] https://github.com/ruanchaves/hashformers
[3] https://ciencias.ulisboa.pt/pt/nlx
[4] https://brat.nlplab.org/

Annotator Agreement — IAA) foi utilizada como principal métrica de avaliação da qualidade das anotações. Sempre que havia divergência entre rótulos atribuídos, o caso era discutido em reunião e a decisão consensual era registrada no manual de diretrizes de anotação (*guideline*). Esse ciclo de anotações independentes, seguido por adjudicação coletiva e atualização do *guideline*, repetiu-se de forma sistemática. Apesar do esforço, os índices de IAA permaneceram em níveis insatisfatórios, até que se diagnosticou que anotações aplicadas a segmentos menores — contendo um ou mais parágrafos sobre um mesmo tópico — permitiam alcançar valores aceitáveis de concordância. A partir dessa constatação, concluiu-se que a tarefa de Segmentação de Tópicos era um pré-requisito essencial para viabilizar a anotação de estruturas argumentativas.

- ***Dataset* de segmentação e classificação de tópicos** — foram iniciados ensaios da tarefa de particionamento do texto em segmentos tematicamente coesos. Essa fase inicial, conduzida por uma equipe reduzida de especialistas e aplicada a um lote pequeno de documentos, resultou na estabilização do índice de IAA em um patamar considerado aceitável. Para viabilizar a anotação em larga escala, foi necessário expandir a equipe por meio de processos de recrutamento, seleção e treinamento de novos anotadores. Outro fator relevante foi o custo estimado da anotação, calculado em R$ 10,00 por documento, o que totalizaria R$ 4.000,00 para cada camada de anotação. Além da segmentação textual, identificou-se a necessidade de classificar cada segmento com base na atribuição de frases-chave. Dessa forma, decidiu-se construir uma camada dupla de anotação: segmentação e classificação de tópicos. O financiamento obtido permitiu cobrir ambas as tarefas, elevando o custo total para R$ 8.000,00. A anotação da segmentação foi finalizada com sucesso. No entanto, a anotação da classificação de tópicos ainda permanece incompleta, em razão de restrições orçamentárias.
- **Ferramenta Argmap** — com o objetivo de apoiar a anotação realizada por uma equipe numerosa de anotadores, foi desenvolvida a ferramenta Argmap. A plataforma centraliza o gerenciamento das tarefas de anotação, incluindo a atribuição de lotes, controle de versões e cálculo automatizado da concordância entre anotadores (*Inter-Annotator Agreement* — IAA). A estratégia adotada consistiu na integração do editor Brat à plataforma, ampliando suas funcionalidades com recursos adicionais, como controle de qualidade baseado em métricas, painel de monitoramento e visualização interativa de árvores argumentativas. O desenvolvimento da Argmap foi fundamental para viabilizar os ensaios de anotação em larga escala e resultou também em uma valiosa oportunidade de formação para um estudante de iniciação científica, que atuou como colaborador técnico no projeto.

Essa etapa da pesquisa proporcionou aprendizados fundamentais que orientaram

os rumos da tese e fundamentaram os estudos da fase seguinte. Entre os principais *insights*, destacam-se:

- **Inviabilidade da anotação completa do *dataset* de Mapeamento de Argumentos** — durante apresentações em seminários do grupo de pesquisa NLX, pesquisadores experientes alertaram para a alta complexidade da tarefa de anotação completa. A recomendação foi reavaliar o escopo da pesquisa, considerando os riscos de inviabilidade no prazo e nos recursos disponíveis. Estimativas práticas indicaram um custo mínimo de R$ 48.000,00 para a finalização da anotação, envolvendo uma equipe de cinco anotadores e um gerente, ao longo de dois anos.
- **Qualidade como requisito central de anotação** — ao longo dos ensaios, diversos mecanismos de controle de qualidade baseados em IAA foram concebidos. Segundo [Klie, Castilho e Gurevych 2024], a qualidade dos dados é essencial para garantir modelos acurados, imparciais e avaliáveis com rigor. Esse princípio norteou a decisão de decompor tarefas complexas em subtarefas mais simples e controláveis. A decomposição mostrou-se decisiva em pontos críticos do processo, especialmente quando o IAA não se estabilizava em níveis aceitáveis.
- **Amadurecimento metodológico via decomposição de tarefas** — a decomposição, orientada por critérios de qualidade, levou à reformulação da tarefa original de Mapeamento de Argumentos em subtarefas específicas, como segmentação e classificação de tópicos, além da detecção de componentes argumentativos. Embora a decomposição torne as tarefas mais manejáveis e confiáveis, também implica aumento de custos e planejamento mais detalhado.

A.3 Fase de Conclusão de Estudos

Nesta fase final da pesquisa, o método de decomposição de tarefas, orientado pela qualidade de anotação, foi aplicado em duas tarefas complementares de menor custo: Identificação de Gênero Textual e Curadoria de Frases-Chave. Ambas foram selecionadas por sua viabilidade técnica e financeira, além do potencial de impacto na organização e consistência das anotações. A seguir, um resumo de cada tarefa:

- **Identificação de Gênero Textual** — essa subtarefa consistiu em atribuir um rótulo de gênero textual (como notícia, blog, índice, entre outros) aos documentos do *dataset* de Mapeamento de Argumentos. Foi motivada pela observação de baixos índices de concordância entre anotadores (IAA) na tarefa de segmentação de tópicos, especialmente em documentos pertencentes a gêneros menos estruturados. A identificação e exclusão desses casos problemáticos permitiu melhorar a consistência da anotação de dados e fornecer aos anotadores expectativas mais claras sobre

a estrutura argumentativa dos textos. Como resultado, houve aumento nos níveis de IAA da tarefa anterior (*upstream task*), além da criação de um *dataset* com 400 documentos classificados em 12 gêneros principais, acompanhado de um *guideline* e de um modelo automático de classificação.

- **Curadoria de Frases-Chave** — esta subtarefa teve como objetivo filtrar frases-chave de alta qualidade a partir de um grande volume de termos ruidosos extraídos automaticamente de corpus. Foi motivada pela necessidade de se obter frases representativas e discriminativas para apoiar a tarefa de classificação de tópicos. Como principais entregas, destacam-se: (i) um *dataset* inédito para curadoria; (ii) uma ferramenta de anotação especializada, a KPC Tool; e (iii) uma métrica original para avaliação da concordância entre agrupamentos (clusters) de frases-chave.

Essa etapa final da pesquisa permitiu consolidar práticas e aprendizados acumulados ao longo do projeto, além de reformular o escopo da tese com base em evidências empíricas. Entre os principais destaques, estão:

- **Design Science Research (DSR)** — a metodologia de Design Science Research foi aplicada de forma transversal ao longo da tese, guiando a construção de artefatos úteis e relevantes com base em problemas reais, sem abrir mão do rigor científico [Peffers et al. 2007]. Embora não tenha sido a abordagem metodológica principal, seus princípios informaram o ciclo iterativo de design, avaliação e refinamento dos artefatos.
- **Consolidação da Decomposição de Tarefas** — o método de decomposição foi empregado sistematicamente em todos os estudos conduzidos, promovendo modularidade, explicabilidade e ganhos em qualidade anotativa. Conforme apontado por [Rokach 2006], a decomposição é uma estratégia central no aprendizado de máquina. Além disso, segundo [Cambria et al. 2023], ela constitui um dos sete pilares para o futuro da Inteligência Artificial. A Figura A.1 ilustra as 11 subtarefas resultantes da decomposição da tarefa original de Mapeamento de Argumentos.
- **Consolidação de práticas de garantia de qualidade** — a preocupação com a qualidade das anotações permeou todo o projeto. Foram desenvolvidos diversos mecanismos de controle baseados em concordância entre anotadores (IAA), monitoramento de adjudicações e análise de padrões de inconsistência. Somente recentemente surgiram revisões sistemáticas sobre o tema, que corroboram e contextualizam muitas das práticas adotadas nesta tese.

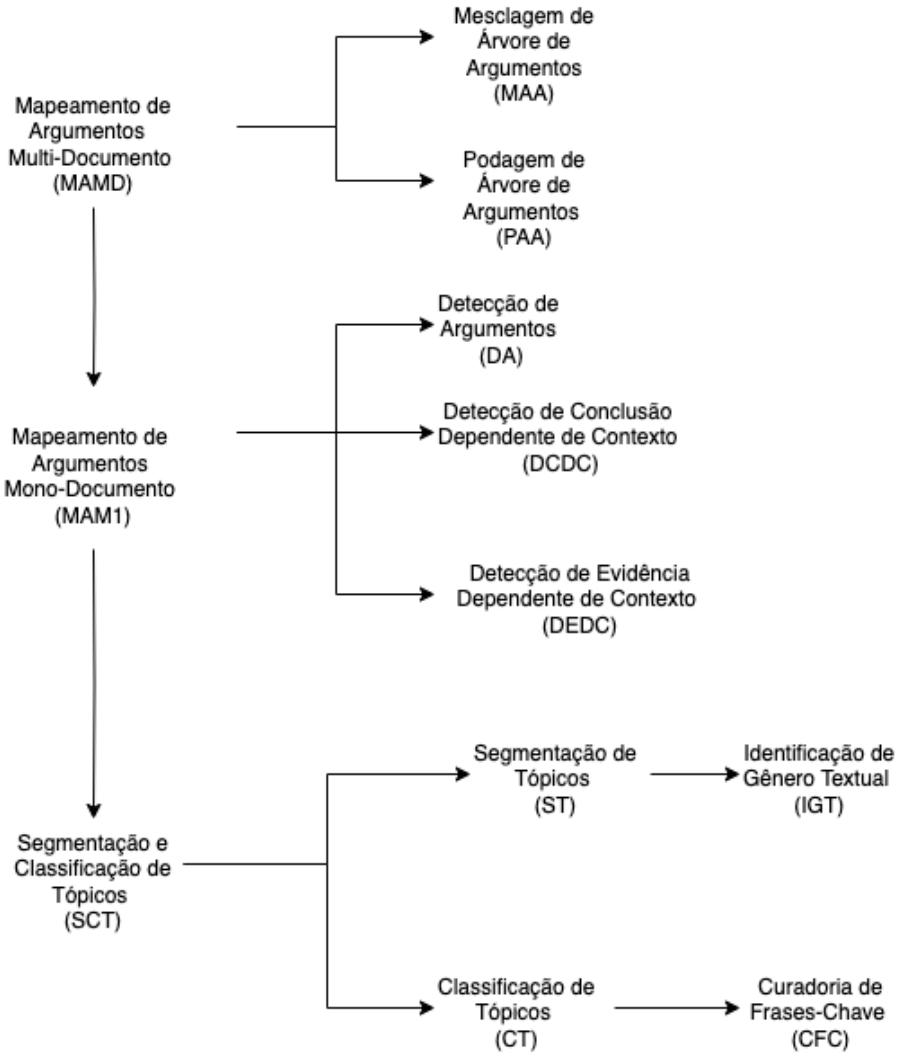


Figura A.1: *Árvore de decomposição em subtarefas da tarefa de Mapeamento de Argumentos.*

O objetivo inicial desta tese era contribuir para o estado da arte em Mapeamento de Argumentos, por meio da construção de *datasets*, modelos de aprendizado de máquina e ferramentas computacionais. Constatar, ao longo do percurso, que esse objetivo não seria plenamente viável poderia ser interpretado, à primeira vista, como um revés. No entanto, é fundamental reconhecer que a pesquisa científica é, por natureza, iterativa e sujeita a reavaliações contínuas. Nesse sentido, os achados emergentes obtidos ao longo do projeto revelaram-se igualmente relevantes para os propósitos da tese.

Um desses achados centrais foi a identificação de padrões de projeto recorrentes nas atividades de anotação. A decomposição em subtarefas, por exemplo, foi aplicada sistematicamente, conforme ilustrado na Figura A.1. Da mesma forma, a estratégia de adjudicação foi incorporada como mecanismo essencial de validação e controle de qualidade. Esses padrões operam como soluções reutilizáveis para problemas recorrentes e, assim como no desenvolvimento de software, podem constituir uma linguagem comum

e estruturante. Esta tese parte do pressuposto de que padrões de projeto também são aplicáveis à anotação de corpus, oferecendo uma base conceitual sólida e replicável para projetos futuros.

A Figura A.1 resume o status de desenvolvimento dos artefatos produzidos ao longo da pesquisa. Cada tarefa foi estruturada com base em uma decomposição funcional, e os artefatos produzidos incluem:

- **Guideline** — manual com diretrizes de anotação;
- ***Dataset*** — corpus anotado com base em rótulos definidos por um esquema documentado no guideline;
- **Ferramenta de Anotação** — interface para anotação manual ou assistida;
- **Métrica de Concordância de Anotação** — métrica utilizada para avaliar a concordância entre anotadores e assegurar a qualidade da anotação.

Tarefas Principais	**Tarefa**	**Guideline**	***Dataset***	**Ferramenta de Anotação**	**Métrica de Concordância**
Mapeamento de Argumentos Multi-Documento (MAMD)	Mesclagem de Árvore de Argumentos (MAA)	◔	○	◑	○
	Podagem de Árvore de Argumentos (PAA)	◔	○	◑	○
Mapeamento de Argumentos Mono-Documento (MAM1)	Detecção de Argumentos (DA)	●	●	●	○
	Detecção de Conclusão Dependente de Contexto (DCDC)	●	◑	●	○
	Detecção de Evidência Dependente de Contexto (DCDC)	●	◑	●	○
Segmentação e Classificação de Tópicos (SCT)	Identificação de Gênero Textual (IGT)	●	●	●	●
	Segmentação de Tópicos (CT)	●	●	●	●
	Classificação de Tópicos (CT)	●	◑	●	●
	Curadoria de Frases-Chave (CFC)	●	●	●	●

Tabela A.1: *Status geral das tarefas, artefatos e métricas*

Legenda: ○ Não Implementado ◔ Iniciado ◑ Parcial ● Completo

Cada tarefa principal da tese resultou na produção de pelo menos um artefato parcial ou completo, refletindo ciclos iterativos de desenvolvimento, avaliação e refinamento. Os três estudos de caso relatados nos Capítulos 5, 6 e 7 ilustram, respectivamente, como as metodologias adotadas foram aplicadas para investigar: (i) a segmentação de hashtags como tarefa de pré-processamento com alto potencial de decomposição; (ii) a curadoria de frases-chave como processo composto por subtarefas interdependentes; e (iii) a decomposição de tarefas complexas de anotação de corpus, com foco em estratégias de controle

de qualidade e reuso de padrões.

A partir da análise cruzada desses estudos, foi possível identificar padrões de projeto reutilizáveis e estratégias eficazes de decomposição de tarefas, contribuindo tanto para o avanço teórico quanto para a prática da anotação em Processamento de Linguagem Natural.

APÊNDICE B

Teorias Complementares

B.1 Padrões em Diversas Áreas de Aplicação

Além dos padrões de projeto, há também diversas outras áreas em que padrões são aplicados, tais como: padrões de processos, padrões com humanos no circuito e padrões para aprendizado de máquina. Nesta seção, exemplos serão apresentados para ilustrar a ampla gama de aplicação do conceito, interligando atividades que envolvem interações: de humanos com humanos; humanos e computadores; e processos computacionais.

Padrões de processo, assim como os *padrões de projeto*, também emergem da aplicação repetida de boas práticas, mas seu foco está em aspectos mais práticos e de execução de processos contínuos. Esses padrões surgem da necessidade de organizar tarefas de maneira eficiente, garantindo a consistência e a confiabilidade em operações repetidas, como a implantação de sistemas, manutenção de infraestrutura ou automação de fluxos de trabalho. Eles fornecem diretrizes e procedimentos que asseguram que operações rotineiras sejam realizadas de maneira uniforme, reduzindo a probabilidade de erros e aumentando a eficiência geral. Ao identificar quais práticas resultam em maior estabilidade e desempenho, essas práticas são transformadas em padrões que podem ser aplicados em diferentes contextos.

Um exemplo bem conhecido de *padrão de processo* é a programação por pares, desenvolvido por [Beck 1999]. Este método, originado na prática de desenvolvimento ágil, envolve dois desenvolvedores trabalhando juntos em uma única estação de trabalho: um codifica enquanto o outro revisa e fornece *feedback* em tempo real. Esse padrão de processo surge de boas práticas voltadas para a melhoria da qualidade do código e da eficiência no desenvolvimento. A prática demonstra benefícios como a redução de erros, a promoção de melhores práticas de codificação e o compartilhamento de conhecimento entre membros da equipe. Formalizada e documentada ao longo do tempo, a programação por pares se tornou um padrão de processo amplamente adotado em equipes de desenvolvimento ágil, contribuindo para a entrega mais rápida e confiável de software.

Com o crescente sucesso e popularidade dos sistemas de aprendizado de máquina, surgiram iniciativas para formalizar padrões de design de engenharia de software aplicados a essa área. Uma pesquisa específica sobre **padrões de projeto para aprendizado de máquina**, conduzida por [Washizaki et al. 2022], busca preencher uma lacuna crítica na implementação desses sistemas, frequentemente marcada por alta complexidade e baixa qualidade. O estudo identificou 15 padrões de *design* que abordam aspectos como arquitetura de sistemas, modularidade de componentes e operações de modelos. Esses padrões oferecem soluções práticas e reutilizáveis que podem melhorar significativamente a manutenibilidade, eficiência e segurança dos sistemas, além de fornecer diretrizes úteis para desenvolvedores e equipes de engenharia.

A metodologia utilizada por [Washizaki et al. 2022] foi uma revisão de literatura multivocal que abrangeu fontes acadêmicas e literatura cinza, seguida de validação com desenvolvedores por meio de pesquisas qualitativas. Os padrões foram classificados em três categorias: topologia de sistemas, design de componentes e operações de modelos. Um exemplo destacado é o padrão *"Distinguish Business Logic from ML Model"* (Distinguir Lógica de Negócio do Modelo de ML), que separa a lógica de negócio dos componentes de aprendizado de máquina, facilitando a manutenção e o ajuste do sistema. Esse padrão sugere a criação de APIs claras entre os componentes tradicionais e de ML, dividindo o sistema em camadas distintas, como lógica de negócios, dados e apresentação. Essa abordagem permite monitorar e ajustar os componentes de ML sem impactar diretamente na lógica de negócios.

Os resultados da pesquisa de [Washizaki et al. 2022] com desenvolvedores indicaram que padrões como *"Distinguish Business Logic from ML Model"* são amplamente utilizados devido à sua capacidade de melhorar a manutenibilidade e simplificar a depuração de sistemas complexos. Apesar de muitos desenvolvedores não estarem familiarizados com todos os padrões, a maioria demonstrou interesse em adotar essas práticas no futuro. O estudo conclui que aumentar a conscientização sobre esses padrões e fornecer documentação mais acessível pode ampliar sua adoção, promovendo maior eficiência e consistência no desenvolvimento de sistemas de aprendizado de máquina.

Além dos padrões que abordam sistemas com interações puramente humanas ou somente computacionais, há outro contexto onde há sistemas com interações humanas e máquinas. Este tipo de sistema é chamado de ***Human-in-the-loop (HITL)***, trata-se de uma abordagem no aprendizado de máquina que integra a intervenção humana em etapas críticas do processo, como anotação de dados, treinamento de modelos e avaliação de resultados. [Wu et al. 2022] destacam que o HITL combina a *expertise* humana com o poder computacional para reduzir custos e melhorar a eficiência, enquanto [Mosqueira-Rey et al. 2023] enfatizam que a implementação pode variar entre aprendizado ativo, aprendizado interativo e *machine teaching*, dependendo do controle humano

no ciclo. Essa abordagem é essencial para superar limitações dos modelos tradicionais, como a falta de explicabilidade e a dificuldade em capturar relações causais, especialmente em aplicações críticas como saúde e segurança. Ao alinhar o julgamento humano às capacidades das máquinas, o HITL possibilita sistemas mais robustos e confiáveis.

Em HITL, assim como em outras áreas, também foram propostos **Padrões de Projeto para Aprendizado de Máquina com Humano no *Loop*** que consistem em soluções reutilizáveis que orientam o desenvolvimento de sistemas baseados em aprendizado de máquina, nos quais humanos desempenham um papel ativo no processo, seja durante o treinamento, a operação ou a colaboração. Andersen e Maalej (2023) propuseram um catálogo com dez padrões que auxiliam na superação de desafios como a confiabilidade limitada dos modelos, alto custo de envolvimento humano e trade-offs entre esforço humano e desempenho. Esses padrões foram definidos a partir de uma revisão semi-sistemática da literatura, aliada à experiência prática dos autores no desenvolvimento de sistemas HITL. Os padrões são divididos em três categorias principais:

- **Padrões de Treinamento**:
  - **Aprendizado Ativo (*Active Learning*)** – Seleciona amostras informativas de dados para minimizar o esforço humano necessário na rotulagem.
  - **Aprendizado Interativo Visual (*Visual Interactive Learning*)** – Utiliza interfaces visuais para permitir que especialistas analisem e selecionem dados de forma mais intuitiva.
  - **Enganar o Modelo (*Trick the Model*)** – Identifica fraquezas do modelo ao testá-lo com entradas desafiadoras e corrige falhas com novos exemplos.
  - **Aprendizado Baseado em Prompt (*Prompt-based Learning*)** – Aproveita modelos pré-treinados para reduzir ou eliminar a necessidade de rotulagem específica da tarefa.
- **Padrões de Operação**:
  - **Sistema de Recomendação (*Recommendation System*)** – Fornece sugestões automáticas para apoiar decisões humanas em tarefas críticas e complexas.
  - **Moderação Ativa (*Active Moderation*)** – Revisa manualmente previsões incertas para melhorar a precisão do modelo em tempo real.
  - **Correção em Tempo Real (*Thumbs Up or Down*)** – Permite que usuários ajustem previsões incorretas diretamente durante o uso do sistema.
  - **Aprendizado Contínuo (*Continuous Learning*)** – Atualiza regularmente o modelo com novos dados para mantê-lo relevante em ambientes dinâmicos.
- **Padrões de Colaboração**:

– **Explicação Baseada em Instâncias (*Instance-based Explanation*)** – Oferece justificativas compreensíveis para decisões do modelo, aumentando a confiança do usuário.
– **Acordo em Multidão (*Crowd Agreement*)** – Utiliza múltiplos rótulos humanos para reduzir vieses e melhorar a consistência e a qualidade das anotações.

Esses padrões fornecem diretrizes práticas para equilibrar esforço humano, custo e desempenho no desenvolvimento de sistemas HITL [Andersen e Maalej 2023].

No contexto relacionado à anotação de corpus e tarefas de Processamento de Linguagem Natural (PLN), os padrões de projeto podem ser metodologias ou abordagens repetíveis, reconhecidamente úteis para resolver desafios específicos no processo de anotação de dados ou na criação de artefatos. Esses padrões podem ser descritos, categorizados e aplicados de forma similar aos padrões de software, mas no contexto de tarefas de PLN.

B.2 Exemplos de Métodos de Decomposição de Tarefas

A presente seção reúne exemplos representativos de métodos de decomposição de tarefas, com o objetivo de ilustrar, de forma concreta, diferentes estratégias adotadas para a fragmentação de problemas complexos em subcomponentes mais manejáveis. Tais métodos abrangem desde abordagens clássicas, baseadas em algoritmos determinísticos — como o paradigma de divisão e conquista — até técnicas contemporâneas baseadas em aprendizado de máquina, mineração de dados e modelos probabilísticos. Em comum, essas abordagens demonstram como a decomposição contribui para reduzir a complexidade computacional, promover modularidade e viabilizar a generalização. As seções subsequentes descrevem, em detalhe, diferentes formas de decomposição, organizadas segundo os tipos de estrutura ou representação explorados, incluindo dados, funções, conceitos intermediários e tópicos latentes.

B.2.1 Decomposição de Tarefas e Divisão e Conquista

A decomposição de tarefas de aprendizado de máquina compartilha similaridades com os algoritmos do **paradigma de divisão e conquista**, que se baseiam em dividir um problema grande e complexo em subproblemas menores e mais manejáveis. Esses subproblemas são resolvidos, geralmente de forma recursiva, e suas soluções são combinadas para obter a solução do problema original. Exemplos clássicos dessa abordagem incluem os algoritmos *Merge Sort* e *Quick Sort*, que, no entanto, operam utilizando funções computacionais determinísticas. A principal distinção entre as duas abordagens está

no fato de que a decomposição de tarefas emprega aprendizado de máquina, caracterizado por maior complexidade, resultados não determinísticos e uma adaptabilidade superior, devido à sua capacidade de generalização. A Tabela B.1 apresenta uma comparação detalhada entre essas abordagens.

Abordagem	**Divisão e Conquista**	**Decomposição de Tarefas**
Definição	Uso de funções computacionais, que resolvem uma operação matemática ou lógica bem definida.	Uso de tarefas de aprendizado de máquina, que fornecem soluções guiadas por objetivos alcançados por modelos aprendendo padrões a partir de dados.
Princípio de Funcionamento	Regras explícitas e algoritmos codificados manualmente.	Dados de entrada e aprendizado de padrões implícitos.
Determinismo	Determinística: mesma entrada sempre gera a mesma saída.	Não determinística: saída depende de aprendizado e generalização.
Adaptabilidade	Estática: não se adapta a novos dados sem modificação explícita.	Dinâmica: pode generalizar para novos dados após o treinamento.
Exemplos	Soma, cálculo de média, Transformada de Fourier.	Classificação de imagens, tradução automática, reconhecimento de fala.
Complexidade	Geralmente mais simples e específica.	Envolve processos complexos, como treinamento de modelos e ajuste de hiperparâmetros.

Tabela B.1: *Comparação entre os paradigmas Divisão e Conquista e Decomposição de Tarefas*

B.2.2 Padrões de Interesse na Mineração de Dados

A Mineração de Dados (Data Mining), também conhecida como Descoberta de Conhecimento em Bancos de Dados (Knowledge Discovery in Databases – KDD), é o processo de extração de conhecimento relevante a partir de grandes volumes de dados [Han Micheline Kamber 2006, p. 6]. Esse conhecimento, frequentemente representado por **padrões de interesse**, pode ser aplicado em diversos contextos, como a análise

de hábitos de consumo em supermercados [Han Micheline Kamber 2006, p. 186]. Por exemplo, por meio da mineração de regras de associação, varejistas podem identificar itens frequentemente comprados em conjunto, como leite e pão. Com base nesses padrões, estratégias comerciais podem ser implementadas, incluindo a reorganização de produtos nas prateleiras para estimular vendas cruzadas ou a criação de promoções específicas para itens correlacionados. Essas iniciativas não apenas aumentam o volume de vendas, mas também promovem uma experiência de compra mais satisfatória para os clientes.

Diversos *métodos de decomposição* foram desenvolvidos na área de mineração de dados, tendo como objetivo principal reduzir a complexidade computacional e otimizar o tempo de processamento. Segundo [Kusiak 2000], a **decomposição em conjunto de dados** (*datasets*) pode ser aplicada tanto no espaço quanto no tempo. A **decomposição no espaço** ocorre em atributos (*features*) ou objetos (*tuples*), enquanto a **decomposição no tempo** é utilizada, por exemplo, em séries temporais, permitindo a identificação de padrões como sazonalidade ou ciclos temporais [Box e Jenkins 1990, p. 367].

Por outro lado, [Bhargava 1999] sugeriu que a mineração de *padrões de interesse* consiste, essencialmente, em encontrar um subconjunto no *espaço de atributos*. Contudo, devido à grande quantidade de atributos, essa tarefa pode levar a uma explosão combinatória, envolvendo bilhões de candidatos. Para mitigar esse problema, o autor propôs o uso de um algoritmo genético, decompondo a busca em três partes interdependentes: busca de atributos, busca de modelos e busca de padrões. Em um estudo de caso utilizando dados médicos de participantes dos EUA no Conflito do Golfo Pérsico de 1991, envolvendo uma base de 150 atributos e 20.000 registros, foram identificados diversos padrões. Entre eles, descobriu-se que participantes que relataram exposição a pesticidas e consumo de alimentos não pertencentes às Forças Aliadas apresentaram 3,9 vezes mais probabilidade de serem diagnosticados com doenças degenerativas das articulações/osteoartrite e apneia do sono. Este tipo de método de decomposição, basicamente, agrupou atributos (features) em subconjuntos, produzindo *padrões de interesse*; desta forma, trata-se de uma **decomposição por atributos**.

B.2.3 Árvores de Decisão no Aprendizado de Máquina

No *aprendizado de máquina*, vários modelos surgiram, e um dos que mais se destacam, daqueles que utilizam *métodos de decomposição*, é a **árvore de decisão**. Esse modelo é baseado em uma estrutura hierárquica que representa decisões e seus possíveis resultados. Cada nó interno da árvore corresponde a uma condição ou pergunta sobre um atributo, enquanto os ramos representam os possíveis caminhos ou respostas, levando a outros nós ou folhas que indicam os resultados finais ou classes. *Árvores de decisão* são amplamente utilizadas em tarefas de classificação e regressão, oferecendo uma abordagem

intuitiva para dividir problemas em subproblemas menores de forma lógica e sequencial, sendo geradas manualmente ou por algoritmos como ID3 [Quinlan 1986] ou CART [Breiman 2017].

Um exemplo da aplicação deste modelo é apresentado na Figura B.1. Trata-se de um problema de classificação utilizando o popular conjunto de dados *Iris Flower Dataset*, que contém 150 instâncias de flores, descritas por quatro atributos (comprimento e largura de sépalas e pétalas) e classificadas em três classes (*Iris setosa*, *Iris virginica* e *Iris versicolor*)[1]. Neste exemplo[2], foi utilizada a classe `DecisionTreeClassifier` do framework Scikit-Learn[3], com o algoritmo CART [Breiman 2017] como padrão. O aprendizado foi realizado com 30% das instâncias separadas para treino, resultando em uma árvore de decisão com quatro nós (`nd1`, `nd2`, `nd3` e `nd4`), definidos pelos critérios de largura e comprimento da pétala. Os valores críticos foram ≤ 0.75, ≤ 1.75, ≤ 4.95 e ≤ 1.55, atribuídos ao primeiro, segundo e último nós (largura da pétala) e ao segundo nó (comprimento da pétala), respectivamente. Cada nó separa as instâncias em subconjuntos associados a uma classe. Por exemplo, o nó `nd1` classificou 38 instâncias como pertencentes à classe *setosa*. Essa árvore de decisão, gerada automaticamente, tornou explícitos os critérios de segmentação do conjunto de dados de forma transparente e compreensível, atingindo uma acurácia de 95%. Sob o ponto de vista de método de decomposição, as tuplas foram separadas em classes; portanto, trata-se de uma **decomposição em espaço de tuplas** ou, simplesmente, **decomposição em tuplas**.

Além da *decomposição em atributos* e *decomposição em tuplas* apresentadas anteriormente, [Rokach 2006] também identificou o método de **decomposição em amostras** (*sample decomposition*), conhecido também como particionamento. Esse método consiste em dividir o conjunto de treinamento de um problema de classificação em subconjuntos menores, chamados de "amostras". Cada subconjunto é utilizado para treinar um classificador separado, e as saídas desses classificadores são combinadas para resolver o problema original. [Rokach, Maimon e Arad 2005] assim definiu formalmente este método, tradução nossa:

> *Dado um método de aprendizado I, um método de combinação C e um conjunto de treinamento S com o conjunto de atributos de entrada $A = \{a_1, a_2, \ldots, a_n\}$ e o atributo alvo y proveniente de uma distribuição D sobre o espaço rotulado de instâncias, o objetivo é encontrar uma decomposição ótima T_{opt} do conjunto de treinamento S em ψ subconjuntos $B_k \subseteq S$, $k =$*

[1] Para mais informações, visite: https://en.wikipedia.org/wiki/Iris_flower_data_set

[2] Baseado em um tutorial disponível em: https://www.geeksforgeeks.org/building-and-implementing-decision-tree-classifiers-with-scikit-learn-a-comprehensive-guide/

[3] Documentação disponível em: https://scikit-learn.org/1.5/modules/generated/sklearn.tree.DecisionTreeClassifier.html

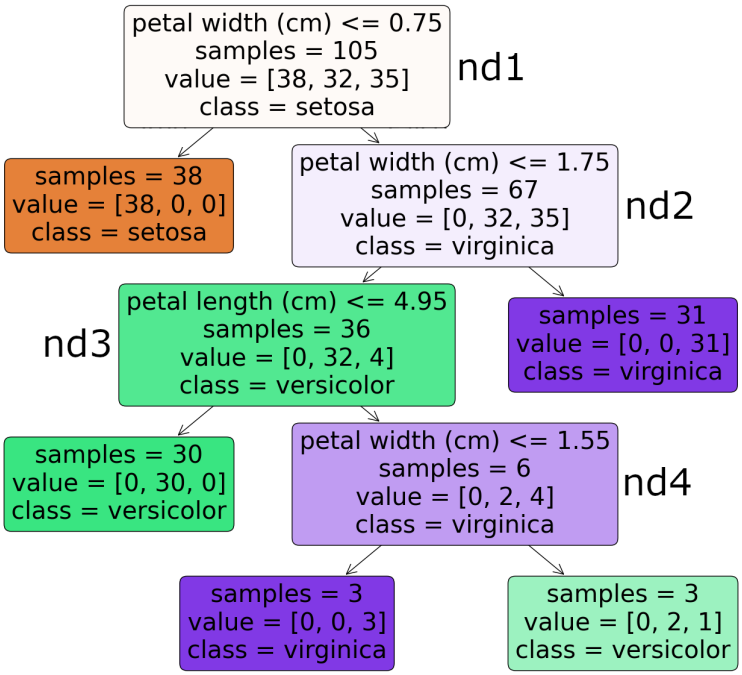


Figura B.1: *Exemplo de aplicação do modelo de aprendizado de máquina de árvore de decisão no conjunto de dados popular "Iris Flower Dataset"*

$1,\ldots,\psi$, *que satisfaçam* $B_i \cap B_j = \emptyset; i, j = 1,\ldots,\psi; i \neq j$, *de forma que a acurácia obtida pelos classificadores combinados* $I_k(S \cap B_k), k = 1,\ldots,\psi$, *usando o método de combinação C, seja maximizada.*

A notação I representa um indutor probabilístico (ou seja, um algoritmo que gera classificadores que também fornecem estimativas da probabilidade condicional da característica alvo, dado os atributos de entrada), e $I(S)$ representa um classificador probabilístico que foi gerado ao ativar o método de indução I no conjunto S.

Um *método de decomposição em amostra* bem conhecido é o método *Bagging* [Breiman 2017] (Bootstrap Aggregating), que melhora a precisão de modelos supervisionados ao gerar várias subamostras do conjunto de treinamento, com reposição, para treinar diferentes modelos. As previsões dos modelos são combinadas, geralmente por votação ou média, reduzindo a variância e aumentando a estabilidade, especialmente em algoritmos instáveis como árvores de decisão.

Outro método também bem conhecido é o método de validação cruzada (*k-fold cross-validation ensembles*), também conhecido como *Cross-Validated Committees* [Gams 1989], que cria uma combinação *ensemble* de classificadores dividindo o conjunto de treinamento em k subconjuntos de tamanho igual. Cada classificador é treinado em $k-1$ subconjuntos e testado no subconjunto restante, repetindo o processo para cada partição. Isso gera k modelos que são combinados, geralmente por votação, para fornecer

a previsão final. Essa abordagem reduz o risco de *overfitting*, melhora a precisão e utiliza eficientemente os dados disponíveis, sendo especialmente útil para algoritmos de alta complexidade.

Outro exemplo de aplicação dessa abordagem é o algoritmo *CBCD (Cluster-Based Concurrent Decomposition)*, proposto por [Rokach, Maimon e Arad 2005], que utiliza o agrupamento por *K-means* [MacQueen 1967] para dividir o conjunto de treinamento em subconjuntos disjuntos e balanceados. Após a divisão, modelos individuais são treinados, e suas saídas são combinadas por votação, similar ao processo utilizado pelo método de *Bagging* [Breiman 2017]. Os experimentos conduzidos por [Rokach, Maimon e Arad 2005] demonstraram que o CBCD superou o *Bagging* em diversos casos, apresentou desempenho estatisticamente superior ao algoritmo C4.5 [Quinlan 2014] em alguns *datasets* e demonstrou robustez ao ajustar seus parâmetros via meta-classificador.

B.2.4 Decomposição por Conceitos Intermediários

Os *métodos de decomposição* anteriores fazem parte daqueles que processam diretamente os dados ou os *conceitos originais*. Na tipologia apresentada por [Rokach 2006], existem também métodos que operam sobre **conceitos intermediários** (*intermediate concepts*), ou seja, *conceitos originais* que passam por algum tipo de transformação. Entre os métodos que utilizam *conceitos intermediários*, destacam-se dois: **agregação de conceitos**, em que conceitos agregados formam novos conceitos, organizados em uma hierarquia; e **decomposição de funções**, que subdivide uma função em uma árvore de funções, na qual cada nó corresponde a uma função, denominada *conceito intermediário*, e cada folha representa um atributo de entrada das funções.

Um conceito importante neste contexto é a **abstração** que consiste no processo de selecionar e isolar aspectos relevantes de um problema para um propósito específico, enquanto se suprimem os aspectos irrelevantes. Diferentes *abstrações* podem ser criadas para o mesmo objeto, dependendo do objetivo [Rumbaugh et al. 1991]. A **agregação**, por sua vez, é uma forma de *abstração* que permite tratar uma relação entre objetos nomeados como um objeto único e nomeado de nível superior [Smith e Smith 1977]. Essa abordagem também se reflete na linguagem natural, onde a *agregação* simplifica a expressão de relações complexas, permitindo representar grupos de atributos como conceitos únicos e abstratos, facilitando a comunicação e a modelagem do mundo real.

Um exemplo dado por [Smith e Smith 1977] para ilustrar a *agregação* na linguagem natural envolve o conceito de **"matrícula"** em uma universidade. Ele explica que a frase:

"O aluno P recebeu a nota G em uma turma de um curso identificado

pelo número C#, com CH horas de crédito e descrição D, ministrado pelo professor I durante o semestre S na sala R"

pode ser representada como uma relação 8-ária entre esses objetos. Para simplificar, ele abstrai *Curso* como uma agregação de $(C\#, CH, D)$, e *Turma* como uma agregação de $(Curso, I, S, R)$. Assim, a relação inicial é reduzida a uma relação 3-ária entre $(Turma, P, G)$, representada como o objeto agregado *"matrícula"*. Em linguagem natural, isso reflete a possibilidade de dizer simplesmente: *"O aluno P se matriculou na turma T e recebeu a nota G"*, sem precisar mencionar todos os detalhes subjacentes.

Por sua vez, [Iivari 1992] classificou esse último tipo de agregação como uma **abstração linguística**, cujos componentes são elementos de uma língua natural, como palavras. A partir desta, ele definiu uma *agregação* em nível superior denominada **abstração representacional**, na qual se constrói uma representação formal e estruturada, como um diagrama UML ou um modelo matemático.

B.2.5 Modelos Gráficos Probabilísticos

Como exemplo de *abstração linguística* e também *abstração representacional*, [Buntine 2000] propôs o uso de **Modelos Gráficos Probabilísticos** (*Probabilistic Graphical Models* ou PGM) para representar documentos de textos através de um modelo de tópicos e subtópicos. [Bishop e Nasrabadi 2006] definiu que um *PGM* consiste em um grafo, no qual cada nó representa uma variável aleatória (ou grupo de variáveis aleatórias), e as arestas expressam relações probabilísticas entre essas variáveis. Há duas variantes principais desse modelo, uma conhecida como **Redes Bayesianas** (*Bayesian Networks*), que utiliza grafos dirigidos; e outra conhecida como **Campo Aleatório de Markov** (*Markov Random Field*) que emprega grafos não dirigidos.

Baseado em *PGM*, [Blei, Ng e Jordan 2003] propuseram um modelo estatístico generativo não supervisionado chamado **Alocação Latente de Dirichlet** (*Latent Dirichlet Allocation (LDA)*) que identifica temas latentes em um corpus com D documentos. Ele assume que cada documento d é uma **mistura de tópicos** representada por uma distribuição probabilística θ_d, onde cada tópico k contribui com uma proporção específica para o documento. Cada tópico k, por sua vez, é descrito por uma **mistura de palavras** associada a uma **distribuição probabilística de Dirichlet** β_k, indicando a probabilidade de cada palavra w pertencer ao tópico. Por exemplo, em um corpus de notícias, um documento sobre avanços tecnológicos pode ser composto por 70% do tópico "Tecnologia"e 30% do tópico "Ciência", enquanto o tópico "Tecnologia"pode conter palavras como *inteligência artificial*, *algoritmo* e *dados*, com diferentes probabilidades. Com base nas coocorrências de palavras nos documentos, o LDA infere essas distribuições (θ_d e β_k), permitindo ex-

plorar os padrões de temas em coleções textuais de forma eficiente. O diagrama de placas ilustrado na Figura B.2 representa o *modelo gráfico LDA*. LDA

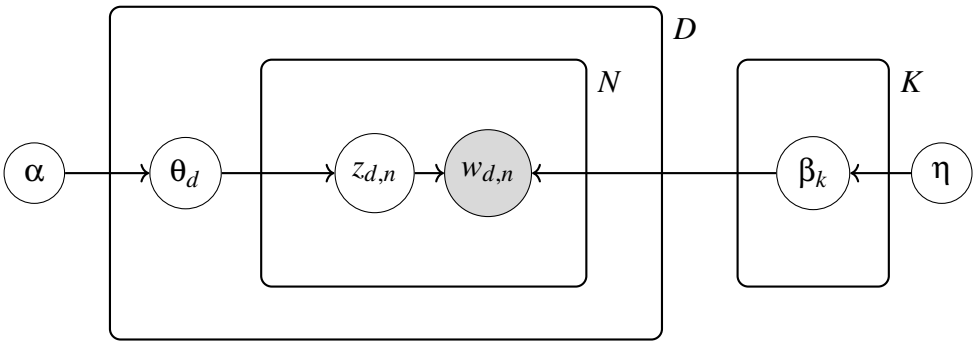


Figura B.2: *Modelo gráfico LDA. Cada nó é uma variável aleatória do modelo. Os nós latentes - proporções de tópicos, atribuições e tópicos - não estão sombreados. os nós observados - as palavras dos documentos - estão sombreados. As constantes vetoriais* α *e* η *são hiperparâmetros que associam pesos para a distribuições* θ_d *e* β_k*. Os retângulos são a notação de "placa", que denota replicação. A placa N denota as palavras da coleção dentro dos documentos; a placa D denota a coleção de documentos dentro da coleção.*

A **Modelagem de Tópicos**, embora tenha origem no campo do **Processamento de Linguagem Natural**, tornou-se um método versátil e amplamente aplicável para a decomposição de tópicos em diversas áreas. Nas revisões de [Blei, Carin e Dunson 2010] e [Blei 2012], os **Modelos de Tópico Probabilísticos** são referenciados como ferramentas eficazes para identificar padrões em imagens, música, áudio, fala, dados genéticos, programas de computador e até em escavações arqueológicas. Além disso, esses modelos têm sido amplamente utilizados na visão computacional, tanto para segmentação e rotulagem de objetos em imagens quanto para análise de vídeos.

Recentemente, [Abdelrazek et al. 2023] revisou e categorizou os métodos de *Modelagem de Tópicos* em quatro abordagens principais: algébrica, difusa (*fuzzy*), probabilística e neural. O estudo avaliou esses métodos em diferentes conjuntos de dados com base nos critérios de coerência, estabilidade, diversidade e eficiência. Com os avanços das **redes neurais**, [Wu, Nguyen e Luu 2024] analisou os **Modelos de Tópicos Neurais** (*Neural Topic Models, NTMs*) e destacou inovações, como o uso de perplexidade para prever tópicos em novos documentos, melhorias na coerência de tópicos, ferramentas de visualização, e o uso de *embeddings* para aprimorar a representação semântica. Outros avanços notáveis são: o uso de metadados para aprendizado supervisionado, NTMs multilíngues, NTMs hierárquicos e NTMs dinâmicos.

Entre os desafios atuais, [Wu, Nguyen e Luu 2024] destacou dificuldades relacionadas à avaliação dos modelos, como a confiabilidade e padronização das métricas, além

da discrepância entre as avaliações automáticas e humanas. Também foram apontadas a ausência de padrões para pré-processamento de *datasets* e problemas de qualidade dos tópicos, como trivialidade, repetição e sensibilidade aos hiperparâmetros.

B.2.6 Decomposição de Conceitos por Tópicos

Além da **decomposição de conceitos por tópicos**, existe outro algoritmo geral de agregação de conceitos que transforma problemas multiclasse em múltiplos problemas binários, ou seja, de duas classes [Dietterich e Bakiri 1994]. Esse método, amplamente conhecido como **ECOC** (*Error Correcting Output Codes*), ou códigos de saída com correção de erros), é utilizado para aumentar a precisão de algoritmos de classificação, especialmente em cenários multiclasse. A abordagem aproveita princípios da teoria da informação relacionados à correção de erros para melhorar a robustez e a confiabilidade dos classificadores.

Por exemplo, na tarefa de reconhecimento de dígitos manuscritos, onde é necessário classificar cada dígito em $k = 10$ classes, o algoritmo *ECOC* codifica cada classe como uma sequência de bits, chamada *wordcode* (ou palavra-código), gerada com redundância. Embora sejam necessários apenas $\log_2 10 \approx 4$ bits para representar as classes, a codificação pode ser feita com 15 bits para maior robustez. Assim, um dígito como "3"pode ser codificado aleatória ou otimamente como 010100111100101. Durante o treinamento, um classificador binário é treinado para prever cada bit da *wordcode*, resultando em 15 classificadores binários. Na etapa de classificação, todos os classificadores são utilizados para prever uma sequência de bits, e a palavra-código mais próxima é identificada com base em uma métrica, como a distância de Hamming [4].

B.2.7 Decomposição por Funções

Outro *método de decomposição* em *conceito intermediário* é a **decomposição em função** que originalmente foi criado para projeto de circuitos digitais, mas que depois foi adaptado por [Zupan et al. 1997] para aprendizado de máquina (tradução nossa):

> *O objetivo é decompor uma função $y = F(X)$ em $y = G(A,H(B))$, onde X é um conjunto de atributos de entrada $x_1,\ldots,x_n$, e y é a variável de classe. As funções F, G e H são parcialmente especificadas por exemplos, isto é, por conjuntos de vetores de atributos-valor com classes atribuídas. A e B são subconjuntos de atributos de entrada tais que $A \cup B = X$. As funções G e H são determinadas no processo de decomposição e não são*

[4] Para mais detalhes, consulte https://en.wikipedia.org/wiki/Hamming_distance

pré-definidas de nenhuma forma. Sua complexidade conjunta (determinada por alguma medida de complexidade) deve ser menor que a complexidade de F. Tal decomposição também descobre um novo ***conceito intermediário*** *$c = H(B)$. Como a decomposição pode ser aplicada recursivamente em H e G, o resultado, em geral, é uma hierarquia de conceitos. Para cada conceito na hierarquia, há uma função correspondente (como $H(B)$) que determina a dependência desse conceito em relação aos seus descendentes imediatos na hierarquia.*

O método foi implementado em uma ferramenta denominada **HINT (Hierarchy INduction Tool)**, cuja aplicação foi demonstrada em [Zupan et al. 1999] por meio do aprendizado da função booleana $y = (x_1 \text{ OR } x_2) \text{ XOR } (x_3 \text{ OR } (x_4 \text{ XOR } x_5))$. Por se tratar de uma função com 5 atributos, o espaço de entrada consiste em $2^5 = 32$ combinações possíveis, cada uma mapeada para uma saída binária $\{0,1\}$. Em 10 experimentos de aprendizado, HINT foi treinado com 24 instâncias selecionadas aleatoriamente (75% do total) e reproduziu corretamente, em 9 casos, a estrutura apresentada na Figura B.3. O método identificou com sucesso as funções intermediárias f1, f2, f3 e f4 como sendo, respectivamente, as funções booleanas OR, XOR, OR e XOR.

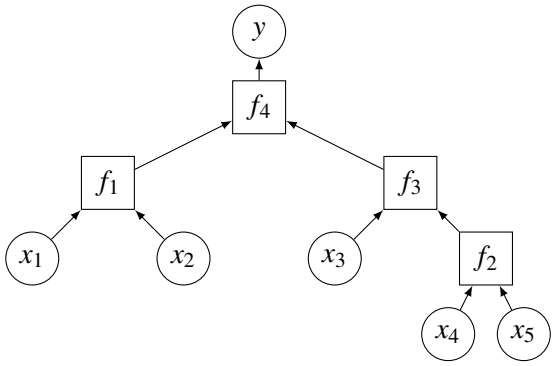


Figura B.3: ***Hierarquia de conceitos*** *intermediários induzidos pelo HINT para o exemplo da função booleana.*

Para avaliar a ferramenta *HINT*, [Zupan et al. 1999] testaram seu desempenho em 14 conjuntos de dados compostos exclusivamente por atributos nominais, comparando os resultados de acurácia com o algoritmo C4.5 [Quinlan 2014]. Os testes mostraram que o HINT apresentou desempenho superior em todos os *datasets*, exceto no SPLICE. O baixo desempenho nesse *dataset* foi atribuído a uma característica específica: seus conceitos intermediários compartilham atributos entre si, formando uma estrutura não hierárquica. Essa particularidade é incompatível com o HINT, que opera com uma estrutura em árvore de conceitos.

Um exemplo da avaliação de uso da ferramenta *HINT* é apresentado na Figura B.4, que ilustra a árvore de conceitos original derivada dos dados de treinamento

do *dataset NURSERY* [Rajkovic 1989]. Este conjunto de dados contém informações provenientes de formulários de solicitação de matrícula de crianças em creches públicas. Na árvore, o nó raiz representa a função-alvo, com uma cardinalidade de 5 — ou seja, o número de categorias que o modelo de classificação deve inferir: *not_recom*, *recommend*, *very_recom*, *priority* e *spec_prior*.

Os *conceitos originais* são representados como nós-folha, correspondendo, neste caso, a oito atributos de entrada. Já os demais nós, denominados *conceitos intermediários*, estão descritos na documentação original [Rajkovic 1989] e aparecem no diagrama como nós-ramo. Quando a *cardinalidade* de um *conceito intermediário* é menor que a soma das cardinalidades de seus nós-filho, isso reflete uma redução na complexidade. Essa redução ocorre pela decomposição de uma função complexa em funções mais simples.

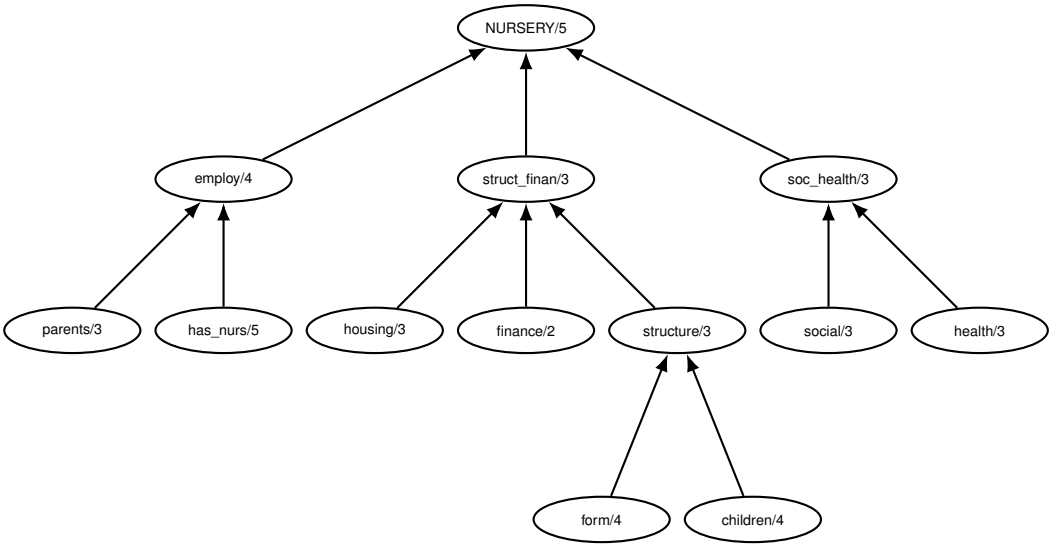


Figura B.4: *Árvore de conceitos original do dataset NURSERY, resultante de dados coletados de formulários de solicitações de matrícula para creches públicas. Cada nó representa uma atributo (nós-folha) ou conceitos intermediários (nós-galho) seguida de sua cardinalidade, que informam a quantidade de valores que um atributo ou conceito intermediário (função) pode assumir.*

A Figura B.5 mostra o resultado da ferramenta *HINT*, que constrói a árvore de conceitos a partir do dataset NURSERY. Espera-se que o algoritmo reconstrua a árvore de conceitos original utilizando dados de treinamento. Alguns conceitos intermediários resultam da abstração direta de atributos, reduzindo a cardinalidade quando um atributo é redundante ou desnecessário; por exemplo, 'social/3' é transformado em 'social'/2'. Outros conceitos intermediários, identificados automaticamente pela ferramenta, foram descobertos, de c1 a c6; por exemplo, o nó 'c5/4' corresponde a 'employ/4'.

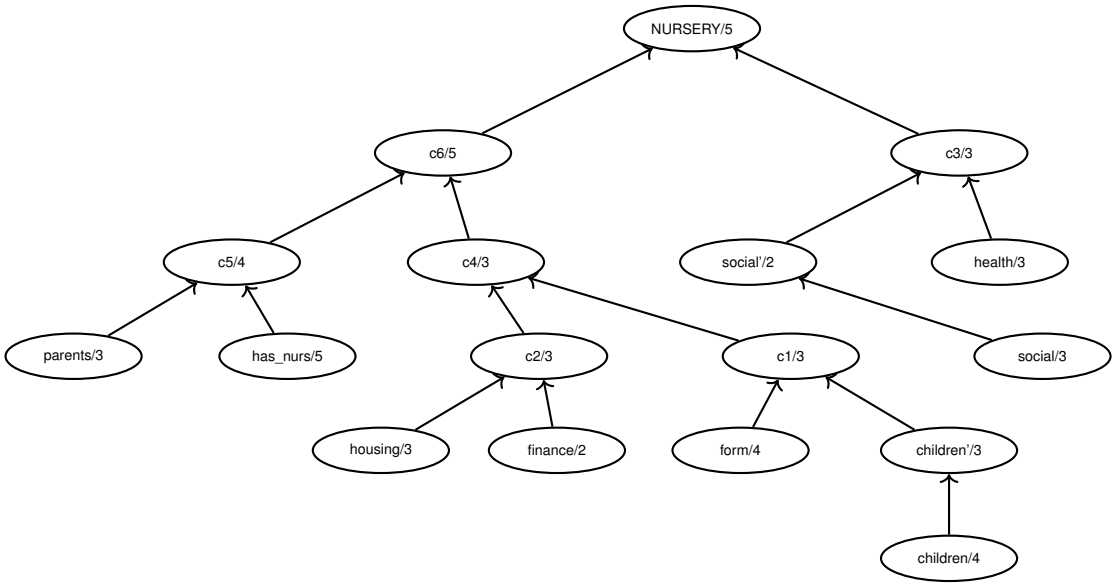


Figura B.5: *Árvore de conceitos descoberta pela ferramenta HINT para o dataset NURSERY*

B.3 Arquitetura de Modelos de Linguagem em Larga Escala (LLMs)

A decomposição de tarefas consolidou-se como uma abordagem fundamental no *Processamento de Linguagem Natural* (*Natural Language Processing* – PLN), especialmente com os avanços proporcionados pelo **aprendizado profundo** (*deep learning*). Desde a introdução da arquitetura *Transformer*, que revolucionou o campo ao introduzir o **mecanismo de atenção** (*attention mechanism*), até o desenvolvimento de **Modelos de Linguagem de Grande Escala** (*Large Language Models* – LLMs), as arquiteturas modernas de redes neurais têm demonstrado capacidades impressionantes para decompor problemas complexos em subcomponentes menores, facilitando o processamento e a resolução eficiente de tarefas.

Esta seção analisa as principais inovações arquiteturais sob o prisma da decomposição de tarefas, com ênfase no papel do *aprendizado de representação* (*representational learning*) como facilitador dessas transformações. O foco recai na evolução das arquiteturas e em como elas permitiram avanços na organização, segmentação e execução de subtarefas em aplicações de PLN, destacando o impacto prático dessas tecnologias na resolução de problemas de alta complexidade.

O aprendizado profundo tem suas raízes nas primeiras redes neurais artificiais, como o Perceptron, proposto por [Rosenblatt 1958], e as *redes de propagação direta (feed-forward networks (FFN))*, que começaram a emergir na década de 1980 com o algoritmo de retropropagação (*backpropagation*) desenvolvido por [Rumelhart, Hinton e Williams 1986]. No entanto, essas arquiteturas iniciais enfrenta-

ram desafios significativos de escalabilidade, especialmente à medida que problemas de classificação se tornavam mais complexos e demandavam modelos com maior profundidade. O aumento do número de camadas em redes neurais revelou limitações críticas, como o problema do desaparecimento do gradiente (*vanishing gradient problem*) [Hochreiter 1991, Bengio, Simard e Frasconi 1994], que dificultava o treinamento efetivo de redes profundas.

Esse desafio limitava a capacidade dos modelos de capturar padrões hierárquicos complexos nos dados, restringindo sua aplicação prática. Foi apenas com o surgimento de técnicas como inicialização de pesos mais eficazes [Glorot e Bengio 2010], funções de ativação não lineares aprimoradas [Nair e Hinton 2010] (*rectified linear unit - ReLU*) e a introdução de arquiteturas avançadas, como Redes Neurais Convolucionais (CNNs) [LeCun et al. 1998], que a escalabilidade começou a ser superada. Essas inovações pavimentaram o caminho para o aprendizado profundo moderno, permitindo avanços significativos no processamento de problemas de classificação em larga escala.

Um dos desafios enfrentados pela área de Processamento de Linguagem Natural (PLN) foi a representação de dados linguísticos para o aprendizado de máquina (AM). Diferentemente de dados tabulares, os dados linguísticos possuem características simbólicas e sequenciais, configurando cadeias complexas de informações, como palavras, frases, parágrafos e documentos. A resposta a esse desafio surgiu com o desenvolvimento de uma subárea do AM denominada *representational learning* (aprendizado de representação), que se concentra na criação de representações úteis e compactas dos dados de entrada para facilitar a resolução de tarefas específicas. Diferentemente das abordagens tradicionais, que dependiam de características manualmente projetadas, o *representational learning* permite que os modelos aprendam automaticamente as representações mais relevantes diretamente dos dados [Bengio, Courville e Vincent 2013]. Essa abordagem está estreitamente relacionada ao *feature learning* (aprendizado de características), cujo objetivo é descobrir automaticamente as representações mais adequadas dos dados em diferentes níveis de abstração, otimizando o desempenho em tarefas específicas.

As representações obtidas por meio de métodos de aprendizado de características são geralmente expressas em formas matemáticas, como vetores em espaços de alta dimensionalidade, e capturam informações essenciais dos dados, preservando padrões que ajudam os algoritmos a realizar tarefas como classificação, tradução ou geração de conteúdo. No contexto do Processamento de Linguagem Natural (PLN), o *representational learning* desempenhou um papel central no avanço de técnicas como *embeddings* de palavras. Um exemplo notável é o *Word2Vec*, que representa cada palavra como um vetor em um espaço contínuo que reflete relações semânticas e sintáticas [Mikolov et al. 2013]. Esses vetores permitem operações vetoriais que capturam relações entre palavras, como a famosa analogia “rei - homem + mulher = rainha”, ilustrando como os *embeddings*

conseguem modelar semelhanças e relações semânticas no espaço vetorial.

Uma das primeiras tentativas de lidar com dados sequenciais foi a criação de *Redes Neurais Recorrentes (RNNs)* (*RNN - Recurrent Neural Networks*). Enquanto as Redes Neurais *Feedforward* (FFNs) processam entradas de forma independente, sem considerar relações temporais, as RNNs lidam com sequências, permitindo que informações de estados anteriores influenciem estados futuros. Em uma FFN, a saída é computada diretamente a partir da entrada sem dependência temporal, seguindo a equação:

$$h = \sigma(W^{xh}x + b^h)$$

$$y = W^{hy}h + b^y$$

Já as RNNs, por sua natureza recorrente, utilizam um estado oculto h_t que se atualiza a cada passo temporal, armazenando memória sobre entradas anteriores. Sua equação é:

$$h_t = \sigma(W^{hx}x_t + W^{hh}h_{t-1})$$

$$y_t = W^{yh}h_t$$

A principal diferença é que a FFN trata cada entrada de forma isolada, enquanto a RNN mantém um histórico das entradas passadas através de sua recorrência ($W^{hh}h_{t-1}$), tornando-a mais adequada para tarefas sequenciais, como modelagem de linguagem e séries temporais.

Outro grande avanço em PLN foram as arquiteturas especializadas para lidar com dados sequenciais. Entre essas, destaca-se a *arquitetura encoder-decoder* que revolucionou aplicações como a tradução automática, permitindo que modelos aprendessem a mapear sequências de entrada em um idioma para sequências de saída em outro idioma de maneira eficiente [Cho et al. 2014]. O *encoder* captura a representação contextual da sequência de entrada, enquanto o *decoder*, com base nessa representação, gera a sequência de saída. A Figura B.6 ilustra o funcionamento dessa arquitetura em alto nível.

A implementação da *arquitetura encoder-decoder* com RNNs enfrentava diversos desafios na prática. Um dos principais problemas era a necessidade de que as sequências de entrada e saída tivessem o mesmo tamanho, além do conhecido problema de *desvanecimento do gradiente*. Para mitigar essas limitações, foram desenvolvidas variantes mais robustas, como as redes *LSTM* (*Long Short-Term Memory*) [Sutskever, Vinyals e Le 2014, Hochreiter e Schmidhuber 1997]. No entanto, mesmo com essas melhorias, vários desafios persistiam, conforme discutido por [Sutskever, Vinyals e Le 2014]. Entre eles, destacava-se a dificuldade de lidar com a falta de correspondência direta entre a ordem dos elementos na entrada e na saída, especial-

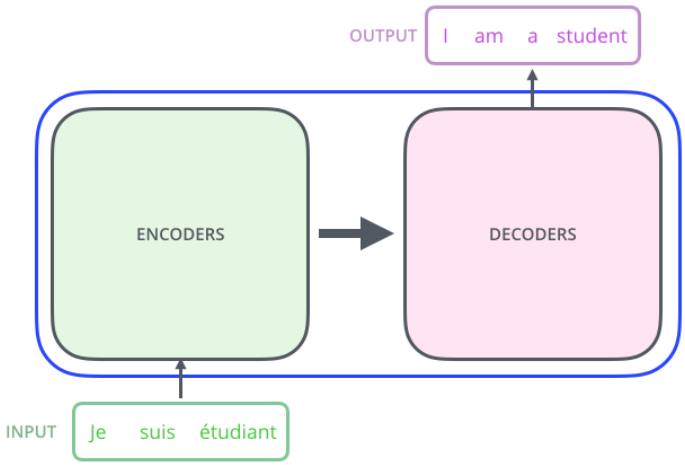


Figura B.6: *Representação em alto nível da arquitetura encoder-decoder, realizando uma tarefa de tradução automática. A entrada consiste em uma frase em francês, "Je suis étudiant", que é processada para produzir a saída traduzida em inglês, "I am a student" [Alammar 2025].*

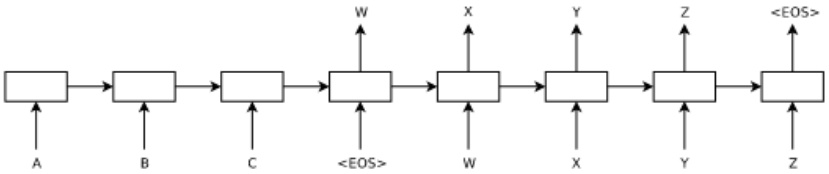


Figura B.7: *O modelo encoder-decoder proposto por [Sutskever, Vinyals e Le 2014] lia sequencialmente cada palavra de uma sequência de entrada "ABC" durante o treinamento para produzir uma sequência de saída "WXYZ".*

mente em tarefas de tradução, onde palavras frequentemente mudam de posição. Além disso, a otimização de sentenças longas era prejudicada por grandes defasagens temporais, enquanto a limitação de vocabulário impunha dificuldades ao lidar com palavras fora do vocabulário (*Out-of-Vocabulary*, OOV). Como solução parcial, reverter a ordem das palavras nas sentenças de entrada mostrou-se eficaz para facilitar a captura de dependências de curto prazo e melhorar a interação entre entrada e saída, conforme ilustrado na Figura B.7.

Para superar as limitações do processamento sequencial das RNNs e seus diversos problemas, foi desenvolvida a arquitetura *Transformer*, que elimina recorrências e adota uma abordagem baseada em mecanismos de atenção combinados com *feed-forward networks*, (FFN). Essa arquitetura permite maior paralelismo e elimina gargalos de desempenho ao conectar cada *decoder* diretamente a todos os *encoders*. A arquitetura do *Trans-*

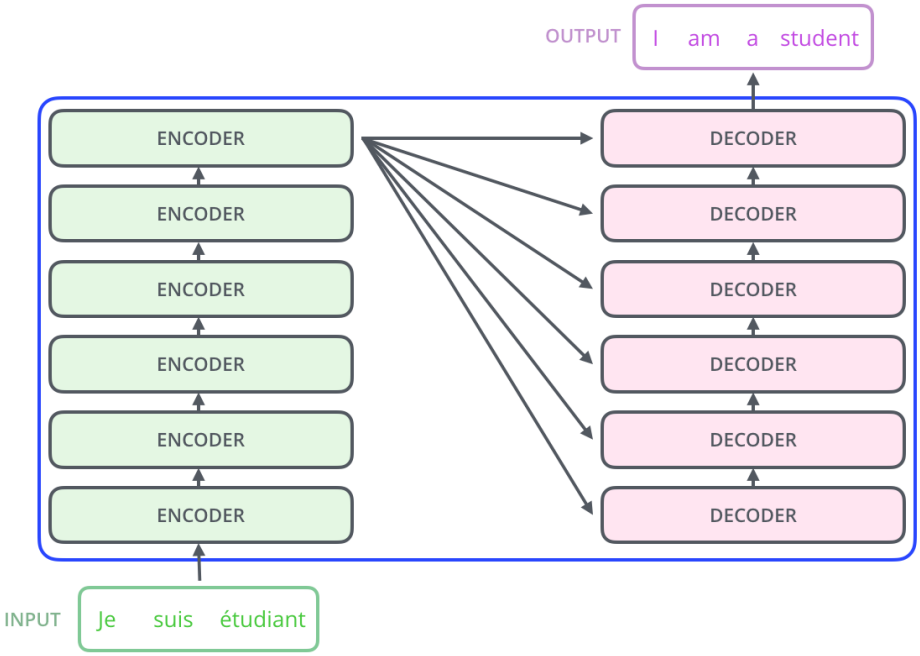


Figura B.8: *Detalhes internos do diagrama ilustrado na Figura B.6. O modelo Transformer consiste em uma pilha de encoders e uma pilha de decoders. Cada encoder processa a entrada e transmite suas representações aos decoders, permitindo uma tradução contextualizada [Alammar 2025].*

former é organizada em duas pilhas principais: uma de *encoders* e outra de *decoders*. Cada *encoder* transforma a entrada em representações abstratas, que são compartilhadas por todos os *decoders*, possibilitando a geração de saídas altamente contextualizadas. Esse *design* elimina a necessidade de processamento sequencial, típico de modelos como as RNNs, viabilizando uma paralelização eficiente. A Figura B.8 ilustra essa estrutura.

Diferentemente de modelos anteriores, como o *Word2Vec*, os *transformers* baseiam-se nos *mecanismos de atenção* [Vaswani et al. 2017], que permitem que o modelo foque em diferentes partes de uma sequência de entrada ao processar informações. Essa abordagem não apenas melhora a captura de dependências contextuais em diferentes níveis de granularidade, mas também expande significativamente as possibilidades de uso de *embeddings*. Por exemplo, enquanto o Word2Vec geraria a mesma representação vetorial para a palavra *banco* tanto no contexto de "sentar no *banco* da praça" quanto no de "abrir uma conta no *banco*", os *transformers* conseguem ajustar o *embedding* da palavra de acordo com o contexto em que ela aparece, preservando nuances semânticas essenciais para tarefas complexas de compreensão textual.

Detalhando a estrutura *encoder-decoder* apresentada na Figura B.6, cada *encoder* possui duas subcamadas: uma camada de autoatenção, seguida de uma rede neural *feedforward* (FFN). A saída de cada *encoder* é passada para o *decoder*, que contém três

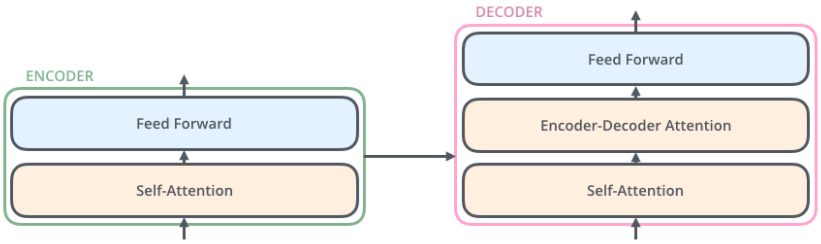


Figura B.9: *Estrutura das subcamadas do encoder e do decoder. Ambos contêm camadas de autoatenção seguidas por uma rede feedforward (FFN). A principal diferença é que o decoder inclui uma camada intermediária de atenção cruzada (encoder-decoder attention), que processa informações provenientes do encoder. [Alammar 2025]*

subcamadas: uma camada de autoatenção, seguida por uma camada de atenção cruzada (*encoder-decoder attention*) e, por fim, uma camada FFN. A Figura 3.13 ilustra essa estrutura.

Cada unidade que compõe as camadas de atenção pode ser expressa pela seguinte função matemática, denominada *Atenção por Produto Escalar Normalizado* (*Scaled Dot-Product Attention*), conforme ilustrado na Figura B.10:

$$\text{Atenção}(Q,K,V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Nesta equação, os principais componentes do mecanismo de atenção são as matrizes Q, K e V, que desempenham papéis distintos na modelagem das relações entre os elementos da entrada:

- **Matriz de consultas** (Q): Representa os vetores de consulta (*queries*), que correspondem a elementos que buscam informações relevantes dentro do mecanismo de atenção. Cada linha de Q é um vetor de dimensão d_k que define uma consulta específica no espaço latente do modelo.
- **Matriz de chaves** (K): Contém os vetores de chave (*keys*), que funcionam como identificadores de informações relevantes. Cada linha de K é um vetor de dimensão d_k, e a similaridade entre Q e K define a importância relativa dos elementos no contexto da atenção.
- **Matriz de valores** (V): Armazena os vetores de valor (*values*), que contêm a informação associada às respectivas chaves K. Cada linha de V é um vetor de dimensão d_v, e a combinação ponderada desses valores, definida pelos pesos de atenção, compõe a saída do mecanismo de atenção.

O mecanismo de atenção é definido pelas seguintes operações matemáticas:

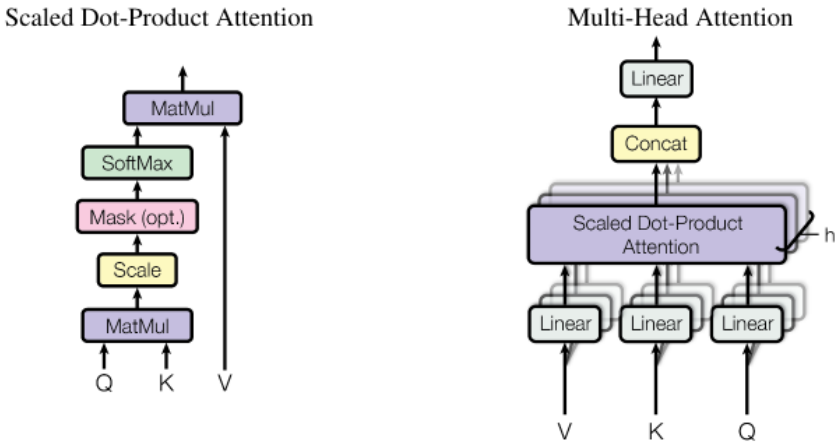


Figura B.10: *Mecanismo de Atenção por Produto Escalar Normalizado: calcula as similaridades entre a matriz de consultas Q e a matriz de chaves K, gerando pesos de atenção que são aplicados à matriz de valores V. Cada mecanismo de atenção é denominado cabeça de atenção, e múltiplas cabeças podem ser concatenadas para formar a Atenção Multi-Cabeça (Multi-Head Attention) [Vaswani et al. 2017].*

1. O produto escalar entre Q e K, dado por $QK^\top$, calcula a similaridade entre as consultas e as chaves.
2. A normalização pela raiz quadrada da dimensão d_k, $\frac{QK^\top}{\sqrt{d_k}}$, evita que os valores dos escores de atenção tenham variâncias muito grandes, estabilizando os gradientes durante o treinamento.
3. A função *softmax* é aplicada aos escores normalizados para gerar uma distribuição de probabilidade sobre as chaves, determinando os pesos de atenção.
4. A multiplicação desses pesos pela matriz V resulta na saída ponderada, destacando os elementos mais relevantes para cada consulta.

Esse mecanismo permite que o modelo aprenda a associar diferentes elementos de uma sequência, capturando dependências contextuais de maneira eficiente.

A Figura B.11 ilustra o mecanismo de *autoatenção*, uma técnica fundamental nos modelos baseados em Transformers, que viabiliza o *aprendizado autossupervisionado* (*self-supervised learning*) de relações entre palavras em uma sequência. O termo "autoatenção" refere-se ao fato de que a atenção é calculada internamente na mesma sequência de entrada, isto é, cada palavra avalia sua relação com todas as outras palavras (incluindo ela mesma) dentro do mesmo conjunto de representações. O funcionamento envolve a aplicação de *transformações lineares* às representações das palavras, gerando as projeções de consulta (Q), chave (K) e valor (V) a partir de matrizes de pesos (W_q, W_k, W_v). As similaridades entre Q e K são computadas, normalizadas pelo *softmax*, resultando no vetor de pesos (α), que reflete a importância relativa de cada palavra no contexto da

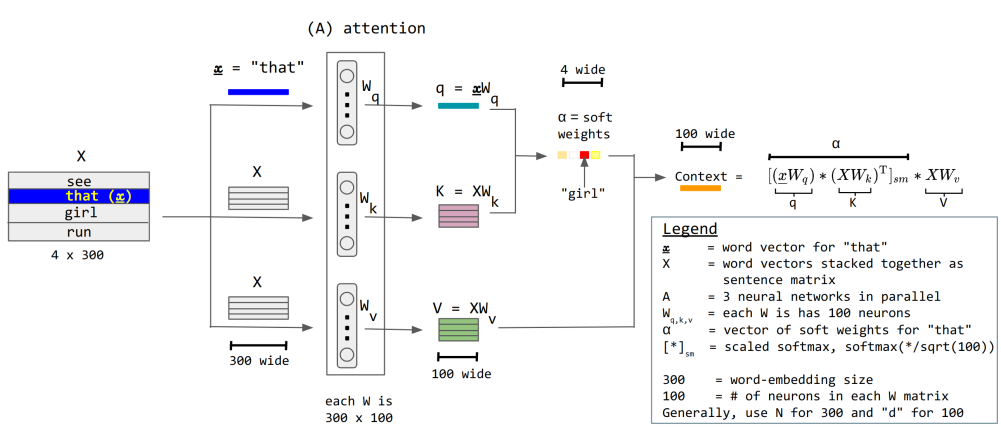


Figura B.11: *Ilustração simplificada do cálculo da função de atenção. A figura demonstra como o vetor de atenção* α *é gerado para a palavra "that"no contexto da frase "See that girl run". A matriz X contém os vetores de palavras da frase, enquanto as projeções de consulta (query), chave (key) e valor (value) são obtidas a partir das matrizes de pesos* W_q*,* W_k *e* W_v*. O vetor* α *destaca as probabilidades de atenção, com maior ênfase em "that"e "girl", refletindo a relação semântica entre essas palavras [Numiri 2023].*

palavra-alvo. Esse vetor de pesos é então utilizado para combinar V, formando o vetor de contexto. Todo o processo é treinado via *backpropagation*, ajustando automaticamente as matrizes de pesos para que o modelo aprenda as dependências semânticas entre palavras em diferentes posições da sequência.

Conforme ilustrado na Figura B.10, o transformer usa *Atenção Multi-Cabeça* (*Multi-head attention*), que expande a atenção por produto escalar normalizado ao aplicar múltiplas cabeças de atenção em paralelo, permitindo capturar diferentes padrões dentro da sequência de entrada. Sua formulação é dada por:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O$$

onde cada cabeça de atenção é computada como:

$$\text{head}_i = \text{Atenção}(QW_i^Q, KW_i^K, VW_i^V)$$

com W_i^Q, W_i^K, W_i^V sendo matrizes de projeção aprendidas. As saídas das cabeças são concatenadas e transformadas por W^O, retornando à dimensão original d_{model}. Esse mecanismo permite que cada cabeça aprenda diferentes aspectos das relações entre *tokens*.

Uma vez compreendidos os fundamentos do mecanismo de atenção, torna-se mais fácil entender outras variantes desse mecanismo na arquitetura do modelo *Trans-*

former, conforme ilustrado na Figura B.12. No diagrama, além da *autoatenção* no codificador, o decodificador apresenta duas variantes do mecanismo de atenção: a *atenção mascarada* (*Masked Attention*), que atribui um valor mascarado (por exemplo, infinito negativo, $-\infty$) aos valores calculados para os *tokens* subsequentes na matriz de autoatenção (QK^T). Isso gera uma matriz triangular inferior, garantindo que a predição de um *token* dependa apenas dos *tokens* anteriores.

A segunda variante é a *atenção cruzada* (*Cross-Attention*), representada pelo bloco *Multi-Head Attention* do decodificador. Nessa abordagem, as chaves (K) e os valores (V) são extraídos do codificador, enquanto as consultas (Q) são geradas pelo próprio decodificador, permitindo que este consulte informações do codificador.

Além dos três mecanismos de atenção mencionados, outras técnicas foram empregadas na arquitetura do *Transformer* para solucionar desafios enfrentados pelas RNNs. Como o processamento no *Transformer* é altamente paralelizado, a ordem dos *tokens* na sequência de entrada não é preservada naturalmente. Para mitigar esse problema, foram introduzidas as *codificações posicionais* (*Positional Encodings*) , que consistem em *embeddings* específicos para sinalizar a posição dos *tokens*.

Outra técnica essencial são as *conexões residuais* (*Residual Connections*), que somam a entrada de uma camada (como atenção multi-cabeça ou FFN) à sua saída, seguida de um processo de normalização, representado pelo bloco *Add & Norm*. Segundo os autores, essa abordagem evita o problema do esvanecimento do gradiente e acelera o treinamento. Para mais detalhes sobre essas técnicas, consulte [Vaswani et al. 2017].

A evolução dos modelos de linguagem foi significativamente impulsionada pela introdução do *Transformer* [Vaswani et al. 2017], que substituiu arquiteturas recorrentes por mecanismos de atenção, permitindo treinar modelos mais eficientes e escaláveis. Essa inovação levou ao desenvolvimento de *Large Language Models* (LLMs), como *BERT* [Devlin et al. 2019] e *GPT* [Radford et al. 2018], que exploraram diferentes estratégias de treinamento: *BERT*, por exemplo, utilizou aprendizado bidirecional e pré-treinamento baseado em mascaramento de palavras, enquanto os modelos da família *GPT* seguiram um paradigma *autoregressivo*, prevendo tokens sequencialmente.

A evolução do *GPT* culminou no *GPT-3* [Brown et al. 2020], que foi posteriormente refinado pelo *Reinforcement Learning with Human Feedback* (RLHF) [Ouyang et al. 2022], técnica na qual anotadores humanos ajudaram a alinhar as respostas do modelo com as expectativas dos usuários. Além disso, o uso de *prompting* tornou-se um elemento central na interação com esses modelos, permitindo que usuários moldassem as saídas do sistema por meio de instruções específicas. Essa abordagem, aliada ao RLHF, resultou no *ChatGPT*, um modelo mais seguro e alinhado às necessidades interativas, estabelecendo um novo padrão para assistentes de *IA* baseados em linguagem natural.

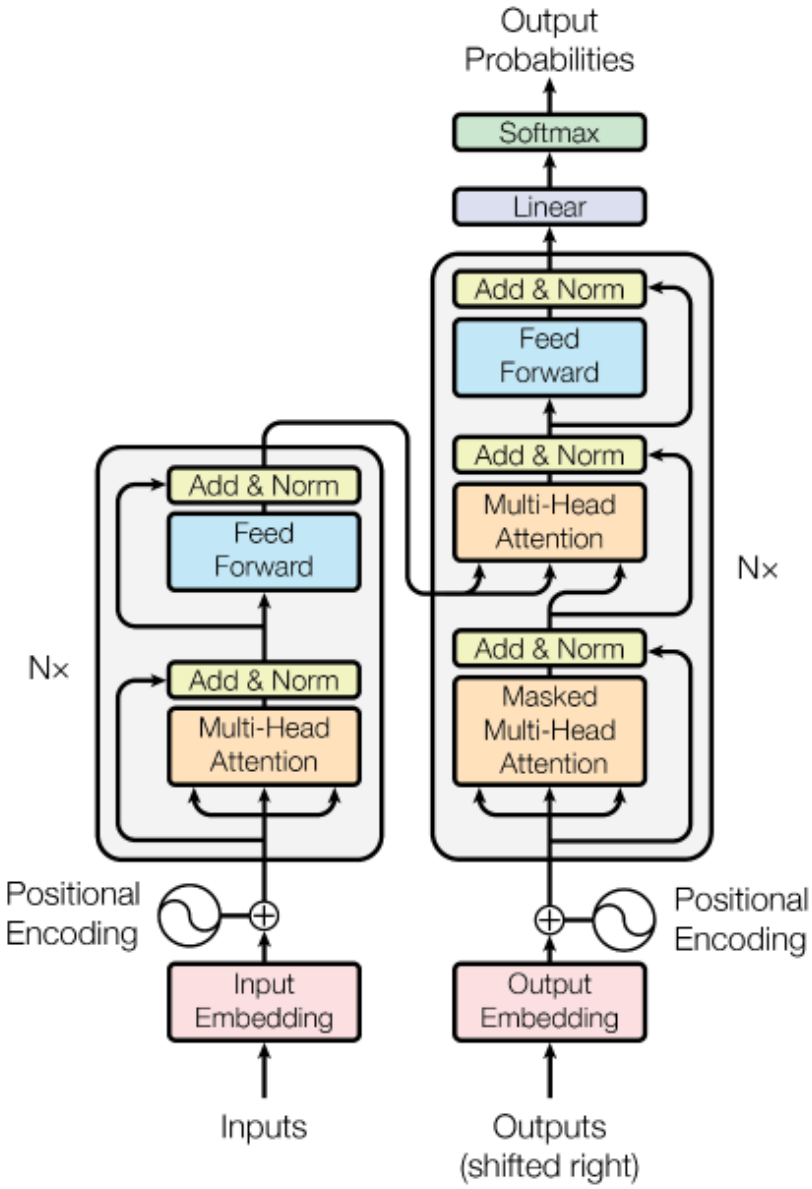


Figura B.12: *Diagrama da arquitetura do modelo Transformer, ilustrando o fluxo de dados entre o codificador (esquerda) e o decodificador (direita). As variáveis-chave incluem as entradas (X), processadas por Input Embedding e Positional Encoding, e as saídas (Y), deslocadas à direita e passadas por Output Embedding e Positional Encoding, com as probabilidades de saída ($P(Y|X)$) geradas após as camadas Linear e Softmax. Os principais componentes incluem atenção multi-cabeça (Multi-Head Attention) para captura de relações entre tokens, atenção multi-cabeça mascarada (Masked Multi-Head Attention) no decodificador para garantir autoregressividade, camadas feed-forward (Feed Forward) para transformação dos embeddings, e Add & Norm para normalização e conexões residuais. O fluxo de dados entre codificador e decodificador ocorre na camada de Multi-Head Attention do decodificador, onde a saída do codificador é utilizada para condicionar a geração das saídas.*

APÊNDICE C

Construções de Argumentos por Indicadores Discursivos

C.1 Estruturas argumentativas de uma sentença

A Tabela C.1 apresenta construções argumentativas organizadas segundo três padrões de estruturação com indicadores discursivos: **PIC** (Premissa leva ao Indicador, que leva à Conclusão — P $\rightarrow$ I $\rightarrow$ C), **CIP** (Conclusão leva ao Indicador, que leva à Premissa — C $\rightarrow$ I $\rightarrow$ P) e **IPC** (Indicador com Premissa leva à Conclusão — I, P $\rightarrow$ C). O padrão **ICP** (Indicador leva à Conclusão, que leva à Premissa — I $\rightarrow$ C $\rightarrow$ P) não está incluído na tabela por conter apenas dois casos recorrentes: *Here is why C: P.* e *In support of C, P.*

C.2 Estruturas argumentativas de duas sentenças

A Tabela C.2 apresenta construções argumentativas formadas por duas sentenças, organizadas segundo dois padrões: **P.IC** (Premissa com Indicador leva à Conclusão) e **C.IP** (Conclusão com Indicador leva à Premissa).

PIC (P → I → C)	CIP (C → I → P)	IPC (I, P → C)
P can cause C.	C, as indicated by P.	As indicated by P, C.
P demonstrates that C.	C, assuming that P.	As shown from P, C.
P guarantees that C.	C, because P.	Assuming that P, C.
P implies that C.	C can be derived from P.	Because P, C.
P indicates that C.	C, considering P.	Convinced by the fact that P, C.
P justifies that C.	C, due to P.	Due to P, C.
P proves that C.	C, due to the reason that P.	Due to the reason that P, C.
P signifies that C.	C, follows from P.	Granted that P, C.
P suggested that C.	C, for the reason that P.	In fact that P, C.
P, consequently C.	C, giving that P.	In light of the fact that P, C.
P, entails that C.	`C, if P.`	In view of the fact that P, C.
P, from which it follows C.	C, in view of the fact that P.	Inasmuch as P, C.
P, in other words, C.	C, insofar as P.	Now that P, C.
P, in that case C.	C is based on P.	On account of the fact P, C.
P, indicating that C.	C is supported by P.	On account of the reason that P, C.
P, indicating that C.	C may be deduced from P.	On the basis of P, C.
P, means that C.	C may be derived from P.	On the grounds that P, C.
P, resulting in C.	C may be inferred from P.	On the hypothesis that P, C.
P, shows that C.	C, on account of the fact P.	Owing to P, C.
P, so that C.	C, on account of the reason that P.	Seeing that P, C.
P, thereby showing that C.	C, on the basis of P.	Supposing that P, C.
P, therefore C.	C, on the grounds that P.	–
P, thus C.P establishes that C.	C, on the hypothesis that P.	–
P, wherefore C.	C, owing to P.	–
P, which allows us to infer C.	C, seeing that P.	–
P, which implies C.	C, since P.	–
P, which leads credence to C.	C, supposing that P.	–
P, which leads to C.	–	–
P, which points to C.	–	–
P, which shows that C.	–	–

Tabela C.1: *Estruturas argumentativas (construções) e seus indicadores discursivos*

P.IC (P. I → C)			C.IP (C. I → P)
P. Accordingly, C.	P. Obviously, C.	P. This proves that C.	C. Its proof is that P.
P. As a result, C.	P. On this account, C.	P. For this reason, C.	C. The reason is that P.
P. As conclusion, C.	P. One can conclude that C.	P. From this it follows that C.	C. This comes from P.
P. Evidently, C.	P. One can deduce that C.	P. From this we can deduce that C.	C. This is shown by P.
P. In conclusion, C.	P. One can infer that C.	P. Hence, C.	–
P. In consequence, C.	P. In short, C.	P. In sum, C.	–
P. In fact, C.	P. In view of that, C.	P. Indeed, C.	–
P. Therefore, C.	P. This is being so C.	–	–

Tabela C.2: *Estrutura de indicador para duas sentenças*

APÊNDICE D

Exemplos de Rótulos para Conclusões, Evidências e seus relacionamentos

Neste anexo, apresentaremos os exemplos contidos no guideline para anotação de conclusões, evidências e seus relacionamentos.

Exemplos de Conclusões Suportadas (SupportedClaim)

Tópico	The sale of violent video games to minors should be banned	
S1	Violent video games can increase children's aggression	**Claim**
S1 [com suporte (Claim)]	"Violent video games can increase children's aggression due to a recurrent and unrestricted exposition to gore and violence." Note que a segunda parte da sentença opera como uma justificativa, ou, para nossa aplicação, uma conclusão ou uma **Claim** secundária	**SupportedClaim**
S2	Video game publishers unethically train children in the use of weapons	**Claim**
S2 [com suporte (Event)]	"Video game publishers unethically train children in the use of weapons as it could be directly linked to the school shooting in Oregon last year." Neste caso, há a busca por um suporte em uma evidência anedótica, operando como um **Event** interno	**SupportedClaim**
S3	Violent games affect children positively.	**Claim**
S3 [com suporte (Qdat)]	"Violent games affect children positively because many studies present a significant reduction of violent tendencies on teenagers after being provided with a way to vent." Neste caso, repare que a sentença busca suporte em um dado quantitativo, semelhante a um **Qdat** interno	**SupportedClaim**
S4	"Violent video games can make violence seem banal."	**Claim**
S4 [com suporte (Def)]	"Violent video games can make violence seem banal as senseless utilization of certain content is the exact definition of banalization." Note que a segunda parte da sentença apresenta uma definição da banalidade apresentada na sentença original, assim, configurando um **Def** interno.	**SupportedClaim**

Tabela D.1: *Exemplos de Claims e suas justificativas internas com diferentes tipos de suporte*

Exemplo de rótulo Rebuttal

Claims com o rótulo de *Rebuttal* indicam refutações ou sentenças de contestação em que uma declaração inicial é colocada para, ainda na mesma sentença, ser impugnada ou desafiada por um argumento oposto.

Podemos observar no exemplo a seguir como o subtópico da "energia nuclear como fonte de energia limpa" declara que esta poderia ser uma conclusão válida, contudo, logo após temos um ponto contra essa possibilidade, o problema de que a energia nuclear ainda produziria lixo nuclear, contradizendo a possibilidade inicial.

> **(Claim [Rebuttal])** Nuclear power could be a source for cleaner energy **but** the production of nuclear waste counteracts this claim.

Tópico	Nuclear power as clean energy
Conclusão	could be a source for cleaner energy
Indicador	but
Conclusão contrária	the production of nuclear waste counteracts this claim

Tabela D.2: *Exemplo de claim com rótulo de Rebuttal*

APÊNDICE E

Co-dependência semântica

Através de elementos coesivos e até como premissa inicial de nossa pesquisa, afirmamos que as sentenças não ganham significação isoladamente, mas em contexto: desta maneira, as evidências e conclusões tornam-se co-dependentes semanticamente. Como uma continuação do conceito de correferência, que se apresenta como um elemento de conectividade formal (morfossintática e lexical) relacionada à coesão entre as sentenças, temos esta co-dependência como um elemento de coerência: constituído sobre conectividade de conteúdo (semântica e argumental).

Tome por exemplo os argumentos a seguir:

> **1a** So what is responsible for this disconnect between popular opinion and medical reality?
> **1b** A big part of it may have to do with the fact that marijuana today is much stronger than it was in previous generations.
> **1c** The average THC level in today's marijuana is approximately three times that of 1990, with some experts saying it's up to six times more potent. (Marijuana legalization/15)

Julgando as duas sentenças isoladamente, tendo em vista a ausência de uma colocação clara na sentença 1a, que apenas apresenta um questionamento sobre as motivações entre percepção pública e objetividade médica, não teríamos 1a como parte constituinte do argumento em 1a:c, sendo rotulado como apenas uma sentença visto que é avaliado como mero artifício retórico. Contudo, ao analisar o mecanismo de chamada e resposta (***question hook***) e encontrar um posicionamento claro em 1b e 1c, 1a retorna para o rol das premissas; em outras palavras: quando vistos em contexto, exemplos como 1a tornam-se constituintes de uma premissa devido ao seu contexto.

A dependência de sentidos demonstradas pela relação 1a→1b em nosso *dataset* é comumente rotulada como `group`, indicando este "sentido incompleto" em uma das sentenças isoladamente, apenas completando-o quando duas sentenças (ou mais) são vistas como um grupo.

E.1 Agrupamento por equivalência

Neste exemplo a seguir, faz-se o agrupamento de duas sentenças devido à sua equivalência de sentido: na prática, ambas as sentenças dizem, essencialmente, a mesma coisa.

Exemplo

Linguisticamente, sentenças como as duas a seguir, que possuem equivalência de sentido, são chamadas de paráfrase.

"The volume of death represented by abortion is staggering."
↔ **Group (paráfrase)** ↔
"Abortion is killing on a genocidal scale."

T	Large quantity of Abortions in the United States
I	Ø
Conclusão	staggering/genocidal scale (Is bad because is large)

E.2 Agrupamento por co-dependência

As codependências semânticas, como mencionado acima, resultam de uma incompletude de sentido de uma sentença isolada, fazendo com que ela busque um tópico ou uma conclusão em outra sentença para concluir seus sentidos. Devido a essa codependência, é bastante comum que sentenças codependentes criem agrupamentos devido às incompletudes de uma das sentenças envolvidas ou de ambas.

Exemplo

No exemplo a seguir temos uma sentença inicial que se basta, afirmando que o risco de uma mulher de ter câncer de mama (tópico) aumenta após um aborto (Qdat - dado/comentário sobre o tópico). No entanto, temos a segunda sentença afirmando que, caso este aborto seja feito na adolescência, os riscos aumentam: repare que, ainda que possamos inferir que "os riscos" se referem aos "riscos de câncer de mama", não temos esta construção autossuficiente na frase uma vez que o tópico completo não está presente.

Desta forma, a segunda aparição do tópico é dependente da primeira ao recuperá-la para especificá-la mais além. Desta maneira, a segunda sentença constrói uma relação de dependência da primeira e, portanto, formará um agrupamento.

(Qdat) A woman's risk of developing breast cancer is 50% higher if she had had an abortion.

↔ **Group (co-dependência)** ↔

(Qdat) If the abortion is done in teenage years then the risk was 100% higher.

Tópico	A woman's risk of developing breast cancer
Indicador	Ø
Qdat	50% higher if she had had an abortion
Tópico (repetição)	T + in teenage years
Qdat	then the risk was 100% higher.

E.3 Questões-gancho

Questões-gancho são construções com base em uma pergunta que funciona para enquadrar um assunto seguida por uma sentença que responde o tópico trazido à tona pela frase anterior.

Exemplo

Assim como podemos observar na questão-gancho a seguir, é possível observar como a questão, por si só, é capaz de resgatar o tópico-geral, sendo o uso da maconha, e coloca um tópico mais específico em voga: "a distorção na opinião pública (sobre o uso de maconha)". Entretanto, neste artifício retórico, apesar de criar-se a expectativa de uma conclusão, esta conclusão está ausente na pergunta, apenas sendo postulada na resposta ao afirmar que o tópico específico (distorção na opinião pública) deve-se a "retratação negativa pela mídia e ao fundo racial" relacionados ao tópico.

Desta forma, reconhecendo que a pergunta em si não declara uma conclusão, essa sentença sozinha fica deficiente e, ao mesmo tempo, ela serve como tópico para a sentença seguinte. Sendo assim, **questões-gancho**, por conta de suas relações de dependência semântica, formarão **agrupamentos**.

So what is responsible for this disconnect between popular opinion and medical reality?

↔ **Group (co-dependência)** ↔

The media's negative depiction and of it the paired with its racial background create a breeding ground for untruthful beliefs

Tópico	**Indicador**	**Conclusão**
... is responsible for this disconnect between popular opinion and medical reality	So what/?	Ø

Da mesma maneira, encontramos a conclusão na sentença 1b. Faz-se possível afirmar que a codependência consistiria na deficiência de **TÓPICO** ou da **CONCLUSÃO** em uma **ÚNICA** sentença, tornando-a dependente de uma outra onde recuperará o constituinte que falta. Delimitamos, em especial, duas configurações semânticas que necessitarão da relação de "group".

APÊNDICE F

Funcionalidades da Ferramenta Argmap

Neste anexo, são apresentadas capturas de tela com as principais funcionalidades da ferramenta Argmap que foi desenvolvida e utilizada para anotar mapas de argumentos provenientes de múltiplos documentos. Para mais detalhes, acesse o guia completo do usuário da ferramenta, como uma subseção do guideline [1].

F.1 Visualização de um documento

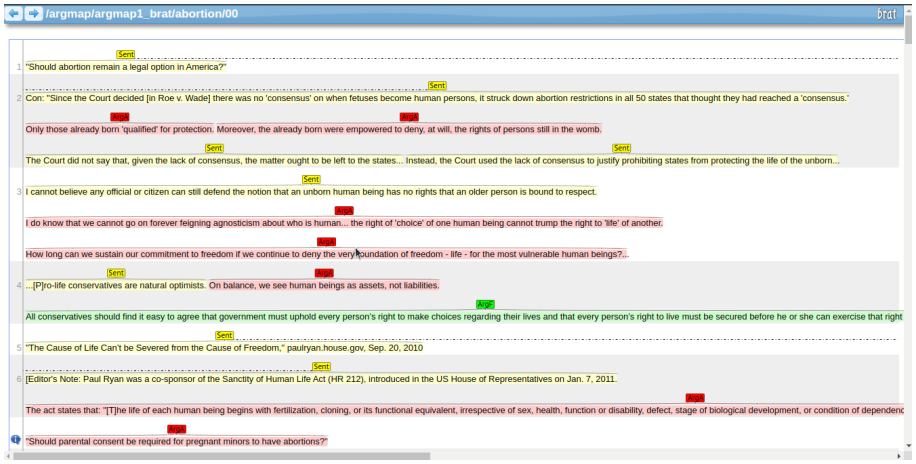


Figura F.1: *Visualização de documento aberto*

[1] Disponível em https://argmap.inf.ufg.br/guideline/tools/

F.2 Aplicação de relação entre UAs

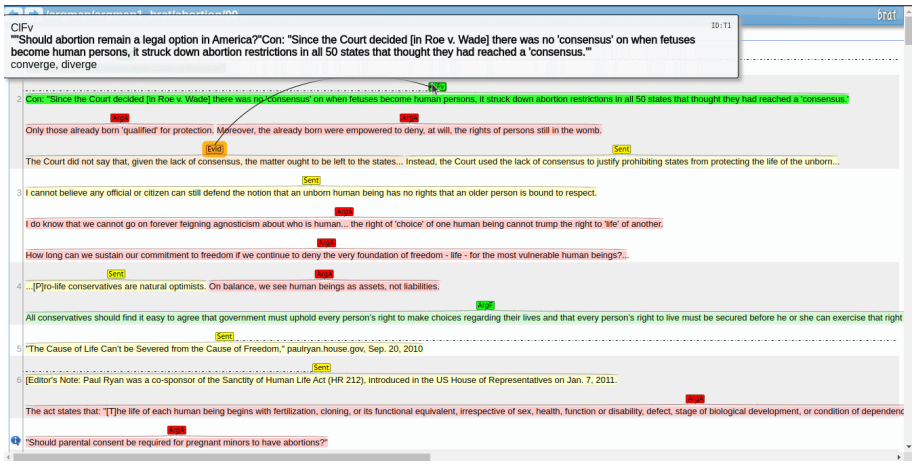


Figura F.2: *Aplicação de uma relação entre UAs*

F.3 Árvore de argumentos resultante

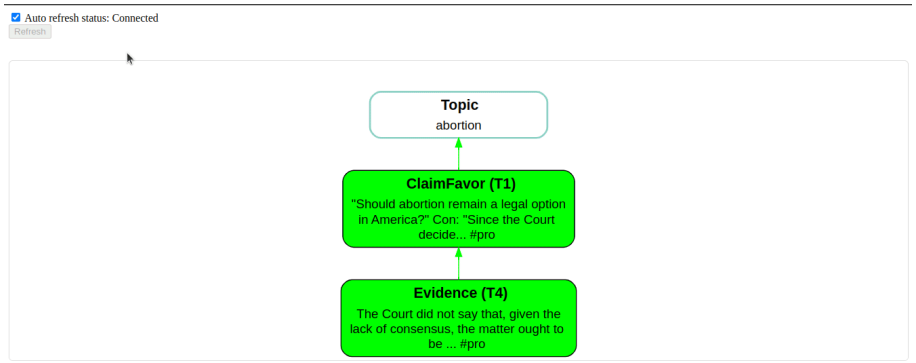


Figura F.3: *Visualização da árvore de argumentos resultante da anotação*

F.4 Visualização da árvore de argumentos em tempo real

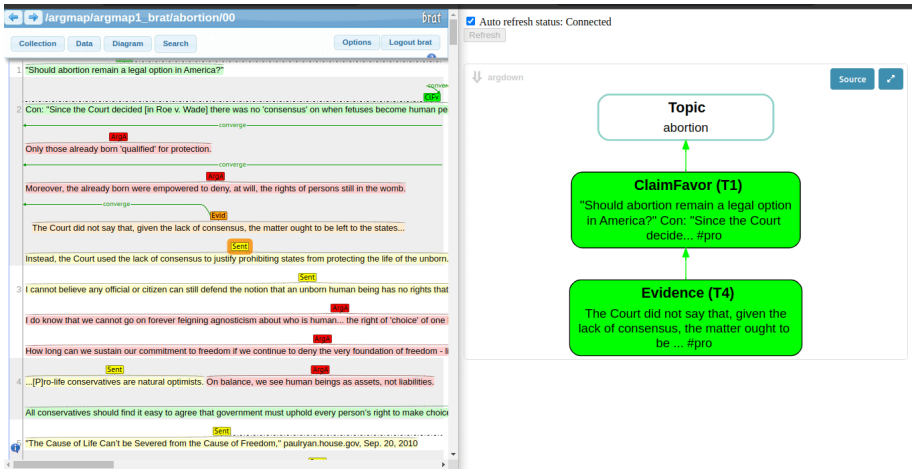


Figura F.4: *Visualização lado a lado da anotação e a árvore resultante em tempo real*

F.5 Anotação de relação entre evidencias e visualização da árvore de argumentos resultante

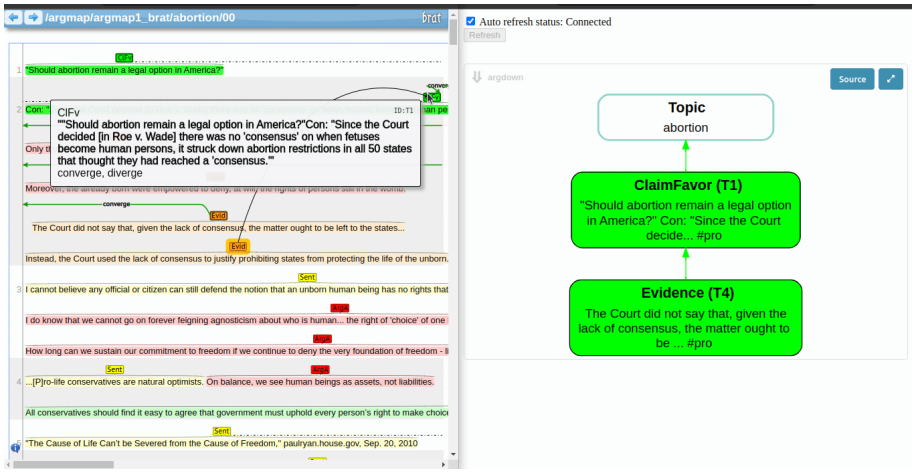


Figura F.5: *Seleção do argumento de origem e destino da relação*

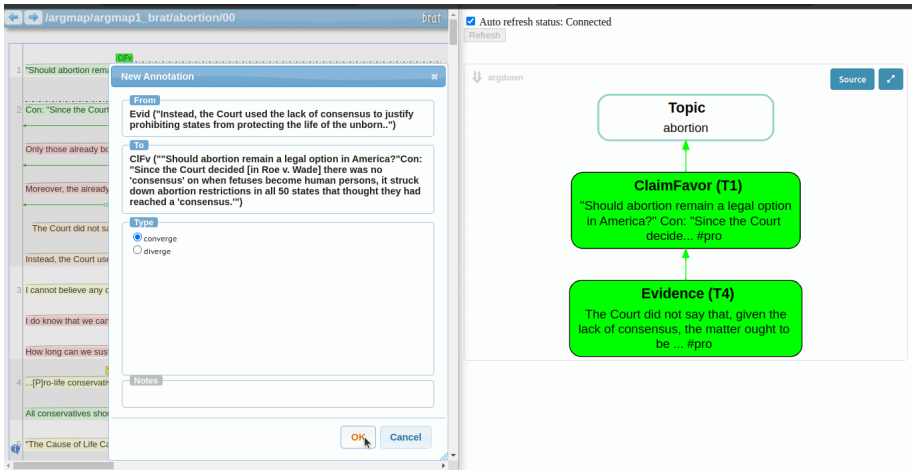


Figura F.6: *Seleção do tipo de relação entre os argumentos selecionados*

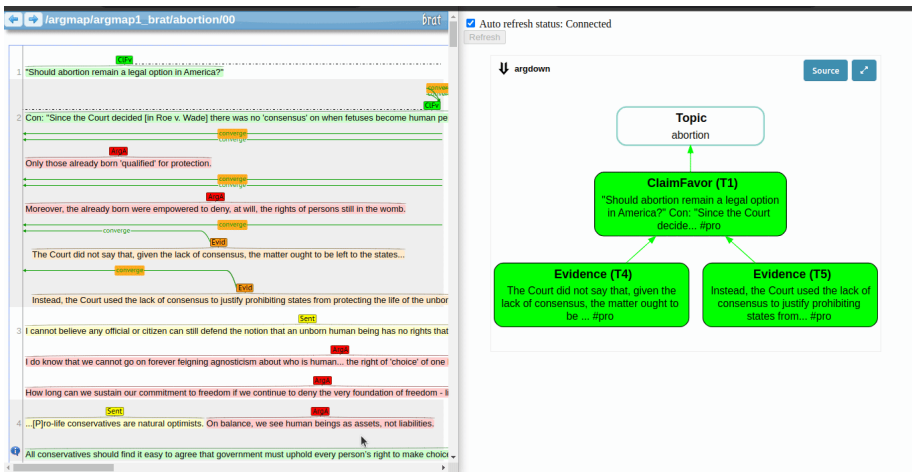


Figura F.7: *Visualização do resultado da anotação*

F.6 Visualização de sentenças agrupadas

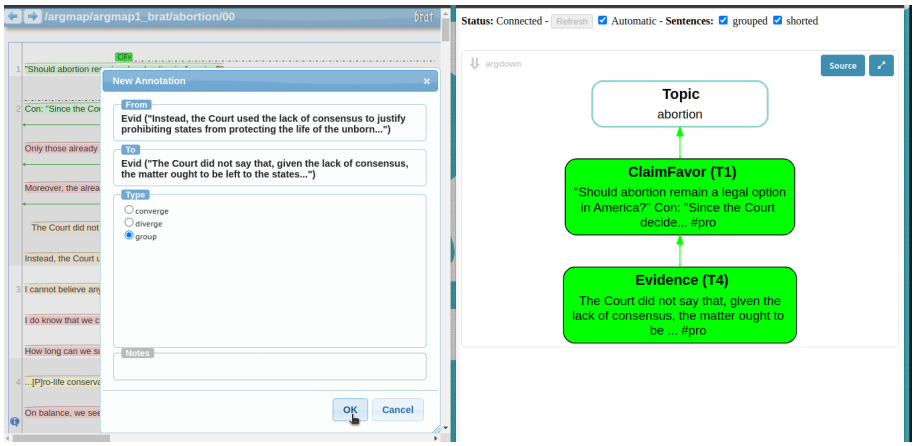


Figura F.8: *Agrupamento de argumentos*

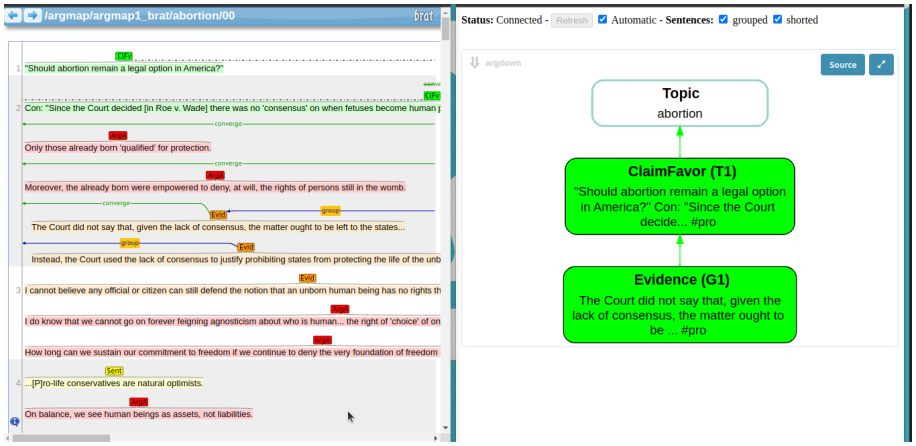


Figura F.9: *visualização dos argumentos agrupados na evidência G1*

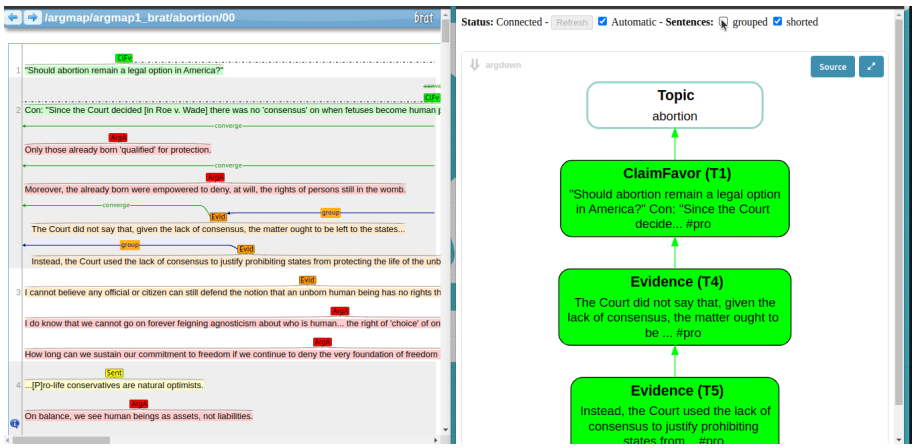


Figura F.10: *visualização dos argumentos de forma desagrupada*

F.7 Visualização de sentenças resumidas

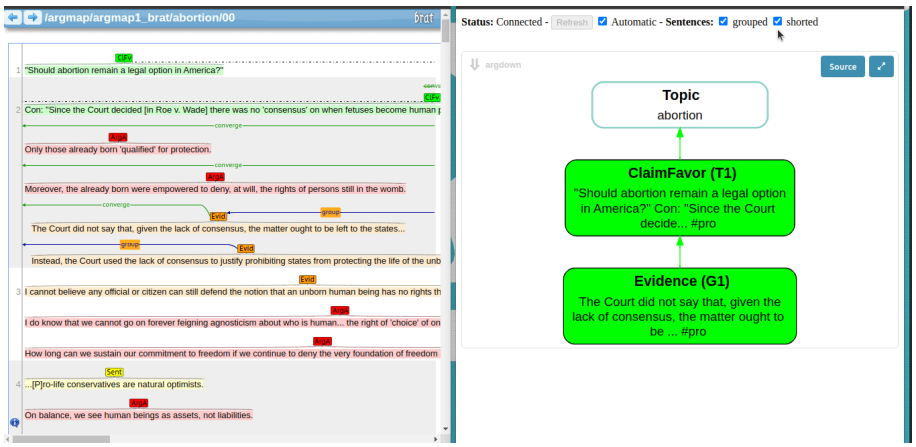


Figura F.11: *visualização da árvore de argumentos com sentenças resumidas*

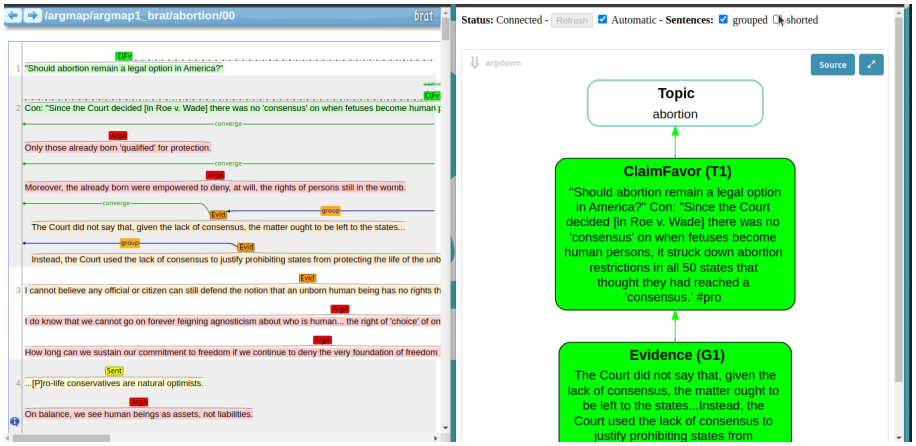


Figura F.12: *visualização da árvore de argumentos com o resumo de sentenças desabilitado*

F.8 Argmap Dashboard

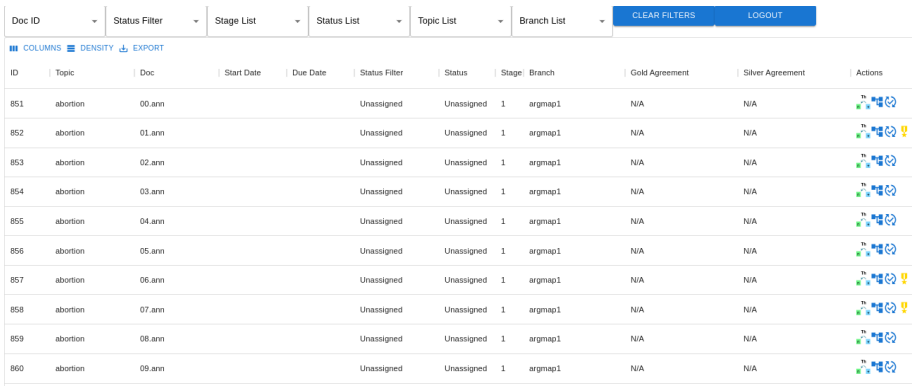


Figura F.13: *Argmap: Dashboard*

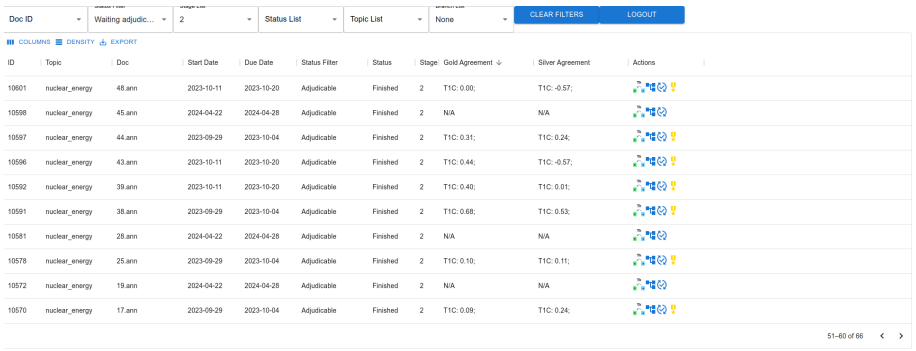


Figura F.14: *Argmap: Dashboard com agreements*

F.9 Argmap Outliner

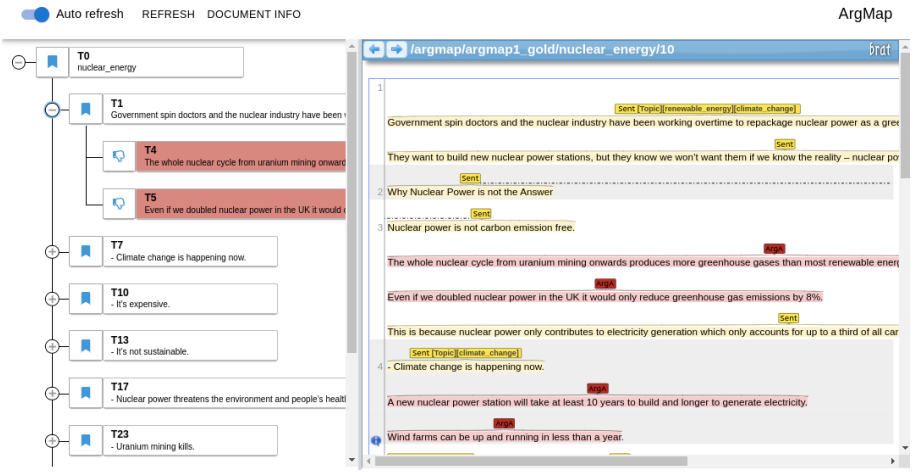


Figura F.15: *Tela de anotação no estágio Segmentação e Classificação de Tópicos*

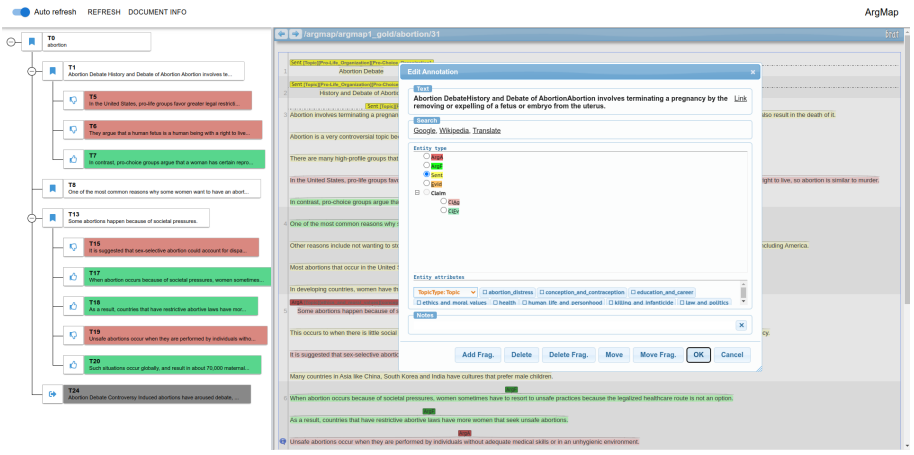


Figura F.16: *Classificação de tópicos sendo realizada*

F.10 Argmap Tree merging

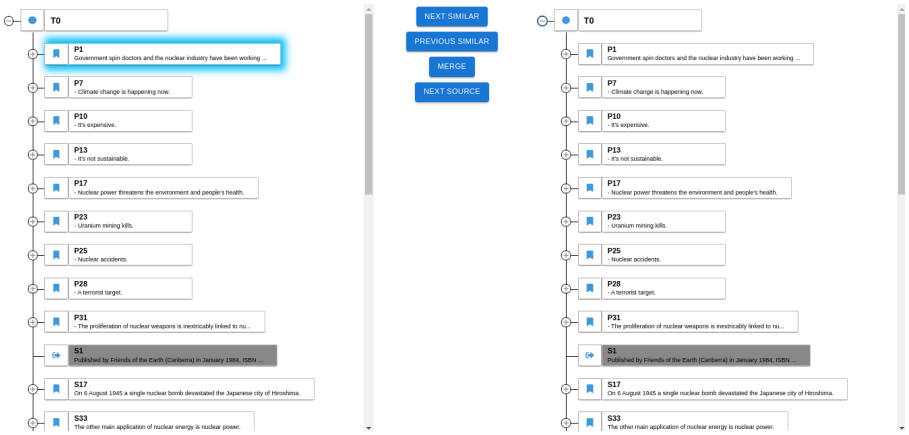


Figura F.17: *Tela de anotação para mesclagem de árvores de argumentos*

APÊNDICE G

Correferência

A correferência consiste na referência ao mesmo índice (pessoa ou entidade do texto) por duas ou mais frases nominais diferentes. Neste contexto, a referência a uma mesma "coisa" em duas circunstâncias no texto representa uma correferência, seja essa referência tecida por um substantivo, artigo, pronome ou expressão de antecedência (ver G) ou consecutividade (ver G).

A árvore sintática a seguir demonstra, em uma sentença, a funcionalidade comum de uma correferência:

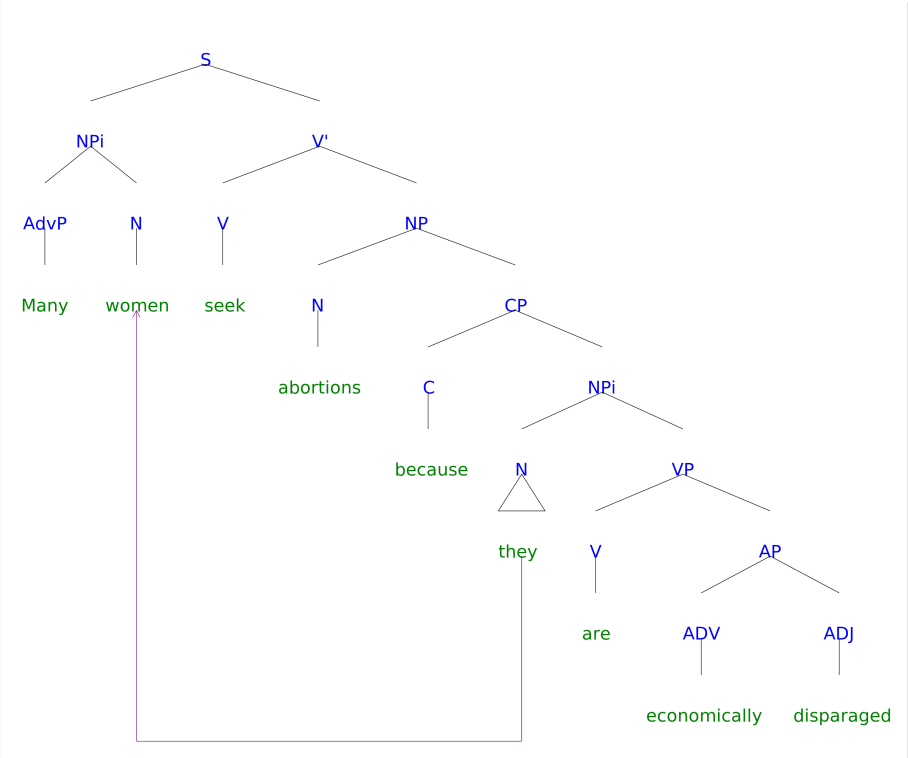


Figura G.1: *Árvore sintática demonstrando correferência*

Como exemplo, a sentença **S** com o sujeito "Many women" (muitas mulheres), se refere novamente a elas com o pronome **THEY**, fazendo de NPi (Many women) o índice da sentença e do pronome **THEY**, em NPi-N, um correferente de NPi.

Uma das consequências da correferencialidade, como pode ser visto, é a perda do sentido do correferente. No exemplo dado, como previsto, **THEY** perde completamente o sentido sem o NPi (Many women).

Outro aspecto comum das correferências é a concordância em número, gênero e pessoa, o que também pode ser visualizado no exemplo:

	Número	Gênero	Pessoa
Many women	plural	neutro	3ª pessoa
They	plural	neutro	3ª pessoa

Tabela G.1: *Concordância em correferência*

Anáfora

Correferências por anáfora se constituem como uma referência a um tema anterior no texto, como é possível ver abaixo:

> Unsafe abortions occur when they are performed by individuals without adequate medical skills or in an unhygienic environment. **SUCH** situations occur globally, and result in about 70,000 maternal deaths and 5 million maternal disabilities per year.

A expressão **SUCH** reintroduz o tema abordado na primeira sentença, voltando a se referir a "Unsafe abortions". Neste caso, "SUCH situations" é um correferente de "Unsafe abortions". Para testar a validade da correferência, substitua o índice pelo referente: note que ao trocar "SUCH situations" por "Unsafe abortions", o sentido da sentença é mantido.

Expressões comuns de antecedência incluem:

- As such
- Such as
- Per se
- And so
- As well

Mais exemplos podem ser vistos em https://en.wikipedia.org/wiki/Pro-form.

Antecedência

Antecedência, em gramática, são elementos que "ligam" orações seguintes a índices já mencionados.

Catáfora

Como contraparte da anáfora, expressões catafóricas antecipam um tema a ser abordado posteriormente:

> To obtain this visa, the worker has to comply with the **FOLLOWING** requirements: to hold a passport valid for more than 6 months.

Expressões comuns catafóricas:

•Following
•As follows
•Hence
•Thus
•In sequence
•Then

Formas Comuns de Correferência

Nominalização

> **POODLES** have become one of the most famous breeds through the past years. **ANIMALS** like these are always beloved by families due to their gentle nature.

Hiperônimos e Hipônimos:

•Hiperônimo: uma classe geral (ex: *Dog* abarca *Poodle*).
•Hipônimo: um item específico da classe (ex: *Poodle* é um hipônimo de *Dog*).

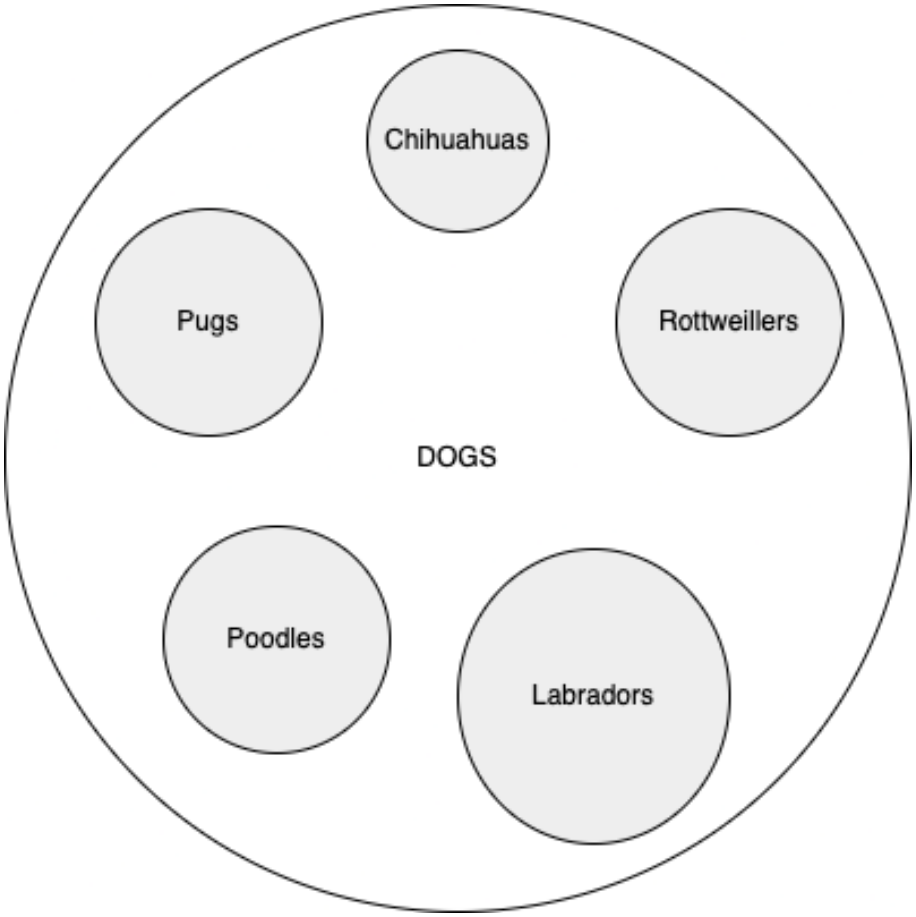


Figura G.2: *Relação de Hiperônimo e Hipônimo*

Pronominalização

In the United States, **PRO-LIFE GROUPS** favor greater legal restrictions on abortion. **THEY** argue that a human fetus is a human being with a right to live...

Também pode ocorrer com demonstrativos:

The truth: Hitler said that if you tell a lie often enough... **THIS** seems to be the tactic used by the pro death crowd...

Tabela de Marcadores

Pronomes de sujeito	I; You; He; She; It; We; They
Pronomes de objeto	Me; Him; Her; Us; Them
Demonstrativos	This; That; These; Those

Tabela G.2: *Pronomes comuns em correferência*

Marcadores Anafóricos	
Exemplificação	For example; As such; Such as; E.g
Adição	Moreover; Additionally; Also; Furthermore
Similaridade	Similarly; Like; Likewise
Ênfase	Indeed; Certainly; Undoubtedly
Retorno	Still; Nevertheless; Even so; Again

Marcadores Catafóricos	
Sequenciação	First; Next; Following; As follows; Below

Expressões

Via de regra, expressões anafóricas aparecem no fim de sentenças, referindo-se a anteriores. Já catafóricas tendem a aparecer no início, antecipando conteúdo posterior.

APÊNDICE H

Guideline para Anotação de Gênero Textual

Este anexo apresenta a guideline utilizada para anotação de gêneros textuais em documentos coletados da web. A taxonomia de gêneros a seguir foi elaborada com base na análise funcional, estrutura argumentativa e natureza da interação discursiva.

Esquema de Gêneros Textuais

Out of Scope

Textos que não apresentam estrutura argumentativa suficiente para análise, sendo considerados fora do escopo. Exemplos:

- **announcement**: anúncios de produtos, serviços, livros etc.
- **index or aggregator**: páginas com índices de resumos ou links para outros conteúdos.
- **presentation**: slides ou apresentações visuais (ex.: PowerPoint, Slideshare).

Monological

Textos opinativos produzidos por um único autor ou grupo com objetivo persuasivo. Caracterizam-se por um ponto de vista predominante e uso de recursos subjetivos.

- **blog post or editorial**: textos publicados em blogs ou editoriais, frequentemente em primeira pessoa, com presença de adjetivos e argumentos subjetivos.
- **talk transcription**: transcrição de falas ou palestras organizadas em perguntas ou tópicos; marcadas por oralidade e ausência de debate direto.
- **manifesto (opcional)**: conteúdo opinativo de indivíduos ou coletivos que demandam ação, geralmente em tom apelativo e em primeira pessoa.

Dialogical

Textos que apresentam mais de um ponto de vista, caracterizando uma interação dialógica entre autores.

- **debate**: transcrição de discussões entre dois ou mais autores, podendo ou não ter mediação.
- **critical review or technical report**: textos baseados em discurso indireto e tipologia expositiva, com seções críticas ou técnicas. Incluem críticas, pareceres técnicos, decisões jurídicas ou cartas ao editor.

Irregular

Gêneros híbridos ou compostos por múltiplas vozes argumentativas, comuns em contextos digitais e de mídia.

- **threaded posts**: sequências de postagens (ex. Twitter), geralmente iniciadas por um único autor, com possibilidade de respostas.
- **interview**: entrevistas estruturadas em turnos de fala extensos por parte do entrevistado, com interposição de perguntas.
- **clipping**: coletâneas de trechos de fontes diversas, caracterizadas pelo uso de discurso indireto.
- **term**: verbetes de enciclopédias ou dicionários online (ex. Wikipedia), com linguagem expositiva e estrutura ontológica.
- **tutorial or guide**: textos instrutivos que organizam argumentos de forma unilateral (ex. guias de refutação) ou pareada (comparativos).
- **news**: textos jornalísticos não-opinativos, com foco em apresentação de fatos, dados e eventos recentes.

Observações Metodológicas

- Em casos de textos híbridos, que combinem mais de um gênero textual (ex.: *blog post* seguido de *threaded post*), a classificação deve considerar o gênero predominante em termos de volume informacional.
- A predominância é determinada pela contagem de palavras ou pela extensão do conteúdo textual de cada segmento.

APÊNDICE I

Funcionalidades da Ferramenta KPCTool

Este anexo apresenta capturas de tela que ilustram as funcionalidades da ferramenta Keyphrase Curation Tool (KPCTool), que é utilizada para a anotação da tarefa de curadoria de frases-chave.

I.1 Agrupamento de Frases-Chave

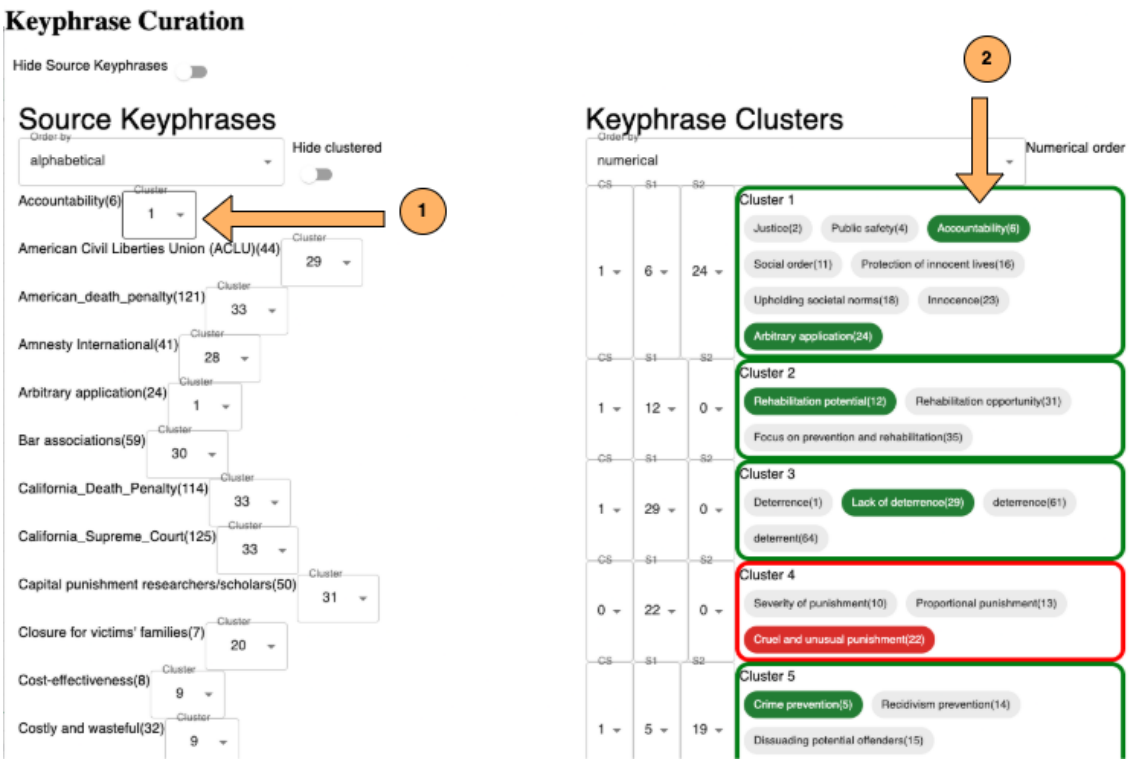


Figura I.1: *Agrupamento de frase-chave: (1) no lado esquerdo da figura, um identificador de cluster é selecionado. (2) no lado direito da figura, a frase-chave é incluída no respectivo cluster do lado direito*

I.2 Seleção de Cluster

Figura I.2: *Seleção de cluster: (1) no lado direito da figura, um cluster é selecionado com valor 1 na caixa de seleção. (2) no lado direito da figura, o cluster selecionado muda para borda verde*

I.3 Seleção de Frase-chave

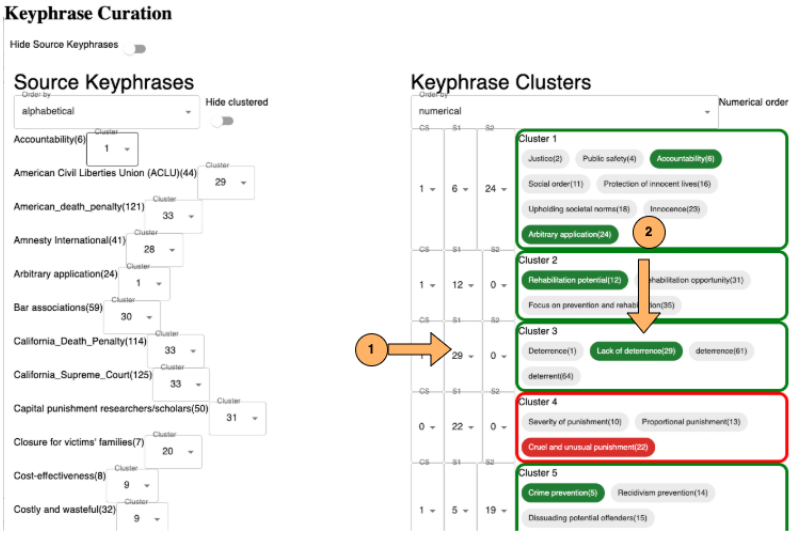


Figura I.3: *Seleção da frase-chave no cluster: (1) no lado direito da figura, uma frase-chave é selecionada informando o seu número identificador na caixa de seleção. (2) no lado direito da figura, a frase-chave é destacada na cor verde*

I.4 Ordenação por Similaridade de *Cluster*

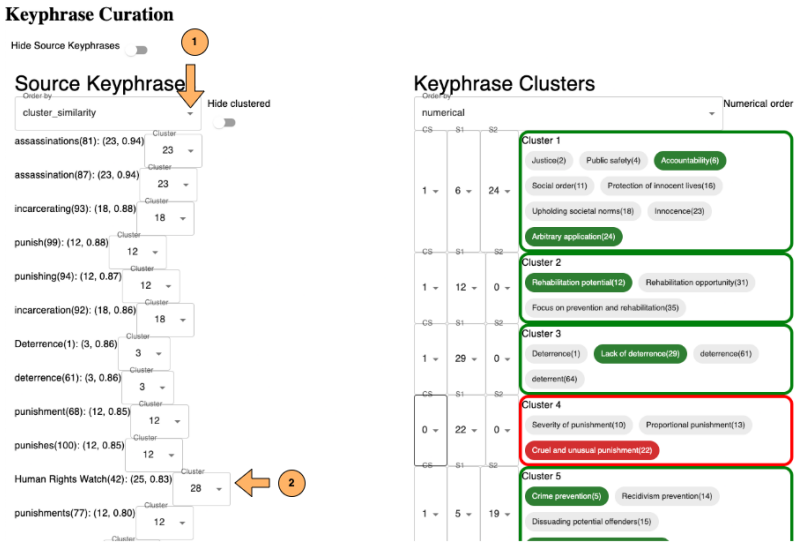


Figura I.4: *Ordenação por similaridade de cluster: (1) no lado esquerdo da figura, a opção "cluster_similarity"da caixa de seleção é selecionada. (2) é calculada a similaridade de cada frase-chave da lista do lado esquerdo com cada cluster do lado direito, sendo que o mais similar é sugerido juntamente com seu grau de similaridade; pode-se selecionar o cluster recomendado ou não, no exemplo, o cluster sugerido é o 25, com grau de similaridade* 0,83*, mas o usuário preferiu incluir a frase-chave no cluster 28*

Na ordenação da Figura I.4, as frases-chave são listadas em ordem decrescente de similaridade. Por exemplo, a primeira frase-chave listada é "assassinations(81): (23,0.94)", ou seja, a frase-chave ‘assassinations‘ com identificador 81 tem maior similaridade com o cluster de número 23, com grau 0,94 (em uma escala que vai de 0 a 1).

I.5 Ordenação por Similaridade em pares

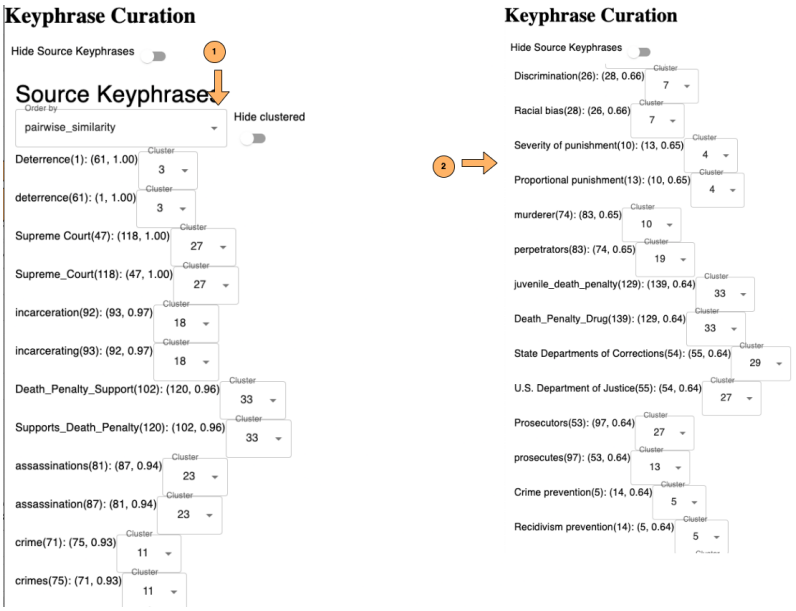


Figura I.5: *Ordenação por similaridade em par: (1) no lado esquerdo da figura, a opção "pairwise_similarity"da caixa de seleção é selecionada. (2) é calculada a similaridade de cada frase-chave da lista do lado esquerdo em pares, sendo que os pares mais similares são sugeridos, juntamente com seu grau de similaridade entre eles. Na imagem, a lista foi rolada para baixo para se visualizar os pares menos similares.*

Na ordenação da Figura I.5, as frases-chave são listadas em pares, em ordem decrescente de similaridade. Por exemplo, o primeiro par de frases-chave da lista à esquerda é "Deterrence(1): (61, 1.00)"e "deterrence(61): (1, 1.00)", ou seja, as duas possuem similaridade 1.00, pois são semanticamente idênticas. A segunda lista mostra pares com menos grau de similaridade, pois foi rolada para baixo. A seta aponta um par "Severity of punishment(10): (13, 0.65)"e "Proportional punishment(13): (10, 0.65)", ou seja, elas são similares em grau 0.65 e ambas foram escolhidas para estarem no mesmo cluster com identificador igual a 4.

I.6 Ordenação por Coesão de Cluster

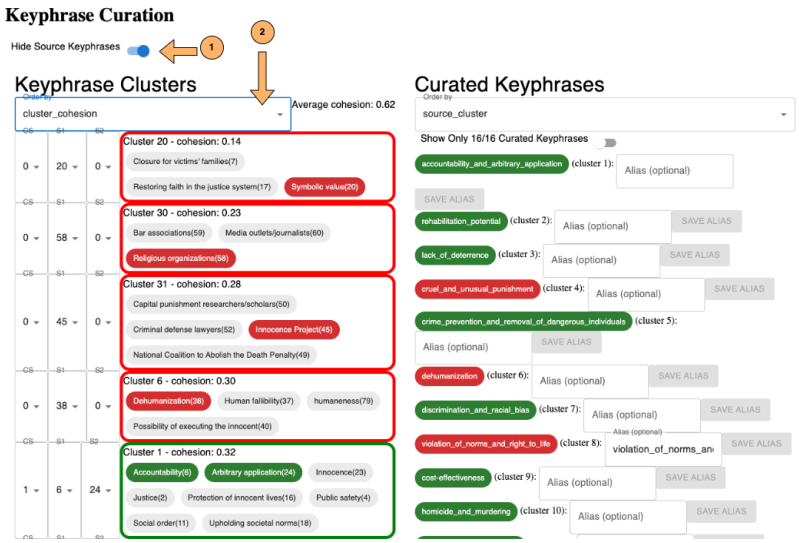


Figura I.6: *Ordenação por coesão de cluster: (1) no lado esquerdo da figura, a chave "Hide Source Keyphrases"é selecionada e por isso, a lista de keyphrases originais são escondidas, a lista de clusters passa para o lado esquerdo da tela e aparece a lista de frases-chave curadas no lado direito da tela. (2) Ao clicar a opção "cluster_cohesion"na caixa de seleção, a lista de agrupamentos é listada em ordem crescente de grau de coesão.*

Na ordenação da Figura I.6, os agrupamentos são listados em ordem crescente de coesão. Por exemplo, o primeiro cluster identificado com o número 20 possui grau de coesão 0.14, sendo que é o que possui menos coesão, ou seja, as frases-chave são pouco similares entre si.

I.7 Ordenação por Similaridade entre Pares de Agrupamentos

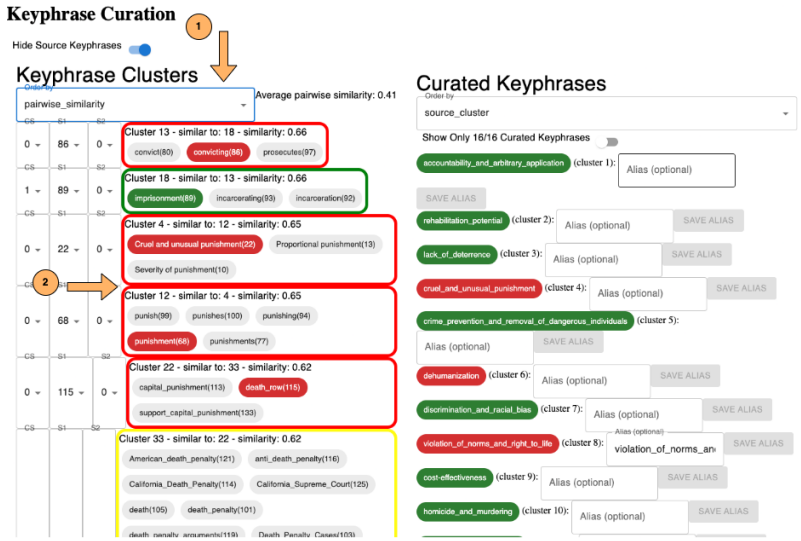


Figura I.7: *Ordenação por similaridade entre pares de agrupamentos: (1) Ao clicar a opção "pairwise_similarity"na caixa de seleção, a lista de agrupamentos ordena de forma decrescente de grau de similaridade entre pares. (2) No exemplo apontado, os clusters 4 e 12 são similares entre si, com grau de similaridade igual a 0,65*

Na ordenação da Figura I.7, os agrupamentos são listados em ordem decrescente de similaridade em pares. Essa ferramenta é útil para ajudar o anotador a não selecionar dois agrupamentos semanticamente muito próximos para evitar redundância no resultado final da curadoria.

I.8 Ordenação por Similaridade com o Centróide

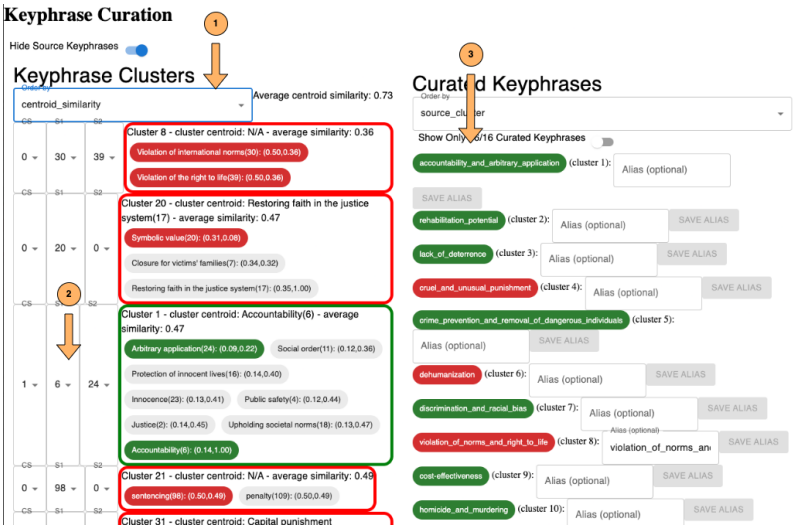


Figura I.8: *Ordenação por similaridade com o centróide: (1) Ao clicar a opção "centroid_similarity"na caixa de seleção, a lista de agrupamentos ordena de forma crescente de grau de similaridade com o centróide, que é informado no título da caixa do agrupameento. (2) Opcionalmente, o anotador pode aceitar o centróide como a frase-chave selecionada; (3) Caso duas frases-chave sejam selecionadas, as duas são concatenadas e assim se torna a frase-chave final composta*

Na ordenação da Figura I.8, os agrupamentos são listados em ordem crescente de similaridade com o centróide. No exemplo mostrado na figura, o centróide é a frase-chave "Accountability(6) (0.14, 1.00)"que possui a maior similaridade média com as demais frases-chave (0.14) e uma similaridade (1.00) com ela mesma. Neste mesmo exemplo, as frases-chave 6 e 24 - "Accountability"e "Arbitrary application", respectivamente - foram selecionadas e, portanto, são listadas como uma frase-chave composta na lista de frases curadas como "accountability_and_arbitrary_application.

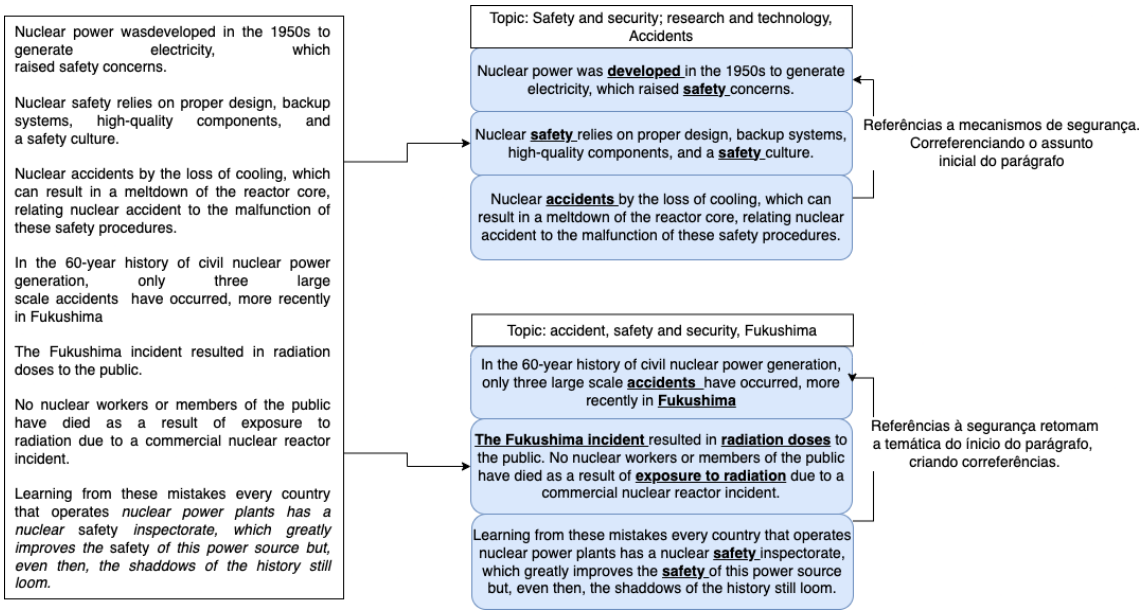
APÊNDICE J

Guideline para Anotação de Segmentação e Classificação de Tópicos

Este anexo apresenta a *guideline* utilizada para anotação da tarefa de segmentação e classificação.

J.1 Rótulos de Segmentação

Para a segmentação, o objetivo se torna recortar e subdividir os textos entre assuntos pertinentes, assim como é feito na imagem a seguir:



Na imagem acima temos **grupos de sentenças** que formam dois **segmentos de texto**. Repare que os segmentos recebem rótulos de "*Topic*", sendo atribuídos três *termos-chave* a cada. O primeiro grupo recebe os atributos *"Safety and security"*, *"research and science"* e *"accidents"*, ao passo que discursa sobre como o desenvolvimento tecnológico da energia nuclear se relaciona com questões de segurança e, subsequentemente, com acidentes nucleares. O segundo grupo, por sua vez, mantém os atributos *"Safety*

and security” e *“accidents”*, mas abandona o desenvolvimento tecnológico para focar em um acidente específico, recebendo o atributo *“Fukushima”* ao invés de *“research and science”*. A mesclagem de tópicos forma um contínuo entre subtópicos que flui gradualmente para novos temas.

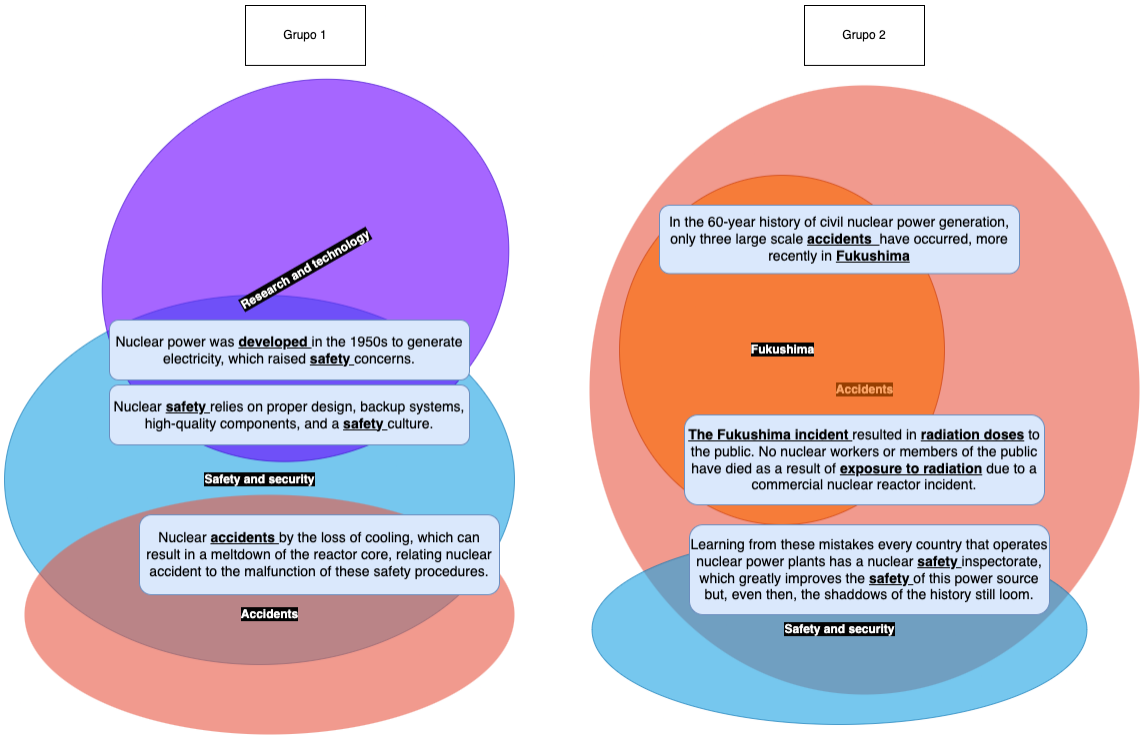


Figura J.1: *Contínuo de tópicos*

Observa-se que os tópicos não são completamente abandonados nas viradas temáticas; novos tópicos frequentemente portam atributos remanescentes das seções anteriores, fruto do poder de coesão textual.

A manifestação de **palavras relacionadas** aos atributos de tópico — como *“Safety”*, *“develop”*, *“radiation”* ou *“accidents”* — é indício relevante da necessidade de atribuição desses rótulos.

Trechos de fechamento de tópico frequentemente fazem referência a sentenças anteriores, sendo bons indicadores da introdução de um novo tópico.

Aspectos formais do texto também sinalizam a divisão temática, como:

- Bullet points
- Estrutura de parágrafos
- Mudança nas palavras-chave manifestadas

J.1.1 TopicType: Topic

Topic	A segmentação de tópicos visa detectar a introdução de subtópicos no texto, uma especificação do assunto principal que se apresenta. Ao visualizar a entrada de assuntos tangentes porém pertinentes, por vezes marcada por bullet points, o anotador deve atribuir o valor “Topic” à sentença.

J.1.2 TopicType: Ignored

Ignored	O atributo “Ignored” pode ser atribuído a sentenças que claramente não tratam do tópico geral ou que representam elementos complementares da formatação do texto (como glossários e referências), sendo consideradas vazias de argumentação pertinente.

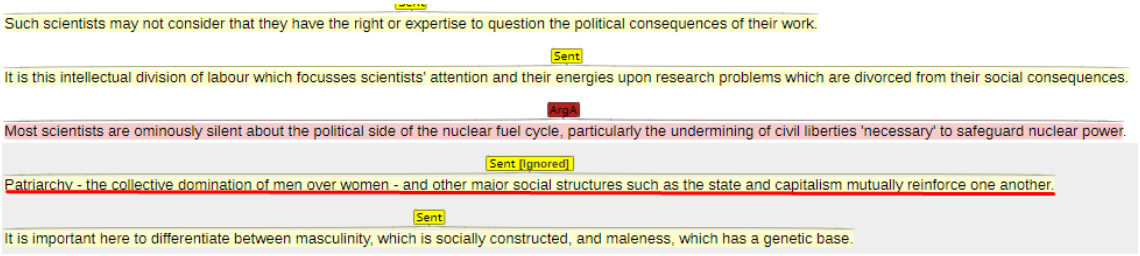


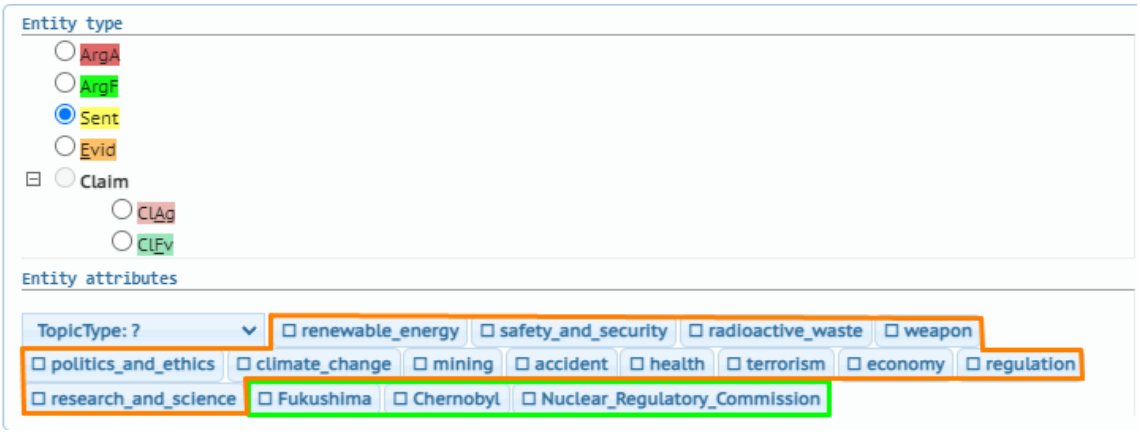
Figura J.2: *Exemplo de segmento a ser ignorado*

Exemplo: Na Figura J.2 a autora inicia sua argumentação dentro do tópico “energia nuclear” e em seguida discorre sobre feminismo e patriarcado, sem conexão com o tema central. Esse trecho deve ser marcado como “TopicType: Ignored”.

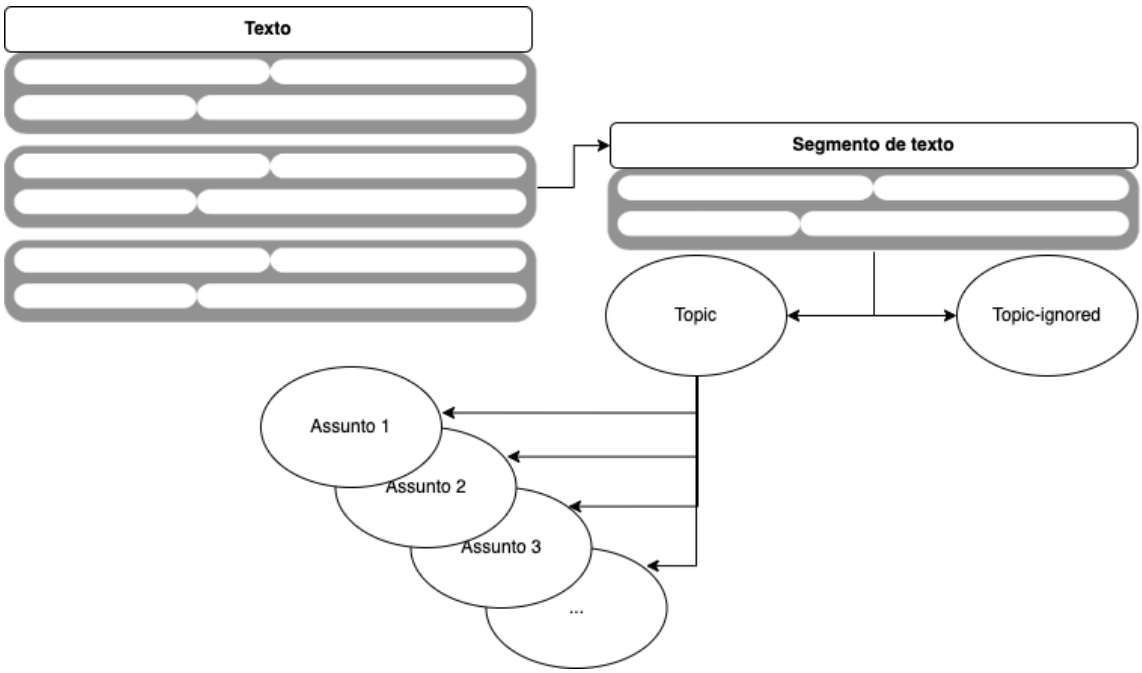
J.2 Rótulos de Classificação

Os rótulos de classificação indicam o assunto nas partes segmentadas do texto. Cada tópico contém cerca de 16 atributos baseados em pontos-chave, subdivididos em:

Palavras-chave	Descrevem de forma sintética um conteúdo relevante do texto. Estão marcadas na imagem a seguir com contorno laranja.
Entidades nomeadas	Referem-se a objetos do mundo real (pessoas, lugares, instituições etc.), marcadas com contorno verde.



Esses atributos devem ser atribuídos a todas as sentenças rotuladas como "Topic". É permitido de um a três rótulos, preferencialmente três. Caso nenhum rótulo seja adequado, pode-se sugerir um novo na seção de comentários.



J.2.1 Regras de Anotação

- Atributos devem ser atribuídos à sentença inicial que introduz um segmento (geralmente com mais de 5 sentenças). Evitar rotular frase a frase.
- Segmentos "Topic" devem conter ao menos uma sentença rotulada como "ArgumentFavor" (verde) ou "ArgumentAgainst" (vermelha).
- Segmentos "Ignored" não devem conter sentenças com rótulo de argumento.
- Segmentos "Topic" geralmente têm mais de 5 sentenças, mas listas com vários tópicos podem ter 2 ou 3 por item.

•Segmentos “Ignored” devem ter ao menos 5 sentenças. Se forem curtos, verificar se há correferência com o segmento anterior ou seguinte e fundir, se aplicável (ex.: “Fonte: Wikipédia”).

J.2.2 Boas Práticas

•Preferir uso de dois atributos por tópico. Um ou três são permitidos em casos excepcionais.
•Evitar segmentar quando o novo tópico repete todos os atributos do anterior.
•Evitar isolar tópicos que não contêm argumentos, salvo absoluta necessidade.
•Evitar separar segmentos com correferência clara entre si.