UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

PEDRO VITOR QUINTA DE CASTRO

Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico

Goiânia
2019

sistema de bibliotecas ufg

PRPG
PRÓ-REITORIA DE PÓS-GRADUAÇÃO

UFG

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO 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 nº 9610/98, 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.

1. Identificação do material bibliográfico: **[X] Dissertação** **[ ] Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do autor: Pedro Vitor Quinta de Castro

Título do trabalho: Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico

3. Informações de acesso ao documento:

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

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Pedro Vitor Quinta de Castro
Assinatura do(a) autor(a)[2]

Ciente e de acordo:

Assinatura do(a) orientador(a)[2]

Data: 06 / 01 / 2020

[1]Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados 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

[2]A assinatura deve ser escaneada.

PEDRO VITOR QUINTA DE CASTRO

Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico

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

Área de concentração: Ciência da Computação.

Orientadora: Profa. Dra. Nádia Félix Felipe da Silva

Co-Orientador: Prof. Dr. Anderson da Silva Soares

Goiânia
2019

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

Quinta de Castro, Pedro Vitor
Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico [manuscrito] / Pedro Vitor Quinta de Castro. - 2019.
CXXV, 125 f.

Orientador: Profa. Dra. Nádia Félix Felipe da Silva; co-orientador Dr. Anderson da Silva Soares.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2019.
Bibliografia.
Inclui siglas, lista de figuras, lista de tabelas.

1. Reconhecimento de Entidades Nomeadas. 2. Processamento de Linguagem Natural. 3. Redes Neurais. 4. Deep Learning. 5. Direito do Trabalho. I. Félix Felipe da Silva, Nádia, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **16/2019** da sessão de Defesa de Dissertação de **Pedro Vitor Quinta de Castro**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos cinco dias do mês de dezembro de dois mil e dezenove, a partir das dezesseis horas, na sala 150 do Instituto de Informática, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico"**. Os trabalhos foram instalados pela Orientadora, Professora Doutora Nádia Félix Felipe da Silva (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professora Doutora Helena de Medeiros Caseli (DC/UFSCar), membra titular externa; cuja participação ocorreu através de videoconferência; Professor Doutor Thierson Couto Rosa (INF/UFG), membro titular interno, e Professor Doutor Anderson da Silva Soares (INF/UFG), membro titular interno, coorientador. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pela Professora Doutora Nádia Félix Felipe da Silva, 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 cinco dias do mês de dezembro de dois mil e dezenove.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Nadia Felix Felipe Da Silva**, **Professor do Magistério Superior**, em 05/12/2019, às 18:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Thierson Couto Rosa**, **Professor do Magistério Superior**, em 05/12/2019, às 18:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 05/12/2019, às 18:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Helena de Medeiros Caseli**, **Usuário Externo**, em 05/12/2019, às 18:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

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 **0948978** e o código CRC **BC8B9E01**.

Referência: Processo nº 23070.034773/2019-80 SEI nº 0948978

Pedro Vitor Quinta de Castro

Graduou–se em Engenharia de Computação na UFG - Universidade Federal de Goiás. Experiência profissional com arquitetura, projeto e desenvolvimento de software, tendo atuado em empresas multinacionais tais como IBM e Indra. Atualmente trabalha no setor privado, desenvolvendo soluções de Processamento de Linguagem Natural voltadas para a área jurídica.

Este trabalho é fruto de todo meu esforço e dedicação à minha esposa e companheira Mariana, e às minhas filhas Gabriela e Isabela, que me dão todo o amor e carinho de que preciso para sempre continuar estudando e me aperfeiçoando. Cada página aqui escrita deve-se a elas.

Dedico este trabalho também ao meu pai Sílvio, que vinte anos atrás me incentivou a trilhar os caminhos da computação.

Dedico também à minha mãe Regina Célia, que depois de quase 50 anos de magistério ainda tem uma força descomunal para enfrentar a sala de aula como aluna, almejando o tão esperado e merecido título de doutora. Seu exemplo não me permite fraquejar por um segundo qualquer.

Agradecimentos

Gostaria de agradecer imensamente aos meus orientadores Nádia Félix Felipe da Silva e Anderson da Silva Soares. Fui extremamente afortunado em ser acolhido pelos dois de volta ao mundo acadêmico, depois de mais de 10 anos distante. Muito obrigado à Nádia pela paciência, enquanto me alfabetizou, me orientou e me desafiou em cada etapa deste processo. Muitíssimo obrigado ao Anderson por ter feito com que eu acreditasse no meu potencial, e pelo papel que desempenhou na renovação da minha carreira. A orientação dos dois contribui para que a minha vida acadêmica seja mais gratificante e estimulante do que qualquer outra coisa.

Meus agradecimentos também ao Caio dos Santos, da Data Lawyer, por ter depositado em mim a confiança de que juntos construiríamos ferramentas inovadoras para o mercado jurídico brasileiro, empreitada na qual temos sido exitosos.

Agradeço também ao grande amigo Jones José da Silva Júnior por ter me estimulado a encarar esta empreitada pelo fascinante mundo da Inteligência Artificial.

Por fim, agradeço aos colegas do grupo de pesquisa Deep Learning Brasil e da Data Lawyer, por criarem um ambiente no qual pesquisar e trabalhar é uma grande satisfação.

Resumo

Quinta de Castro, Pedro Vitor. **Aprendizagem Profunda para Reconhecimento de Entidades Nomeadas em Domínio Jurídico**. Goiânia, 2019. 125p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Reconhecimento de Entidades Nomeadas (REN) é uma tarefa desafiadora em Processamento de Linguagem Natural, para uma língua tão rica quanto o Português. Quando aplicada em um domínio específico, a tarefa adquire uma nova camada de complexidade, por tratar de um léxico muito particular ao domínio trabalhado. O domínio estudado neste trabalho é o do Direito, voltado especificamente para a Justiça do Trabalho do Brasil. Arquiteturas baseadas em Aprendizado Profundo, com representações de palavras baseadas em vetores estáticos de palavras e modelos de linguagem, têm demonstrado um desempenho em nível de estado da arte para a tarefa de REN. Neste trabalho é utilizado um modelo baseado em Redes Neurais Profundas, avaliando diferentes formas de representação de palavras. São avaliados modelos tanto para o domínio do Direito quanto para a língua portuguesa em um contexto geral. Para tanto, foram treinados modelos de linguagem baseados na arquitetura *ELMo* para os dois domínios, assim como vetores estáticos de palavras específicos para o domínio do Direito. Neste trabalho também verificou-se os melhores tipos de vetores para cada domínio, a partir de uma série de análises comparativas entre os vetores aplicados na tarefa de REN. Para os treinos dos modelos de REN, *ELMo* e vetores estáticos do domínio jurídico foram produzidos e anotados em *corpora* específicos deste domínio, a partir da coleta de documentos públicos da Justiça do Trabalho do Brasil. Para o modelo de REN do domínio geral da língua portuguesa, atingiu-se um novo estado da arte no benchmark do HAREM, com 83.22% de *F-Score* para o cenário seletivo, e 78.04% para o cenário total. Para o domínio trabalhista brasileiro, foi obtido um modelo com 93.81% de *F-Score*.

Palavras–chave

Reconhecimento de Entidades Nomeadas, Processamento de Linguagem Natural, *Deep Learning*, Redes Neurais, Língua Portuguesa, Direito do Trabalho

Abstract

Quinta de Castro, Pedro Vitor. **Deep Learning for Named Entity Recognition in Legal Domain**. Goiânia, 2019. 125p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Named Entity Recognition (NER) is a challenging Natural Language Processing task for a language as rich as Portuguese. When applied to a specific domain, the task acquires a new layer of complexity, handling a lexicon particular to the domain in question. In this work, it is studied the Legal domain, targeting specifically the Brazilian Labor Law. Architectures based on Deep Learning, with word representations based on static word embeddings and language models have shown state-of-the-art performance for the NER task. In this work it is used a model based on Deep Neural Networks, evaluating different forms of word representations. The evaluated models are applied to Portuguese language, for both Legal and general domains. To this end, language models based on the *ELMo* architecture were trained for both domains, as well as static word embeddings, specific for the Legal domain. In this work, it is verified the best type of pre-trained word embeddings for each domain, after performing a comparative study between the types of word embeddings applied to the NER task. For the training of the Legal domain NER models, *ELMo* and static word embeddings, two different *corpora* were produced and annotated, based on a collection of public documents from the Brazilian Labor Court. For the Portuguese general domain NER model, a new state-of-the-art result was achieved for the HAREM benchmark, with 83.22% F-Score for the selective scenario, and 78.04% for the total scenario. For the Brazilian Labor Law domain, a model with 93.81% F-Score was obtained.

Keywords

Named Entity Recognition, Natural Language Processing, *Deep Learning*, Neural Networks, Portuguese Language, Labor Law

Lista de Abreviaturas e Siglas

AP - Aprendizado Profundo
biLM - *bidirectional Language Model*
biLSTM - *bidirectional Long Short-Term Memory*
CBoW - *Continuous Bag-of-Words*
CC - Código Civil
CD - Coleção Dourada
CF - Constituição Federal
CLT - Consolidação das Leis do Trabalho
CNJ - Conselho Nacional de Justiça
CNN - *Convolutional Neural Networks*
CoNLL - *Computational Natural Language Learning*
CPC - Código Processual Civil
CRF - *Conditional Random Fields*
ELMo - *Embeddings from Language Model*
ER - Extração de Relações
LSTM - *Long Short-Term Memory*
MLP - *Multilayer Perceptron*
NILC - Núcleo Interinstitucional de Linguística Computacional
PJe - Processo Jurídico Eletrônico
PLN - Processamento de Linguagem Natural
REN - Reconhecimento de Entidades Nomeadas
ReLU - *Rectified Linear Unit*
RL - Regressão Logística
RNA - Redes Neurais Artificiais
RNN - *Recurrent Neural Networks*
RNP - Redes Neurais Profundas
SVM - *Support Vector Machines*
TJ - Tribunal de Justiça
TRT - Tribunal Regional do Trabalho
TST - Tribunal Superior do Trabalho

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

A extração de informação é o processo de obtenção de informação relevante (dados estruturados) a partir de fontes que não podem ser interpretadas diretamente por máquinas, como textos [63]. Atualmente, este processo tem uma importância muito grande, pois vive-se em uma época de abundância de informações textuais, que em sua maioria, são não-estruturadas. De acordo com dados levantados[1], existem cerca de 4.2 bilhões de usuários da Internet. Diariamente são registrados no mundo inteiro: 478 milhões de *tweets*[2], 4 milhões de *posts* em *blogs* e 4.2 bilhões de pesquisas no Google[3].

O processo de estruturação das informações é de grande importância, pois à medida em que informações são estruturadas, elas passam a ser mais facilmente localizadas por sistemas de busca indexada, como o Google, ou tornam-se mais interpretáveis por outros sistemas de processamento de dados. Para realizar tal extração de dados estruturados, é imprescindível realizar uma tarefa que faça o reconhecimento de dados de uma determinada estrutura, seguido pela classificação destes dados, dentro de tal estrutura. Esta tarefa é o Reconhecimento de Entidades Nomeadas (REN).

1.1 Definição da Tarefa

A tarefa de REN visa identificar entidades dentro de um texto e classificá-las em um determinado conjunto de categorias de interesse, tais como pessoas, organizações ou lugares [63]. Um exemplo de entidades reconhecidas pela API de linguagem natural do Google[4] seria: <Tipo=Pessoa **Dilma Roussef**> foi a <Tipo=pessoa **presidente**> do <Tipo=Local **Brasil**> eleita pelo <Tipo=Organização **Partido dos Trabalhadores**> nas <Tipo=Evento **eleições**> de <Tipo=Número **2014**>.

Li et al. [56] formalizou a tarefa como sendo uma função que produz um conjunto de tuplas da forma $<I_i,I_f,t>$ a partir de uma sentença $s = \{p_1, p_2, p_3, ..., p_N\}$, tal que I_i

[1] Dados obtidos a partir do site `http://www.internetlivestats.com/` em 30/04/2019
[2] `http://www.twitter.com`
[3] `http://www.google.com`
[4] `https://cloud.google.com/natural-language/`

$\in [1, N]$ e $I_f \in [1, N]$. Nestas tuplas, I_i e I_f são os índices iniciais e finais das entidades reconhecidas, e t representa os tipos das entidades, a partir do conjunto de categorias predefinido. O Exemplo 1.1 mostra a identificação de quatro entidades por um modelo qualquer de REN:

Exemplo 1.1: *Exemplo de entidades reconhecidas por um modelo de REN.*

$$Jair_{p_1}\ Messias_{p_2}\ Bolsonaro_{p_3}\ foi_{p_4}\ eleito_{p_5}\ presidente_{p_6}\ do_{p_7}\ Brasil_{p_8}\ nas_{p_9}\ eleições_{p_{10}}\ de_{p_{11}}\ 2018_{p_{12}}\ pelo_{p_{13}}\ Partido_{p_{14}}\ Social_{p_{15}}\ Liberal_{p_{16}}\ .{}_{p_{17}}$$

$$\downarrow$$

Modelo de REN

$$\downarrow$$

$$\text{<1, 3, PESSOA>} \longrightarrow \textit{Jair Messias Bolsonaro}$$
$$\text{<8, 8, LOCAL>} \longrightarrow \textit{Brasil}$$
$$\text{<12, 12, NUMERO>} \longrightarrow \textit{2018}$$
$$\text{<14, 16, ORGANIZACAO>} \longrightarrow \textit{Partido Social Liberal}$$

A partir destes exemplos, pode-se perceber que esta tarefa implica em fazer uma classificação sequencial de cada palavra p da sentença s, da seguinte forma:

- A palavra p é uma entidade?
- Se a palavra p é uma entidade, a que tipo de entidade t ela pertence?

Além da classificação de cada palavra, o modelo de REN também deve delimitar as fronteiras das entidades reconhecidas de forma correta. Considerando que o exemplo 1.1 seja totalmente correto, os Exemplos 1.2 e 1.3 mostram diferentes erros de delimitação das fronteiras da entidade *Jair Messias Bolsonaro*. No Exemplo 1.2, a palavra *Bolsonaro* foi excluída da entidade reconhecida, e sua fronteira foi delimitada na palavra *p2* ao invés de *p3*. Já no Exemplo 1.3, cada palavra foi reconhecida como uma entidade diferente, produzindo três entidades reconhecidas ao invés de uma.

Exemplo 1.2: *Exemplo de identificação errada da fronteira, com a entidade reconhecida de forma parcial.*

$$\text{<1, 2, PESSOA>} \longrightarrow \textit{Jair Messias}$$

Exemplo 1.3: *Exemplo de erro na identificação das fronteiras da entidade, com cada palavra sendo reconhecida como uma entidade diferente.*

$$\text{<1, 1, PESSOA>} \longrightarrow \textit{Jair}$$

<2, 2, PESSOA> $\longrightarrow$ *Messias*
<3, 3, PESSOA> $\longrightarrow$ *Bolsonaro*

A tarefa de REN é predecessora a outras tarefas de extração de informações, tais como Extração de Relações (ER) [19] e Vínculo de Entidades (ou Desambiguação de Entidades) [63]. A tarefa de ER visa classificar o relacionamento entre duas entidades reconhecidas em um texto, enquanto a tarefa de vínculo de entidades visa associar uma entidade reconhecida a uma base de conhecimento, atribuindo à entidade um identificador único nesta base, promovendo sua desambiguação. Um exemplo de extração de relações seria: <Entidade=Pessoa **Dilma Roussef**> é <Relação=Afiliação **afiliada**> ao <Entidade=Organização **Partido dos Trabalhadores**>. Um exemplo de vínculo e desambiguação de entidades seria saber diferenciar entre Java[5], a linguagem de programação e Java[6], a ilha do arquipélago da Indonésia, e vincular a ocorrência de cada entidade específica a um identificador próprio.

A tarefa de REN é aplicável não só em conteúdo textual de caráter geral, mas como também em acervos de domínios específicos, tais como Jornalismo [79], Biomedicina [22] e Geologia [25]. Quando se aplica extração de entidades em domínios específicos, tem-se a possibilidade de extrair diferentes tipos de informações, que são de particular relevância para os domínios em questão. No domínio do Direito, por exemplo, pode-se usar REN para extrair nomes de ***juízes*** e de ***advogados*** de documentos legais, ao invés de extrair somente nomes de ***pessoas*** em geral, atribuindo mais semântica às informações obtidas.

1.2 Justificativa

1.2.1 Relevância na Língua Portuguesa

O termo ***Named Entity*** (Entidade Nomeada) foi cunhado em 1996 [69], para a ***VI Message Understanding Conference (MUC-6)***[7], que teve como foco tarefas de Extração de Informações. Apesar desta nomenclatura ter surgido neste ano, pode-se rastrear os primeiros trabalhos de estudo acerca de identificação de entidades nomeadas ao ano de 1991, na ***VII Conferência da IEEE de Aplicações de Inteligência Artificial*** [8].

Apesar de já ser estudada há tanto tempo, a tarefa de REN ainda é um problema em aberto para a comunidade científica, especialmente para a língua portuguesa [62, 19]. O enorme volume de informações textuais na língua inglesa, assim como a abundância de

[5] https://pt.wikipedia.org/wiki/Java_(linguagem_de_programa%C3%A7%C3%A3o))
[6] https://pt.wikipedia.org/wiki/Java
[7] http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html
[8] http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=331

pesquisas e ferramentas desenvolvidas para este idioma, faz com que estes problemas de Processamento de Linguagem Natural (PLN) tenham soluções mais eficientes e avançadas para a língua inglesa do que para a língua portuguesa. Em trabalhos existentes focados na língua portuguesa, como em [19], há indicativos de que, mesmo na comunidade científica, modelos existentes para a língua portuguesa ainda têm muita dificuldade em alcançar o estado da arte, considerando-se o que já foi alcançado para a língua inglesa.

Os autores também evidenciam as dificuldades em atingir boa performance para problemas de REN e ER em virtude da performance sub-ótima de modelos predecessores[9], necessários para a execução destas tarefas. Como, para alguns modelos, há dependência de um pré-processamento do texto a partir da aplicação de etiquetagem morfossintática (***Part-of-Speech tags***), dentre outros, se o modelo utilizado para a aplicação das *tags* tiver uma performance ruim, isso afetará a performance das tarefas dependentes. Desta forma, a escassez de recursos necessários para o desenvolvimento dos modelos fundamentais de PLN também é um fator limitante para evoluir modelos de extração de texto, como os de REN.

1.2.2 No Domínio da Justiça do Trabalho Brasileira

Apesar de haverem vários trabalhos de REN voltados para o domínio do Direito [80, 88, 3, 4, 8, 28, 60], existem poucos voltados não só para a língua portuguesa, mas, também, para a própria justiça brasileira. Ademais, não foi identificado nenhum trabalho que trate especificamente da Justiça do Trabalho brasileira. O Conselho Nacional de Justiça[10] (CNJ) mantém um relatório anual chamado ***Justiça em Números*** [17], que, dentre outras, faz a análise da litigiosidade no Brasil. De acordo com a série histórica de casos pendentes no sistema judiciário brasileiro (Figura 1.1)[11], ao final de 2018, 78.7 milhões de processos estavam pendentes. Neste mesmo ano, a justiça conseguiu baixar 3.8 milhões de processos a mais do que a quantidade total de novos processos que surgiram. Embora isso evidencie que há um aumento de produtividade, também mostra que a diferença para o total de processos acumulados ainda é muito grande. De acordo com a Figura 1.2, a justiça estadual é a que mais possui processos pendentes na justiça brasileira, sendo 63 milhões de processos pendentes. Para a justiça do trabalho, a Figura 1.3 mostra que este número é de 4.9 milhões de processos.

[9] Predecessores aqui indica uma relação de dependência. Por exemplo, para que um modelo de ER possa ser treinado, é necessário aplicar um modelo de REN antes para etiquetagem dos dados de treino do modelo de ER, fazendo do modelo de REN um insumo do modelo de ER.

[10] `http://cnj.jus.br/`

[11] Processos baixados são processos finalizados.

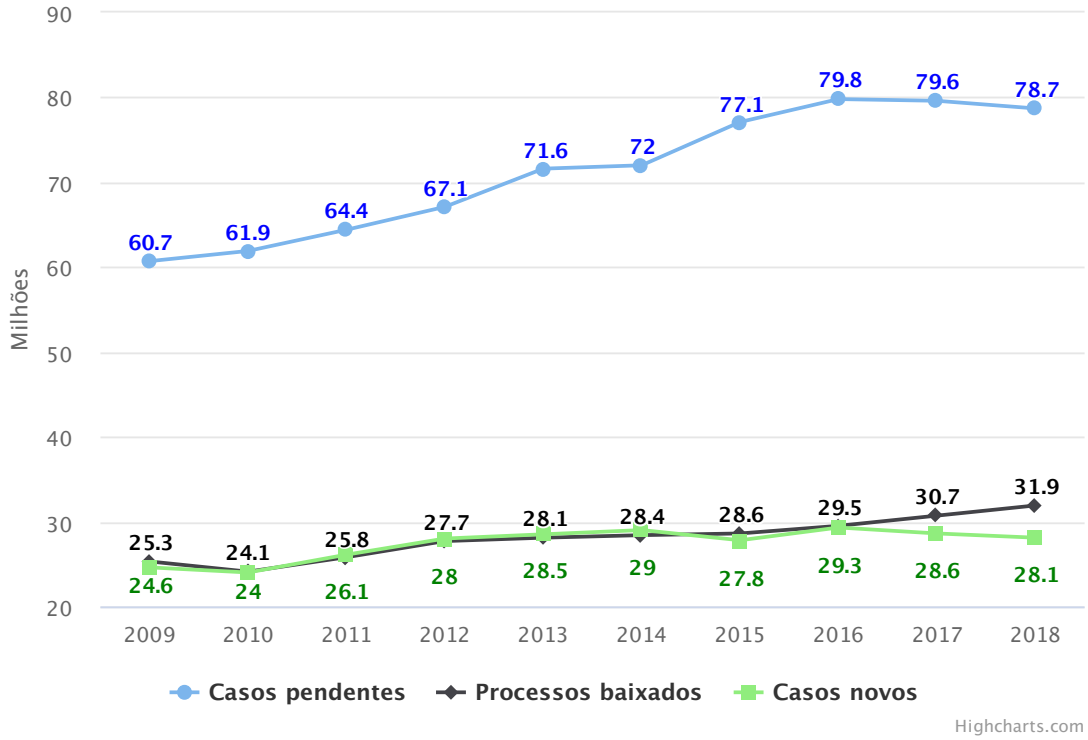

Figura 1.1: *Série histórica de total de casos pendentes na justiça brasileira. Dados obtidos através do relatório do* ***Justiça em Números****.*

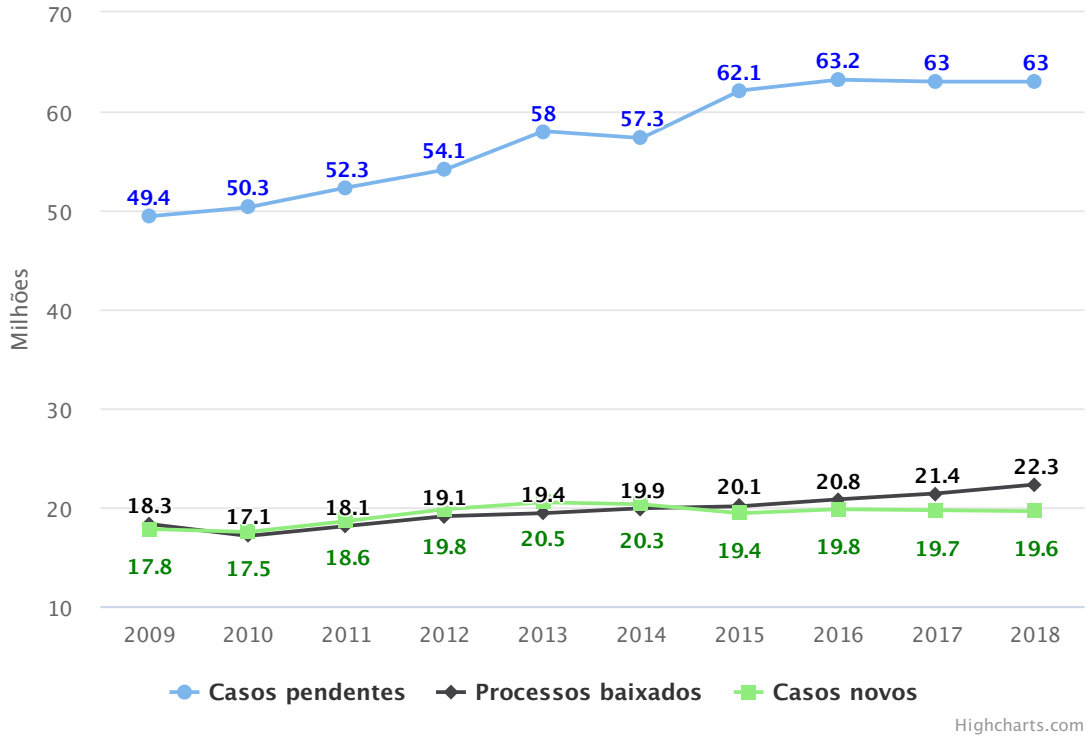

Figura 1.2: *Série histórica de casos pendentes, baixados e novos na justiça estadual brasileira. Dados obtidos através do relatório do* ***Justiça em Números****.*

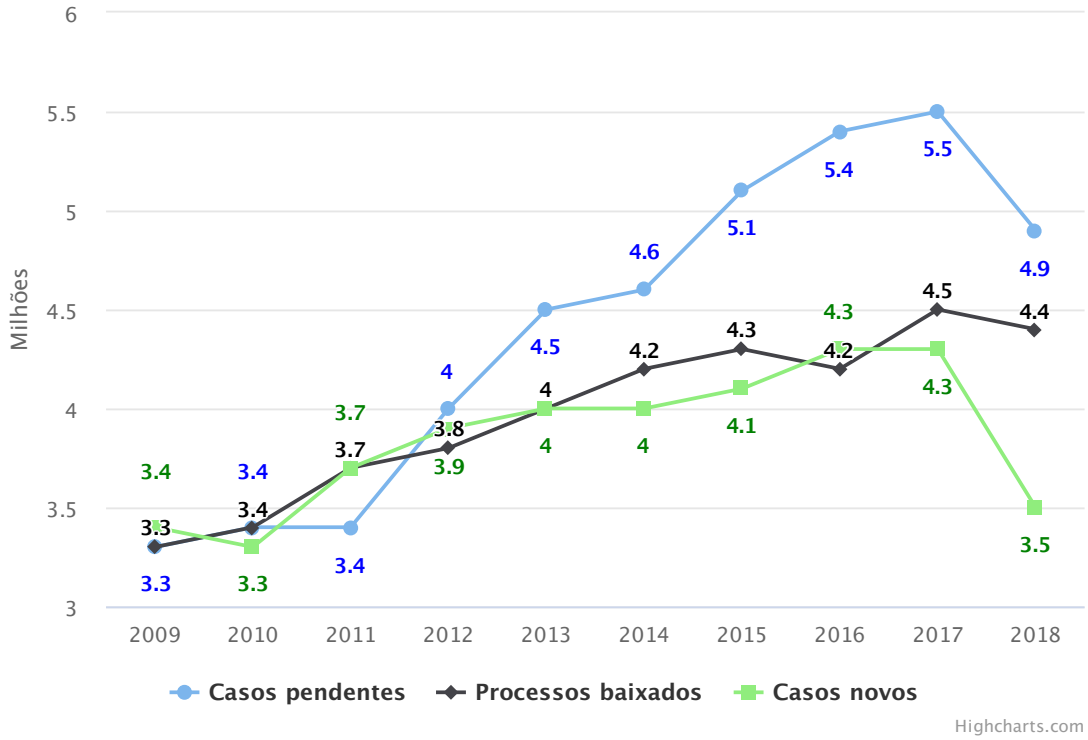

Figura 1.3: *Série histórica de casos pendentes, baixados e novos na justiça trabalhista brasileira. Dados obtidos através do relatório do* ***Justiça em Números****.*

O relatório do CNJ [17] ainda mostra que a esfera trabalhista é mais aderente à distribuição de novos processos em meio eletrônico do que a justiça estadual. Como pode ser visto na Figura 1.4, enquanto em 2018 a média de novos processos eletrônicos na justiça estadual foi de 82.6%, na justiça do trabalho, 97.7% dos novos processos já são distribuídos diretamente no sistema de ***Processo Judicial Eletrônico*** (***PJe***) [78] de cada tribunal das 24 regiões trabalhistas. Isso faz com que seja mais viável trabalhar com informações textuais dos processos da justiça trabalhista, visto que quase todos os documentos e dados estruturados dos processos eletrônicos estão acessíveis publicamente através do PJe, de forma digitalizada e padronizada. Além de ter menos processos distribuídos em meio eletrônico, a justiça estadual também não é tão padronizada quanto a trabalhista, pois, enquanto os Tribunais Regionais do Trabalho (TRTs) do Brasil inteiro usam o PJe, os Tribunais de Justiça (TJs) usam diferentes sistemas, tais como o e-SAJ[12], Projudi[13] e eproc[14], além do próprio PJe, em alguns estados.

[12] https://www.softplan.com.br/solucoes/saj-tribunais/
[13] https://pt.wikipedia.org/wiki/PROJUDI
[14] https://eproc.trf4.jus.br

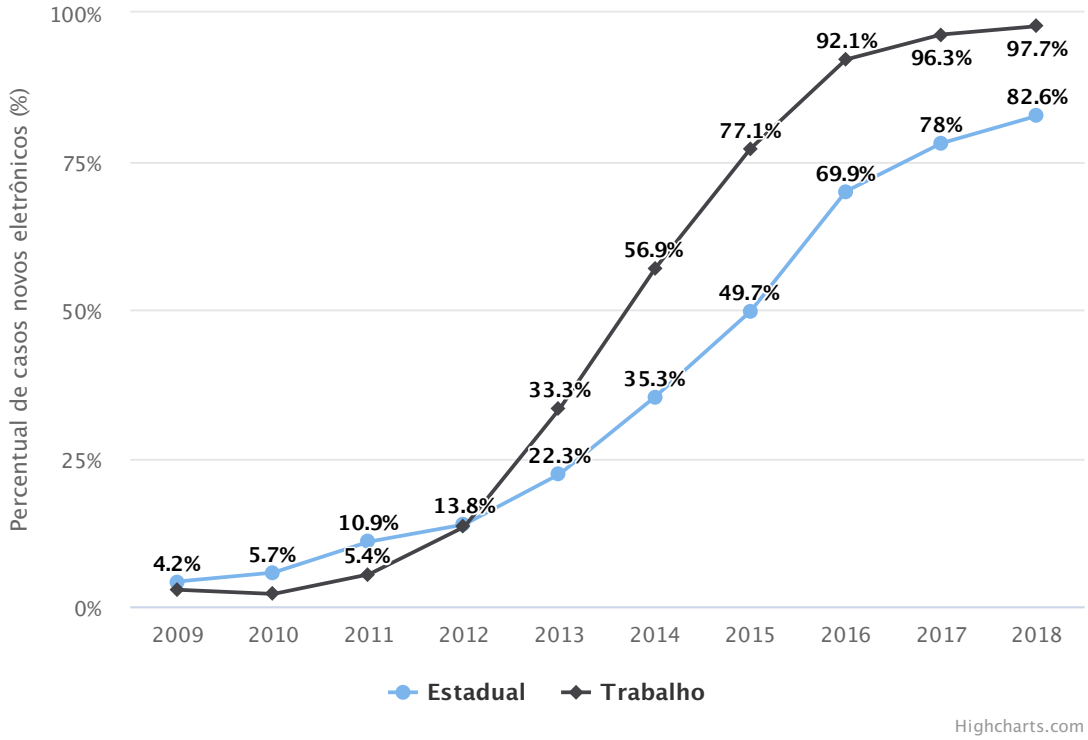

Figura 1.4: *Percentual de distribuição de novos processos em meio eletrônico nas justiças estadual e do trabalho. Dados obtidos através do relatório do* ***Justiça em Números****.*

Como não foram encontrados trabalhos voltados para a extração de informações da justiça trabalhista brasileira, e como é possível trabalhar com mais categorias de informações do que as que foram investigadas em outros trabalhos voltados para o domínio do Direito [60], definiu-se como escopo deste trabalho abordar este domínio, voltado para a justiça do trabalho brasileira.

1.3 Objetivos

Os objetivos deste trabalho são avaliar arquiteturas baseadas em Redes Neurais Profundas e desenvolver um modelo de Reconhecimento de Entidades Nomeadas para a língua portuguesa e para o domínio da Justiça do Trabalho do Brasil. Nas seções a seguir serão definidos os objetivos geral e específicos deste trabalho, no contexto que foi estabelecido.

1.3.1 Objetivo Geral

O objetivo principal desta dissertação é verificar como o Aprendizado Profundo pode ser aplicado para o Reconhecimento de Entidades Nomeadas no domínio da justiça do trabalho brasileira.

1.3.2 Objetivos Específicos

Dentre os objetivos específicos, que irão suportar o objetivo geral, pode-se elencar:

- Realizar um estudo acerca de arquiteturas de Reconhecimento de Entidades Nomeadas, baseadas em Redes Neurais Profundas;
- Construir um *corpus* trabalhista de REN baseado em documentos da justiça do trabalho do Brasil, com a ajuda de especialistas, tendo em vista que não existe na literatura correlata este recurso para o Português;
- A partir dos estudos de arquiteturas realizados, avaliar um modelo de REN para a língua portuguesa treinado em *corpus* de referência do idioma;
- A partir dos estudos de arquiteturas realizado, avaliar um modelo de REN para a justiça do trabalho brasileira treinado no *corpus* construído;
- Analisar os resultados obtidos a partir dos modelos de REN propostos.

Além de avaliar modelos de REN, também é objetivo deste trabalho avaliar modelos de representação numérica de palavras. Para isso, os objetivos específicos a seguir são propostos para verificar quais formas de representação de palavras fornecerão os melhores resultados para a tarefa de REN nos domínios em questão:

- Realizar um estudo acerca das formas de representação de palavras em modelos de REN, especificamente as que são baseadas em vetores de palavras e modelos de linguagem;
- Treinar vetores estáticos de palavras baseados em um acervo jurídico de documentos públicos obtidos a partir dos tribunais regionais do trabalho;
- Treinar um modelo de linguagem baseado em textos públicos da língua portuguesa obtidos na Internet;
- Treinar um modelo de linguagem baseado em um acervo jurídico de documentos públicos obtidos a partir dos tribunais regionais do trabalho;
- Analisar os resultados obtidos a partir das diferentes formas de representação de palavras avaliadas nos modelos de REN.

1.4 Hipóteses de Pesquisa

O Reconhecimento de Entidades Nomeadas para línguas ricas em recursos (como *corpora* e ferramentas automáticas para extração de características), como o Inglês, vem apresentando resultados melhores em diversas aplicações com o passar dos anos [44, 15, 54, 61, 75, 23, 1]. Entretanto, para outras línguas como o Português, em que há um deficit de recursos, o assunto ainda é pouco explorado [27, 24, 26, 25, 21, 77].

Com o intuito de contribuir com as pesquisas de REN para o Português, duas frentes de trabalho são desenvolvidas nesta dissertação. A primeira frente busca entender o cenário de Reconhecimento de Entidades Nomeadas para a língua portuguesa, usando arquiteturas e representações de palavras baseadas em Redes Neurais Profundas. Nesta primeira frente de trabalho não é explorada a definição de um domínio em específico.

Hipótese 1: É possível melhorar a acurácia dos modelos de Reconhecimento de Entidades Nomeadas para a língua portuguesa, utilizando arquiteturas e representações de palavras baseadas em Redes Neurais Profundas propostas para outras línguas.

A segunda frente de trabalho envolve desenvolver um modelo de Reconhecimento de Entidades Nomeadas específico para o domínio de Direito, em particular, voltado para a Justiça do Trabalho do Brasil. O objetivo é treinar este modelo usando a mesma arquitetura proposta para a língua portuguesa, mas em *corpora* específico deste domínio.

Hipótese 2: É possível usar as mesmas arquiteturas do modelo de REN para a língua portuguesa de domínio geral e criar um modelo de extração de entidades para o domínio do Direito, com um desempenho avaliado pela Medida F (*F-Score*, definida na Seção 2.1) de pelo menos 80%.

1.5 Contribuições

São contribuições deste trabalho:

- Um *corpus* de documentos da justiça do trabalho do Brasil, anotado com entidades nomeadas;
- Vetores de palavras estáticos para a língua portuguesa, treinados em um acervo de documentos públicos dos tribunais regionais do trabalho;
- Modelo de linguagem para a língua portuguesa, treinado em acervo de domínio geral;
- Modelo de linguagem para a língua portuguesa, treinado em um acervo de documentos públicos dos tribunais regionais do trabalho;
- Modelo de extração de entidades treinado nos *corpora* do HAREM, de domínio geral;
- Modelo de extração de entidades para o domínio do Direito, especificamente a justiça do trabalho brasileira;

1.6 Organização da Dissertação

O restante da dissertação será organizado da seguinte forma:

- O Capítulo 2 trata da fundamentação teórica do trabalho, abordando diferentes *benchmarks* de avaliação de sistemas de REN; formas de representação de palavras para modelos de REN; e arquiteturas de Aprendizado Profundo (AP) para modelos de REN;
- O Capítulo 3 descreve os trabalhos relacionados nas áreas abordadas, cobrindo modelos de REN para a língua portuguesa, modelos de REN baseados em AP e trabalhos de extração de informações no domínio do Direito;
- No Capítulo 4 será apresentado um estudo do domínio, abordando fundamentos do sistema judicial brasileiro na esfera trabalhista, e como este estudo foi determinante para a seleção das categorias de entidades utilizadas;
- O Capítulo 5 descreve o processo de construção do *corpus* de REN de documentos da justiça trabalhista do Brasil;
- No Capítulo 6 é descrita a modelagem dos métodos que são fundamentais para o modelo de REN proposto, cobrindo tanto a arquitetura quanto a forma de representação do conhecimento na aprendizagem do modelo;
- No Capítulo 7 são apresentados os resultados obtidos a partir do processo de avaliação dos modelos obtidos, fazendo também uma discussão acerca dos resultados e uma análise dos erros;
- Encerra-se no Capítulo 8 apresentando as conclusões atingidas, as principais contribuições realizadas, e as publicações produzidas, assim como as que estão em andamento.

CAPÍTULO 2

Fundamentação Teórica

De acordo com [63], a tarefa de Reconhecimento de Entidades Nomeadas (REN) consiste na identificação de nomes próprios em conteúdo textual, seguida da classificação dos mesmos em categorias pré-determinadas, tais como pessoas, lugares e organizações. Para que um modelo de REN seja desenvolvido, ele precisa passar por um processo que o torne capaz de converter uma forma de representação das palavras em uma saída que atribua a cada palavra o tipo de categoria a que ela pertence (inclusive palavras que **não** são entidades). Maiores detalhes sobre a avaliação de modelos de REN serão fornecidos na seção 2.1.

Abordagens clássicas de REN trabalhavam com sistemas baseados em regras e com modelos treinados de maneira tanto supervisionada como não-supervisionada [56]. Sistemas baseados em regras dependem do desenvolvimento e manutenção de um conjunto de regras de forma manual. Exemplos de artefatos usados nestes sistemas são *gazetteers*[1] e regras baseadas em análises sintáticas e léxicas. Modelos baseados em aprendizado não-supervisionado usam algoritmos que tentam extrair entidades realizando o agrupamento de palavras, baseados em algum tipo de similaridade entre os membros dos grupos. Modelos não-supervisionados não dependem de dados rotulados para realizar seu aprendizado.

Os modelos de REN treinados de maneira supervisionada dependem tanto da disponibilidade de dados rotulados, quanto de alguma forma de representação das palavras que possam produzir um vetor de características para treino. Para que estes vetores forneçam a melhor representatividade possível das palavras – por meio de um melhor desempenho do modelo – é necessário realizar um trabalho manual de seleção de características. Este trabalho é bem oneroso, pois requer a anotação manual de cada palavra do *corpus* de treino com as características que o anotador julgar relevantes para a identificação e classificação de entidades. Desta forma, é necessário realizar todo um ciclo de anotações das características de cada palavra toda vez que se desejar avaliar um conjunto

[1] Listas de entidades mantidas manualmente, usadas como um dicionário que os sistemas consultam para identificar entidades, verificando se as palavras do texto sendo processado se encontram nas listas.

diferente de características. Por exemplo, em um primeiro ciclo de experimentos o pesquisador pode desejar usar a classe gramatical como característica de cada palavra. Para isso, seria necessário anotar cada palavra do *corpus* com a sua respectiva classe gramatical e implementar o seu modelo em função da conversão da combinação destes dados na predição da categoria de entidade de cada palavra do *corpus*. Para cada nova característica que o pesquisador quiser experimentar (como, por exemplo, a função sintática das palavras), é necessário refazer todas as anotações de cada palavra com a nova característica, e depois ajustar o seu modelo para contemplar este novo vetor de representação das palavras. Um exemplo de algoritmo de aprendizado supervisionado comumente usado na tarefa de REN é *Conditional Random Fields* (CRF), que será mais detalhadamente explicado na seção 2.3.

Abordagens baseadas em Aprendizado Profundo (AP) oferecem alternativas a estas abordagens clássicas e são o foco deste trabalho. As arquiteturas de AP abordadas neste estudo serão descritas na seção 2.4.

2.1 Avaliação de Sistemas de Reconhecimento de Entidades Nomeadas

Existem diferentes *benchmarks* de avaliação conjunta[2] com o objetivo de regulamentar a tarefa de REN. A regulamentação desta tarefa consiste em vários pontos, tais como:

- Especificação de um conjunto de categorias de referência nas quais as entidades identificadas serão classificadas;
- Formalização do conceito de ***Entidade*** na avaliação, bem como estabelecimento das diretrizes a serem consideradas para atribuir entidades às categorias especificadas;
- Descrição de uma métrica de qualidade de referência para avaliação de modelos;
- Disponibilização de um *corpus* anotado de acordo com os parâmetros estabelecidos na avaliação conjunta, conhecido como ***Coleção Dourada (CD)***.

No que se refere às métricas de avaliação de REN, o desempenho de um modelo pode ser medido avaliando a correspondência de detecção das entidades tanto de forma exata quanto flexível [56]. Como a tarefa consiste em identificar as fronteiras das entidades[3], assim como as suas classificações, estes níveis de exatidão se referem a

[2] Avaliações realizadas por diferentes pessoas e/ou instituições com o objetivo de estabelecer um conjunto único de definições, critérios e normas que descrevam uma determinada tarefa, assim como os diversos parâmetros relacionados a ela.

[3] A fronteira de cada entidade seria delimitada pelo primeiro *token* constituinte da entidade, até o último.

como são quantificadas as identificações e classificações das palavras. Na correspondência exata, para que uma entidade seja considerada correta, os limites inicial e final da fronteira precisam estar idênticos aos da CD. Já na correspondência flexível, uma entidade pode ser classificada como correta mesmo que as suas fronteiras não sejam exatamente iguais às do valor real da CD.

Uma vez adotado o critério de correspondência, o valor final do desempenho do modelo é calculado por meio da Medida F (***F-Score***), obtida a partir da Precisão (***P***) e Abrangência (***A***) do modelo. As Equações (2-1), (2-2) e (2-3) mostram como calcular o ***F-Score*** a partir da quantificação das entidades classificadas pelo modelo de REN.

$$\boldsymbol{P} = \frac{\boldsymbol{VP}}{\boldsymbol{VP}+\boldsymbol{FP}} \tag{2-1}$$

$$\boldsymbol{A} = \frac{\boldsymbol{VP}}{\boldsymbol{VP}+\boldsymbol{FN}} \tag{2-2}$$

$$\boldsymbol{F\text{-}Score} = \frac{2 \times \boldsymbol{P} \times \boldsymbol{A}}{\boldsymbol{P}+\boldsymbol{A}} \tag{2-3}$$

onde ***VP*** (Verdadeiro Positivo) é a quantidade de entidades corretamente identificadas e classificadas, ***FP*** (Falso Positivo) é a quantidade de entidades identificadas e classificadas de maneira errônea, e ***FN*** (Falso Negativo) é a quantidade de entidades que deixaram de ser identificadas pelo modelo.

O ***F-Score*** é uma medida relevante para modelos de classificação como REN, pois faz a média harmônica entre a precisão e a abrangência do modelo, medindo tanto o seu desempenho para reconhecer as entidades corretas, por meio da abrangência, quanto para classificar corretamente, por meio da precisão. Como normalmente se trabalha com várias classes de entidades, também é importante considerar as diferentes formas de ponderação do ***F-Score*** para realizar a composição final da métrica do modelo, a partir da média do cálculo de desempenho de cada classe. Esta média pode ser feita nos modos *micro* e *macro*, sendo que, no primeiro, a precisão e abrangência são calculadas de uma só vez, considerando todos os valores de cada uma das classes; enquanto no modo *macro* a precisão e abrangência de cada classe são calculadas separadamente, para depois calcular a média entre os valores obtidos. A implicação disso é que no modo *macro* as contribuições de desempenho de cada classe têm o mesmo peso, independente da quantidade total de entidades de cada classe; enquanto no modo *micro*, por agregar todos os valores no mesmo cálculo, quem têm o mesmo peso são as entidades. Por este motivo, caso o *corpus* de treino do modelo de REN não tenha uma quantidade balanceada de entidades entre as classes, é recomendável usar o modo *micro*, pois o modo *macro* é insensível ao desequilíbrio entre as classes, dando o mesmo peso para todas. As Equações

(2-4) a (2-9) mostram como é feito o cálculo de cada um destes modos:

$$\textbf{\textit{Micro-P}} = \frac{\sum_{i=1}^{n} \mathbf{VP_i}}{\sum_{i=1}^{n} \mathbf{VP_i + FP_i}} \tag{2-4}$$

$$\textbf{\textit{Micro-A}} = \frac{\sum_{i=1}^{n} \mathbf{VP_i}}{\sum_{i=1}^{n} \mathbf{VP_i + FN_i}} \tag{2-5}$$

$$\textbf{\textit{Micro-F-Score}} = \frac{2 \times \textbf{\textit{Micro-P}} \times \textbf{\textit{Micro-A}}}{\textbf{\textit{Micro-P}} + \textbf{\textit{Micro-A}}} \tag{2-6}$$

$$\textbf{\textit{Macro-P}} = \frac{\sum_{i=1}^{n} \mathbf{P_i}}{n} \tag{2-7}$$

$$\textbf{\textit{Macro-A}} = \frac{\sum_{i=1}^{n} \mathbf{A_i}}{n} \tag{2-8}$$

$$\textbf{\textit{Macro-F-Score}} = \frac{2 \times \textbf{\textit{Macro-P}} \times \textbf{\textit{Macro-A}}}{\textbf{\textit{Macro-P}} + \textbf{\textit{Macro-A}}} \tag{2-9}$$

onde n é o número de classes avaliadas, **VP** é o número de verdadeiros positivos, **FP** é o número de falsos positivos, **FN** é o número de falsos negativos, **P** é a precisão e **A** é a abrangência.

De acordo com levantamento feito em [56], os *benchmarks* mais comumente utilizados como avaliação de referência de modelos de REN na língua inglesa são da ***Conference on Computational Natural Language Learning (CoNLL-2003)*** [91] e o ***OntoNotes 5.0*** [42]. Ambos disponibilizaram textos de domínio geral, baseados em notícias e conteúdos obtidos na Internet. Enquanto o CoNLL especifica 4 categorias de entidades (Pessoa, Local, Organização e Diversos), o OntoNotes, na versão 5.0, especifica 18 tipos de entidades, incluindo tipos mais específicos tais como Produto, Evento, Obra de Arte, Linguagem e Lei. Em relação à métrica de qualidade, o CoNLL e o OntoNotes adotaram a correspondência exata para calcular o ***F-Score***, no modo *micro* [91].

Para a língua portuguesa, embora existam *corpora* tais como o **WikiNER** [72], o **LeNER-Br** [60] e o **Paramopama** [48], a avaliação conjunta de referência mais utilizada [26, 25, 77, 60, 21] é o **HAREM**, um evento de avaliação conjunta da tarefa de REN que teve o objetivo de regular esta tarefa neste idioma. Houveram 2 edições organizadas

Corpus	Entidades	Palavras	Documentos
CD HAREM I [73]	5270	92830	129
CD MiniHAREM [73]	3858	62461	128
CD HAREM II [68]	3851	89241	129

Tabela 2.1: *Dados quantitativos de cada Coleção Dourada do HAREM.*

pela Linguateca[4]: **HAREM I** [87], em 2005 e o **HAREM II** [68], em 2008, além de um evento intermediário em 2006, o **MiniHAREM** [73], que foi uma repetição da primeira avaliação. Cada um destes eventos produziu uma CD diferente para avaliação de sistemas de REN. A Tabela 2.1 apresenta dados quantitativos do *corpus* de cada edição. No HAREM foram adotadas 10 categorias de entidades: Pessoa, Organização, Local, Valor, Tempo, Abstração, Obra, Acontecimento, Coisa e Outro.

Em relação às métricas de avaliação adotadas no HAREM, em [87] são descritos dois cenários distintos de avaliação dos modelos submetidos no HAREM: absoluto e relativo. No cenário absoluto, a métrica de desempenho dos modelos considera a correspondência exata, enquanto no cenário relativo a métrica é mais flexível, considerando diferentes critérios de acerto completo ou parcial, assim como diferentes tipos de erro também. O *script* de avaliação que implementa as métricas de avaliação do HAREM é chamado de **SAHARA**[5].

2.2 Esquemas de Anotações

O objetivo de um modelo de REN é classificar cada palavra de um texto, determinando se ela é uma entidade ou não, e, caso seja, que tipo de entidade é. Com isso, é necessário que cada *token*[6] do *corpus* de treino seja anotado com o seu rótulo. Neste trabalho os experimentos serão conduzidos com os esquemas de anotação **IOB2** [85] e **IOBES** [51]. O esquema IOB2 funciona da seguinte forma:

- *Tokens* que não fazem parte de uma entidade são rotulados com **O**;
- *Tokens* que fazem parte de uma entidade são rotulados com o prefixo **B** ou **I**, concatenado à categoria da entidade, por meio de um hífen, por exemplo **B-PESSOA**;
- O uso do prefixo **B** indica que o *token* marca o início (*beginning*) de anotação de uma entidade;

[4] `https://www.linguateca.pt/`

[5] `https://www.linguateca.pt/HAREM/avaliador/avaliador.html`

[6] O termo *token* será usado de forma equivalente a *palavra*, pois não necessariamente cada segmento do texto de um *corpus* é uma palavra, podendo ser, também, pontuações ou fragmentos de palavras, dependendo da forma como o texto foi segmentado.

Palavra	IOB2	IOBES
Nasceu	O	O
em	O	O
Nova	B-LOCAL	B-LOCAL
Iorque	I-LOCAL	E-LOCAL
,	O	O
nos	O	O
EUA	B-LOCAL	S-LOCAL
,	O	O
a	O	O
11	B-TEMPO	B-TEMPO
de	I-TEMPO	I-TEMPO
Janeiro	I-TEMPO	I-TEMPO
de	I-TEMPO	I-TEMPO
1842	I-TEMPO	E-TEMPO
.	O	O

Tabela 2.2: *Exemplo de uma sentença anotada nos esquemas IOB2 e IOBES.*

- O uso do prefixo **I** indica que o *token* está dentro (*inside*) de uma entidade anotada;

A diferença do esquema IOBES em relação ao IOB2 é que o IOBES é mais específico em relação a alguns casos. No IOBES, o prefixo **E** é usado para denotar o fim da anotação de uma entidade (*ending*) e o prefixo **S** é usado para entidades que possuem somente um *token* (*single*). A Tabela 2.2 mostra um exemplo de uma sentença anotada nos dois esquemas.

2.3 *Conditional Random Fields*

Conditional Random Fields [53] (CRF) é um método de modelagem estatística que é frequentemente aplicado em reconhecimento de padrões e aprendizado de máquina, em particular para classificação sequencial. Ao classificar sequências de palavras com CRF, o modelo aprende a correlação entre palavras e rótulos que ocorrem próximos uns dos outros, isto é, ele usa o rótulo das palavras vizinhas para determinar qual deve ser o rótulo de uma determinada palavra alvo. Como exemplo de classificação de REN usando o esquema de anotação IOB2, uma palavra não poderia ser classificada como *I-PESSOA* se vier logo depois de uma palavra da classe *O*. Isto porque **I** indica um *token* interno de uma entidade, e **O** indica um *token* que não é uma entidade, o que significa que qualquer *token* depois de um *O* só poderia ser outro *O* ou então o início de uma nova entidade, que deveria ser de uma classe iniciada com **B**.

Dada uma sequência de entrada representada por $\mathbf{x} = \{x_1, x_2, x_3, ..., x_n\}$, e uma sequência de saída $\mathbf{y} = \{y_1, y_2, y_3, ..., y_n\}$, x_i e y_i representam, respectivamente, o vetor de entrada e a classe do *i*-ésimo *token*. [61] usou as seguintes fórmulas para explicar a intuição por trás de um modelo probabilístico CRF, em que as entradas são os pesos **W** e os vieses **b** (*biases*) de uma rede neural:

$$p(\boldsymbol{y} \mid \mathbf{x}; \mathbf{W}, \mathbf{b}) = \frac{\prod_{i=1}^{n} \psi_i(y_{i-1}, y_i, \mathbf{x})}{\sum_{y\prime \in \Upsilon(\mathbf{x})} \prod_{i=1}^{n} \psi_i(y\prime_{i-1}, y\prime_i, \mathbf{x})} \tag{2-10}$$

$$\psi_i(y\prime, y, \mathbf{x}) = \exp(\mathbf{W}^T_{y\prime,y}\mathbf{x}_i + \mathbf{b}_{y\prime,\,y}) \tag{2-11}$$

$$L(\mathbf{W}, \mathbf{b}) = \sum_i \log(p(\boldsymbol{y} \mid \mathbf{x}; \mathbf{W}, \mathbf{b})) \tag{2-12}$$

$$\boldsymbol{y}^* = \underset{y \in \Upsilon(\mathbf{x})}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \mathbf{x}; \mathbf{W}, \mathbf{b}) \tag{2-13}$$

onde:

- $p(\boldsymbol{y} \mid \mathbf{x}; \mathbf{W}, \mathbf{b})$ é um conjunto de probabilidades condicionais das classes $\boldsymbol{y}$, dados os *tokens* da entrada **x**;
- $\Upsilon(\mathbf{x})$ representa o conjunto de todas as possíveis sequências de classes para a entrada **x**;
- *y′,y* são pares de classes obtidos do conjunto $\Upsilon(\mathbf{x})$;
- $\mathbf{W}^T_{y\prime,y}$ e $\mathbf{b}_{y\prime,\,y}$ são os vetores de peso e *bias* para o par *y′,y*, respectivamente, e eles representam a entrada necessária para calcular as probabilidades condicionais das classes;
- ψ_i na Equação (2-11) denota a função exponencial da entrada, usada em (2-10) para normalizar a probabilidade condicional de cada par de classes de $\Upsilon(\mathbf{x})$;
- (2-12) é a função de máxima verossimilhança que é maximizada durante o treino;
- (2-13) representa a busca pela sequência de classes $\boldsymbol{y}$ com a maior probabilidade condicional, dada a entrada **x**.

Como REN é uma tarefa de classificação sequencial, pois cada *token* é classificado levando em consideração os *tokens* da sequência, então o uso de CRF para classificação de entidades é importante pois ele maximiza a probabilidade de ser feita uma classificação que respeite as "regras" do esquema de anotação [85]. Caso fosse utilizada

uma função *softmax*[7], a mesma atribuiria uma probabilidade para cada classe sem levar em consideração as sequências de classes fornecidas pelo conjunto de treino.

2.4 Aprendizado Profundo para Reconhecimento de Entidades Nomeadas

As Redes Neurais Artificiais (RNA) vieram de uma inspiração biológica, na tentativa de criar um modelo matemático que fosse capaz de replicar o funcionamento de neurônios biológicos a partir de uma analogia com seus três elementos principais: dendrito, corpo celular e axônio [84]. Assim como um neurônio biológico recebe estímulos elétricos através de seus dendritos, um artificial recebe suas entradas, que são ponderadas, agregadas e processadas, da mesma forma como são processados os estímulos elétricos no corpo celular. O processamento das entradas de um neurônio artificial gera um potencial de ativação, que uma vez verificado pela função de ativação, produz uma saída que pode vir a sensibilizar camadas de neurônios seguintes. O mesmo comportamento é verificado nos neurônios biológicos, que podem ter seus estímulos elétricos propagados por seus axônios de acordo com a ação de neurotransmissores.

A Figura 2.1 mostra um exemplo de como funciona um neurônio artificial. Cada uma das entradas são representadas por um valor x_i, que são multiplicadas por pesos w_i e somados a um viés b_i (ou *bias*). O somatório da multiplicação das entradas pelos pesos, somados aos *biases*, é passado para a função de ativação, que calcula se o estímulo dessas entradas será propagado adiante na rede, de acordo com o valor resultante da função. Este tipo de neurônio é chamado de ***Perceptron***, e redes mais complexas, chamadas de ***Multilayer Perceptrons*** (MLP) são formadas por várias camadas, cada qual contendo vários perceptrons [40]. Nas redes MLP têm-se 3 tipos de camadas: a de entrada, as escondidas (ou ocultas) e a de saída, conforme exibido na Figura 2.2. Cada uma destas camadas possui um valor de entrada, sendo que a camada de entrada representa os valores de treino (representados por x_i na Figura 2.1); a primeira camada oculta recebe estes valores da camada de entrada; cada camada oculta seguinte recebe o valor da função de ativação dos neurônios da camada oculta anterior; e a camada de saída da rede emite os dados finais resultantes de todo o processamento da rede. Toda entrada de um neurônio está associada a um peso e um *bias* (representados por w_i e b_i na Figura 2.1), e estes pesos e os *biases* são os valores a serem aprendidos no processo de treino da rede neural.

[7]Função que distribui um vetor de números reais em uma distribuição de probabilidade entre 0 e 1.

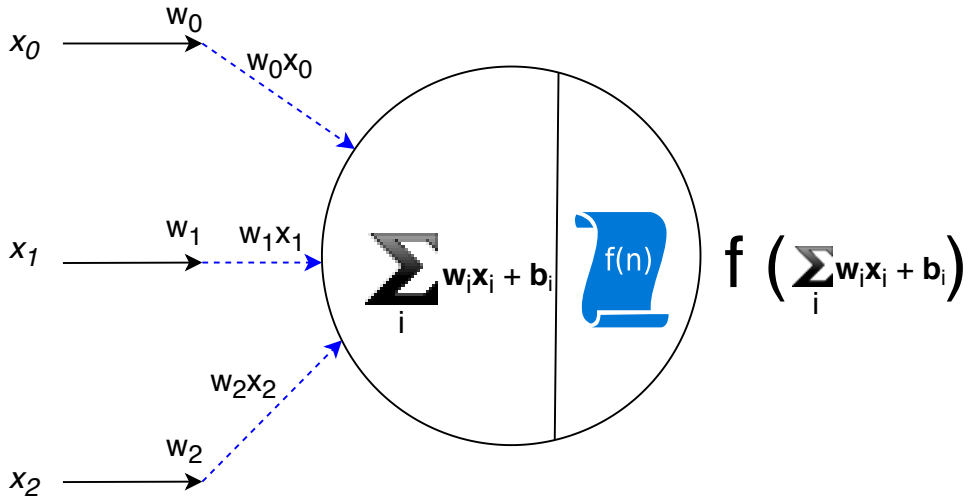

Figura 2.1: *Esquema de funcionamento de um neurônio artificial.*

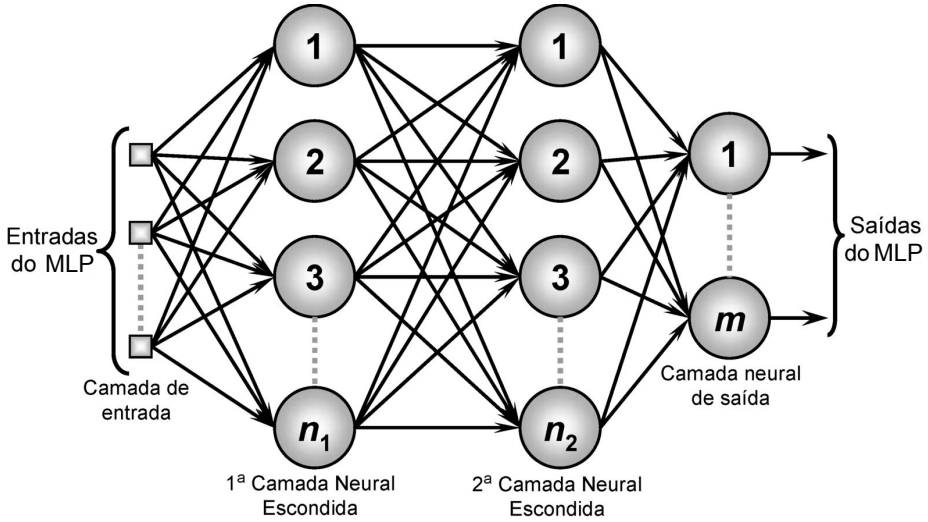

Figura 2.2: *Exemplo de uma rede MLP, obtido de [67]*

Uma época do processo de treino supervisionado de uma rede MLP pode ser resumida da seguinte forma:

- Os pesos **w** e os *biases* **b** são inicializados;
- São calculados todos os valores de somatório das multiplicações das entradas pelos pesos de cada neurônio, conforme exemplificado em 2.1, para todos os neurônios;
- Todos os valores calculados na etapa anterior são passados pela função de ativação do neurônio para determinar o valor que é propagado para a camada seguinte;
- Os valores de saída da rede são medidos em função do erro dos valores resultantes, em relação aos valores esperados, de acordo com os dados rotulados do conjunto de treino;
- Para que o erro seja reduzido, os valores dos pesos são ajustados;

O algoritmo que permite este ajuste dos pesos em redes MLP é chamado de ***Retropropagação*** ou ***Backpropagation*** [55]. Ele funciona a partir das derivadas parciais da função que calcula o erro que se deseja reduzir (também chamada ***função de custo***). A partir do cálculo destas derivadas, é calculado um gradiente, que indica em que sentido os pesos devem ser ajustados (para mais ou para menos). No algoritmo de retropropagação, este cálculo é realizado a partir da última camada, até a primeira camada. Isto significa que é calculado o erro da rede na última camada e, em seguida, a partir das derivadas parciais deste erro, calcula-se a contribuição de cada neurônio de cada camada para este erro, desde a última camada até a primeira. No entanto, como tipicamente as funções de ativação dos neurônios são funções que produzem valores pequenos (como a função logística, que produz valores entre 0 e 1, e a função tangente hiperbólica, que produz valores entre -1 e 1), o cálculo das derivadas parciais dos erros acaba produzindo valores muito reduzidos, fazendo com que o gradiente vá desaparecendo a cada camada. Este problema é chamado de ***Vanishing Gradient***, e ele mostrou a dificuldade de se treinar redes MLP com muitas camadas, pois depois de 1 ou 2 camadas o cálculo do gradiente para ajuste dos pesos desta forma é impossibilitado. Com isso, o treino de Redes Neurais Profundas (RNP) no Aprendizado Profundo (AP) só foi possível quando foram propostas soluções que permitiram o cálculo do gradiente em profundidade.

As primeiras arquiteturas de RNA baseadas em Aprendizado Profundo (AP) a serem aplicadas em tarefas de Processamento de Linguagem Natural (PLN) foram baseadas em Redes Neurais Convolucionais (***Convolutional Neural Networks - CNN***) [18, 26], e mais adiante começaram a ser desenvolvidos trabalhos baseados em Redes Neurais Recorrentes (***Recurrent Neural Networks - RNN***) [37, 44, 15, 54, 61]. Estas arquiteturas serão explicadas nas Seções 2.4.1 e 2.4.2.

O motivo pelo qual RNAs são uma alternativa para modelos que dependem de seleção de características é que elas mesmas realizam o aprendizado das características, durante um tipo de processo de treino em que ocorre *extração de características*. Este processo será descrito na Seção 2.4.3.

2.4.1 Redes Neurais Convolucionais

Apesar das Redes Neurais Convolucionais (CNN) terem sido teorizadas desde o início da década de 1980 [32], o treino delas só foi possível a partir de alguns avanços, como o algoritmo de retropropagação [55] de 1989, para ajuste dos pesos; e depois com o uso de uma função de ativação que amenizou o problema do *vanishing gradient*, a ReLU (*Rectified Linear Unit*) [46, 70, 52].

CNNs são redes neurais que têm o seu funcionamento baseado no que é chamado processo de ***Convolução***. A convolução foi concebida para que as redes neurais pudessem

aprender diferentes níveis de características, observando os dados de entrada do modelo em mais de uma dimensão. Isto é feito a partir de ***filtros convolucionais***, que funcionam da seguinte forma:

- Um filtro de uma determinada dimensão percorre os dados de entrada, usando um passo[8] de um tamanho escolhido;
- Para cada passo do filtro, é feita uma multiplicação dos dados de entrada, na mesma dimensão do filtro, pelos dados do próprio filtro;
- O valor resultante é armazenado em um mapa de características;
- Opcionalmente, uma operação de amostragem (*Pooling*) é aplicada no mapa de características resultante. Esta operação pode ser de máximo, mínimo, média, ou alguma outra operação. O *pooling* também é aplicado em passos, observando os dados de uma dimensão determinada;
- Os valores resultantes da operação de *pooling* são armazenados em outro mapa de características.

A Figura 2.3 mostra o processo de convolução até a etapa em que é aplicado um filtro de dimensão 3x3 e passo de tamanho 1. A Figura 2.4 mostra o exemplo de um processo de *pooling* usando a operação de *máximo* [93], observando uma janela de dados de dimensão 2x2 e passo de tamanho 2.

Normalmente são aplicados muitos filtros diferentes em uma mesma camada de dados, produzindo vários mapas de características diferentes, com o objetivo de aprender diferentes características. Caso a CNN seja utilizada em um processo de classificação, os mapas de características são concatenados e redistribuídos em uma camada unidimensional (*flattening*), para que os neurônios dessa camada possam ser conectados a uma camada para classificação, que normalmente é do tipo *softmax*. Neste contexto, a camada *softmax* aprende uma distribuição de probabilidade a partir dos resultados das convoluções, projetados na camada unidimensional.

[8] O tamanho do passo é o tamanho do deslocamento de um filtro, conforme ele se move no eixo horizontal da matriz de dados.

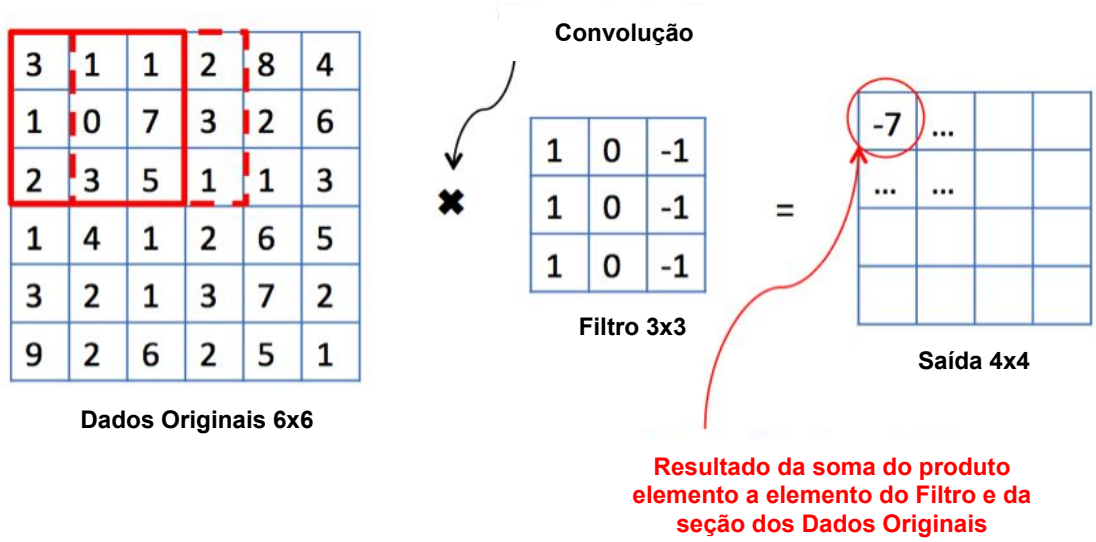

Figura 2.3: *Exemplo de uma convolução, adaptado e traduzido de [9]*

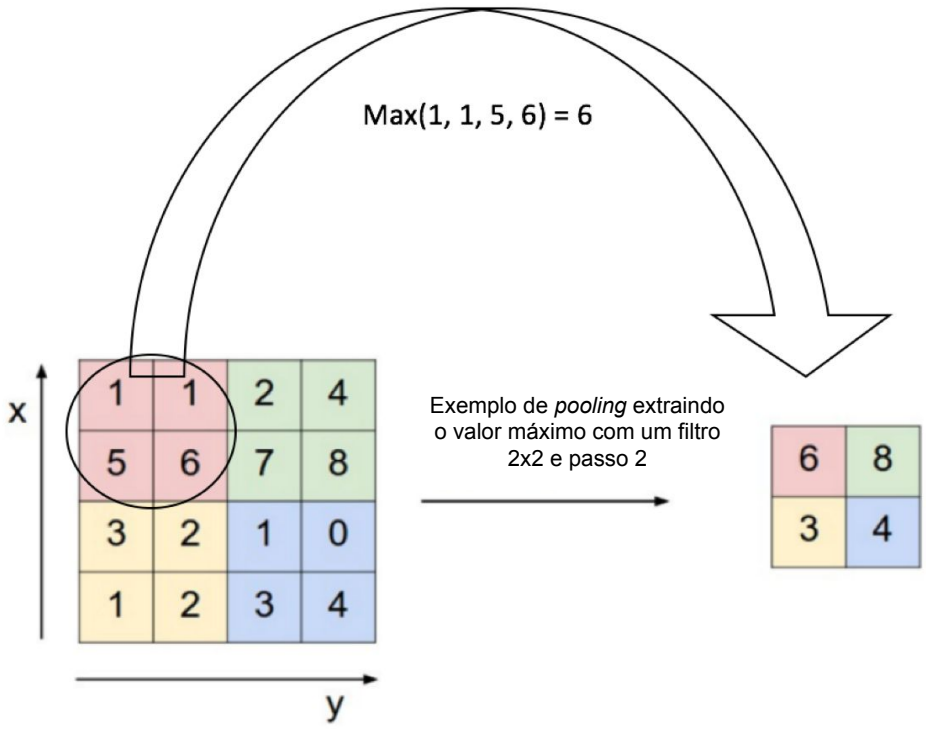

Figura 2.4: *Exemplo de uma operação de pooling, usando a função de máximo, adaptado e traduzido de [49]*

Em tarefas de PLN, redes CNN normalmente são utilizadas para o aprendizado de características de palavras em nível de caracteres. Arquiteturas que usam redes CNN

desta forma serão detalhadas na Seção 2.4.3.

2.4.2 Redes Neurais Recorrentes

Redes Neurais Recorrentes (RNNs) representam uma classe de RNP que são mais aplicáveis a dados sequenciais, como textos. Redes como MLP, e até mesmo CNNs, são limitadas no sentido de que elas recebem uma entrada de tamanho fixo, e produzem uma saída também de tamanho fixo. RNNs, por outro lado, suportam sequências de vetores, sendo aptas a receber entradas de tamanhos variáveis, e também produzindo saídas de tamanhos variáveis.

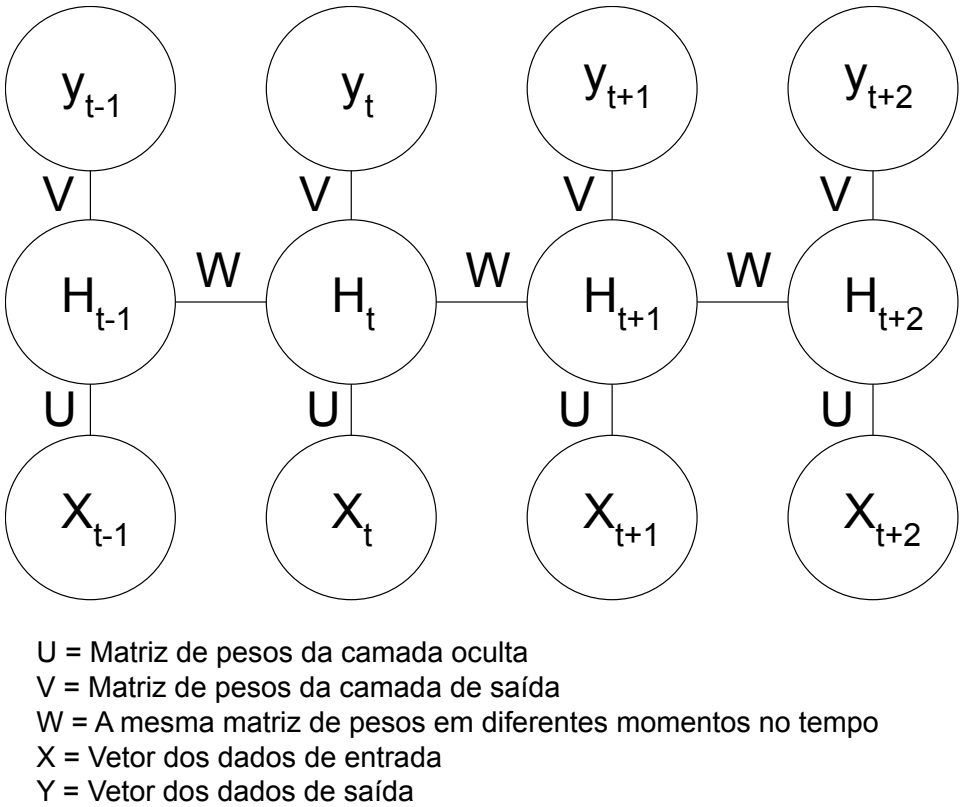

Figura 2.5: *Exemplo do desdobramento de uma RNN, adaptado e traduzido de [86]*

O desdobramento de uma RNN pode ser visualizado como na Figura 2.5. A particularidade dela em relação a outros tipos de RNAs é que ela tem o funcionamento semelhante a um laço temporal, de forma que ela percorre cada parte dos dados de entrada em uma iteração do laço, de forma sequencial. Em um dado momento t, a saída y_t é obtida a partir da entrada X_t e do estado interno da rede da iteração anterior H_{t-1}. A matriz de pesos W de uma RNN é a mesma para todas as iterações, enquanto as matrizes de pesos V e U, de cada entrada e cada saída, são diferentes em cada iteração.

Na teoria, RNNs foram concebidas para capturar dependências de longo prazo em sequências grandes, mas, na prática, isso não foi possível devido aos problemas de *vanishing* e *exploding gradient*[9] [6]. Para superar esta limitação, [41] propôs a rede ***Long Short-Term Memory*** (LSTM), um tipo de RNN na qual os neurônios das camadas ocultas são incrementados com três portões multiplicativos que controlam o esquecimento e a propagação da informação, a cada momento do tempo. Estes três portões são: portão de atualização, portão de esquecimento e portão de saída. As Equações (2-14) a (2-19) mostram as fórmulas utilizadas na atualização de uma unidade LSTM em um instante de tempo t.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{h}_{t-1} + \mathbf{U}_i\mathbf{x}_t + \mathbf{b}_i) \tag{2-14}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{h}_{t-1} + \mathbf{U}_f\mathbf{x}_t + \mathbf{b}_f) \tag{2-15}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c\mathbf{h}_{t-1} + \mathbf{U}_c\mathbf{x}_t + \mathbf{b}_c) \tag{2-16}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t\text{-}1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \tag{2-17}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{h}_{t-1} + \mathbf{U}_o\mathbf{x}_t + \mathbf{b}_o) \tag{2-18}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{2-19}$$

onde $\mathbf{i}_t$ representa o portão de atualização, $\mathbf{f}_t$ representa o portão de esquecimento e $\mathbf{o}_t$ representa o portão de saída, todos os três em um instante de tempo t. $\mathbf{c}_t$ e $\tilde{\mathbf{c}}_t$ representam o estado da célula e o estado candidato da célula da unidade LSTM, em um instante de tempo t. $\mathbf{W}$ representa as matrizes de pesos do estado oculto $\mathbf{h}$, $\mathbf{U}$ representa as matrizes de pesos da entrada $\mathbf{x}$, e $\mathbf{b}$ representa os vetores de *bias*. σ representa a função logística elemento a elemento e $\odot$ representa o produto elemento a elemento.

Em problemas de PLN, normalmente as redes LSTM são utilizadas de forma bidirecional [37, 44, 15, 54, 61, 43, 75, 1]. Como os dados de entrada são textuais, é utilizada uma rede LSTM para processar o texto da esquerda para direita, e outra rede LSTM para processar o texto da direita para esquerda. Desta forma, durante o processo de treinamento, a rede aprende com acesso a informação tanto do início quanto do fim da entrada, adquirindo uma contextualização maior.

[9] Problema semelhante ao *vanishing gradient*, mas neste caso ocorre a explosão ao invés do sumiço do gradiente, isto é, ao invés de ficar muito pequeno, ele fica grande demais.

2.4.3 Representações das Palavras

Conforme descrito no início deste capítulo, abordagens clássicas de REN precisavam fazer um trabalho de seleção de características que pudessem ser utilizadas nos treinos dos modelos desta tarefa. Além de ser uma tarefa que demanda muito tempo, ela também acaba produzindo um modelo que fica dependente das características experimentadas. Ademais, para que o mesmo modelo pudesse ser aplicado em outros idiomas, todo este ciclo de estudo linguístico, seleção de características e desenvolvimento do modelo precisaria ser refeito. O mesmo aconteceria caso fosse necessário um modelo para o mesmo idioma, mas para um domínio específico. À vista disso, modelos baseados em arquiteturas de RNA e Aprendizado Profundo oferecem uma alternativa para a representação de informações textuais na tarefa de REN.

Tipicamente, os modelos de representação de palavas baseados em aprendizado profundo são obtidos a partir de um pré-treino em uma quantidade massiva de texto. Um *corpus* muito utilizado para este fim é uma descarga completa do Wikipedia[10], que para a língua inglesa é um arquivo compactado de 16 gigabytes de informação[11], e para a língua portuguesa, de 1,6 gigabytes[12]. O uso de uma quantidade tão grande de texto permite o entendimento do contexto em que as palavras tendem a ocorrer [64]. Modelos treinados desta forma podem gerar representações de palavras tanto de forma estática, com vetores de palavras [64, 65, 74, 58, 7, 47], quanto de forma contextual, com modelos de linguagem [43, 75, 81, 23, 1].

As seções a seguir detalham os três tipos de representações de palavras avaliados neste trabalho: vetores de palavras, modelos de linguagem e representação por caracteres.

Vetores de Palavras

Vetores de palavras são vetores multidimensionais que representam características aprendidas automaticamente por meio de treino não-supervisionado. Estas características são latentes e representam informações morfológicas, sintáticas e semânticas acerca das palavras, distribuídas em cada dimensão do vetor. Os algoritmos usados para o treino destes vetores são baseados em modelos preditivos [64, 58, 7, 47] ou em modelos estatísticos [74], que consideram a co-ocorrência das palavras no *corpus*.

Word2Vec [64, 65], ***Wang2Vec*** [58] e ***FastText*** [7, 47] são modelos treinados de acordo com a predição de palavras em determinados contextos, em dois algoritmos diferentes. Um dos algoritmos é o ***Continuous Bag-of-Words (CBoW)***, que é treinado fazendo a predição de uma palavra alvo, dada uma lista de palavras ao redor da mesma

[10] `https://dumps.wikimedia.org/`
[11] `https://dumps.wikimedia.org/enwiki/latest/` acessado em 24/10/2019.
[12] `https://dumps.wikimedia.org/ptwiki/latest/` acessado em 24/10/2019.

(predição da palavra, dado o seu contexto). O outro algoritmo é o ***Skip-Gram***, que é treinado prevendo quais palavras estariam ao redor de uma determinada palavra (predição do contexto, dada a palavra). A arquitetura da rede neural utilizada por [64] é baseada em uma rede MLP, na qual a camada de entrada tem dimensão *V*, igual ao tamanho do vocabulário de treino. A entrada é projetada em uma camada oculta *h*, de dimensão *N*, onde *N* é o tamanho da dimensão dos vetores a serem treinados, ou seja, a quantidade de características latentes desejadas. A camada oculta, finalmente, é ligada a uma camada de saída que também tem o tamanho do vocabulário *V*. A Figura 2.6 exemplifica esta arquitetura para o algoritmo *Skip-Gram*, em que é usada a função *Softmax* na camada de saída para prever a distribuição de probabilidade das possíveis palavras **y** a estarem no contexto de uma palavra **X**.

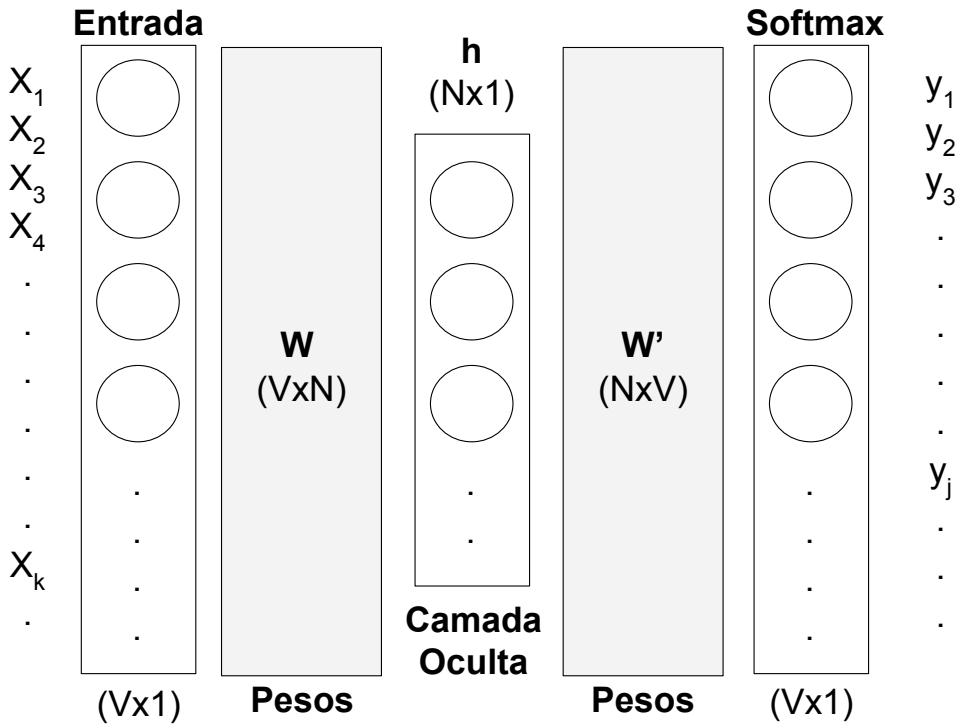

Figura 2.6: *Visualização da arquitetura do algoritmo* ***Skip-Gram****, adaptado e traduzido de [13]*

A Figura 2.7 ilustra a mesma arquitetura para o algoritmo *CBoW*. Nesta imagem é mostrado como as camadas de entrada e de saída são codificadas para maximizar a probabilidade de se prever, na camada de saída, qual a palavra **y** é inserida no contexto (ativada com o valor 1) das palavras de entrada **X** (também ativadas com o valor 1).

A diferença entre os algoritmos ***Word2Vec*** e ***Wang2Vec*** [58] é que o segundo fez uma modificação na arquitetura para levar em consideração a ordem das palavras, com o objetivo de melhorar o aprendizado de características sintáticas das palavras. No *Wang2Vec* foram propostos os modelos ***Structured Skip-Gram***, equivalente ao *Skip-*

Gram, e o ***Continuous Window***, equivalente ao *Continuous Bag-of-Words*, com ambos realizando alterações para que fosse considerada a ordem das palavras. Já o ***FastText*** [7, 47] se diferencia dos dois anteriores pois ele cria sub-representações das palavras a partir de n-gramas de seus caracteres, de forma que o vetor distribuído de cada palavra é obtido a partir da soma destes sub-vetores.

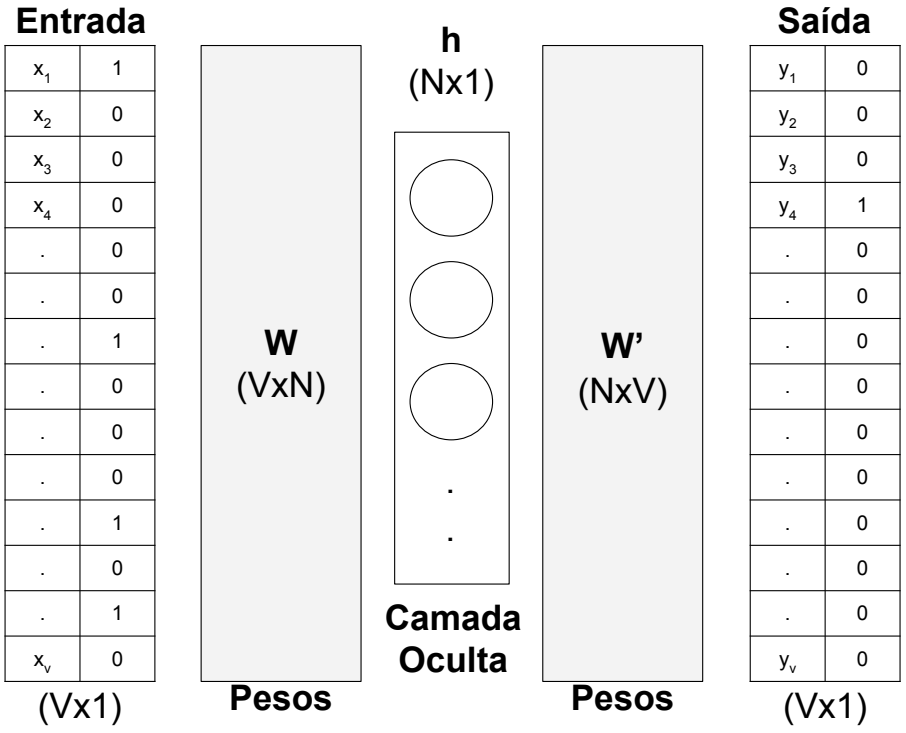

Figura 2.7: *Visualização da arquitetura do algoritmo* ***Continuous Bag-of-Words***

O ***GloVe*** [74] (*Global Vectors*) é um modelo que cria uma matriz quadrada de co-ocorrências de todas as palavras do *corpus*, contabilizando a quantidade de vezes em que cada palavra ocorre no mesmo contexto que as demais. O objetivo do modelo é tentar criar uma representação distribuída das palavras de forma que as características aprendidas reflitam a relevância entre palavras de acordo com as taxas de probabilidades condicionais de ocorrências entre elas. A Figura 2.8 mostra a matriz de co-ocorrência, e a Equação (2-20) mostra a função de regressão de quadrados mínimos que é minimizada para se obter a representação *GloVe* para as palavras do vocabulário de treino.

$$GloVe = -\sum_{i=1}^{V}\sum_{j=1}^{V} f(X_{\mathrm{i,j}})(\log X_{\mathrm{i,j}} - w_i^T w_{\mathrm{j}})^2 \tag{2-20}$$

onde V é o tamanho do vocabulário de treino; $f(X_{\mathrm{i,j}})$ é a função que calcula o peso da co-ocorrência das palavras i e j; $\log X_{\mathrm{i,j}}$ é a probabilidade de co-ocorrência das palavras i e j; e

$w_i^T w_j$ são os pesos da função do modelo que tenta prever a probabilidade de co-ocorrência entre as palavras.

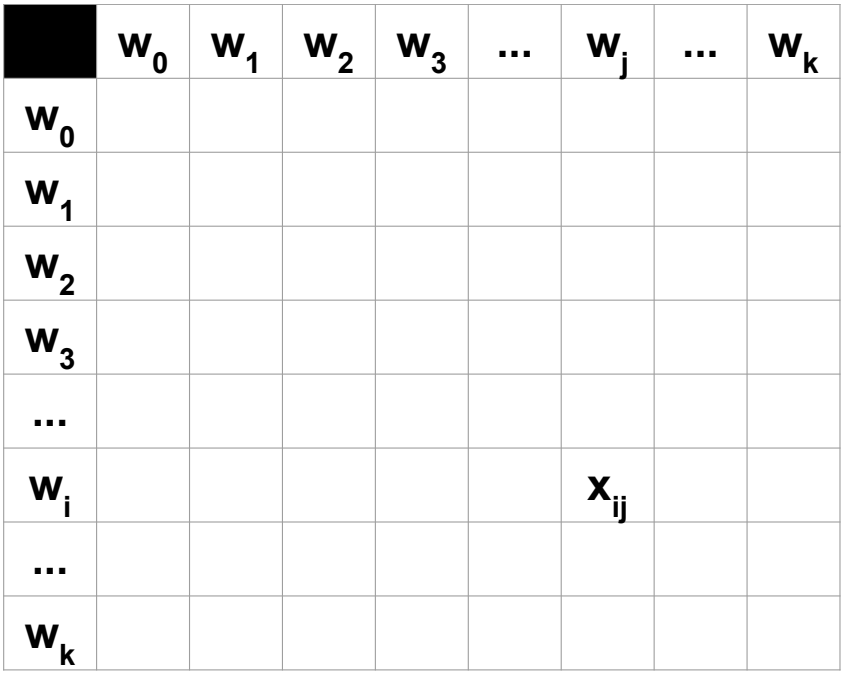

Figura 2.8: *Matriz de co-ocorrência do **GloVe**, adaptado e traduzido de [66]*

Com estas representações estáticas na forma dos vetores de palavras, é possível projetar as palavras em um plano da dimensionalidade dos vetores e efetuar operações de distância entre as mesmas, de forma que palavras que possuam um contexto semelhante sejam situadas próximas umas das outras. A Figura 2.9 ilustra essa característica dos vetores. As palavras equivalentes a nomes de países são projetadas em regiões próximas, enquanto suas respectivas capitais seriam projetadas em outra região, mas seguindo uma distribuição semelhante. Da mesma forma, é possível estabelecer analogias entre palavras, aplicando operações de soma e subtração dos vetores envolvidos. Usando como exemplo as mesmas palavras da Figura 2.9, seria possível obter o vetor da palavra ***Rússia*** a partir de operações sobre os vetores de ***China***, ***Pequim*** e ***Moscou*** da seguinte forma:

$$\overrightarrow{\textbf{China}} - \overrightarrow{\textbf{Pequim}} + \overrightarrow{\textbf{Moscou}} = \overrightarrow{\textbf{Rússia}}$$

Apesar das arquiteturas utilizadas nos vetores de palavras mencionados nesta seção não serem exatamente de redes neurais *profundas*, tais vetores vêm sendo mencionados no contexto de *Aprendizado Profundo* [35, 36]. Isto se deve à dimensionalidade da representação numérica distribuída que eles possibilitam para as características de palavras em modelos voltados para tarefas de PLN.

Figura 2.9: *Exemplo da projeção de vetores estáticos de palavras projetados a partir de uma redução a duas dimensões*

Modelos de Linguagem

Considerando $w_{1:n}$ uma sequência de palavras de tamanho n, um modelo de linguagem tenta fazer a predição da probabilidade condicional de uma palavra w, dada uma sequência de palavras que a antecederam. Esta predição pode ser representada por meio da Equação (2-21) [35].

$$P(w_{1:n}) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_{1:2})...P(w_n \mid w_{1:n-1}) \tag{2-21}$$

que pode ser reescrita em 2-22:

$$P(w_{1:n}) \approx \prod_{i=1}^{n} P(w_i \mid w_{1:i-1}) \tag{2-22}$$

Assim como as RNNs, modelos de linguagem também podem ser bidirecionais. Desta forma, o modelo faria duas predições diferentes. No sentido *direto* ele faz a previsão conforme indicada pela Equação (2-22), enquanto no sentido *inverso*, a previsão condicional é de que qual palavra teria ocorrido, dada uma sequência de palavras que a sucederam. Esta predição pode ser representada por meio da Equação (2-23):

$$P(w_{1:n}) \approx \prod_{i=1}^{n} P(w_i \mid w_{i+1:n}) \tag{2-23}$$

A métrica de avaliação de um modelo de linguagem é a ***Perplexidade***, que indica quão bom um modelo de probabilidade é em prever uma amostra desconhecida [35]. A perplexidade de um modelo de linguagem (***ML***) em um *corpus* de tamanho n pode ser

calculada pela Equação (2-24) [35]:

$$2^{-\frac{1}{n}\sum_{i=1}^{n}\log_2 ML(w_i|w_{1:i-1})} \tag{2-24}$$

o que significa que quanto maior forem as probabilidades do modelo ***ML*** de prever as palavras, então menor será a sua perplexidade. Portanto, quanto menor a perplexidade, melhor é o desempenho de um modelo de linguagem.

Em tarefas de processamento de linguagem natural, os modelos de linguagem têm sido utilizados como uma forma de representação contextualizada de palavras [43, 75, 81, 23, 82, 1]. Isso significa que os vetores de palavras usados como representação deixam de ser *valores estáticos* para serem obtidos a partir de uma *função*, capaz de fornecer representações *dinâmicas*. Para que essa representação contextual possa ser obtida, a função recebe não uma palavra, mas uma sentença completa, de forma que as representações de cada palavra da sentença sejam calculadas pelo modelo levando em consideração as outras palavras da sentença.

A Tabela 2.3 mostra um exemplo de similaridade de representações obtidas através de dois modelos diferentes: uma representação da palavra *play* obtida a partir do ***GloVe*** para a língua inglesa, e duas representações da palavra *play* obtidas a partir do modelo de linguagem ***biLM*** [75], também treinado para a língua inglesa. A representação obtida do *GloVe* é estática, e as representações mais próximas da palavra *play* foram todas mantidas em um contexto de jogos, o que significa que o sentido de *play* que o *GloVe* aprendeu foi no contexto de jogos. Já com as representações obtidas a partir do *biLM*, cada uma delas foi contextual, com sentidos diferentes, explorando a polissemia[13] da palavra. No primeiro exemplo, a sentença de exemplo usou *play* em um contexto de jogo, e a sentença que usou a palavra *play* com a representação mais próxima também foi em um contexto de jogo. Já no segundo exemplo, a sentença de exemplo foi no contexto de uma peça teatral, e a sentença mais próxima também foi no mesmo contexto.

O modelo ***biLM*** faz parte da solução proposta neste trabalho e sua arquitetura será mais detalhada no Capítulo 6.

Representação por Caracteres

Uma outra forma de representação de palavras muito adotada em arquiteturas de AP é aprender características de palavras em nível de caracteres. Esta forma de representação é eficiente no aprendizado de características morfológicas e ortográficas, captando formas das palavras tais como prefixos, sufixos e capitalização [26, 54]. Representação de palavras em nível de caracteres são capazes de fornecer uma alternativa para represen-

[13] Diferentes sentidos para a mesma palavra.

	Exemplo	Vizinhos mais Próximos
GloVe [74]	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM [75]	Chico Ruiz made a spectacular play on Alusik's grounder {. . . }	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch, as well as his all-round excellent play.
	Olivia De Havilland signed to do a Broadway play for Garson {. . . }	{. . . } they were actors who had been handed fat roles in a successful play, and had talent enough to fill the roles competently, with nice understatement.

Tabela 2.3: *Comparação de vizinhos mais próximos de representações para a palavra **play** obtidas do **GloVe** e do modelo de linguagem **biLM** [75]. Os exemplos foram aplicados em [75] para a língua inglesa, e como a idéia é mostrar representações contextuais enfatizando a polissemia das palavras, os textos não foram traduzidos.*

tação em nível de palavras para termos fora do vocabulário. Supondo que uma tarefa de PLN qualquer esteja trabalhando com uma representação de caracteres e outra de palavras, como o *Word2Vec*, caso o processo de treino precise lidar com uma palavra desconhecida pelo vocabulário do *Word2Vec*, a representação em nível de caracteres é capaz de prover uma representação eficiente e amenizar a falta da representação do vetor estático.

Estas características ortográficas são especialmente relevantes em uma língua tão morfologicamente rica como o Português. Por exemplo, caracteres como "ç" e vogais acentuadas são muito comuns no léxico deste idioma. No contexto de Reconhecimento de Entidades Nomeadas, o aprendizado de características referentes à capitalização das palavras é particularmente relevante, pois é comum que nomes próprios tenham a primeira letra de cada palavra na forma maiúscula.

Normalmente o aprendizado desta representação se dá através do uso de redes convolucionais [26, 15, 61, 50, 75], em um esquema que pode ser delineado da seguinte forma:

- Um vocabulário de palavras $\boldsymbol{V}$, de dimensão v, com todas as palavras encontradas no *corpus* de treino;
- Um vocabulário de caracteres $\boldsymbol{C}$, de dimensão c, com todos os caracteres encontrados no *corpus* de treino;
- Uma dimensão d escolhida para os vetores dos caracteres;
- Uma dimensão p escolhida para a projeção da representação final de uma palavra a partir da convolução dos caracteres;
- Uma matriz $\boldsymbol{C}^{\boldsymbol{c}}$ dos vetores dos caracteres de dimensão d x c;
- Uma palavra qualquer $w \in \boldsymbol{V}$, composta dos caracteres $\{c_1, c_2, c_3, ..., c_l\}$, em que l é o tamanho da palavra.

Neste esquema, a representação da palavra *w* de acordo com seus caracteres é iniciada a partir de uma matriz $\boldsymbol{C^w}$ de dimensão *d* x *l*, sendo que cada caractere de *w* tem uma representação de dimensão *d*. As representações dos caracteres são obtidas a partir de uma convolução na matriz $\boldsymbol{C^w}$. Neste ponto, cada arquitetura adota a quantidade de filtros a serem aplicados na convolução, bem como o tamanho dos mesmos. Em tarefas de PLN os filtros normalmente são de dimensão 1 x $\boldsymbol{F}$, onde $\boldsymbol{F}$ é um tamanho qualquer escolhido na modelagem. A estratégia de *pooling* também varia de acordo com a arquitetura. A Figura 2.10 ilustra o processo de obtenção da representação de uma palavra a partir do processo de convolução da matriz $\boldsymbol{C^w}$, em que w = *Brasil*, e são aplicados *p* filtros de dimensão 1 x *o*. Na Figura, é aplicado o processo de *pooling* para selecionar uma amostra de *p* características a partir dos mapas de características resultantes da convolução, formando a representação final da palavra *Brasil*.

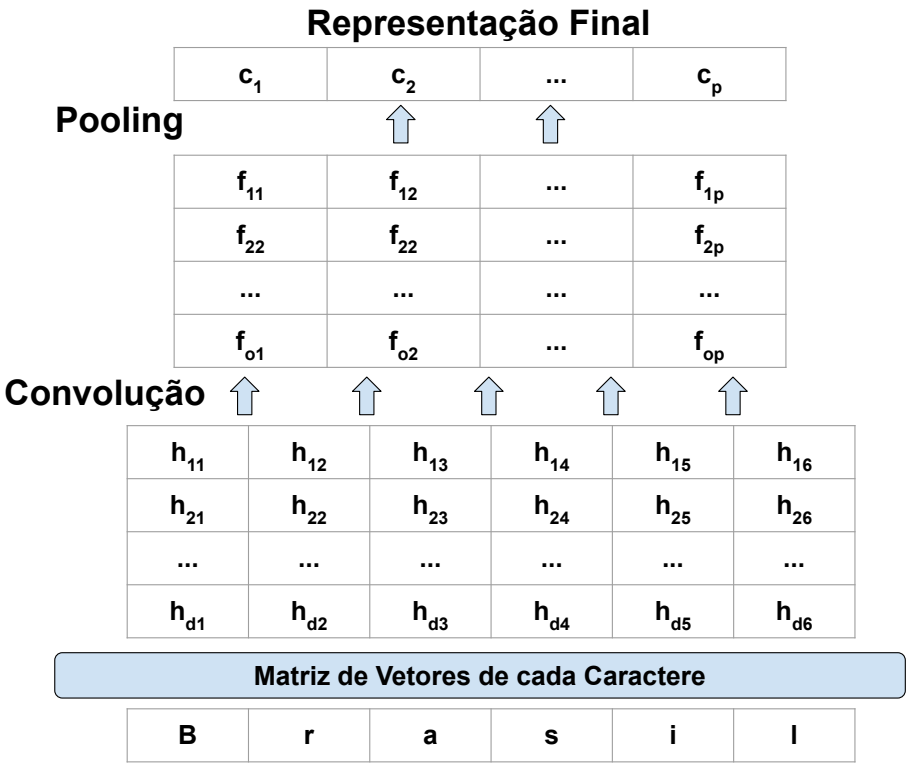

Figura 2.10: *Exemplo da obtenção de um vetor de características de caracteres a partir de uma rede convolucional.*

CAPÍTULO 3

Trabalhos Relacionados

Neste trabalho são propostos modelos de Reconhecimento de Entidades Nomeadas para a língua portuguesa tanto em domínio geral, quanto em domínio específico do Direito, voltado para a Justiça do Trabalho do Brasil. Os modelos propostos serão baseados em arquiteturas de Aprendizado Profundo. Dessa forma, este capítulo está organizado em três seções: na Seção 3.1 serão apresentados trabalhos relacionados de REN e de representação de palavras baseados em arquiteturas de Aprendizado Profundo; na Seção 3.2 serão apresentados os trabalhos relacionados de REN para a língua portuguesa; e na Seção 3.3 serão descritos trabalhos existentes no domínio de Direito, voltados para extração de informações.

3.1 Reconhecimento de Entidades Nomeadas baseado em Aprendizado Profundo

Uma das principais vantagens de se usar uma arquitetura baseada em AP, que use representações distribuídas em nível de palavras e caracteres para o treino de modelos de REN é a independência do modelo de características específicas de linguagens, uma vez que as representações distribuídas são aprendidas automaticamente. Portanto, é possível usar a mesma arquitetura para treinar modelos de REN para diferentes línguas e domínios, contanto que seja fornecido um *corpus* para cada idioma ou domínio, assim como os recursos necessários para as representações das palavras, sejam vetores estáticos ou modelos de linguagem pré-treinados. [26] usou a mesma rede neural para treinar modelos para Português e Espanhol, e [54] treinou modelos para Inglês, Holandês, Alemão e Espanhol.

Dos Santos e Guimarães [26] propuseram a arquitetura ***CharWNN***, baseada no modelo de Collobert et al. [18], que é uma arquitetura de rede neural para classificação sequencial, concebida para diferentes tarefas de PLN. [18] desenvolveu uma arquitetura de rede *Multilayer Perceptron* que recebe sentenças como entradas e aprende várias camadas de extração de características. A primeira camada aprende características e produz repre-

sentações para cada palavra. A segunda camada extrai características de múltiplas palavras, tratando a entrada como uma sequência, considerando a ordem das palavras. Para a segunda camada, os autores usaram duas abordagens: uma de janela, e outra de convolução de sentenças. A idéia desta camada é produzir decisões de classificação para cada palavra da entrada, de acordo com o esquema IOB [85]. A abordagem de janela considera as características das palavras vizinhas, em uma janela de tamanho fixo, concatenando-as e fornecendo-as para camadas de classificação. A abordagem de sentença extrai as características da sentença inteira por meio de uma camada convolucional. O modelo de [18] foi salvo e disponibilizado como um vetor estático de palavras de dimensão 50 chamado ***SENNA***[1].

Depois dos trabalhos baseados em arquiteturas MLP e CNN de [18, 26], os próximos trabalhos de REN foram mais voltados para arquiteturas baseadas em Redes Neurais Recorrentes bidirecionais – também mencionados neste trabalho como *biLSTM* – , com algumas diferenças arquiteturais entre eles. Huang et al. [44] usou uma rede biLSTM, com uma seleção de características ortográficas definidas manualmente, concatenadas a vetores de palavras do *SENNA* [18]. Para classificação sequencial das palavras foi utilizada uma camada CRF. Chiu e Nicols [15] utilizaram uma rede biLSTM sem utilizar CRF para classificação, e experimentaram usar representações em nível de caracteres a partir de uma rede CNN. Também experimentaram os vetores de palavras pré-treinados *GloVe* [74], *Word2Vec* [64] e *SENNA*, tendo obtido o melhor modelo com este último. Além destas características, [15] também utilizou características adicionais de capitalização definidas manualmente. Lample et al. [54] e Ma e Hovy [61] utilizaram abordagens bem semelhantes baseadas em biLSTM-CRF, com a diferença de que em [54] foi utilizada uma biLSTM para capturar características das palavras em nível de caracteres e *Word2Vec* [64] de 100 dimensões, enquanto [61] usou uma rede CNN para as características em nível de caracteres e experimentou vetores de palavras com *GloVe* de 100 dimensões, *SENNA* de 50, *Word2Vec* de 300 e um vetor inicializado aleatoriamente de 100 dimensões, tendo obtido melhores resultados com o *GloVe*.

Com estes trabalhos, notou-se que o padrão arquitetural de modelos de REN (assim como de outras tarefas de PLN de classificação de sequências) baseado em redes LSTM bidirecionais virou uma referência, e o foco de outros trabalhos que experimentaram com esta tarefa passou a ser não na arquitetura do modelo de REN, mas na de representação das características das palavras, em especial as que são baseadas em modelos de linguagem.

Peters et al. [75], Devlin et al. [23] e Akbik et al. [1] desenvolveram arquiteturas diferentes de representação contextual de palavras, baseadas em modelos de linguagem,

[1] `https://ronan.collobert.com/senna/`

e avaliaram o desempenho de seus modelos na tarefa de REN, assim como em outras tarefas de PLN. [75] e [1] utilizaram uma arquitetura biLSTM-CRF de referência para avaliar as representações obtidas, enquanto [23] avaliou a tarefa de REN acrescentando somente uma camada neural ao seu modelo de linguagem para realizar as classificações das sequências de palavras. O modelo de linguagem ***biLM*** usado por [75] é baseado em 2 redes biLSTM, com 2 camadas cada. A entrada deste modelo é uma representação das palavras em nível de caracteres, produzida por uma rede CNN com 2048 filtros convolucionais, de tamanhos unidimensionais variando de 1 a 7. [23] criou um modelo baseado na arquitetura ***Transformer*** [92], que é baseada no mecanismo neural de ***atenção***. Tal mecanismo procura fazer com que o aprendizado da rede descubra as informações mais relevantes dos dados de treino, e Devlin et al. criou um modelo de linguagem bidirecional cujo treinamento é baseado em vários níveis diferentes de atenção. [1] criou uma representação contextual de palavras baseada em um modelo de linguagem bidirecional em nível de caracteres, sendo que o objetivo do seu modelo não é fazer predição de palavras, dado as suas antecedentes, mas fazer predição de *caracteres*, dado os caracteres que o antecederam. A arquitetura deste modelo de linguagem também é baseado em uma rede biLSTM.

A Tabela 3.1 lista os trabalhos apresentados nesta seção, com seus respectivos valores publicados de *F-Score* obtidos no *benchmark* do CoNLL-2003 [91]. Como ressalvas destes resultados publicados, [1] e [15] usaram tanto o conjunto de treino quanto o de validação para o treino de seus modelos, enquanto os outros trabalhos usaram somente o conjunto de treino. Além disso, [1] também não utilizou o *script* oficial do CoNLL-2003 para calcular o seu *F-Score*, não usando os mesmos critérios de correspondência exata da avaliação.

Trabalho	***F-Score***	**Ano**
Akbik et al. [1]	93,09%*	2018
Devlin et al. (*BERT Large*) [23]	**92,80%**	2018
Devlin et al. (*BERT Base*) [23]	92,40%	2018
Peters et al. [75]	92,22%	2018
Chiu e Nichols [15]	91,62%*	2016
Ma e Hovy [61]	91,21%	2016
Lample et al. [54]	90,94%	2016
Huang et al. [44]	90,10%	2015
Collobert et al. [18]	89,59%	2011

Tabela 3.1: *Trabalhos de REN avaliados na língua inglesa no* ***benchmark*** *do* ***CoNLL-2003****. * Usaram tanto o conjunto de treino quanto o de validação para o treino de seus modelos, enquanto os outros trabalhos usaram somente o conjunto de treino.*

3.2 Reconhecimento de Entidades Nomeadas para a Língua Portuguesa

O primeiro trabalho de REN para a língua portuguesa baseado em AP foi a arquitetura *CharWNN* proposta por Dos Santos e Guimarães em [26]. Esta arquitetura foi uma adaptação da abordagem de janela utilizada por [18], de forma que os autores adicionaram uma camada convolucional para extrair uma representação em nível de caracteres para as palavras do *corpus* de treino. Além desta representação, os autores também utilizaram vetores de palavras a partir do próprio pré-treino não-supervisionado que foi feito usando o algoritmo de *Skip-Gram* de [64], usando a ferramenta do Word2Vec[2]. Para o pré-treino destes vetores foram utilizados uma descarga do Wikipedia em Português[3], o *corpus* CETENFolha[4] e o *corpus* CETEMPúblico[5]. [26] experimentou em dois cenários de categorias do HAREM: total e seletivo. Para o *total* foram usadas as 10 categorias anotadas nos *corpora* do HAREM, enquanto no cenário *seletivo* foram consideradas somente as 5 categorias mais representativas[6]: Pessoa, Organização, Local, Tempo e Valor.

Até o trabalho [26], os trabalhos de REN para o Português eram baseados em algoritmos de aprendizado de máquina clássico, sistemas de regras e seleção de características de forma manual. Dos Santos e Guimarães usaram a arquitetura *CharWNN* para superar resultados que haviam sido apresentados por Dos Santos e Milidiú em [27], no qual foi usado um modelo chamado ***ETL - Entropy Guided Transformation Learning***, baseado em regras e características selecionadas manualmente.

Do Amaral [24] criou um modelo baseado em CRF e usou 17 características diferentes para cada palavra do seu *corpus* de treino, tais como etiquetagem morfossintática, capitalização, e as próprias palavras, considerando uma janela de contexto de tamanho 2[7]. Em [25], Do Amaral usou a mesma arquitetura usada em [24] em um *corpus* criado para o domínio da Geologia. Neste trabalho foram usadas novas características, tais como prefixos, sufixos e *gazetteers* específicos do domínio estudado.

Em [77], Pirovani usou uma abordagem híbrida de aprendizado de máquina com características linguísticas manualmente definidas, propondo um modelo baseado em CRF e Gramáticas Locais (***Local Grammars - LG***).

Da Costa e Paetzold [21] utilizou uma abordagem baseada em rede LSTM bidirecional com CRF, usando como representação das palavras vetores pré-treinados

[2] `https://code.google.com/archive/p/word2vec/`

[3] `https://dumps.wikimedia.org/ptwiki/latest/`

[4] `https://www.linguateca.pt/cetenfolha/`

[5] `https://www.linguateca.pt/cetempublico/`

[6] Em termos de quantidades de entidades anotadas por categoria.

[7] Isto é, o vetor de características de uma palavra considerava as características da própria palavra, assim como as de cada uma de suas vizinhas imediatas à sua esquerda e à sua direita.

baseados no algoritmo do *FastText*, combinados a um vetor distribuído em nível de caracteres obtido por meio de redes biLSTM. A diferença da abordagem de [21] e da arquitetura biLSTM-CRF proposta por [54] é que em [21] foi feito a concatenação das representações das palavras (em nível de palavra e caractere) e a alimentação das mesmas em uma rede LSTM antes de fazer a classificação na camada CRF; enquanto [54] aplicou as redes biLSTM nas duas representações para depois concatená-las.

Utilizando como referências os *corpora* disponíveis do HAREM [87, 73, 68] e o *benchmark* do CoNLL-2003 [91], os resultados para a língua portuguesa ainda estão em um nível bem inferior aos obtidos para idiomas como Inglês ou Espanhol. Enquanto o melhor resultado para a língua inglesa é de 92.80% para o *F-Score* (Tabela 3.1), os melhores resultados reportados para a língua portuguesa são 69.14% para o cenário total de categorias e 71.23% para o cenário seletivo (conforme definido em [26]), usando o mesmo *script* CoNLL. Considerando as diferentes formas de usar os *corpora* disponíveis do HAREM, assim como os diferentes *scripts* de avaliação dos resultados [62], pode-se dizer que não há um *benchmark* padronizado para Português, sendo difícil fazer comparações exatas entre as diferentes abordagens de REN para este idioma. A Tabela 3.2 mostra diferentes resultados de modelos de REN para a língua portuguesa, reportados pelos seus autores.

Trabalho	***Corpus* de Treino**	***Corpus* de Teste**	***Script* de Avaliação**	**Cenário**	***F-Score***
Dos Santos e Milidiú [27]	HAREM I	MiniHAREM	SAHARA	Total	63.56%
				Seletivo	70.72%
Do Amaral [24]	HAREM I	HAREM II	SAHARA	Total	48.43%
Dos Santos e Guimarães [26]	HAREM I	MiniHAREM	SAHARA	Total	**71.41%**
				Seletivo	**77.93%**
			CoNLL	Total	65.41%
				Seletivo	**71.23%**
Da Costa e Paetzold [21]	HAREM I	MiniHAREM	CoNLL	Total	**69.14%**
Pirovani [77]	HAREM I	MiniHAREM	CoNLL	Seletivo	60.36%

Tabela 3.2: *Resultados reportados em diferentes configurações de avaliações realizadas nos corpora do HAREM. São destacados aqui os melhores resultados de cada cenário, em cada script de avaliação.*

3.3 Extração de Informações na Área do Direito

Muitos trabalhos foram desenvolvidos para extração de informações legais. Em [80], Quaresma usou Máquinas de Vetores de Suporte (***Support Vector Machines - SVM***) para identificar conceitos legais, e também desenvolveu um sistema de REN baseado em

análise sintática e regras gramaticais, com o objetivo de criar sistemas de recuperação de informações em níveis mais detalhados. Este sistema foi baseado em uma árvore de estrutura sintática de sentenças que traduzia os atributos morfossintáticos das palavras em categorias de locais, organizações, datas e referências a documentos e artigos.

Savelka [88] desenvolveu um sistema de REN baseado em CRF para identificar diferentes tipos de agentes em documentos legais, tais como advogados, juízes, peritos, testemunhas, partes, júri, legisladores e tribunais. Posteriormente, o mesmo autor usou CRF em [89] para segmentar documentos em partes funcionais e contextuais, visando a compreensão de decisões judiciais.

Em [11], Chalkidis criou modelos de extração de elementos de contratos legais usando RNP, que foram mais eficientes em relação aos que ele já havia desenvolvido em [12], usando Regressão Logística (RL) e SVM. Em [12], Chalkidis havia criado um *corpus* para *benchmark*, constituído de 3.500 contratos na língua inglesa, anotados com 11 tipos de elementos: Título, Partes, Data de Início, Data Efetiva, Data de Término, Período, Valor, Legislação, Jurisdição, Referências para Legislações e Titulos de Cláusulas. Neste trabalho também foram produzidos e utilizados vetores de palavras de dimensão 200 treinados com *Word2Vec* [64], em um *corpus* de cerca de 750.000 contratos. Em [11], o autor usou o mesmo *benchmark* e vetores de palavras do trabalho anterior, e experimentou com diferentes arquiteturas de biLSTM: uma biLSTM com uma camada de classificação baseada em RL, uma biLSTM com 2 camadas com classificação por RL, e uma biLSTM com classificação por CRF. Seus melhores resultados foram obtidos com as duas últimas configurações, cada uma em um cenário diferente. Nos dois trabalhos, Chalkidis define zonas de extração de determinados elementos, através de expressões regulares, para usar como características do seu modelo, buscando induzir o aprendizado de cada tipo de elemento anotado.

Em [3, 4, 8, 28] foram desenvolvidos sistemas de REN e vinculação de entidades para extrair diferentes tipos de entidades relevantes para o meio jurídico e enriquecer ontologias legais. Angelidis [3] treinou seus próprios vetores de palavras de dimensão 100 usando *Word2Vec* [64] em um acervo jurídico grego, e também experimentou algumas características manuais para capitalização das palavras. Os tipos de entidades identificadas e classificadas em [3] compreendem legislações, pessoas, organizações, entidades geopolíticas, marcos geográficos e referências a documentos públicos. Angelidis também experimentou com as mesmas variações de biLSTM que foram propostas por Chalkidis [11], e no seu cenário, a arquitetura que produziu os melhores resultados foi a biLSTM de 2 camadas, com RL para classificação.

Para a legislação brasileira, Luz de Araújo et al. [60] propôs o LeNER-Br, um *corpus* construído a partir de 70 documentos legais. Destes documentos, 4 são legislações e 66 são documentos de processos coletados a partir do Supremo Tribunal Federal (STF),

Superior Tribunal de Justiça (STJ), Tribunal de Justiça de Minas Gerais (TJ-MG) e Tribunal de Contas da União (TCU). O autor também propôs um modelo de REN baseado na arquitetura biLSTM-CRF de [54] para estabelecer um *benchmark* de referência para o *corpus* proposto. Seu modelo utilizou como vetores de palavras um *GloVe* para a língua portuguesa disponibilizado por [39].

Neste trabalho será desenvolvido um modelo de REN baseado na arquitetura biLSTM-CRF para o domínio específico da justiça trabalhista brasileira. Para este modelo, serão avaliados diferentes algoritmos de vetores estáticos de palavras. Além destes vetores, também serão experimentadas representações de palavras em nível de caracteres e representações contextuais a partir de um modelo de linguagem. Também serão produzidos dois *corpora* a partir de um acervo de documentos desta esfera da justiça: um para treino do modelo de REN e outro para treino dos vetores estáticos e modelo de linguagem. A seleção de categorias de entidades a serem etiquetadas no *corpus* de REN visa produzir uma ontologia que possa ser utilizada em análises estatísticas.

CAPÍTULO 4

Estudo do Domínio

Em Ciência da Computação, uma ontologia é definida por [38] como sendo a especificação de um conjunto de conceitos e relações que sejam relevantes para a modelagem de um domínio, definindo um vocabulário que represente o conhecimento do domínio em questão. Neste Capítulo, será descrito o domínio do Direito, no contexto da Justiça do Trabalho do Brasil, que será o objeto da criação de uma ontologia abastecida pelo modelo de Reconhecimento de Entidades Nomeadas resultante deste trabalho. Vale ressaltar que a criação desta ontologia não faz parte do escopo deste trabalho, mas é retratada aqui como uma possível aplicação prática do modelo de REN que de fato é produto deste trabalho. Este modelo será treinado em um *corpus* formado por documentos públicos obtidos a partir de processos trabalhistas, que serão explicados na Seção 4.1. As categorias selecionadas para anotação neste *corpus* serão descritas na Seção 4.2.

4.1 Direito e a Justiça do Trabalho Brasileira

Reale [83], define *Direito* como o conjunto de regras obrigatórias às quais uma sociedade está sujeita, impondo limites aos seus membros e garantindo o convívio social entre eles. A primeira forma como a ciência do Direito foi dividida foi entre *Público* e *Privado* [83]. No Direito Público, existem duas divisões que são escopo deste trabalho: *Direito Processual* e *Direito do Trabalho*.

O Direito Processual retrata o Estado como um prestador de serviços à sociedade, na medida em que deve exercer um papel de conciliador em conflitos que ocorrem entre os seus membros [83]. Desta forma, o objetivo do Direito Processual é definir bem a forma como o Estado deve desempenhar o seu papel, por meio de um sistema de regras e princípios a serem observados e cumpridos. Este sistema é regido por um conjunto de procedimentos conhecido como ***processo***. O restante deste Capítulo descreverá os elementos e procedimentos de um processo, no contexto de uma ação do Direito do Trabalho.

De acordo com Reale [83], o Direito do Trabalho é mais uma manifestação do Direito Público, cujo foco é regulamentar as relações entre empregadores e empregados.

No Brasil, o Estado se manifesta no Direito do Trabalho na figura da ***Justiça do Trabalho***, que foi organizada em 1943 através da *Consolidação das Leis do Trabalho* (CLT) [5]. Basile [5] assim descreve a divisão da jurisdição[1] na Justiça do Trabalho em *instâncias* (ou *graus*):

Primeira Instância: *Juízes* do Trabalho que atuam em ***Varas do Trabalho*** (VT), sendo que cada vara é composta por um juiz titular e um juiz substituto (este segundo, se o orçamento do tribunal assim permitir).

Segunda Instância: ***Tribunais Regionais do Trabalho*** (TRT), distribuídos em 24 *regiões* pelo território nacional, cada um composto por, no mínimo, 7 *juízes*.

Instância Extraordinária: ***Tribunal Superior do Trabalho***, composto por 27 *ministros*.

Basile [5] define os elementos de uma ação trabalhista da seguinte forma:

- *Partes* são os sujeitos envolvidos no conflito: o que se sentiu prejudicado e iniciou a ação, contra quem teria lhe prejudicado, que é alvo da ação. Estes dois sujeitos são também chamados de *polo ativo* e *polo passivo* do processo, respectivamente. No contexto de reclamações trabalhistas (outra nomenclatura para a ação trabalhista ordinária), estes papéis também são normalmente referidos como *reclamante* e *reclamado*.
- *Pedido*, também chamado de *objeto* da ação. O pedido de uma ação representa um direito do polo ativo do processo, conferido a ele pela legislação trabalhista, e que de alguma forma lhe teria sido negado pelo polo passivo. É possível realizar vários pedidos em um único processo, conforme estabelecido no artigo 292 do *Código Processual Civil* (CPC) [20].
- *Causa de pedir*, que deve ser descrita em forma de fatos e ***fundamentos jurídicos*** que justifiquem e baseiam o pedido. Esta fundamentação pode ser feita de forma argumentativa, apontando e descrevendo ilicitudes provocadas pelo polo passivo do processo, assim como por meio de citação de ***dispositivos legais*** (tais como legislação e jurisprudência) que constituem as regras do Direito do Trabalho relevantes na ação.

Existem ainda outras pessoas envolvidas em um processo trabalhista, desempenhando outros papéis, além do ***juiz*** e das ***partes***. As partes são representadas por seus ***advogados***, que possuem o conhecimento acerca do trâmite do processo, para orientar seus clientes na conduta da ação. ***Testemunhas*** contribuem para a fundamentação das partes, para que possam afirmar ou refutar alguma alegação das mesmas. ***Peritos*** desempenham

[1] Representação do Estado por meio de um juiz em um processo [5].

a função de fazer uma validação assertiva e embasada de alguma prova ou alegação, provendo subsídio para que o juiz tome sua decisão.

Ao iniciar uma ação trabalhista, a parte reclamante deve apresentar, por meio do advogado que a representa, um documento chamado de ***petição inicial***. A petição deve enumerar os pedidos e a fundamentação de cada um, indicando também a parte reclamada que lhe teria causado prejuízo. De acordo com o artigo 319 do CPC [20], na petição inicial deve ser definido o valor de cada pedido pretendido, seja este valor preciso ou estimado, de acordo com o cenário. A soma de todos os valores pedidos é chamado de ***valor da causa***. Ao tomar sua decisão, o juiz defere ou não cada pedido que tenha sido feito pelo reclamante. Da mesma forma, mesmo para pedidos que tenham sido deferidos, pode ser que o valor concedido para cada um não seja correspondente ao valor pedido. À soma dos valores deferidos pelo juiz para cada pedido, dá-se o nome de ***valor de condenação***. Em qualquer momento do processo as partes podem se conciliar e chegar a um acordo, que deve ser homologado pelo juiz. Ao valor acordado entre as partes, a ser pago pelo reclamante ao reclamado, dá-se o nome de ***valor do acordo***. Há ainda o ***valor de custas processuais***, que é correspondente à soma das despesas decorrentes da tramitação do processo, devida ao Poder Judiciário, pela prestação do serviço público. A Justiça do Trabalho define o cálculo do valor das custas no artigo 789 da CLT [16].

De acordo com Basile [5], os julgamentos proferidos por juízes são descritos em ***sentenças***, enquanto os julgamentos proferidos por tribunais são documentos chamados de ***acórdãos***. Assim como a petição inicial, as decisões dos juízes e tribunais também devem ser devidamente fundamentadas, por meio de argumentação e citação de dispositivos legais. O artigo 93, inciso IX da *Constituição Federal* (CF) [10] indica que os processos podem ser anulados caso as decisões não sejam bem fundamentadas pelos responsáveis. Durante o trâmite do processo, as audiências realizadas entre as partes são mediadas por juízes do trabalho e documentadas em ***atas de audiência***.

4.2 Determinação das Entidades Jurídicas e suas Classes

Dada esta contextualização teórica acerca de como funciona a justiça trabalhista, e com todos estes dados pertinentes ao cenário de um processo do trabalho, nesta seção serão descritas as informações que serão extraídas pelo modelo de Reconhecimento de Entidades Nomeadas que será aplicado neste domínio.

Conforme mencionado no início deste Capítulo, uma possível aplicação do modelo de extração de informações desenvolvido neste trabalho seria a criação de uma ontologia para a Justiça do Trabalho do Brasil. No Direito, existe uma disciplina que descreve a aplicação de modelos estatísticos em informações jurídicas, chamada de

Jurimetria. O objetivo da Jurimetria é mapear o comportamento da justiça por meio de uma análise quantitativa de dados jurídicos [45].

A seleção de categorias de entidades a serem etiquetadas no *corpus* produzido neste trabalho visa produzir uma ontologia que possa ser utilizada em análises jurimétricas no escopo da justiça trabalhista. Como o objetivo da Jurimetria é mapear comportamento judicial, as informações extraídas pelo modelo de REN devem apoiar este tipo de análise. Uma possível análise é mapear como juízes e advogados usam dispositivos legais em suas fundamentações no âmbito de um processo trabalhista. Também seria possível fazer uma análise quantitativa em relação aos valores descritos neste cenário, tais como:

- Valor médio de causa dos processos de cada unidade federativa do Brasil;
- Valor médio de condenação arbitrado por cada juiz, vara ou tribunal;
- Valor médio de acordos de processos em que houve conciliação entre as partes;
- Valor médio de custas processuais, e como elas impactam no custo do sistema judiciário enquanto mediador de conflitos trabalhistas.

A partir destas análises, optou-se por definir as classes de entidades abaixo, a serem identificadas pelo modelo de REN proposto. Os tipos de documentos que terão estas classes etiquetadas neste trabalho serão *atas de audiência*, *acórdãos* e *sentenças*.

- ***Função***: função da pessoa no processo, como *reclamante*, *reclamado*, *advogado*, *juiz*, *testemunha*, etc.;
- ***Fundamento***: dispositivo legal usado como fundamentação, como "*artigo 795 da CLT*"ou "*artigo 1º da Lei n.º 6.858, de 24 de Novembro de 1980*";
- ***Local***: localidades geográficas e entidades geopolíticas;
- ***Organização***: nomes de pessoas jurídicas;
- ***Pessoa***: nomes de pessoas físicas;
- ***Tribunal***: nomes de tribunais;
- ***Vara***: nomes de varas;
- ***Valor de Acordo***, ***Valor de Causa***, ***Valor de Condenação***, ***Valor de Custas***: conforme explicados na Seção 4.1.

O Capítulo 5 descreve de forma detalhada a criação do *corpus* anotado com estas classes de entidades e que foi utilizado para o treino do modelo de REN do domínio jurídico trabalhista. Serão apresentados os critérios de anotação de cada uma das categorias, bem como exemplos de cada uma delas.

CAPÍTULO 5

Corpus da Justiça Trabalhista

5.1 Criação do *Corpus*

Em um processo da justiça do trabalho, vários tipos de documentos - que tipicamente são de natureza textual não-estruturada - fazem parte do trâmite do mesmo. Suas funções vão desde documentar etapas do ciclo de vida do processo até o provimento de argumentos, tanto por parte dos advogados na sustentação de seus pleitos e defesas; quanto dos juízes ao fundamentarem as decisões que tomam nas lides. Exemplos de documentos que possuem função de documentação são as atas de audiência. Em relação a documentos de caráter argumentativo, petições e contestações são de responsabilidade de advogados das partes, enquanto sentenças e acórdãos são de autoria dos juízes.

Considerando o crescente nível de litigiosidade da sociedade brasileira [17], fazem-se necessárias ferramentas que visem não só a celeridade dos processos na justiça, mas também a compreensão acerca dos motivos que fazem com que os brasileiros procurem cada vez mais o sistema judiciário para a solução de seus conflitos. Ademais, também é importante identificar as informações dos processos de forma sistematizada, para que a aplicação eficiente de métodos estatísticos em uma larga escala de dados possa viabilizar o mapeamento do panorama legal. Conforme explicado no Capítulo 4, à aplicação de análise estatística em dados legais dá-se o nome de Jurimetria [45]. A jurimetria é um recurso importante para prover diversas análises que possam descrever o funcionamento do sistema judiciário, tanto de forma quantitativa quanto qualitativa.

Neste contexto, uma técnica de extração de informações como REN pode possibilitar a identificação de diversos dados não-estruturados em documentos trabalhistas, viabilizando a construção de um banco de dados no qual possam ser aplicadas análises de jurimetria. Com isso, para que um modelo de REN pudesse ser aplicado neste tipo de problema, foi necessária a criação de um *corpus*, etiquetado com informações relevantes do ponto de vista da jurimetria.

5.2 Composição do *Corpus*

O *corpus* produzido é composto por 1305 documentos obtidos do PJe [78], distribuídos entre ***atas de audiência***, ***sentenças*** e ***acórdãos***. Estes documentos foram selecionados a partir de processos distribuídos em todas as 24 regiões da justiça do trabalho brasileira, entre os anos de 2008 e 2018. A Tabela 5.1 mostra a quantidade de documentos anotados de cada tipo. Os documentos foram descarregados e indexados por softwares desenvolvidos pela empresa ***Data Lawyer***[1].

Como critério de seleção dos documentos, procurou-se por amostras de cada um dos tipos mencionados, oriundos de cada região, no intervalo de anos determinado. Os documentos foram selecionados a partir de consultas onde estavam indexados[2], selecionando-se os maiores documentos[3] de cada agrupamento de [*tipo de documento*, *ano* e *região*]. As Tabelas 5.2 e 5.3 mostram as quantidades de documentos por ano e região. Como o PJe é um sistema que foi lançado em 2011, e sua adoção por parte de cada região da justiça do trabalho foi lenta, não foram disponibilizados muitos documentos anteriores à 2011[4]. O desbalanço que se verifica entre a quantidade de documentos de processos distribuídos nos demais anos após 2011 é justificado por restrições de disponibilidade dos documentos por parte da Data Lawyer.

Tipo de Documento	Quantidade
Acórdão	430
Ata de Audiência	427
Sentença	448

Tabela 5.1: *Quantidade de documentos anotados por tipo*

Ano	Quantidade
2008	1
2011	192
2012	91
2013	868
2014	83
2018	70

Tabela 5.2: *Quantidade de documentos anotados por ano*

[1] `https://www.datalawyer.com.br/`

[2] O conteúdo textual dos documentos estava salvo e indexado no Elasticsearch (`https://www.elastic.co/`).

[3] Consulta ordenada de forma decrescente pelo tamanho dos documentos, medido em quantidade de caracteres.

[4] `http://www.cnj.jus.br/tecnologia-da-informacao/processo-judicial-eletronico-pje`

Região	Quantidade
01	39
02	60
03	58
04	50
05	51
06	58
07	60
08	60
09	58
10	60
11	60
12	60
13	60
14	60
15	60
16	60
17	47
18	20
19	24
20	60
21	60
22	60
23	60
24	60

Tabela 5.3: *Quantidade de documentos anotados por região*

5.3 Processo de Anotação

A anotação dos documentos foi realizada por uma estudante do 10º período do curso de bacharelado em Direito, da Pontifícia Universidade Católica de Goiás. A revisão, realizada pelo autor desta dissertação, visou garantir a aderência dos critérios e padrões de anotação, além da inerente detecção e correção de erros de anotação. A ferramenta utilizada para anotação foi o ***WebAnno***[5].

O processo de anotação dos documentos foi realizado de maneira semi-supervisionada: foram anotados e revisados 76 documentos. Em seguida, estes documentos foram usados para treinar a primeira versão do modelo de extração de entidades jurídicas. Posteriormente, tal modelo foi utilizado para realizar a anotação automática dos 1229 documentos restantes. Isso propiciou à anotadora uma maior agilidade na anotação dos documentos, considerando que o processo de anotação passou a ser mais uma revisão do que uma anotação propriamente dita, fazendo somente uma correção dos erros de

[5] `https://webanno.github.io/webanno/`

anotação automática cometidos pelo modelo. Para os resultados documentados neste trabalho, foram revisados mais 68 documentos (dentre os 1229 anotados automaticamente), totalizando 144 documentos anotados e revisados.

5.4 Categorias das Entidades Anotadas

Foram anotadas as seguintes informações no *corpus* produzido:

5.4.1 Função

A categoria corresponde à função ou papel das pessoas mencionadas nos documentos. A Tabela 5.4 apresenta exemplos de funções anotadas no WebAnno. Foi adotado como critério fazer a anotação das funções somente quando elas acompanham algum nome de pessoa encontrado no documento, correspondentes à categoria PESSOA. O objetivo deste critério é induzir o modelo a reconhecer as funções somente quando elas podem ser atribuídas às pessoas que as possuem. A Figura 5.1 ilustra este critério. Na linha 11, o termo ***exeqüente*** não foi anotado por não estar acompanhando o nome da pessoa que é o exeqüente de fato, mas ***advogado(a)*** foi anotado logo em seguida, por estar acompanhando o nome da advogada. Também foi adotado o critério de não incluir nas anotações de função os pronomes de tratamento, tal como é mostrado na linha 10, a não anotação do termo ***Exma.***.

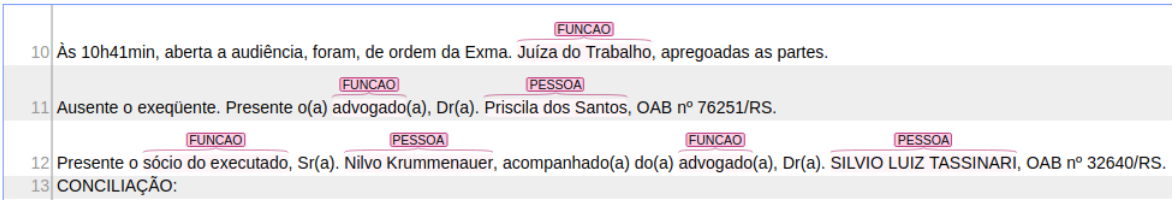

Figura 5.1: *Exemplo de funções anotadas no WebAnno*

5.4.2 Fundamento

Fundamento é a categoria atribuída a todo e qualquer dispositivo jurídico que possa ser referenciado nos documentos para fundamentar os pleitos dos advogados e as decisões dos magistrados. A Tabela 5.5 contém exemplos de diferentes fundamentos anotados no WebAnno. Adotou-se como critério anotar qualquer referência a legislação, jurisprudência ou doutrina, desde que os termos anotados permitam a identificação do fundamento. O critério adotado para considerar a identificação mínima do fundamento é especificar a lei, jurisprudência ou doutrina de referência. Por exemplo: ***CPC*** (Código Processual Civil), ***Lei nº 6858/80***, ***CLT*** (Consolidação das Leis do Trabalho), ***Constituição da República***. No caso das jurisprudências, deve ser identificada a numeração e o

Função
advogado
advogado (a)
Desembargador
Desembargador Relator
EXEQÜENTE
Juiz
Julgador de Primeiro Grau
Juíza do Trabalho
Juíza do Trabalho Substituta
patrono
preposto do (a) reclamado (a)
presidente / relator
Procuradora Regional do Trabalho
RECLAMADO
reclamante
Rel . Des .
Rel . Juiz
Rel . Min .
Relator (a)
Relatora Ministra
Relª Juíza
Relª Juíza Conv .
Secretário da Fazenda do Estado do Ceará

Tabela 5.4: *Exemplos de funções anotadas no WebAnno*

tribunal de origem, como em ***Súmula 381 do C. TST***. Além da especificação do dispositivo, deve-se, também, ter algum elemento que identifique ao menos um subconjunto de normas do mesmo. Por exemplo, anotar somente ***CPC***, ***CLT*** ou ***Constituição Federal*** não são de grande valia como referência de fundamentação, visto que tratariam de toda uma legislação, ao invés da parte dela que é relevante para o contexto. A Figura 5.2 mostra exemplos de anotações que não foram realizadas por não especificar um subconjunto dos dispositivos: ***CÓDIGO CIVIL DE 1916*** e ***Código Civil de 2002***. Note que a anotação ***art. 2028 do Código Civil de 2002*** se enquadra no critério perfeitamente, por especificar um subconjunto (o artigo 2028) do Código Civil de 2002.

Um outro critério foi anotar termos contíguos dentro de uma mesma referência, que permitiriam a identificação de mais de um fundamento, caso a separação dos mesmos em anotações distintas impedissem a identificação de um deles. As Figuras 5.3 e 5.4 ilustram este critério. Na Figura 5.3 os dois fundamentos estão mencionados de forma contígua, mas podem ser anotados separadamente, pois cada anotação individual identifica cada fundamento de forma bem definida: os dois dispositivos estão claros, sendo o primeiro, da Constituição da República, e o segundo, do Código Civil. Apesar de ***186 do***

CC não especificar que 186 é um artigo, é dedutível para o profissional do Direito, pelo padrão de nomenclatura da legislação brasileira [6], que seria um artigo, não prejudicando a identificação do dispositivo.

Fundamento
186 do CC
AC 1.0024.03.088345 - 8 / 001
AIRR181940 - 85.2008.5.18.0002
art . 112, do C . Civil Brasileiro
art . 12, II, do CPC
art . 1º, da Lei nº 6858 / 80
art . 2028 do Código Civil de 2002
art . 23, § 5º, da Lei nº 8.036 / 90
art . 236 da Constituição Federal de 1988
art . 397, do CPC
art . 48 da Lei nº 8.935 / 1994
Art . 6º do Código de Processo Civil
art . 795 da CLT
art . 7º da Constituição Federal de 1988
art . 85, do Código de 1916
artigo 1 º da Lei n . º 6.858, de 24 de Novembro de 1980
artigo 114, inciso VIII, da Constituição Federal
artigo 195, I, a, e II, da Constituição da República
artigo 267, incisos IV ou VI, do Código de Processo Civil
artigo 3º da Lei n.º 11.457 / 2007
arts . 326 e 327 do CPC
arts . 7º, XXVIII, da Constituição da República
CPC, art . 396
Curso de Direito Processual Civil, Vol . I, 14ª edição, Ed . Forense, pág . 57
Código de Processo Civil, no artigo 295, parágrafo único
EMENDA CONSTITUCIONAL Nº 45 / 04
inciso XXIX do art . 7º da Constituição
RR - 268100 - 66.2005.5.04.0404
Súmula 381 do C . TST
§ 1º, do art . 840 da CLT

Tabela 5.5: *Exemplos de fundamentos anotados no WebAnno*

[6] `http://www.planalto.gov.br/ccivil_03/leis/lcp/lcp95.htm`

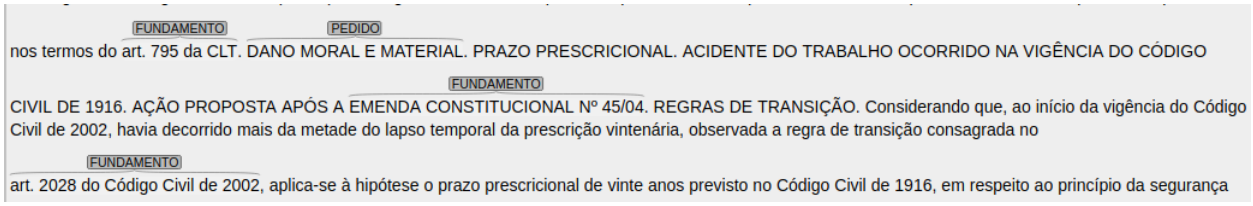

Figura 5.2: *Exemplo de fundamentos incompletos para anotação*

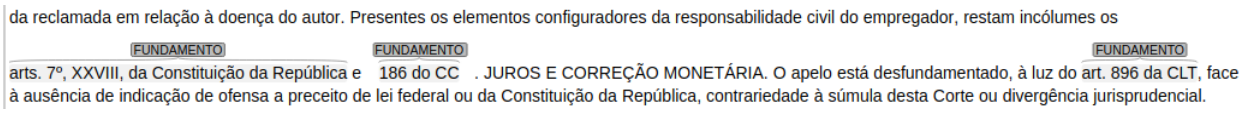

Figura 5.3: *Exemplo de dois fundamentos anotados separadamente no WebAnno*

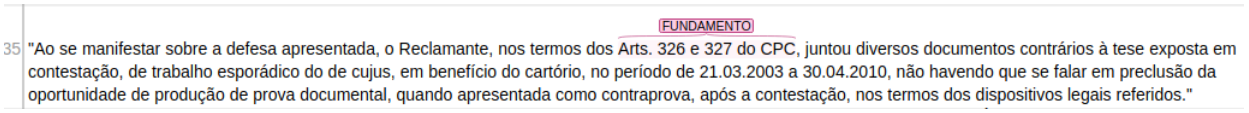

Figura 5.4: *Exemplo de dois fundamentos na mesma anotação*

5.4.3 Local

A Tabela 5.6 mostra exemplos de locais anotados no WebAnno. Os critérios foram a identificação de elementos contíguos que definissem logradouros, bairros, cidades, estados (com ou sem siglas), ou qualquer combinação destes, que definissem um endereço parcial ou completo.

Local
Avenida Deputado Raimundo Holanda
Bahia
Bairro de Fátima, Piracuruca-PI
Bebedouro-SP
Belo Horizonte
Francisco Cassiano de Brito
MACEIO
Morro da Saudade
Rua Luis Torquato da Silva, 35 - Vinght Rosado - Mossoró / RN
Santa Bárbara d' Oeste
VITORIA - ES
VITORIA / ES

Tabela 5.6: *Exemplos de locais anotados no WebAnno*

5.4.4 Organização

Para anotação de organizações, manteve-se critério semelhante ao de fundamentos, no intuito de manter as palavras que possibilitam a identificação de uma entidade específica. A Figura 5.5 mostra um exemplo deste critério: nas linhas 99 e 100 ***Cartório*** e ***Cartório de Registro de Imóveis*** não foram anotadas, pois não identificam um cartório em específico; já na linha 96, ***Cartório de Registro de Imóveis do 2º Ofício de Aracati*** foi anotado por especificar claramente um cartório. Como exceção a este critério, foram anotadas entidades que representam alguma organização abstrata, tal como ***Estado do Ceará*** ou ***JUSTIÇA DO TRABALHO***. A Tabela 5.7 mostra exemplos de organizações anotadas no WebAnno.

Organização
ARACATI - CARTÓRIO DO REGISTRO DE IMÓVEIS 2º OFÍCIO
ARACATI CARTORIO DO REGISTRO DE IMOVEIS 2 OFICIO
ARACATI CARTÓRIO DO REGISTRO DE IMÓVEIS 2 º OFÍCIOS
Banco Itaú
BNH
Caixa Econômica Federal
CARTÓRIO DE REGISTRO DE IMÓVEIS DO 2º OFÍCIO DE ARACATI
CEF
Estado do Ceará
INSS
Instituto de Previdência do Estado do Ceará
IPEC
JUSTIÇA DO TRABALHO
Ministério da Ação Social
Neniva Cereais e Transportes Ltda .
Receita Federal
Secretaria da Receita Federal
Secretaria da Receita Federal do Brasil
SISTEMA ÚNICO DE PREVIDÊNCIA SOCIAL DOS SERVIDORES PÚBLICOS CIVIS E MILITARES, DOS AGENTES PÚBLICOS E DOS MEMBROS DE PODER DO ESTADO DO CEARÁ
SUPSEC
VIVO S / A

Tabela 5.7: *Exemplos de organizações anotadas no WebAnno*

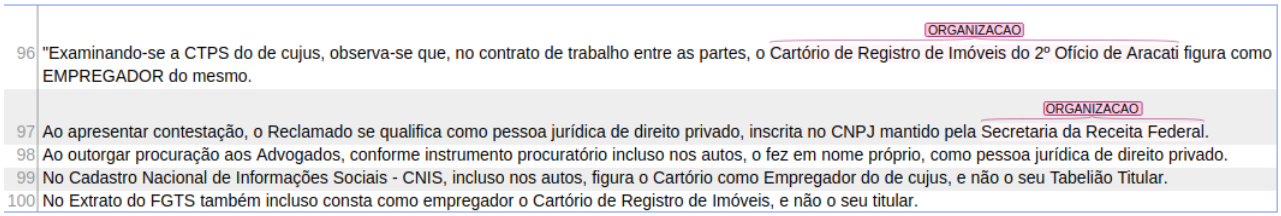

Figura 5.5: *Exemplo de organizações anotadas no WebAnno, visando a identificação das mesmas*

5.4.5 Pessoa

Esta categoria consiste em realizar a anotação de qualquer nome de pessoa física, esteja o nome completo ou incompleto no texto. Por *incompleto*, entende-se as ocorrências isoladas de primeiros nomes ou de sobrenomes de pessoas.

5.4.6 Tribunal e Vara

Por se tratar de domínio jurídico, optou-se por anotar dois tipos específicos de organizações: ***TRIBUNAL*** e ***VARA***. Os critérios de anotação de tribunais e varas também visam a identificação de um órgão específico, em detrimento de termos genéricos. As Tabelas 5.8 e 5.9 mostram exemplos de tribunais e varas anotadas no WebAnno.

Tribunal
STF
STJ
Supremo Tribunal Federal
TJMG
Tribunal de Justiça do Ceará
Tribunal Regional do Trabalho
TRIBUNAL REGIONAL DO TRABALHO 21ª REGIÃO
TRIBUNAL REGIONAL DO TRABALHO DA 15ª REGIÃO
Tribunal Superior do Trabalho
TRT 17ª R .
TRT 4ª R
TRT da 4ª Região
TST

Tabela 5.8: *Exemplos de tribunais anotados no WebAnno*

5.4.7 Valores de Acordo, Causa, Condenação e Custas

Os últimos tipos de entidades específicas que foram anotadas foram correspondentes a diferentes tipos de valores de processos trabalhistas. O critério para se anotar cada um destes tipos de valores foi considerar o contexto em que cada valor é mencionado, ao

Vara
10ª Vara do Trabalho desta Capital
11ª Vara do Trabalho de Recife
14ª VARA DO TRABALHO DE FORTALEZA / CE
22ª VARA DO TRABALHO DE PORTO ALEGRE
6ª Vara do Trabalho de Maceió - AL
8ª VARA DO TRABALHO DE CUIABÁ-MT
TERCEIRA VARA DO TRABALHO DE MOSSORÓ / RN
vara do trabalho de Atalaia
Vara do Trabalho de Bebedouro
VARA DO TRABALHO DE BRAGANCA PAULISTA
vara do trabalho de Maceió
Vara do Trabalho de Pirassununga
Única Vara do Trabalho de Aracati

Tabela 5.9: *Exemplos de varas anotadas no WebAnno*

invés do valor absoluto em si. Como exemplo, supondo que ***R$ 10.000,00*** seja o valor de condenação arbitrado e mencionado em uma sentença, não é anotada toda ocorrência do valor ***R$ 10.000,00*** no texto, de forma indiscriminada. Como o objetivo é fazer com que o modelo a ser treinado aprenda o contexto em que estes valores são mencionados nos documentos, anota-se somente as ocorrências em que o autor do documento deixa claro que este valor é de condenação. O objetivo disso é evitar não só a confusão com outros tipos de valores, mas, também, com quaisquer outros valores monetários mencionados nos documentos dos processos trabalhistas, que são muito frequentes. A Figura 5.6 mostra a anotação do valor do acordo ***R$ 210.000,00*** em um contexto em que ele foi mencionado claramente que era o valor total de uma conciliação. Isso pode ser percebido pela linha 13 do documento, por meio do termo ***CONCILIAÇÃO:***, e na linha 14, pelos termos ***a importância líquida e total de***. Para conformidade com o critério estabelecido, é importante ressaltar que os valores ***R$ 15.000,00***, ***R$ 10.500,00*** e ***R$ 4.500,00*** na linha 14 não foram anotados, assim como o valor ***R$ 15.000,00*** na linha 15. Os três primeiros não foram anotados por se tratar de: valor total da primeira parcela, valor da primeira parcela creditado ao reclamante e valor da primeira parcela creditado ao procurador da parte autora. Já o valor da linha 15 também não foi anotado por se tratar de um valor de parcela. As Figuras 5.7 e 5.8 mostram exemplos de valores de custas anotados em função dos valores de condenação e causa, respectivamente.

Também foi determinado o critério de se anotar somente valores numéricos, não realizando a anotação de valores escritos por extenso, como se pode ver na Figura 5.7: o valor ***R$ 30.000,00*** foi anotado na linha 279 como condenação, mas o ***trinta mil reais*** ao lado dele não foi anotado. O mesmo pode ser percebido na anotação do valor de custas de ***R$ 600,00***, na linha 280 da mesma figura.

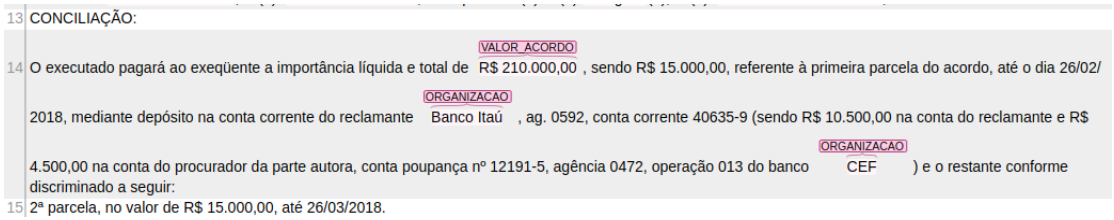

Figura 5.6: *Exemplo de anotação de valor de acordo e de valores monetários não anotados*

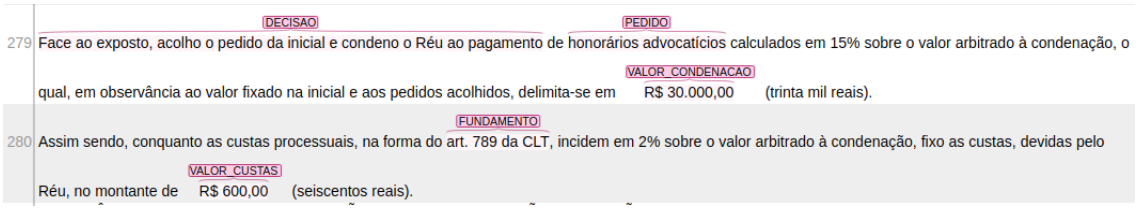

Figura 5.7: *Exemplo de anotação de valor de custas em função de valor de condenação*

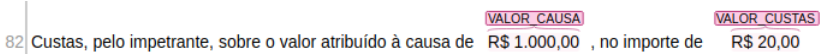

Figura 5.8: *Exemplo de anotação de valor de custas em função de valor de acordo*

5.5 Resultado das Anotações

A Tabela 5.10 mostra as quantidades de informações anotadas no processo de criação do *corpus*. A primeira etapa de anotação, com os 76 documentos anotados manualmente, e revisados, teve um total de 4.578 entidades anotadas, com 20.908 *tokens* selecionados na etiquetagem de cada entidade. Com o modelo temporário treinado para fazer as anotações automáticas dos 1.229 documentos restantes, chegou-se ao total de 132.289 entidades anotadas. A quantidade total de *tokens* anotados desta forma foi de 590.653. O *corpus* apresentado neste trabalho conta com a revisão adicional de 68 documentos (dentre os 1.229 que foram anotados automaticamente), totalizando 144 documentos anotados e revisados.

Após a revisão destes 144 documentos, constatou-se que a quantidade de entidades correspondentes aos valores de acordo, causa, condenação e custas eram bem inferiores em relação às quantidades das outras entidades. Isto se deve ao fato de que estes valores tendem a ocorrer somente uma ou duas vezes nos documentos trabalhistas utilizados, e ainda assim somente de acordo com o contexto. Valores de acordo normalmente são apresentados somente em atas de audiência em que foi homologado acordo entre as partes. Já os valores de condenação são mais mencionados em sentenças e alguns acórdãos. Valores de causa tendem a ocorrer em uma seção em que o juiz faz um relatório do

	Anotação Manual	Anotação Manual + Automática	Anotação Apresentada	Anotação Apresentada + Valores
Documentos	76	1.305	144	144 *
Sentenças	4.057	184.633	10.792	12.171
Tokens	173.541	8.895.520	327.358	466.597
Entidades	4.578	132.289	8.558	12.536
Tokens de entidades	20.908	590.653	38.342	53.701

Tabela 5.10: *Quantidades de informações anotadas em cada etapa do processo de criação do corpus:* ***Documentos*** *- quantidade de documentos anotados;* ***Sentenças*** *- quantidade total de sentenças nos documentos;* ***Tokens*** *- quantidade total de tokens nos documentos;* ***Entidades*** *- quantidade total de entidades anotadas;* ***Tokens de entidades*** *- quantidade de tokens das entidades anotadas. * O documento com sentenças de valores não conta como um documento adicional, sendo um aglomerado de sentenças avulsas.*

processo, e valores de custas são normalmente citados nos finais das decisões, em que o juiz relata o valor devido pelo custo do processo, e de quem é a responsabilidade de arcar com ele.

Para compensar esta baixa amostragem de valores, foram extraídas sentenças avulsas dos outros 1.161 documentos não revisados, em que o modelo usado para classificação automática identificou entidades classificadas nestas quatro categorias de valores. Estas sentenças foram consolidadas em um arquivo a parte, e em seguida foram revisadas para que pudessem ser incorporadas aos dados de treino do modelo de REN proposto. Ao todo, 1.379 sentenças foram identificadas desta forma. A Tabela 5.10 mostra o total de entidades anotadas e revisadas desta forma.

A Tabela 5.11 mostra a quantidade final de entidades anotadas de cada categoria, apresentando as informações do *corpus* anotado e revisado manualmente, e a versão final acrescida das informações obtidas a partir das sentenças avulsas. Para as entidades VALOR_ACORDO, VALOR_CAUSA, VALOR_CONDENACAO e VALOR_CUSTAS é possível perceber o balanceamento resultante do acréscimo das entidades obtidas a partir das sentenças avulsas que foram adicionadas ao *corpus*.

Para formação dos *corpora* de treino, teste e validação que foram utilizados nos treinos dos modelos de REN deste trabalho, o seguinte processo foi realizado:

1. Combinação de todos os documentos anotados em um único arquivo;
2. Segmentação do conteúdo do arquivo combinado em sentenças[7], a partir de quebra

[7] "Sentença"aqui utilizada no sentido linguístico de frases, não é o tipo de documento legal.

Categoria	Anotação Manual		Anotação Manual + Valores	
	Entidades	**Tokens**	**Entidades**	**Tokens**
FUNCAO	1.435	2.881	1.689	3.327
FUNDAMENTO	2.639	19.547	3.347	24.475
LOCAL	271	965	392	1.284
ORGANIZACAO	1.773	5.690	2.317	7.791
PESSOA	1.580	5.566	2.064	7.158
TRIBUNAL	530	1.927	630	2.530
VALOR_ACORDO	23	68	337	1.010
VALOR_CAUSA	4	11	236	708
VALOR_CONDENACAO	30	84	457	1.358
VALOR_CUSTAS	64	176	829	2.465
VARA	209	1427	238	1.595
Total	**8.558**	**38.342**	**12.536**	**53.701**

Tabela 5.11: *Quantidade de entidades e tokens anotados para cada categoria, após as diferentes etapas de revisão do corpus.*

de linhas e pontuação final das frases;

3. Considerando uma divisão de treino-teste-validação de 70%-15%-15%, são selecionadas aleatoriamente uma quantidade de frases para cada tipo de conjunto;
4. Os conjuntos de teste e validação são formados a partir das frases selecionadas na etapa anterior.

A tabela 5.12 apresenta a quantidade de entidades distribuídas para cada um dos conjuntos de treino, validação e teste utilizados para o modelo treinado no domínio trabalhista, de acordo com o processo descrito anteriormente.

Categoria	Treino	Validação	Teste
FUNCAO	1.124	266	299
FUNDAMENTO	2.320	508	519
LOCAL	287	46	56
ORGANIZACAO	1.651	356	310
PESSOA	1.400	344	319
TRIBUNAL	435	107	89
VALOR_ACORDO	250	50	37
VALOR_CAUSA	166	38	33
VALOR_CONDENACAO	312	73	71
VALOR_CUSTAS	577	139	113
VARA	166	35	38
Total	**8.688**	**1.962**	**1.884**

Tabela 5.12: *Quantidade de entidades atribuídas a cada um dos conjuntos de treino, validação e teste.*

CAPÍTULO 6

Modelagem do Método

Conforme indicado na seção 3.1, arquiteturas híbridas baseadas em redes biLSTM-CRF tornaram-se uma referência para tarefas de classificação sequencial [44, 54, 15, 61, 21, 75, 1, 3, 60], tal como Reconhecimento de Entidades Nomeadas. Neste trabalho, os experimentos foram conduzidos avaliando um modelo para REN usando esta mesma arquitetura, que será detalhada na Seção 6.1.

Como formas de representação dos dados textuais de treino, foram avaliadas três abordagens: por convolução de caracteres com uma rede CNN [18, 26, 15, 61], por vetores estáticos pré-treinados de palavras [54, 61, 21, 60, 3], e por modelo de linguagem [75, 1, 23]. Os vetores estáticos utilizados serão discutidos na Seção 6.2. Detalhes da rede CNN utilizada para convolução de caracteres serão fornecidos na próxima seção. Para modelo de linguagem, foi utilizada a arquitetura ***ELMo (Embeddings from Language Models)***, proposta no trabalho de Peters et al.[75], que será detalhadamente explicada na Seção 6.3.

Os experimentos realizados neste trabalho, para o processo de obtenção de um modelo de REN tanto para a língua portuguesa, quanto para o domínio da Justiça do Trabalho do Brasil, neste mesmo idioma, serão descritos na Seção 6.4.

6.1 Arquitetura LSTM-CRF

A arquitetura ***LSTM-CRF*** proposta por [54], conforme mostrada na Figura 6.1, é baseada em duas intuições: (i) Classificação de *tokens* em um texto é algo que deve ser baseado em informações contextuais, dependendo da relação entre as palavras das sentenças; (ii) Para determinar se um *token* é um nome, é importante considerar evidências ortográficas e distributivas. Evidências ortográficas são relacionadas à forma da palavra (as características que determinam a aparência da palavra), e evidências distributivas são relacionadas a como as palavras se situam no texto (as características relacionadas às palavras vizinhas nas sentenças e no *corpus*).

Considerando uma sentença de entrada representada por $\{x_1, x_2, x_3,...,x_n\}$, com n palavras codificadas como um vetor de d dimensões, uma unidade LSTM bidirecional

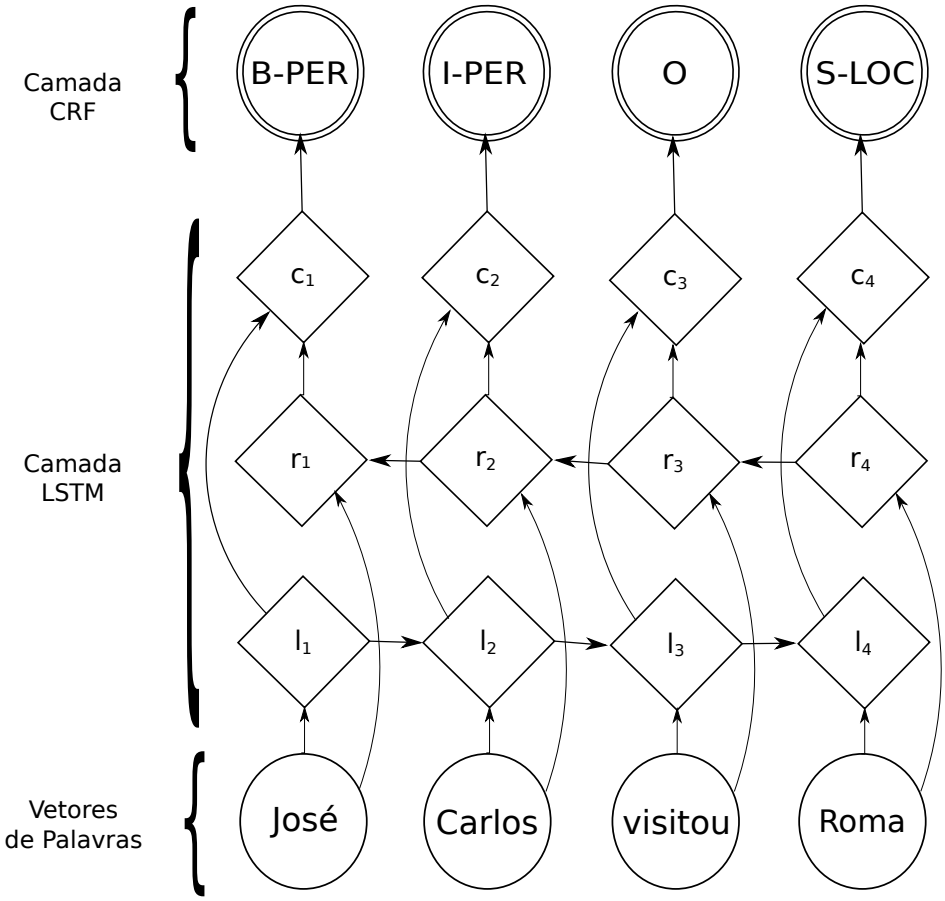

Figura 6.1: *As representações das palavras são alimentadas em uma rede LSTM bidirecional. l_i representa a palavra i e seu contexto à esquerda, r_i representa a palavra i e seu contexto à direita. As duas representações são concatenadas, resultando em uma representação da palavra i em seu contexto, c_i. Adaptado de [54]*

calcula uma representação $\mathbf{l}_i$ para o contexto esquerdo de cada palavra i e uma representação $\mathbf{r}_i$ para o contexto direito, conforme representado na Figura 6.1. As duas representações são concatenadas produzindo o contexto $\mathbf{c}_i$ de cada palavra. Os contextos de cada palavra são utilizados pela LSTM bidirecional para produzir uma representação única da sentença, dada por $h_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$, de forma que a Equação (2-19) mostra como $\boldsymbol{h}$ pode ser obtido a partir de $\boldsymbol{c}$. Na arquitetura LSTM-CRF, esta representação h_t é fornecida a uma camada ***CRF*** [53] para classificação sequencial das palavras. Em [54] foi utilizada uma segunda rede biLSTM para representação das palavras em nível de caracteres, para constituírem parte da representação h_t.

Neste trabalho foi utilizada uma implementação da arquitetura LSTM-CRF do *framework* **AllenNLP**[30], seguindo a mesma parametrização realizada em [75] para a tarefa de REN. Nesta implementação, a configuração do modelo foi feita para utilizar uma representação de palavras em nível de caracteres usando uma rede ***CNN***, conforme

feito em [75]. A configuração da rede CNN na implementação usa vetores de dimensão 16 e 128 filtros convolucionais de tamanho 3, ativados por uma função *ReLU* [46, 70]. Para codificação das representações das palavras, a rede biLSTM utilizada tem 2 camadas e 200 unidades ocultas. A Figura 6.2 mostra a dimensionalidade das representações de palavras por caracteres e vetores estáticos, e como eles são alimentados às redes biLSTM de 2 camadas, também com suas dimensionalidades destacadas.

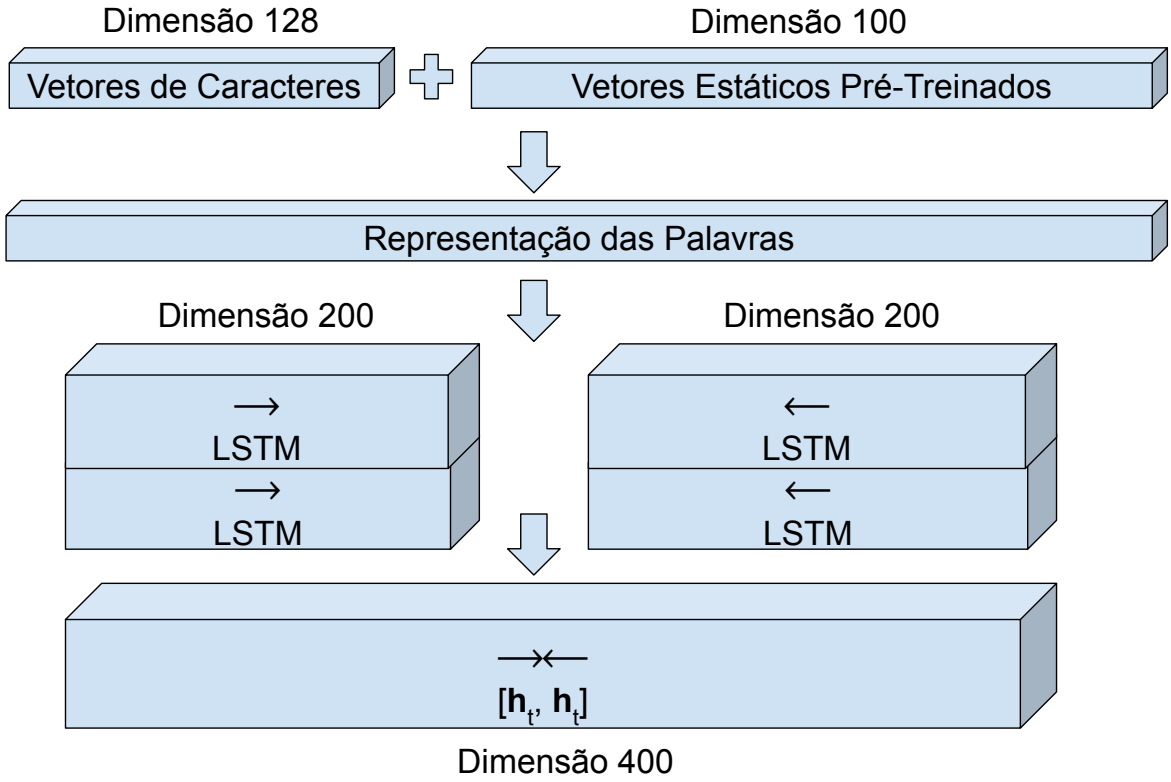

Figura 6.2: *Representação das palavras na arquitetura, com as dimensionalidades das entradas e das unidades de cada LSTM utilizada para bidirecionalidade da representação criada nos estados h_t.*

6.2 Vetores Estáticos de Palavras Pré-Treinados

Os vetores de palavras pré-treinados utilizados neste trabalho foram obtidos do Repositório de *Word Embeddings* do *Núcleo Interinstitucional de Linguística Computacional (NILC)* [71]. Neste repositório foram disponibilizados 4 tipos de vetores de palavras pré-treinados para o Português: *Word2Vec* [64], *Wang2Vec* [58], *FastText* [7, 47] e *GloVe* [74], em 5 dimensões diferentes: 50, 100, 300, 600 e 1000. Com exceção do *GloVe*, que usa o treinamento baseado na matriz de co-ocorrência das palavras, os outros 3 tipos foram disponibilizados nos algoritmos *Skip-Gram* e *Continuous Bag-of-Words*. Todos os vetores pré-treinados disponibilizados pelo NILC utilizaram o mesmo pré-processamento, de

acordo com [39], resultando em uma matriz de vetores com um vocabulário de 934.963 palavras. Seguindo [26] e [54], neste trabalho os experimentos foram conduzidos com os vetores de dimensão 100, treinados com o algoritmo de *Skip-Gram*.

Em relação ao modelo de REN voltado para o domínio da justiça trabalhista, foram experimentados novos vetores estáticos, pré-treinados em um acervo jurídico constituído de sentenças, acórdãos e atas de audiência de todos os tribunais regionais trabalhistas do Brasil, coletados entre os anos de 1997 e 2018, totalizando 278.160.851 frases e 6.029.035.996 *tokens*. Os *embeddings* foram treinados utilizando as implementações do *Word2Vec* de [34], *Wang2Vec* de [57], *FastText* de [33] e *GloVe* de [90]. Para *Word2Vec* e *FastText* foram treinados os algoritmos de *Skip-Gram* e *Continuous Bag-of-Words* e para o *Wang2Vec* foram treinados os algoritmos de *Structured Skip-Gram* e *Continuous Window*. Todos foram treinados com 100 dimensões e uma janela de tamanho 5, sendo que o *GloVe* utilizou uma janela de tamanho 15. Foi aplicado um pré-processamento combinando a limpeza do texto realizada em [71] e a *tokenização* realizada em [14]. Foram mantidas todas as frases que continham mais de 3 palavras, o que resultou em um descarte de 72.392.045 frases, resultando em um total de 205.768.806 frases. Os modelos foram configurados para manter as palavras que tiveram um mínimo de 5 ocorrências no *corpus*, o que reduziu o vocabulário de 2.337.222 para somente 729.651 palavras diferentes, sendo esta a quantidade de linhas da matriz pré-treinada. Esta redução representa a manutenção de somente 31% do vocabulário original.

6.3 Modelagem de Linguagem com ELMo

Além de vetores estáticos de palavras, neste trabalho também foram treinados modelos de linguagem para obtenção de representação contextual de palavras, visando suporte à polissemia. O modelo de linguagem utilizado neste trabalho foi o ***ELMo (Embeddings from Language Model)*** [75], que foi treinado na arquitetura ***biLM (bidirectional Language Model***) [75]. Peters et al. [75] atingiram o estado da arte para REN na língua inglesa utilizando uma rede biLSTM-CRF com representações de palavras a partir de CNN, vetores pré-treinados do *GloVe* [74] e seu modelo de linguagem *ELMo* proposto.

O motivo pelo qual o modelo do ELMo foi escolhido neste trabalho é pela sua viabilidade de ser treinado no idioma Português, demandando a utilização de menos recursos computacionais em relação ao ***BERT*** [23], que foi treinado na língua inglesa utilizando TPUs[1] [23]. Já o ***Flair*** de Akbik et al.[1] não foi avaliado por ter reportado resultados de treinos que foram realizados utilizando o conjunto de treino e o de validação

[1] `https://en.wikipedia.org/wiki/Tensor_processing_unit`

do *benchmark* do CoNLL-2003 [91], o que contamina a sua comparação com estas arquiteturas (tabela 3.1).

6.3.1 Modelo de Linguagem biLM

A arquitetura de modelo de linguagem *biLM* é baseada em duas redes biLSTM, cada uma responsável por uma direção do modelo de linguagem bidirecional: uma para manter uma representação da rede a partir da predição de palavras no sentido direto de leitura do texto (Equação 2-22), e outra mantendo uma representação da predição do sentido inverso (Equação 2-23). A Figura 6.3 ilustra as camadas da arquitetura do modelo biLM. Cada rede biLSTM recebe como entrada um vetor de características dos caracteres das palavras, produzidas por uma rede CNN nos dois sentidos de leitura do texto. Esta rede CNN é inicializada com uma dimensão de 16, para a matriz de representações do vocabulário de caracteres, e possui um total de 2048 filtros convolucionais, com tamanhos variando de 1 a 7. As duas representações da rede CNN constituem uma representação bidirecional de dimensão 4096, que é conectada à entrada da primeira camada de cada rede biLSTM. Cada camada das redes biLSTM faz uma projeção dos 4096 neurônios para uma representação de dimensão 512, de tal forma que a composição de cada camada de cada rede produz uma representação de dimensão 1024 por camada. Da mesma forma, uma terceira projeção de dimensão 1024 é produzida a partir da representação das palavras obtidas pela camada anterior, resultando em 3 camadas de dimensão 1024. Desta forma, o modelo biLM é caracterizado por manter um estado interno da rede que é produto de uma combinação linear destas 3 camadas. O tamanho do vocabulário de treino do biLM determina a quantidade de palavras que tentarão ser preditas na camada *Softmax* do modelo, exibida na Figura 6.3.

6.3.2 Pré-Processamento para Treino

Foram realizados três treinos do biLM neste trabalho: dois para o domínio geral e um para o domínio da justiça trabalhista, todos para a língua portuguesa. Foi aplicado o mesmo pré-processamento utilizado em [75, 14], que consiste nas seguintes etapas:

1. Remoção de todo conteúdo XML (*tags*) que possa ter no *corpus*;
2. Ordenação de todas as frases do *corpus* para que sejam eliminadas as repetidas;
3. Remoção de frases formadas somente por números, pontuação ou por caracteres fora do alfabeto português. Esta etapa foi acrescentada neste trabalho com o objetivo de aplicar uma limpeza no *corpus*, eliminando linhas como as da Tabela 6.1.
4. Aleatorização das frases restantes, para que a linguagem seja aprendida sem se ater a uma ordem específica das frases;

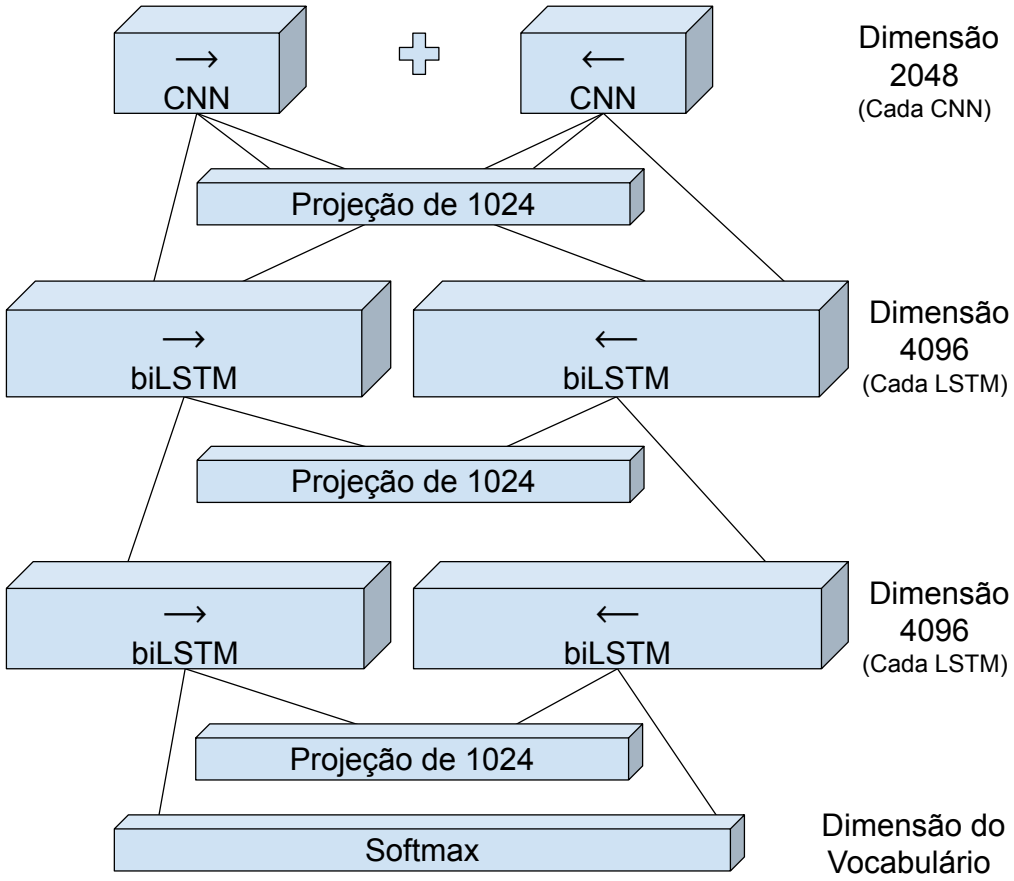

Figura 6.3: *Representação das camadas da arquitetura biLM e de suas conexões entre as camadas e as projeções de cada uma. Note que as setas $\rightarrow$ e $\leftarrow$ nas camadas LSTM indicam o sentido da função objetivo do modelo de linguagem bidirecional, e não das redes LSTM, que também são bidirecionais. Cada rede biLSTM de duas camadas é empregada neste esquema como um modelo de linguagem unidirecional, e a composição das duas funciona como o modelo final bidirecional.*

5. Normalização de pontuação e *tokenização* do texto;
6. Criação de um vocabulário mantendo somente as palavras que se repetiram pelo menos 3 vezes no *corpus*;
7. Segmentação do *corpus* resultante em um conjunto de treino, composto por 99% do total de frases, e um conjunto para validação, formado pelo 1% restante.

Para os modelos do domínio geral, foram usados dois *corpora* diferentes. O primeiro foi realizado a partir de uma descarga da versão portuguesa do Wikipedia[2], contendo 15.200.645 linhas, das quais 5.471.463 foram mantidas depois do pré-processamento. O *corpus* processado tem 267.310.316 *tokens* de treino, com um voca-

[2] `https://dumps.wikimedia.org/`

´
‘
∼
<
<!–
<!–
<!– ==
+———–+—-+————————+
⋈
_
__
__ _*(#(#(
10032 47622 20521 94645 10536 24505 12988 43087 13444 39507
1004
100	4	1
12/02/1990
1203
1’20"399
1 + 21 + 20 + 15 + 11 + 18 + 9 + 22 + 20 + 15 + 7 + 18 + 1 + 6 + 9 + 1 = 194

Tabela 6.1: *Exemplo de linhas removidas do Wikipedia português, formadas somente por números ou caracteres fora do alfabeto português.*

bulário de 2.387.159 *tokens* diferentes. Ao filtrar os que ocorrem menos de 3 vezes, o vocabulário final obtido cai para o tamanho de 811.468. O segundo *corpus* de treino utilizado foi o ***brWaC*** [29], que é composto por mais de 2.7 bilhões de tokens. Ele contém 349.382.930 linhas, das quais somente 38.740.390 foram mantidas depois do pré-processamento. A redução tão acentuada de linhas no pré-processamento se deve a um erro na execução do *script* de pré-processamento, mas que só foi identificada depois do pré-treino do modelo biLM. Este *corpus* teve um total de 1.148.467.764 de *tokens* de treino, e o vocabulário final, após manter somente as palavras de frequência mínima igual a três, teve tamanho 1.516.187.

O *corpus* usado para o domínio trabalhista é o mesmo que foi utilizado para treino dos vetores estáticos, que na sua forma original possui 90.639.814 frases (1 frase por linha) e 1.681.407.606 *tokens*. No caso do acervo jurídico, uma etapa adicional de filtragem de *tokens* foi adicionada ao pré-processamento, com o objetivo de descartar *tokens* correspondentes a números de processos, valores monetários e identificadores de documentos[3]. Isto se deve ao fato de que nos documentos dos processos trabalhistas em questão, os valores envolvidos, identificadores de documentos e os números de processos

[3]Valores alfanuméricos utilizados pelo PJe para identificar os documentos que são carregados na plataforma.

são mencionados de forma repetitiva, o que faz com que não sejam descartados pelo critério de frequência mínima. Assim como no domínio geral, a frequência mínima para manutenção das palavras do vocabulário de treino também foi 3. Ao final do pré-processamento, são descartadas 62.213.324 linhas, restando 28.426.490. O *corpus* trabalhista processado ficou com 1.225.378.678 *tokens* de treino, e um vocabulário de 7.667.712 *tokens* distintos, dos quais 774.340 foram mantidos após o descarte de acordo com os critérios estabelecidos.

6.3.3 Representação de Palavras com ELMo

Conforme exibido na Figura 6.3, a arquitetura biLM mantém projeções de dimensão 1024, a partir de cada camada da rede: da camada de entrada alimentada pela rede CNN e pelas 2 camadas das redes biLSTM. O ELMo é a representação composta a partir destas 3 camadas de projeções, e pode ser definida de acordo com a Equação (6-1) [75]. Uma palavra w teria uma representação R_w, em uma rede biLM de L camadas, dada por:

$$R_{\mathrm{w}} = \{x_w^{LM}, \overrightarrow{\mathbf{h}}_{w,l}^{LM}, \overleftarrow{\mathbf{h}}_{w,l}^{LM} \mid l = 1, ..., L\} = \{\mathbf{h}_{w,l}^{LM} \mid l = 0, ..., L\} \tag{6-1}$$

onde x_w^{LM} é a representação criada pela projeção da camada de entrada através da rede CNN, $\overrightarrow{\mathbf{h}}_{w,l}^{LM}$ é a representação de cada camada, de dimensão 512, criada pela rede biLSTM no sentido direto e $\overleftarrow{\mathbf{h}}_{w,l}^{LM}$ é a representação criada pela rede biLSTM no sentido inverso. Desta forma, são criadas $2L + 1$ representações em R_w. Como cada representação de cada biLSTM constitui metade da projeção de 1024, então a quantidade de projeções de dimensão 1024 que formam R_w é $L + 1$. Na segunda igualdade, x_w^{LM} é representado como $\mathbf{h}_{w,0}^{LM}$.

O ELMo de uma palavra w é um vetor obtido a partir desta representação R_w, que deve ser aplicado em uma tarefa fim de NLP[4]. O vetor *ELMo* é uma combinação linear de cada representação $\mathbf{h}_{w,l}^{LM}$ em R, de forma que cada uma delas passa por uma normalização *softmax* durante o processo de treino da tarefa fim, o que significa que cada camada recebe um peso p, e a soma destes pesos é 1. Além disso, o processo de treino também otimiza um parâmetro γ que também é específico da tarefa, atribuindo um peso a todo o *ELMo*, correspondente à contribuição do vetor no treinamento do modelo de NLP. A Equação (6-2) define a composição da representação do vetor *ELMo*, ilustrada na Figura 6.4.

$$ELMo_w^{tarefa} = \gamma^{tarefa} \sum_{l=0}^{L} p_l^{tarefa} \mathbf{h}_{w,l}^{LM} \tag{6-2}$$

[4]Neste trabalho, a tarefa fim é Reconhecimento de Entidades Nomeadas.

$$\mathbf{ELMo}_{w}^{REN} = \gamma^{REN} \times \sum \begin{cases} p_{2}^{REN} \times \mathbf{h}_{w2}^{LM} \\ p_{1}^{REN} \times \mathbf{h}_{w1}^{LM} \\ p_{0}^{REN} \times \mathbf{h}_{w0}^{LM} \end{cases}$$

Figura 6.4: *ELMo específico da tarefa de Reconhecimento de Entidades Nomeadas, com seus parâmetros aplicados na tarefa e na projeção de cada camada do modelo de linguagem bidirecional.*

A Figura 6.5 ilustra a representação das palavras aplicadas no modelo de REN proposto, contemplando a representação contextual do *ELMo* além das representações de caracteres da rede CNN e das representações de vetores estáticos pré-treinados.

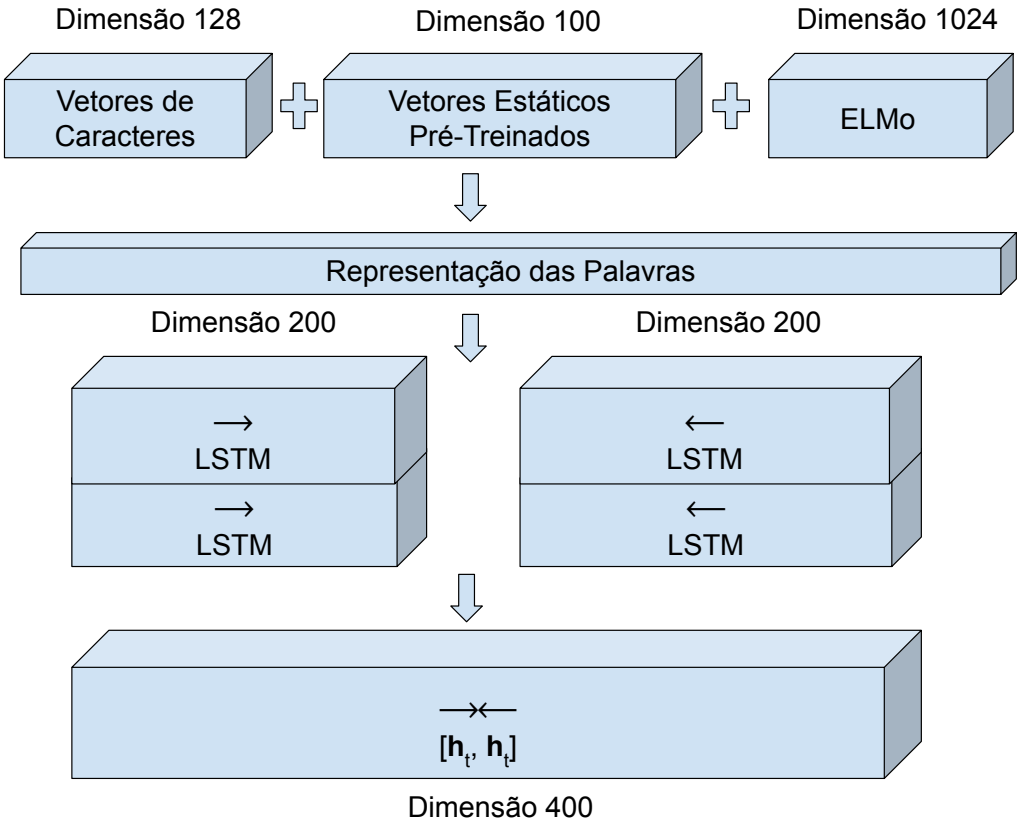

Figura 6.5: *Representação das palavras na arquitetura do modelo de REN conforme a Figura 6.2, acrescentando a representação do* ***ELMo****.*

Antes que o *ELMo* seja aplicado na tarefa fim, o modelo de linguagem *biLM* pode passar, ainda, por um processo de ajuste fino (*fine tuning*). Isto é feito aplicando o mesmo pré-processamento no *corpus* da tarefa que foi realizado no *corpus* no qual o modelo de linguagem foi treinado. Em seguida, o processo de treino do modelo *biLM* é retomado a partir do último ponto de verificação (*checkpoint*) dos pesos do modelo pré-

treinado. Com isso, o modelo ajustado é aperfeiçoado nos textos da tarefa em que ele foi aplicado, fornecendo uma melhor representatividade das palavras da mesma.

6.4 Configuração dos Experimentos

Nesta seção são descritas as configurações dos experimentos conduzidos para obtenção dos modelos de linguagem e REN de cada domínio proposto neste trabalho. A obtenção do modelo de cada um deve passar por duas etapas: treino do modelo de linguagem *biLM* e treino do modelo de REN.

O treino dos modelos de linguagem foi realizado por meio da implementação em [31]. O pré-treino do modelo de cada domínio foi feito por 10 épocas, utilizando o vocabulário e o pré-processamento conforme descrito em 6.3.2.

6.4.1 Modelo de REN para o Domínio Geral

Para o modelo de REN de domínio geral, foram avaliadas as melhores combinações de representações de palavras para verificar qual delas apresentou o melhor resultado, isto é, o melhor *F-Score* no *corpus* de teste do HAREM. Para isso, foi realizada uma validação cruzada usando o método *10-fold*, dividindo os dados de treino em 10 combinações diferentes de treino e validação. Cada combinação foi executada 5 vezes para cada tipo de representação de palavras avaliado:

1. *ELMo* [75];
2. *ELMo* [75] + *FastText* [7, 47] no modo *Continuous Bag of Words*;
3. *ELMo* [75] + *FastText* [7, 47] no modo *Skip-Gram*;
4. *ELMo* [75] + *GloVe* [74];
5. *ELMo* [75] + *Wang2Vec* [58] no modo *Continuous Window*;
6. *ELMo* [75] + *Wang2Vec* [58] no modo *Structured Skip-Gram*;
7. *ELMo* [75] + *Word2Vec* [64] no modo *Continuous Bag of Words*;
8. *ELMo* [75] + *Word2Vec* [64] no modo *Skip-Gram*;
9. *ELMo* [75] + CNN;
10. *ELMo* [75] + CNN + *FastText* [7, 47] no modo *Continuous Bag of Words*;
11. *ELMo* [75] + CNN + *FastText* [7, 47] no modo *Skip-Gram*;
12. *ELMo* [75] + CNN + *GloVe* [74];
13. *ELMo* [75] + CNN + *Wang2Vec* [58] no modo *Continuous Window*;
14. *ELMo* [75] + CNN + *Wang2Vec* [58] no modo *Structured Skip-Gram*;
15. *ELMo* [75] + CNN + *Word2Vec* [64] no modo *Continuous Bag of Words*;
16. *ELMo* [75] + CNN + *Word2Vec* [64] no modo *Skip-Gram*.

Foram realizados dois treinos adicionais, sem validação cruzada[5], com o objetivo de criar uma base de referência para a contribuição do *ELMo* na tarefa de REN. Os treinos adicionais foram: (i) usando somente CNN para representação das palavras e (ii) usando somente vetores pré-treinados para representação das palavras. O tipo de vetor pré-treinado utilizado aqui foi o *GloVe*. O motivo pelo qual foi utilizado este vetor é que somente um treino deve ser suficiente para estabelecer uma referência, e o *GloVe* é uma escolha arbitrária comum em experimentos de modelos de REN [15, 54, 61, 75, 1].

Todos os vetores estáticos foram obtidos de [71], e os *ELMos* foram os pré-treinados neste trabalho, para este domínio. Estas combinações de representações resultam em um total de 800 execuções[6]. Foram avaliados quatro modelos de *ELMo* distintos: dois treinados no Wikipedia e dois treinados no brWaC, sendo que para cada *corpus* foi avaliado um modelo com ajuste fino no *corpora* do HAREM, e outro modelo sem este ajuste. O objetivo disso é avaliar o impacto do ajuste na tarefa de REN deste domínio. Com isso, um total de 3.200 treinos de modelos de REN foram realizados para o domínio geral em Português. Estes treinos foram realizados por 10 épocas, no cenário *seletivo* do HAREM, e seus resultados foram considerados para determinar as melhores representações para o modelo de REN, por meio da média dos *F-Score* de cada grupo de resultados obtido. As melhores representações avaliadas foram utilizadas em um novo treino final de 50 épocas, que consistirá em 10 execuções de treino para o cenário *seletivo* e 10 para o *total*, sem validação cruzada. As médias dos *F-Score* resultantes destes treinos finais foram utilizadas como resultados finais.

Acompanhando o *benchmark* mais comum dos modelos voltados para o Português, o HAREM [27, 26, 21, 77], também foi utilizado o *corpus* do HAREM I [87] como treino e o do MiniHAREM [73] como teste. O *script* de avaliação utilizado para métrica de desempenho dos modelos é o CoNLL [91].

Pré-Processamento do HAREM

Os arquivos que constituem o *corpora* do HAREM I [87] e MiniHAREM [73] são em formato XML e podem ser obtidos em [59]. Os arquivos originais possuem documentos que são rotulados com muitas informações desnecessárias para o escopo deste trabalho, tais como tipos e subtipos de categorias. O pré-processamento realizado do *corpora* consistiu nas etapas a seguir, executadas depois de alguns ajustes nos *scripts* disponibilizados[7] por [76].

[5] Utilizando todo o conjunto de treino para o treino, sem conjunto de validação.
[6] 16 alternativas de representações dos dados x 5 execuções por *fold* x 10 *folds*.
[7] `https://github.com/arop/ner-re-pt`

1. Limpeza no arquivo XML, filtrando todas as marcações desnecessárias, mantendo somente as categorias das entidades. Nesta etapa são produzidos dois arquivos diferentes, para cada cenário avaliado: um arquivo XML com todas as 10 categorias, para o cenário *total*, e outro arquivo XML contendo somente as 5 categorias do cenário *seletivo*.
2. *Tokenização* dos arquivos resultantes da etapa anterior, usando o *PTBTokenizer*[8] do *Stanford CoreNLP*[9]. O *PTBTokenizer* foi executado com parâmetro *tokenizeNLs=true*. Esta *tokenização* produz um arquivo de texto com um único *token* do arquivo XML em cada linha. O uso do *tokenizeNLs=true* permite a criação de linhas vazias entre as sentenças, nas etapas subsequentes.
3. A próxima etapa é remover qualquer conteúdo XML remanescente da etapa de *tokenização* anterior. Ao final desta etapa tem-se um arquivo de texto com nada além dos *tokens* e suas respectivas categorias de entidades.
4. As categorias obtidas na etapa anterior ainda são as mesmas do XML original do HAREM, sem estar no esquema de anotação IOB [85]. Nesta etapa é executado um *script* que processa os rótulos das categorias, prefixando-os de acordo com o esquema de anotação IOB2.
5. Nesta etapa é executada uma limpeza final, convertendo sequências de mais de uma linha em branco em uma linha só, mantendo somente as que são necessárias para separar cada sentença do *corpus*.
6. Como os arquivos XML do HAREM são codificados no padrão *ISO-8859-1*, a etapa final é converter para a codificação *UTF-8*, para prevenir erros em relação ao uso de caracteres especiais e acentuações da língua portuguesa.

6.4.2 Modelo de REN para a Justiça do Trabalho do Brasil

O procedimento seguido no treino do modelo de REN para a justiça trabalhista foi semelhante ao do domínio geral. A avaliação também buscou a combinação de representação de palavras que apresentou o melhor *F-Score* no conjunto de teste do *corpus* trabalhista apresentado no Capítulo 5. Foram executados 12 treinos para cada uma das combinações de representação enumeradas a seguir:

1. *ELMo* [75];
2. *ELMo* [75] + *Word2Vec* [64] obtido de [71] na versão *Skip-Gram*;
3. *ELMo* [75] + *Word2Vec* [64] obtido de [71] na versão *Continuous Bag of Words*;
4. *ELMo* [75] + *FastText* [7, 47] obtido de [71] na versão *Skip-Gram*;

[8] `https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/process/PTBTokenizer.html`

[9] `https://stanfordnlp.github.io/CoreNLP/`

5. *ELMo* [75] + *FastText* [7, 47] obtido de [71] na versão *Continuous Bag of Words*;
6. *ELMo* [75] + *Wang2Vec* [58] obtido de [71] na versão *Structured Skip-Gram*;
7. *ELMo* [75] + *Wang2Vec* [58] obtido de [71] na versão *Continuous Window*;
8. *ELMo* [75] + *GloVe* [74] obtido de [71];
9. *ELMo* [75] + *Word2Vec* [64] treinado em acervo jurídico usando [34] na versão *Skip-Gram*;
10. *ELMo* [75] + *Word2Vec* [64] treinado em acervo jurídico usando [34] na versão *Continuous Bag of Words*;
11. *ELMo* [75] + *FastText* [7, 47] treinado em acervo jurídico usando [33] na versão *Skip-Gram*;
12. *ELMo* [75] + *FastText* [7, 47] treinado em acervo jurídico usando [33] na versão *Continuous Bag of Words*;
13. *ELMo* [75] + *Wang2Vec* [58] treinado em acervo jurídico usando [57] na versão *Structured Skip-Gram*;
14. *ELMo* [75] + *Wang2Vec* [58] treinado em acervo jurídico usando [57] na versão *Continuous Window*;
15. *ELMo* [75] + *GloVe* [74] treinado em acervo jurídico usando [90].
16. *ELMo* [75] + CNN;
17. *ELMo* [75] + CNN + *Word2Vec* [64] obtido de [71] na versão *Skip-Gram*;
18. *ELMo* [75] + CNN + *Word2Vec* [64] obtido de [71] na versão *Continuous Bag of Words*;
19. *ELMo* [75] + CNN + *FastText* [7, 47] obtido de [71] na versão *Skip-Gram*;
20. *ELMo* [75] + CNN + *FastText* [7, 47] obtido de [71] na versão *Continuous Bag of Words*;
21. *ELMo* [75] + CNN + *Wang2Vec* [58] obtido de [71] na versão *Structured Skip-Gram*;
22. *ELMo* [75] + CNN + *Wang2Vec* [58] obtido de [71] na versão *Continuous Window*;
23. *ELMo* [75] + CNN + *GloVe* [74] obtido de [71];
24. *ELMo* [75] + CNN + *Word2Vec* [64] treinado em acervo jurídico usando [34] na versão *Skip-Gram*;
25. *ELMo* [75] + CNN + *Word2Vec* [64] treinado em acervo jurídico usando [34] na versão *Continuous Bag of Words*;
26. *ELMo* [75] + CNN + *FastText* [7, 47] treinado em acervo jurídico usando [33] na versão *Skip-Gram*;
27. *ELMo* [75] + CNN + *FastText* [7, 47] treinado em acervo jurídico usando [33] na versão *CBoW*;
28. *ELMo* [75] + CNN + *Wang2Vec* [58] treinado em acervo jurídico usando [57] na versão *Structured Skip-Gram*;

29. *ELMo* [75] + CNN + *Wang2Vec* [58] treinado em acervo jurídico usando [57] na versão *Continuous Window*;
30. *ELMo* [75] + CNN + *GloVe* [74] treinado em acervo jurídico usando [90].

Os *ELMos* utilizados foram os pré-treinados neste trabalho, tanto para o domínio da justiça do trabalho, quanto para o domínio geral, com o objetivo de avaliar a diferença de desempenho do domínio das representações. Assim como nos experimentos do domínio geral, foram avaliadas as mesmas combinações de representações de vetores pré-treinados em conjunto com CNN e *ELMo*, de forma que os vetores foram avaliados tanto na forma disponibilizada em [71] para o domínio geral, quanto na forma pré-treinada em acervo jurídico, produzida neste trabalho. Estas combinações de representações resultam em um total de 360 execuções[10]. A exemplo do cenário geral, neste domínio também foi observado o impacto do ajuste fino no pré-treino do *ELMo*, incorrendo em mais 360 treinos para o *ELMo* sem este ajuste. O total de 1.440 treinos foi realizado por 5 épocas, e os melhores resultados também foram aplicados em 10 treinos finais de 50 épocas. O *script* de avaliação aplicado também foi o CoNLL [91]. Como o tempo de treino dos modelos deste domínio são bem maiores[11], os resultados apresentados para este domínio não foram provenientes do procedimento de validação cruzada.

[10] 30 alternativas de representações dos dados x 12 execuções.

[11] Uma época de treino do modelo no *corpus* trabalhista é cerca de cinco vezes mais demorada do que uma época de treino do HAREM, devido a diferença de tamanho entre os dois *corpora*.

CAPÍTULO 7

Resultados

Os treinos dos modelos de linguagem e de REN deste trabalho foram realizados nas seguintes máquinas:

- *Oracle*[1] com processador *Intel Xeon Gold 5120* de 2.20 GHz e 56 núcleos, 192 GB de memória RAM, e 2 placas de vídeo *Tesla P100 SXM2* de 16 GB de memória e 3584 núcleos *CUDA*
- *IBM Power System AC922*[2] com processador Power9 de 3.6 GHz e 128 núcleos, 1 TB de memória RAM, e 4 placas de vídeo *Tesla V100 SXM2* de 16 GB de memória e 5.120 núcleos *CUDA*
- *NVIDIA DGX-1*[3] com processador *Intel Xeon E5-2698 v4* de 2.20 GHz e 80 núcleos, 512 GB de memória RAM, e 8 placas de vídeo *Tesla V100 SXM2* de 32 GB de memória e 5.120 núcleos *CUDA*
- Máquina[4] com processador *Intel Core i7-8700* de 3.20 GHz e 12 núcleos, 16 GB de memória RAM, e 1 placa de vídeo *RTX 2080ti* de 11 GB de memória e 4352 núcleos *CUDA*

O treino do modelo de linguagem para o domínio geral, usando o *corpus* do Wikipedia, foi realizado em 5 dias, utilizando-se as duas *GPUs* disponíveis na máquina Oracle. O treino do mesmo modelo utilizando o *corpus* do brWaC teve duração de 10 dias, utilizando 7 GPUs disponíveis na DGX-1. Para o treino do *biLM* no domínio trabalhista foi disponibilizada somente uma GPU na mesma máquina, o que, somado ao fato de que o *corpus* é 4,53 vezes maior do que o do Wikipedia, fez com que o treino levasse 45 dias.

Para o treino dos modelos de Reconhecimento de Entidades Nomeadas, os treinos do domínio geral do *corpus* do HAREM foram executados na máquina com a GPU RTX 2080ti, e o tempo médio de treino foi de 9 minutos e 13 segundos. Para o

[1] Gentilmente cedida pela empresa Americas Health.
[2] Gentilmente cedida pela IBM do Brasil.
[3] Gentilmente cedida pelo laboratório de *Deep Learning* do Instituto de Informática da UFG.
[4] Gentilmente cedida pela empresa Data Lawyer.

Idioma	Domínio	*Corpus*	Perplexidade	Vocabulário	*Tokens* de Treino
Português	Geral	Wikipedia	36.38	811.468	267.310.316
Português	Geral	brWaC	62.49	1.516.187	1.148.467.764
Português	Trabalhista	Trabalhista	11.64	774.340	1.206.765.738
Inglês (Peters et al. [75])	Geral	1 Billion Word Language Model Benchmark	39.70	793.471	768.648.884

Tabela 7.1: *Perplexidades obtidas nos modelos treinados em Português, em cada domínio, comparando com o modelo original do **ELMo** treinado em [75].*

domínio trabalhista, os treinos foram executados na máquina IBM, e o tempo médio de treino foi de 29 minutos e 19 segundos.

Nas próximas seções são apresentados os resultados para os modelos de linguagem *biLM* e para os modelos de REN treinados em cada domínio. A métrica utilizada para avaliação dos modelos de linguagem foi a perplexidade (explicada na Seção 2.1), enquanto a métrica dos modelos de REN foi o *F-Score* (explicado na Seção 2.4.3).

7.1 Resultados dos Modelos de Linguagem

A perplexidade dos modelos de linguagem de cada domínio foi calculada em um conjunto de testes, criado a partir de uma amostra de 1% das sentenças de cada *corpus*. Para os modelos *biLM* treinados no domínio geral, as perplexidades obtidas foram **36.38** para o *corpus* do Wikipedia e **62.49** para o *corpus* do brWaC. Para o domínio trabalhista, a perplexidade obtida no acervo jurídico utilizado foi de **11.64**. A Tabela 7.1 apresenta os resultados de perplexidade obtidos em cada cenário avaliado neste trabalho, apresentando como referência a perplexidade do modelo *biLM* original introduzido em [75].

O critério restritivo de criação do vocabulário de treino do domínio trabalhista talvez possa explicar a perplexidade reduzida do modelo de linguagem deste domínio, tendo sido aplicado neste *corpus* o descarte de *tokens* correspondentes a números de processos, valores monetários e identificadores de documentos. Como a quantidade de palavras do vocabulário indica a quantidade de neurônios na camada *softmax* final da rede, a dificuldade de predição do modelo torna-se menor, pois ele precisa aprender a prever uma quantidade menor de palavras. Para o domínio geral, utilizando o *corpus* do Wikipedia, a perplexidade observada foi semelhante à do modelo de referência de Peters et al. [75] para a língua inglesa. Já o modelo resultante do *corpus* do brWaC, que possui um vocabulário quase duas vezes maior do que o do Wikipedia, mostra este nível maior de dificuldade do aprendizado apresentando, também, uma perplexidade quase duas vezes maior.

Corpus	Ajuste Fino	*F-Score*	Mínimo	Máximo	Desvio Padrão
brWaC	Sim	**81.60%**	79.64%	82.94%	0.58%
brWaC	Não	81.06%	78.66%	82.71%	0.77%
Wikipedia	Sim	80.80%	77.84%	82.56%	0.69%
Wikipedia	Não	80.70%	78.24%	82.92%	0.66%

Tabela 7.2: *Resultados obtidos para os 3.200 treinos no domínio geral da língua portuguesa, agrupados pelo* ***corpus*** *em que o pré-treino do ELMo foi realizado e pela realização de ajuste fino do ELMo no* ***corpus*** *do HAREM.* ***Não*** *indica o uso do modelo de linguagem sem ajuste fino, e* ***Sim*** *indica o uso do modelo de linguagem com ajuste fino. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

Mais resultados em relação a estes modelos serão apresentados nas próximas seções, em que serão discutidas as suas utilizações como representação de palavras na tarefa de REN para os domínios geral e da justiça do trabalho.

7.2 Resultados de REN para o Domínio Geral

Para determinar o melhor modelo de REN para o domínio geral da língua portuguesa, foram realizados 3.200 treinos no *corpora* do HAREM, avaliando: (i) o desempenho de cada ELMo em relação ao *corpus* em que foi pré-treinado, (ii) o desempenho de cada ELMo em relação ao ajuste fino no *corpora* do HAREM, e (iii) as diferentes combinações de representações de palavras experimentadas. Cada treino nesta avaliação foi realizado por 10 épocas. O *F-Score* de cada treino foi calculado no *corpus* do MiniHAREM [73], utilizando o *script* CoNLL [91]. A Tabela 7.2 apresenta os resultados agrupados por *corpus* e ajuste fino do modelo de linguagem. Percebe-se que o melhor desempenho foi o dos modelos treinados no brWaC, com um desempenho médio 0.71% acima dos modelos treinados no Wikipedia, possivelmente pela diferença de tamanho entre os dois *corpora*. Em relação ao ajuste fino no *corpora* do HAREM, para os modelos do brWaC ele contribuiu para um aumento de desempenho médio de 0.66%, e para o Wikipedia este aumento foi de 0.12%.

A Tabela 7.3 apresenta os resultados agrupados por tipo de representação de palavras avaliados. Percebe-se que os resultados em relação a este agrupamento são inconclusivos. A diferença entre a melhor (ELMo combinado com vetores estáticos) e a pior (ELMo combinado com vetores estáticos e convolução de caracteres) representação é de somente 0.27%. No entanto, o melhor desempenho individual ainda foi de um modelo usando a representação de pior desempenho médio, com 83.10%.

Representação	*F-Score*	Mínimo	Máximo	Desvio Padrão
ELMo+Vetor	**81.16%**	77.84%	82.85%	0.70%
ELMo	81.09%	79.36%	82.43%	0.74%
ELMo+CNN	81.03%	78.40%	82.65%	0.77%
ELMo+CNN+Vetor	80.94%	78.24%	83.10%	0.79%

Tabela 7.3: *Resultados obtidos para os 3.200 treinos no domínio geral da língua portuguesa, agrupados pelo tipo de representação utilizada no treino. O **F-Score** de cada grupo é a média dos resultados de cada grupo.*

Tipo de Vetor	*F-Score*	Mínimo	Máximo	Desvio Padrão
GloVe	**81.24%**	78.68%	82.85%	0.68%
Skip-Gram	81.13%	78.90%	83.10%	0.72%
Sem Vetor	81.06%	78.40%	82.65%	0.75%
CBoW	80.90%	77.84%	82.82%	0.78%

Tabela 7.4: *Resultados obtidos para os 3.200 treinos no domínio geral da língua portuguesa, agrupados pelo tipo de algoritmo dos vetores de palavras utilizados nos treinos. **Sem Vetor** indica o agrupamento de treinos que não fizeram uso de vetores de palavras. Os tipos **Structured Skip-Gram** e **Continuous Window** do **Wang2Vec** foram contabilizados como **Skip-Gram** e **CBoW**, respectivamente. O **F-Score** de cada grupo é a média dos resultados de cada grupo.*

A Tabela 7.4 faz uma comparação entre os tipos de algoritmos de vetores estáticos utilizados. O valor médio de *F-Score* dos treinos utilizando vetores baseados em *GloVe* teve o melhor desempenho em relação aos demais modelos. Modelos pré-treinados usando o algoritmo de *Skip-Gram* tiveram um desempenho médio de 0.28% acima dos modelos que utilizaram *Continuous Bag of Words*. Modelos que não utilizaram nenhum vetor de palavras (somente ELMo ou ELMo com CNN) também foram melhores do que modelos baseados em *Continuous Bag of Words*.

A Tabela 7.5 apresenta os resultados dos treinos agrupados por tipo de vetor de palavras. Os melhores modelos foram os baseados em *GloVe* e *Wang2Vec*, que tiveram uma diferença de somente 0.02% entre eles. *FastText* é o tipo de vetor que mais se destoou dos demais, sendo o único que teve desempenho médio inferior a 81%. A diferença entre usar *Word2Vec* e não usar nenhum vetor estático de palavras foi somente de 0.01%.

A Tabela 7.6 apresenta o desempenho médio dos modelos treinados considerando todos os agrupamentos apresentados: *corpus*, realização do ajuste fino, representação das palavras, tipo de vetor de palavras, e os vetores de palavras. O melhor resultado médio é dos modelos que: (i) usaram o ELMo pré-treinado no *corpus* do brWaC,

Vetor	*F-Score*	Mínimo	Máximo	Desvio Padrão
GloVe	**81.24%**	78.68%	82.85%	0.68%
Wang2Vec	81.22%	79.37%	83.10%	0.69%
Sem Vetor	81.06%	78.40%	82.65%	0.75%
Word2Vec	81.05%	78.72%	82.94%	0.74%
FastText	80.78%	77.84%	82.78%	0.79%

Tabela 7.5: *Resultados obtidos para os 3.200 treinos no domínio geral da língua portuguesa, agrupados pelo vetor de palavras pré-treinado utilizada no treino.* ***Sem Vetor*** *indica o agrupamento de treinos que não fizeram uso de vetores de palavras. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

(ii) utilizaram o ELMo **com** ajuste fino, (iii) usaram como representação a combinação ELMo+CNN+Vetor, (iv) utilizaram o *Wang2Vec*, e (v) utilizaram o algoritmo do *Structured Skip-Gram* (na tabela representado por *Skip-Gram*). Este modelo teve o desempenho médio de 81.96%. Considerando-se os melhores resultados individuais de cada um dos critérios analisados nas Tabelas 7.2 a 7.5, a melhor combinação entre eles seria utilizar ELMo-brWaC com ajuste fino, e sem representação por CNN, utilizando o *GloVe* como vetor estático de palavras. Esta combinação é a que teve o 7º melhor desempenho individual na Tabela total 7.6, com desempenho médio de 81.74%.

Escolhidos os modelos *ELMo-brWaC+CNN+Wang2Vec-Skip-Gram* e *ELMo-brWaC+GloVe*, ambos com ajuste fino, como os melhores modelos para o *benchmark* do HAREM na língua portuguesa, foram realizados dez novos treinos com cada uma das duas opções, nos cenários seletivo e total do HAREM, para modelos finais treinados em 50 épocas. A Tabela 7.7 apresenta os dados atualizados dos *benchmarks* da língua portuguesa considerando estes modelos. Utilizando como referência o melhor modelo *ELMo-brWaC+CNN+Wang2Vec-Skip-Gram* foi possível elevar o estado da arte dos cenários total e seletivo em 12.87% e 16.83%, respectivamente.

7.2.1 Contribuição do *ELMo* no Desempenho de REN

Conforme indicado na Seção 6.4.1, foram realizados dois treinos adicionais, com o objetivo de avaliar o desempenho de um modelo de REN utilizando como representação somente convolução de caracteres através de uma rede CNN, e outra representação utilizando somente um tipo de vetor pré-treinado de palavras, o *GloVe* [74]. O objetivo destes dois treinos foi determinar se o uso de alguma representaçao sem o *ELMo* poderia ser descartado na validação cruzada, considerando a pretensão de se obter o melhor desempenho possível no modelo de REN. O resultado de cada treino foi: 49.38% de *F-Score* para o modelo que utilizou somente a representação de caracteres, e 73.05% para

Corpus	Ajuste Fino	Representação	Vetor	Tipo de Vetor	*F-Score*	Mínimo	Máximo	Desvio Padrão
brWaC	Sim	ELMo+CNN+Vetor	Wang2Vec	Skip-Gram	**81.96%**	80.91%	82.88%	0.49%
brWaC	Sim	ELMo+CNN+Vetor	Wang2Vec	CBoW	81.82%	80.87%	82.77%	0.41%
brWaC	Sim	ELMo+CNN+Vetor	FastText	Skip-Gram	81.81%	79.79%	82.78%	0.53%
brWaC	Sim	ELMo+CNN+Vetor	Word2Vec	Skip-Gram	81.80%	80.79%	82.94%	0.48%
brWaC	Sim	ELMo+CNN+Vetor	Word2Vec	CBoW	81.75%	80.82%	82.82%	0.52%
brWaC	Sim	ELMo+Vetor	Wang2Vec	Skip-Gram	81.74%	80.20%	82.82%	0.57%
brWaC	Sim	ELMo+Vetor	GloVe	GloVe	**81.74%**	80.09%	82.85%	0.50%
brWaC	Sim	ELMo+CNN+Vetor	GloVe	GloVe	81.73%	81.21%	82.77%	0.35%
brWaC	Não	ELMo+Vetor	Word2Vec	Skip-Gram	81.63%	80.69%	82.35%	0.44%
brWaC	Não	ELMo+Vetor	Wang2Vec	CBoW	81.59%	80.31%	82.57%	0.51%
brWaC	Não	ELMo+Vetor	FastText	Skip-Gram	81.56%	80.05%	82.30%	0.53%
brWaC	Sim	ELMo+Vetor	FastText	Skip-Gram	81.55%	80.34%	82.32%	0.53%
brWaC	Sim	ELMo+Vetor	Word2Vec	CBoW	81.55%	80.27%	82.43%	0.57%
brWaC	Sim	ELMo+Vetor	Word2Vec	Skip-Gram	81.54%	79.80%	82.65%	0.66%
brWaC	Sim	ELMo	Sem Vetor	Sem Vetor	81.54%	80.78%	82.35%	0.46%
brWaC	Sim	ELMo+Vetor	Wang2Vec	CBoW	81.53%	79.64%	82.66%	0.62%
brWaC	Não	ELMo+Vetor	Word2Vec	CBoW	81.52%	80.51%	82.55%	0.50%
brWaC	Não	ELMo+Vetor	GloVe	GloVe	81.51%	80.19%	82.37%	0.49%
brWaC	Não	ELMo+Vetor	Wang2Vec	Skip-Gram	81.47%	79.72%	82.37%	0.49%
brWaC	Não	ELMo	Sem Vetor	Sem Vetor	81.45%	79.47%	82.43%	0.68%
...								
Wikipedia	Não	ELMo+Vetor	Word2Vec	Skip-Gram	80.78%	79.26%	81.65%	0.55%
Wikipedia	Não	ELMo+CNN	Sem Vetor	Sem Vetor	80.76%	79.81%	81.64%	0.43%
Wikipedia	Não	ELMo+Vetor	FastText	Skip-Gram	80.73%	78.95%	81.93%	0.71%
Wikipedia	Sim	ELMo+CNN+Vetor	Wang2Vec	CBoW	80.72%	79.62%	82.34%	0.70%
Wikipedia	Sim	ELMo+CNN+Vetor	FastText	Skip-Gram	80.71%	79.30%	81.78%	0.63%
Wikipedia	Não	ELMo+CNN+Vetor	FastText	Skip-Gram	80.68%	78.90%	82.16%	0.70%
Wikipedia	Sim	ELMo+CNN+Vetor	FastText	CBoW	80.64%	79.70%	81.69%	0.49%
Wikipedia	Não	ELMo+Vetor	Word2Vec	CBoW	80.64%	78.72%	81.87%	0.67%
Wikipedia	Sim	ELMo+CNN+Vetor	Word2Vec	CBoW	80.58%	79.25%	81.93%	0.64%
brWaC	Não	ELMo+CNN+Vetor	FastText	Skip-Gram	80.58%	79.40%	82.07%	0.65%
Wikipedia	Não	ELMo+CNN+Vetor	Word2Vec	CBoW	80.52%	79.13%	82.51%	0.68%
brWaC	Não	ELMo+CNN+Vetor	Word2Vec	CBoW	80.49%	79.10%	82.27%	0.64%
Wikipedia	Sim	ELMo	Sem Vetor	Sem Vetor	80.46%	79.36%	81.85%	0.58%
brWaC	Não	ELMo+CNN+Vetor	Word2Vec	Skip-Gram	80.46%	79.30%	81.89%	0.65%
Wikipedia	Não	ELMo+CNN+Vetor	Word2Vec	Skip-Gram	80.45%	79.00%	81.67%	0.63%
Wikipedia	Sim	ELMo+CNN	Sem Vetor	Sem Vetor	80.39%	78.40%	81.76%	0.78%
Wikipedia	Sim	ELMo+Vetor	FastText	CBoW	80.27%	77.84%	81.93%	0.78%
Wikipedia	Não	ELMo+Vetor	FastText	CBoW	80.19%	78.75%	81.08%	0.48%
brWaC	Não	ELMo+CNN+Vetor	FastText	CBoW	79.97%	78.66%	81.55%	0.71%
Wikipedia	Não	ELMo+CNN+Vetor	FastText	CBoW	79.92%	78.24%	81.36%	0.71%

Tabela 7.6: *Resultados obtidos para os 3.200 treinos no domínio geral da língua portuguesa, agrupados para cada cenário avaliado nos treinos.* ***Sem Vetor*** *indica o agrupamento de treinos que não fizeram uso de vetores de palavras. Os tipos* ***Structured Skip-Gram*** *e* ***Continuous Window*** *do* ***Wang2Vec*** *foram contabilizados como* ***Skip-Gram*** *e* ***CBoW****, respectivamente. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

Trabalho	***Corpus* de Treino**	***Corpus* de Teste**	***Script* de Avaliação**	**Cenário**	***F-Score***
Dos Santos e Milidiú [27]	HAREM I	MiniHAREM	SAHARA	Total	63.56%
				Seletivo	70.72%
Do Amaral [24]	HAREM I	HAREM II	SAHARA	Total	48.43%
Dos Santos e Guimarães [26]	HAREM I	MiniHAREM	SAHARA	Total	**71.41%**
				Seletivo	**77.93%**
			CoNLL	Total	65.41%
				Seletivo	71.23%
Da Costa e Paetzold [21]	HAREM I	MiniHAREM	CoNLL	Total	69.14%
Pirovani [77]	HAREM I	MiniHAREM	CoNLL	Seletivo	60.36%
ELMo-af-brWaC+GloVe	HAREM I	MiniHAREM	CoNLL	Total	77.63%
				Seletivo	82.89%
ELMo-af-brWaC+CNN+Wang2Vec-SG	HAREM I	MiniHAREM	CoNLL	Total	**78.04%**
				Seletivo	**83.22%**

Tabela 7.7: *Resultados obtidos neste trabalho, para a língua portuguesa, em comparação com os resultados existentes nos diferentes benchmarks do HAREM.* ***ELMo-af*** *indica a versão do ELMo com ajuste fino no corpora do HAREM, e SG é o modelo com Skip-Gram.*

o modelo que utilizou somente o *GloVe*. Como são valores significativamente inferiores aos apresentados na validação cruzada, conclui-se que o uso do *ELMo* realmente contribui para o melhor desempenho do modelo de REN no estudo comparativo de representações que foi realizado.

7.2.2 Análise dos Erros

A Figura 7.1 apresenta uma matriz de confusão produzida a partir das predições do melhor modelo **ELMo-af-brWaC+CNN+Wang2Vec-SG** treinado para o cenário seletivo do HAREM. Pela matriz, pode-se perceber que os erros mais frequentes são de identificação, quando o modelo deixa de prever que algum *token* faz parte de uma entidade, classificando-o como *O*[5]. Em relação aos erros de classificação, os mais comuns foram classificar LOCAL como ORGANIZACAO, ORGANIZACAO como LOCAL e ORGANIZACAO como PESSOA.

No Exemplo 7.1 é mostrado um erro de classificação de uma pessoa, anotada de acordo com as diretivas do HAREM[87]. O modelo reconheceu o *token "1º"* como VALOR, não identificou *"Ciclo do Ensino Básico da"* como parte da entidade, e reconheceu *"Escola de Codeçoso"* como sendo uma organização. Já no Exemplo 7.2, o modelo reconheceu as formas de tratamento *senhora* e *senhor* como parte das entidades do tipo PESSOA, o que está aderente às diretrizes de anotação do HAREM [87]. Embora no

[5] *Outside*, de acordo com o esquema de anotação IOB2.

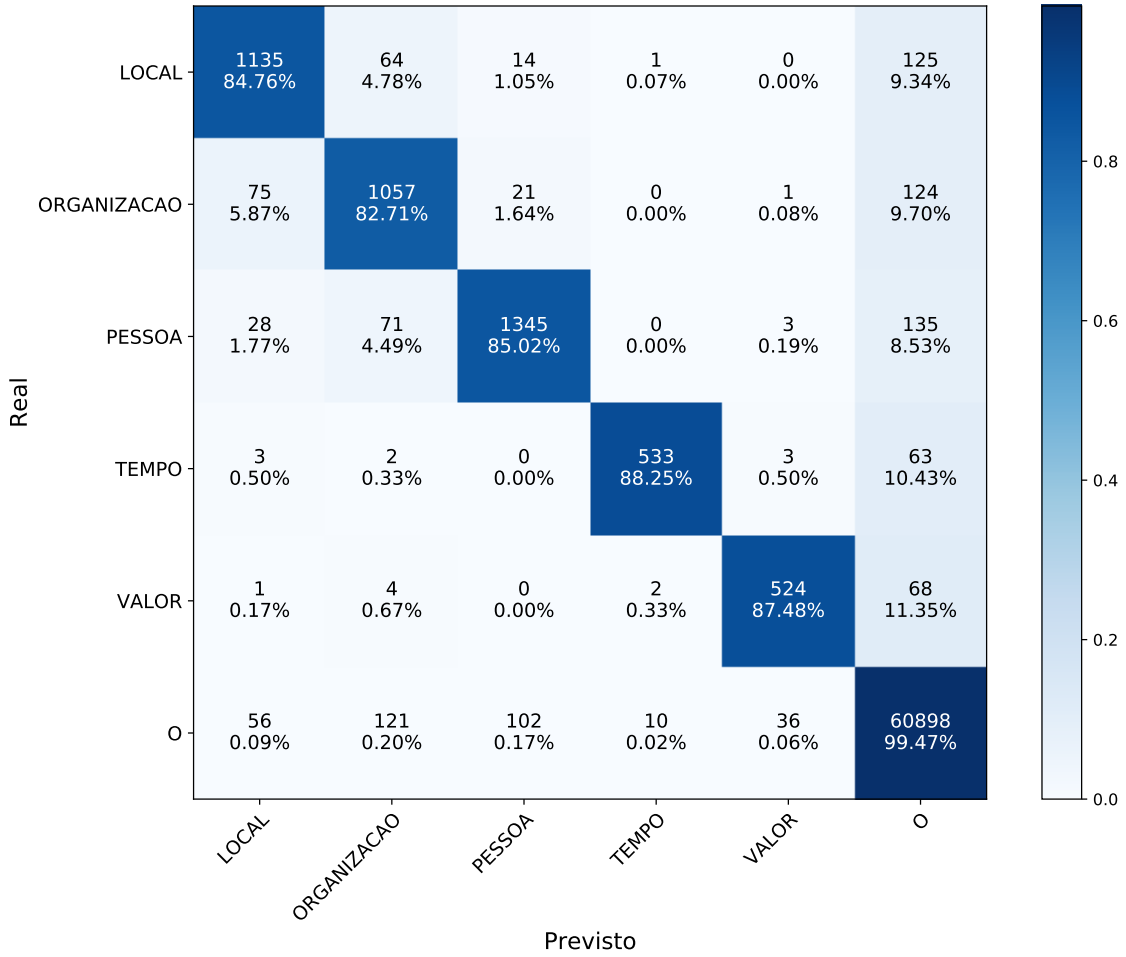

Figura 7.1: *Matriz de confusão do melhor modelo* ***ELMo-af-brWaC+CNN+Wang2Vec-SG*** *treinado no cenário seletivo do HAREM.*

corpus estes *tokens* tenham sido rotulados como *O*, pode-se encontrar outros casos em que formas de tratamentos foram rotuladas como PESSOA[6], conforme demonstrado no Exemplo 7.3. Um local que foi reconhecido como organização pelo modelo é mostrado no Exemplo 7.4.

Exemplo 7.1: *Exemplo de erros de classificação no cenário seletivo do HAREM, em que uma pessoa foi reconhecida como organização. Ao lado de cada* ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido na Coleção Dourada do HAREM. Foram destacados somente os erros de classificação.*

"Em o passado dia 7 de Dezembro , os alunos de o 1º (U-VALOR,B-PESSOA) Ciclo (O,I-PESSOA) do (O,I-PESSOA) Ensino (O,I-PESSOA) Básico (O,I-PESSOA) da (O,I-

[6]Especificamente de acordo com o critério **Tipo INDIVIDUAL** do HAREM [87].

PESSOA) Escola (B-ORGANIZACAO,I-PESSOA) de (I-ORGANIZACAO,I-PESSOA) Codeçoso (L-ORGANIZACAO,L-PESSOA) , sob a orientação de a professora , em o âmbito de o concurso promovido por a Biblioteca Municipal de Montalegre , decidiram entrevistar um par de idosos de o lugar de Codeçoso , fregueisa de a Venda Nova , sobre o Natal de os seus tempos de juventude , assim como algumas recordações que lhes deixaram saudade ."

Exemplo 7.2: *Exemplo de erros de classificação no cenário seletivo do HAREM, em que o modelo treinado reconheceu as formas de tratamento como parte das entidades do tipo PESSOA. Ao lado de cada* ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido na Coleção Dourada do HAREM. Foram destacados somente os erros de classificação.*

"Trata - se de a senhora (B-PESSOA,O) Teresa Gonçalves Bastos , de 88 anos e de o senhor (B-PESSOA,O) António Gonçalves Bastos de 90 anos . "

Exemplo 7.3: *Exemplo de sentenças obtidas da Coleção Dourada do HAREM, em que formas de tratamento foram anotadas como parte das entidades do tipo PESSOA. As palavras destacadas em* ***negrito*** *são as que constituem cada entidade de PESSOA nestes exemplos.*

"Antes de dar a palavra ao ***senhor deputado Watts*** *sobre o mesmo assunto, gostaria apenas de os informar que, após essa terrível catástrofe, escrevi evidentemente ao* ***Presidente do Parlamento*** *grego, o* ***senhor Kaklamanis****, para lhe comunicar em vosso nome a minha profunda tristeza e a nossa solidariedade para com as famílias das vítimas."*

Exemplo 7.4: *Exemplo de erros de classificação no cenário seletivo do HAREM, em que um local foi reconhecido como organização. Nesta sentença também tem um exemplo de uma pessoa que deixou de ser identificada pelo modelo, no* ***token*** *"Autor". Ao lado de cada* ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido na Coleção Dourada do HAREM. Foram destacados somente os erros de classificação.*

"Novo modelo de apólice cobrirá safra agrícola Data : 25/02/2000 Fonte : Jornal (B-ORGANIZACAO,B-LOCAL) do (I-ORGANIZACAO,I-LOCAL) Commercio (L-ORGANIZACAO,L-LOCAL) Autor (O,U-PESSOA) : Matéria : O Governo quer disponibilizar o novo modelo de o seguro agrícola para os agricultores brasileiros já em a próxima safra de grãos e frutas , que começa a ser cultivada em julho . "

A matriz de confusão da Figura 7.2 mostra os resultados obtidos para o melhor modelo treinado no cenário total do HAREM. Pode-se perceber pelos resultados da matriz que as categorias ABSTRACCAO, ACONTECIMENTO, COISA, OBRA e OUTRO, que complementam o cenário total do HAREM, são as que tiveram um maior grau de dificuldade por parte do modelo proposto. Para a categoria OUTRO, que teve somente 14 exemplos de testes, o modelo errou a predição de todos os exemplos. Dois casos destes erros são exibidos no Exemplo 7.5. É possível que a dificuldade em reconhecer as entidades desta categoria possa ser explicada por: (i) ela é a que possui menos exemplos de treino, com um total de 40, enquanto ACONTECIMENTO e COISA, que foram as próximas duas com menos exemplos, tiveram 129 e 135 exemplos, respectivamente; (ii) é uma categoria sem critérios específicos, criada para realizar a anotação de entidades que não se enquadraram nas diretrizes de anotação das demais categorias, fazendo com que o modelo neural tenha tido uma maior dificuldade em aprender algum tipo de padrão para estas anotações. A Tabela 7.8 mostra a quantidade de entidades de cada categoria, nos *corpora* de treino e teste utilizados neste trabalho.

Categoria	**Exemplos de Treino**	**Exemplos de Teste**
ABSTRACCAO	406	204
ACONTECIMENTO	129	59
COISA	135	170
LOCAL	1239	878
OBRA	196	191
ORGANIZACAO	926	599
OUTRO	40	14
PESSOA	1034	834
TEMPO	434	358
VALOR	467	326
Total	**5.006**	**3.633**

Tabela 7.8: *Quantidade de entidades anotadas nos conjuntos de treino e de teste, para cada categoria do cenário total do HAREM.*

A análise da matriz do cenário total também indica que o modelo teve uma dificuldade em distinguir entidades do tipo ABSTRACCAO e ORGANIZACAO, classificando mais de 44% dos *tokens* do tipo ABSTRACCAO como ORGANIZACAO. O Exemplo 7.6 mostra quatro destes erros em uma única sentença. Para os *tokens* das categorias ABSTRACCAO, ACONTECIMENTO, COISA, OBRA e OUTRO, somente a categoria OBRA teve um desempenho acima de 50%, mesmo tendo menos da metade da quantidade de exemplos de ABSTRACCAO, que é a categoria destas cinco que possui mais exemplos. Isso pode ser explicado pelo fato de que os critérios de anotação de enti-

dades do tipo OBRA são mais objetivos do que os do tipo ABSTRACCAO, que precisou ser simplificada entre o HAREM I [87] e o HAREM II [68].

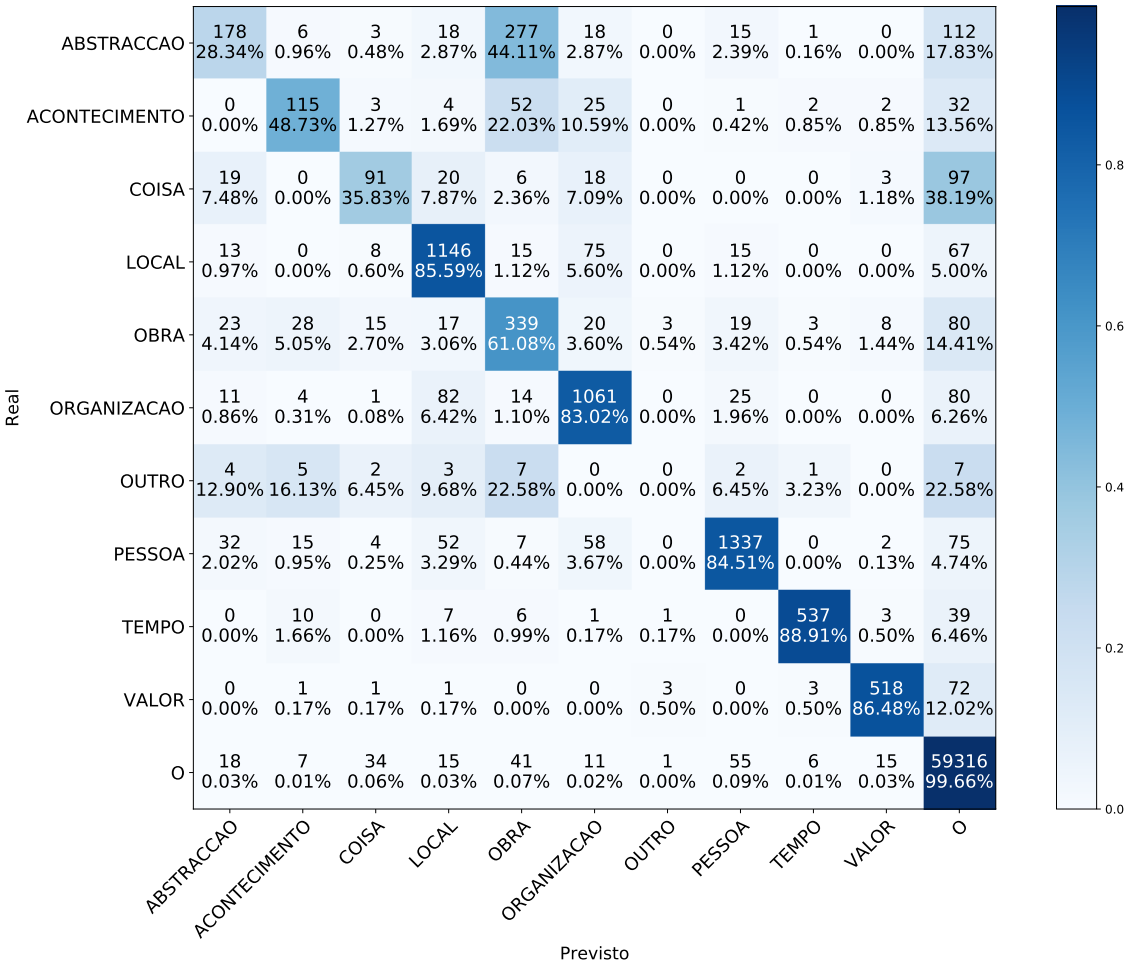

Figura 7.2: *Matriz de confusão do melhor modelo **ELMo-af-brWaC+CNN+Wang2Vec-SG** treinado no cenário total do HAREM.*

Exemplo 7.5: *Exemplo de erros de classificação no cenário total do HAREM, em que tipos OUTRO foram reconhecidos como COISA. Ao lado de cada **token** classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido na Coleção Dourada do HAREM. Foram destacados somente os erros de classificação.*

"Variava entre um 4-3-3 (U-COISA,U-OUTRO) e um 4-2-4 (U-COISA,U-OUTRO) ."

Exemplo 7.6: *Exemplo de erros de classificação no cenário total do HAREM, em que tipos ABSTRACCAO foram reconhecidos como ORGANIZACAO. Ao lado de cada **token** classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo*

definido na Coleção Dourada do HAREM. Foram destacados somente os erros de classificação.

"Consciente de o hiato existente entre o ensino colegial e os estudos universitários , as Faculdades do Sagrado Coração o Primeiro (B-ORGANIZACAO,B-ABSTRACCAO) Ciclo (L-ORGANIZACAO,L-ABSTRACCAO) , cujo objetivo é o de superar as insuficiências de o ensino pré-universitário e de esta forma , promover o ajustamento de o vestibulando as novas exigências de os Cursos (B-ORGANIZACAO,B-ABSTRACCAO) Superiores (L-ORGANIZACAO,L-ABSTRACCAO) ."

7.3 Resultados de REN para a Justiça do Trabalho do Brasil

Para o domínio trabalhista, o melhor modelo de REN foi determinado a partir da realização de 12 iterações de 120 treinos, compreendendo os cenários enumerados na Seção 6.4.2. Os treinos foram realizados no *corpus* trabalhista criado neste trabalho, contendo somente os 144 documentos revisados, complementados por 1.362 sentenças avulsas contendo valores monetários. Em relação aos modelos do ELMo utilizados, foram avaliados dois modelos pré-treinados em *corpus* jurídico, com e sem ajuste fino no *corpora* de REN deste mesmo domínio, e dois modelos correspondentes aos utilizados nas avaliações do domínio geral, pré-treinados nos *corpora* do Wikipedia e do brWaC. Estes dois modelos do domínio geral não foram avaliados aplicando ajuste fino no *corpus* jurídico. Para avaliação das representações estáticas de palavras, optou-se por avaliar vetores pré-treinados em acervo jurídico, comparando-os com os respectivos vetores de domínio geral [71]. Também foram realizados treinos utilizando somente ELMo e ELMo+CNN. Os treinos deste domínio foram realizados por 5 épocas. O *F-Score* de cada treino foi calculado utilizando o *script* CoNLL [91]. A Tabela 7.9 apresenta os resultados agrupados por tipo de ELMo utilizado. Percebe-se que o ajuste fino realizado no ELMo, no *corpus* de treino do modelo de REN trabalhista, pouco contribui para o desempenho dos modelos, possivelmente porque o ELMo jurídico já tenha sido treinado em documentos semelhantes aos que constituem os *corpora* de REN. Da mesma forma, percebe-se o aumento de desempenho ao usar um ELMo que tenha sido pré-treinado em acervo do mesmo domínio, com um aumento de cerca de 2.3% em relação ao ELMo treinado no Wikipedia. Ao contrário do que foi observado no domínio geral, no domínio jurídico os modelos de REN utilizando o ELMo treinado no brWaC tiveram um desempenho inferior aos que utilizaram o ELMo treinado no Wikipedia.

A Tabela 7.10 apresenta os resultados agrupados pelos tipos de representação de

Modelo	*F-Score*	Mínimo	Máximo	Desvio Padrão
ELMo-af Jurídico	**88.28%**	87.13%	89.15%	0.37%
ELMo Jurídico	88.25%	86.93%	89.12%	0.37%
ELMo Wikipedia	86.27%	83.38%	87.58%	0.83%
ELMo brWaC	85.70%	84.07%	87.36%	0.71%

Tabela 7.9: *Resultados obtidos para os 1.440 treinos no domínio jurídico trabalhista, agrupados por tipo de modelo de linguagem utilizado.* ***ELMo-af*** *indica que foi realizado ajuste fino no* ***corpora*** *de treino de REN. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

Representação	*F-Score*	Mínimo	Máximo	Desvio Padrão
ELMo+Vetor	**87.72%**	85.25%	89.06%	0.89%
ELMo+CNN+Vetor	87.56%	84.31%	89.15%	1.10%
ELMo	87.19%	84.54%	88.92%	1.89%
ELMo+CNN	86.86%	83.38%	88.75%	1.99%

Tabela 7.10: *Resultados obtidos para os 1.440 treinos no domínio jurídico trabalhista, agrupados pelo tipo de representação utilizada no treino. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

palavras avaliados. Percebe-se que as representações utilizando vetores estáticos possuem um desempenho superior em relação às que não utilizaram. A Tabela 7.11 faz uma comparação entre os tipos de algoritmos dos vetores estáticos utilizados. Assim como na Tabela 7.10, os valores médios de *F-Score* aqui apresentados também mostram o ganho de desempenho ao usar vetores estáticos neste domínio. Os vetores do *GloVe* e aqueles treinados usando *Skip-Gram* apresentam um desempenho semelhante. Assim como no domínio geral, neste domínio os vetores treinados utilizando *Continuous Bag of Words* também mostram ser uma pior opção em relação ao *Skip-Gram*.

A Tabela 7.12 apresenta os resultados dos treinos agrupados por tipo de vetor de palavras, e se foram pré-treinados no domínio trabalhista. O melhor resultado foi do *GloVe* de domínio geral, seguido pelo *Word2Vec* neste mesmo domínio, que teve desempenho 0.01% acima do *Wang2Vec* treinado no domínio trabalhista. De forma geral, houve uma pequena diferença entre os vetores de *GloVe*, *Word2Vec* e *Wang2Vec*, para os dois domínios. Os modelos treinados com *FastText* e sem vetores de palavras tiveram um resultado pior em relação aos demais modelos.

A Tabela 7.13 apresenta o desempenho médio dos modelos treinados considerando todos os agrupamentos apresentados: tipo do ELMo utilizado, representação das palavras, tipo de vetor de palavras e o respectivo algoritmo, e se os vetores de palavras foram treinados no domínio específico. O melhor resultado é dos modelos que: (i) usa-

Tipo de Vetor	*F-Score*	Mínimo	Máximo	Desvio Padrão
GloVe	**87.87%**	85.51%	89.15%	1.01%
Skip-Gram	87.76%	85.45%	89.00%	0.84%
CBoW	87.44%	84.31%	89.07%	1.11%
Sem Vetor	87.02%	83.38%	88.92%	1.94%

Tabela 7.11: *Resultados obtidos para os 1.440 treinos no domínio jurídico trabalhista, agrupados pelo tipo de algoritmo dos vetores de palavras utilizados nos treinos.* ***Sem Vetor*** *indica o agrupamento de treinos que não fizeram uso de vetores de palavras. Os tipos* ***Structured Skip-Gram*** *e* ***Continuous Window*** *do* ***Wang2Vec*** *foram contabilizados como* ***Skip-Gram*** *e* ***CBoW****, respectivamente. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

ram o ELMo jurídico **com** ajuste fino, (ii) usaram como representação a combinação ELMo+Vetor, (iii) utilizaram o *Word2Vec*, (iv) utilizaram o algoritmo do *Skip-Gram*, e (v) que tiveram o vetor estático pré-treinado no domínio trabalhista. Este modelo teve o desempenho médio de 88.77%. Considerando-se os melhores resultados individuais de cada um dos critérios analisados nas Tabelas 7.9 a 7.12, a melhor combinação entre eles seria utilizar ELMo jurídico com ajuste fino, e sem representação por CNN, utilizando o *GloVe* como vetor estático de palavras, sem ter sido pré-treinado no domínio trabalhista. Esta combinação é a que teve o 3º melhor desempenho individual na Tabela total 7.13, com desempenho médio de 88.68%.

A Tabela 7.14 apresenta os resultados finais deste domínio. Escolhidos os modelos *ELMo-af-Jurídico* combinado com o *Word2Vec* treinado em domínio jurídico com *Skip-Gram*, e o mesmo ELMo combinado com *GloVe* treinado em domínio geral do Português, foram realizados dez novos treinos de 50 épocas com cada uma das duas opções, para avaliação dos resultados finais. Antes da realização deste treino final, foram realizadas várias correções do *corpus* de REN trabalhista, o que incorreu em uma melhora no desempenho dos modelos, em relação ao *benchmark* realizado para avaliação das formas de representação. O modelo treinado em *GloVe* teve o melhor desempenho médio, com 93.81%, enquanto o *Word2Vec* jurídico teve 93.41%.

7.3.1 Contribuição do *ELMo* no Desempenho de REN no Domínio Jurídico

Assim como no domínio geral, foram realizados dois treinos adicionais com o intuito de estabelecer uma referência para a contribuição da representação contextual provida pelo *ELMo* treinado no acervo jurídico. Para isso, o primeiro treino realizado foi

Vetor	Domínio Específico	*F-Score*	Mínimo	Máximo	Desvio Padrão
GloVe	Não	**88.00%**	85.51%	89.15%	0.97%
Word2Vec	Não	87.80%	85.45%	89.07%	0.96%
Wang2Vec	Sim	87.79%	85.94%	88.78%	0.74%
GloVe	Sim	87.75%	85.75%	89.14%	1.04%
Wang2Vec	Não	87.69%	85.56%	88.96%	0.94%
Word2Vec	Sim	87.66%	85.18%	89.00%	0.93%
FastText	Não	87.39%	84.55%	88.88%	1.09%
FastText	Sim	87.28%	84.31%	88.64%	1.16%
Sem Vetor	-	87.02%	83.38%	88.92%	1.94%

Tabela 7.12: *Resultados obtidos para os 1.440 treinos no domínio jurídico trabalhista, agrupados pelo tipo de vetor de palavras e se foram pré-treinados no domínio específico.* ***Sem Vetor*** *indica o agrupamento de treinos que não fizeram uso de vetores de palavras. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

utilizando como representação de palavras somente a convolução de caracteres através de uma rede CNN. Para o segundo treino, foi utilizado como representação de palavras somente o *GloVe*[7], [74] como vetor de palavras. Da mesma forma, não foi realizada nenhuma validação cruzada. O resultado de cada treino foi: 63.49% de *F-Score* para o modelo que utilizou somente a representação de caracteres, e 78.47% para o modelo que utilizou somente o *GloVe*. Os dois resultados são bastante inferiores a todos os outros apontados na Tabela 7.13, justificando o descarte destes cenários.

7.3.2 Análise dos Resultados

A Figura 7.3 mostra os resultados por categoria das classificações realizadas pelo melhor modelo treinado em 50 épocas, baseado no ELMo-af-Jurídico e *GloVe*. Destaca-se o desempenho de 100% do modelo na identificação e classificação de entidades do tipo VALOR_CUSTAS e VARA. Assim como no domínio geral, os erros mais frequentes também são relativos à identificação de *tokens* que fazem parte das entidades, sendo as categorias FUNDAMENTO e LOCAL as mais desafiadoras neste sentido, dado o tamanho e diversidade das formas de se escrever fundamentos jurídicos e locais.

O Exemplo 7.7 mostra um caso de uma entidade do tipo LOCAL que não foi devidamente identificada e classificada pelo modelo. De toda a entidade *"D. João VII - Village Residencial"*, somente os dois últimos *tokens* foram identificados como entidades e classificados como LOCAL. A cidade *"Ariquemes-RO"* também foi corretamente clas-

[7]Escolhido este tipo aqui, pelo mesmo motivo em que foi escolhido no mesmo teste realizado no domínio geral.

Modelo	Representação	Vetor	Tipo de Vetor	Domínio Específico	*F-Score*	Mínimo	Máximo	Desvio Padrão
ELMo-af Jurídico	ELMo+Vetor	Word2Vec	Skip-Gram	Sim	**88.77%**	88.24%	88.88%	0.20%
ELMo-af Jurídico	ELMo+Vetor	GloVe	GloVe	Sim	88.68%	88.22%	89.06%	0.19%
ELMo-af Jurídico	ELMo+Vetor	GloVe	GloVe	Não	**88.68%**	88.64%	88.86%	0.07%
ELMo-af Jurídico	ELMo+CNN+Vetor	GloVe	GloVe	Não	88.67%	88.45%	89.15%	0.19%
ELMo Jurídico	ELMo+Vetor	Wang2Vec	CBoW	Sim	88.67%	88.40%	88.78%	0.10%
ELMo Jurídico	ELMo+CNN+Vetor	GloVe	GloVe	Não	88.67%	88.13%	89.12%	0.28%
ELMo-af Jurídico	ELMo+CNN+Vetor	Wang2Vec	Skip-Gram	Não	88.61%	88.46%	88.96%	0.15%
ELMo Jurídico	ELMo+Vetor	Word2Vec	CBoW	Sim	88.61%	88.08%	88.78%	0.26%
ELMo Jurídico	ELMo+Vetor	Word2Vec	Skip-Gram	Não	88.57%	88.53%	88.75%	0.06%
ELMo-af Jurídico	ELMo+CNN+Vetor	Word2Vec	CBoW	Não	88.55%	88.04%	89.07%	0.30%
ELMo-af Jurídico	ELMo	Sem Vetor	Sem Vetor	Não	88.54%	88.18%	88.92%	0.24%
...								
ELMo Wikipedia	ELMo+CNN+Vetor	FastText	Skip-Gram	Sim	86.29%	85.73%	86.77%	0.35%
ELMo Wikipedia	ELMo+CNN+Vetor	GloVe	GloVe	Não	86.26%	85.51%	86.82%	0.31%
ELMo Wikipedia	ELMo+CNN+Vetor	Wang2Vec	CBoW	Não	86.22%	85.56%	86.70%	0.38%
ELMo Wikipedia	ELMo+CNN+Vetor	Word2Vec	CBoW	Sim	86.11%	85.18%	86.74%	0.50%
ELMo Wikipedia	ELMo+CNN+Vetor	Word2Vec	CBoW	Não	86.03%	85.56%	86.48%	0.28%
ELMo Wikipedia	ELMo+Vetor	FastText	CBoW	Não	86.02%	85.25%	86.09%	0.24%
ELMo Wikipedia	ELMo+Vetor	FastText	CBoW	Sim	85.38%	85.38%	85.38%	0.00%
ELMo Wikipedia	ELMo+CNN+Vetor	FastText	CBoW	Não	85.21%	84.55%	85.93%	0.50%
ELMo Wikipedia	ELMo+CNN+Vetor	FastText	CBoW	Sim	84.76%	84.31%	85.80%	0.40%
ELMo Wikipedia	ELMo	Sem Vetor	Sem Vetor	Não	84.56%	84.54%	84.76%	0.06%
ELMo Wikipedia	ELMo+CNN	Sem Vetor	Sem Vetor	Não	84.13%	83.38%	84.96%	0.46%

Tabela 7.13: *Resultados obtidos para os 1.440 treinos no domínio jurídico trabalhista, agrupados para cada cenário avaliado nos treinos.* ***ELMo-af*** *indica que foi realizado ajuste fino no corpora de treino de REN.* ***Sem Vetor*** *indica o agrupamento de treinos que não fizeram uso de vetores de palavras. Os tipos* ***Structured Skip-Gram*** *e* ***Continuous Window*** *do* ***Wang2Vec*** *foram contabilizados como* ***Skip-Gram*** *e* ***CBoW****, respectivamente. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

ELMo	Ajuste Fino do ELMo	Vetor	Tipo de Vetor	Domínio Específico	*F-Score*
ELMo Jurídico	Sim	GloVe	GloVe	Não	**93.81%**
ELMo Jurídico	Sim	Word2Vec	Skip-Gram	Sim	93.41%

Tabela 7.14: *Resultados obtidos para os 20 treinos finais no domínio jurídico trabalhista, agrupados pelos dois melhores modelos avaliados.* ***Ajuste Fino do ELMo*** *indica que foi realizado ajuste fino do modelo biLM no corpora de treino de REN.* ***Domínio Específico*** *indica se os vetores foram pré-treinados em acervo jurídico. O* ***F-Score*** *de cada grupo é a média dos resultados de cada grupo.*

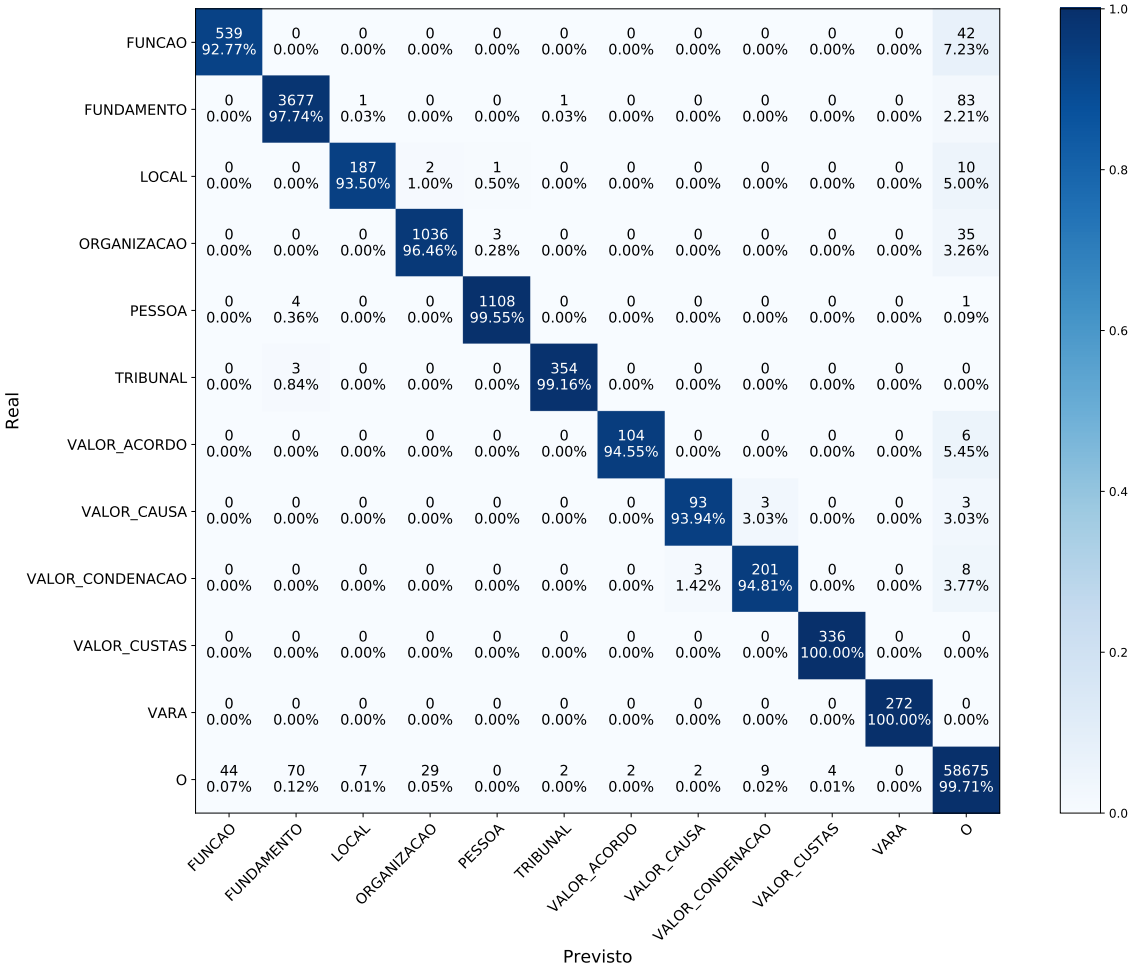

Figura 7.3: *Matriz de confusão do melhor modelo* ***ELMo-af-Jurídico-GloVe*** *treinado no corpus de REN jurídico.*

sificada. O Exemplo 7.8 mostra casos de entidades complexas do tipo LOCAL que foram identificadas e classificadas corretamente pelo modelo. São endereços completos, de 14 e 20 *tokens*, que tiveram suas fronteiras corretamente delimitadas pelo modelo.

Exemplo 7.7: *Exemplo de erros de classificação no domínio jurídico, em que tipos*

LOCAL não foram corretamente identificados. Ao lado de cada ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido no* ***corpus*** *de teste. Foram destacadas as classificações somente dos erros de classificação. Os* ***tokens*** *em* ***negrito*** *correspondem a entidades do tipo LOCAL que foram corretamente classificadas pelo modelo.*

"Pelo exposto , conheço da ação de embargos de terceiro ajuizados por RONALDO PIO DE ALMEIDA em face de UNIÃO FEDERAL vinculada aos autos de processo n. 0000219-44.2016.5.14.0161 , uma vez que preenchidos os pressupostos legais para sua admissibilidade , e , no mérito , DEFIRO o pedido formulado pela embargante para levantamento da penhora realizada sobre o bem assim descrito : "D. (O,B-LOCAL) João (O,I-LOCAL) VII (O,I-LOCAL) - (O,I-LOCAL) ***Village Residencial*** *", situado na cidade de* ***Ariquemes-RO*** *, matrícula nº 22203"*

Exemplo 7.8: *Exemplos de entidades complexas do tipo LOCAL que foram classificadas corretamente em sua totalidade. Estas entidades possuem 14 e 20* ***tokens****, respectivamente.*

"Avenida Deputado Raimundo Holanda , 347 – Morro da Saudade - Piripiri - Piauí"
"Alameda das Carnaubeiras , 833 , Presidente Costa e Silva , MOSSORO - RN - CEP : 59625 - 410"

No Exemplo 7.9 são exibidos erros de identificação e classificação de *tokens* do tipo FUNDAMENTO. No primeiro exemplo, o modelo reconheceu a entidade *LEI Nº 7.347 / 1985 , ART . 2º*, mas errou na delimitação da fronteira, incluindo na entidade um *token* "."a mais. A entidade *CÓDIGO DE DEFESA DO CONSUMIDOR , ART . 93* foi classificada de forma totalmente correta, mas *Res . 186 / 2012 , DEJT* deixou de ser identificada pelo modelo como entidade. O Exemplo 7.10 mostra um caso de uma sentença em que quatro entidades do tipo FUNDAMENTO distintas foram corretamente identificadas e classificadas pelo modelo, com suas fronteiras delimitadas de forma correta. Cada uma destas entidades possui um padrão diferente das demais: um artigo (*art. 37 , da CF*), uma circular de serviço (*CDS 66 / 86 de 6 / 9 / 1984*), uma resolução (*Resolução n.º 15 / 87*) e uma norma interna (*ADMPE / 12 de 1992*), mostrando a capacidade do modelo de generalizar diferentes padrões. O Exemplo 7.11 mostra um caso de um FUNDAMENTO complexo de 20 *tokens*, do tipo doutrina, que foi identificado e classificado corretamente, em todos os *tokens* que a constituem.

Exemplo 7.9: *Exemplos de erros de classificação no domínio jurídico, em que tipos*

FUNDAMENTO não foram corretamente identificados. Ao lado de cada ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido no* ***corpus*** *de teste. Foram destacadas as classificações somente dos erros de classificação. Os* ***tokens*** *em* ***negrito*** *correspondem a entidades do tipo FUNDAMENTO que foram corretamente classificadas pelo modelo.*

"130 . AÇÃO CIVIL PÚBLICA . COMPETÊNCIA . LOCAL DO DANO . ***LEI Nº 7.347 / 1985 , ART . 2º*** *. (I-FUNDAMENTO,O)* ***CÓDIGO DE DEFESA DO CONSUMIDOR , ART . 93*** *(redação alterada na sessão do Tribunal Pleno realizada em 14.09.2012) - Res (O,B-FUNDAMENTO) . (O,I-FUNDAMENTO) 186 (O,I-FUNDAMENTO) / (O,I-FUNDAMENTO) 2012 (O,I-FUNDAMENTO) , (O,I-FUNDAMENTO) DEJT (O,L-FUNDAMENTO) divulgado em 25 , 26 e 27.09.2012"*

"Fixo os juros conforme art. (O,B-FUNDAMENTO) ***1º-F da Lei 9.494 / 1997*** *. Imposto de renda e contribuição previdenciária , como de lei . "*

Exemplo 7.10: *Exemplo de uma sentença com quatro exemplos de entidades do tipo FUNDAMENTO que foram classificadas corretamente em sua totalidade. Os tokens em* ***negrito*** *destacam tais entidades.*

"Pleiteia o reclamante a reintegração ao emprego alegando a nulidade da dispensa . Afirma que foi contratado mediante concurso público pelo Banco do Estado do Paraná S / A. , sociedade de economia mista , e para a sua demissão , não foi observado o princípio da motivação , preconizado no ***art. 37 , da CF*** *. Aduz , ainda , que seu empregador originário editou a* ***CDS 66 / 86 de 6 / 9 / 1984*** *, a* ***Resolução n.º 15 / 87*** *e a* ***ADMPE / 12 de 1992*** *, estabelecendo a necessidade de motivação para a aplicação de penalidades , dentre as quais , a demissão sem justa causa ."*

Exemplo 7.11: *Exemplo de um tipo complexo de entidade do tipo FUNDAMENTO, uma doutrina. O modelo identificou corretamente todos os 20* ***tokens*** *da entidade.*

"Humberto Theodoro Junior , com efeito , ensina que "legitimados ao processo são os sujeitos da lide , isto é , os titulares dos interesses em conflito . A legitimação ativa caberá ao titular do interesse afirmado na pretensão , e a passiva ao titular do interesse que opõe ou resiste à pretensão "(in ***Curso de Direito Processual Civil , Vol . I , 14ª edição , Ed . Forense , pág . 57*** *)"*

O Exemplo 7.12 mostra casos de VALOR_ACORDO que foram corretamente identificados e classificados pelo modelo. A maioria dos padrões de acerto do modelo se

deram nestes padrões. No primeiro padrão é indicado objetivamente que o réu pagará ao autor do processo o valor acordado, como descrito em *"A 1ª ré pagará a importância líquida e total de..."*. Já no segundo padrão, o valor de acordo é definido juntamente com o valor de custas, declarado em função do valor acordado. A indicação do modelo neste cenário se dá pelo contexto do documento e da frase, visto que as custas são *"pelo exeqüente"*[8]. Caso fosse um valor de condenação, as custas seriam de responsabilidade do réu do processo. Já no Exemplo 7.13 são mostrados alguns erros do modelo, nos quais ele não foi capaz de interpretar o contexto das sentenças para detectar que os valores mencionados eram de acordo. A matriz de confusão não indica que o modelo confundiu valores de acordo com outros tipos de valores, como é o caso de VALOR_CAUSA e VALOR_CONDENACAO. Os Exemplos 7.14 e 7.15 mostram casos em que o modelo confundiu valores de causa com valores de condenação. No primeiro exemplo o modelo entendeu um valor de causa como sendo de condenação, enquanto no segundo exemplo ele entendeu o contrário. A dificuldade do Exemplo 7.15 se deve ao fato de que o valor de condenação do processo foi exatamente igual ao valor pretendido da causa, e o modelo não foi capaz de detectar este contexto.

Exemplo 7.12: *Exemplos de padrões de VALOR_ACORDO corretamente classificados pelo modelo. Valores em* ***negrito*** *destacam os valores de acordo identificados.*

"A 1ª ré pagará a importância líquida e total de ***R $ 2247,45*** *(R $ 1954,30 ao autor e R $ 293,15 de honorários advocatícios) , sendo R $ 1123,72 , referentes à primeira parcela do acordo , em 17 / 04 / 2014 , e o restante conforme discriminado a seguir :"*

"Custas pelo exeqüente no importe de R $ 4.200,00 , calculadas sobre ***R $ 210.000,00*** *, dispensadas na forma da lei ."*

Exemplo 7.13: *Exemplos de erros de classificação no domínio jurídico, em que tipos VALOR_ACORDO não foram corretamente identificados. Ao lado de cada* ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido no* ***corpus*** *de teste. Foram destacadas as classificações somente dos erros de classificação.*

"Para fins de quantificação do dano moral , utilizo o parâmetro já utilizado pelo MPT quando da celebração do acordo com a primeira reclamada . Foi celebrado acordo no valor de R (O,B-VALOR_ACORDO) $ (O,I-VALOR_ACORDO) 226.298,87 (O,L-

[8] **Exeqüente** é um tipo de polo ativo (autor) de um processo de Execução, sendo que o polo passivo (réu) se chama **Executado**.

VALOR_ACORDO) (Id 838b40b) ."

"VALOR LÍQUIDO : R (O,B-VALOR_ACORDO) $ (O,I-VALOR_ACORDO) 7.000,00 (O,L-VALOR_ACORDO) (sete mil reais) VALOR LÍQUIDO : R $ (Contribuição Sindical) e R $ (Honorários advocatícios incidindo sobre o valor de R $ 7.000,00)"

Exemplo 7.14: *Exemplo de erro de classificação de VALOR_CAUSA, em que o modelo inferiu que o valor em questão era VALOR_CONDENACAO. Ao lado de cada **token** classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido no **corpus** de teste. Foram destacadas as classificações somente dos erros de classificação.*

"Custas , pela requerida , no importe de R $ 200,00 , calculadas sobre o valor atribuído à causa , R (B-VALOR_CONDENACAO,B-VALOR_CAUSA) $ (I-VALOR_CONDENACAO,I-VALOR_CAUSA) 10.000,00 (L-VALOR_CONDENACAO,L-VALOR_CAUSA) , a serem recolhidas em 48 horas após a citação ."

Exemplo 7.15: *Exemplo de erro de classificação de VALOR_CONDENACAO, em que o modelo inferiu que o valor em questão era VALOR_CAUSA. Ao lado de cada **token** classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido no **corpus** de teste. Foram destacadas as classificações somente dos erros de classificação.*

"Custas de R $ -2.230,45 (dois mil , duzentos e trinta reais e quarenta e cinco centavos) , pelas rés , calculadas sobre o valor dado à causa de R (B-VALOR_CAUSA,B-VALOR_CONDENACAO) $ (I-VALOR_CAUSA,I-VALOR_CONDENACAO) -111.522,88 (L-VALOR_CAUSA,L-VALOR_CONDENACAO) (cento e onze mil , quinhentos e vinte e dois reais e oitenta e oito centavos)."

Pela matriz de confusão da Figura 7.3, nota-se que a categoria com a qual o modelo teve mais dificuldade foi a do tipo FUNCAO. Percebe-se que os tokens do tipo FUNCAO não foram confundidos com nenhuma outra categoria. Os erros desta categoria foram: (i) por não terem sido identificadas, (ii) por terem sido identificados em *tokens* que não eram parte de entidades, de acordo com as anotações. O Exemplo 7.16 mostra alguns dos diferentes tipos de erros em relação a esta categoria. Apesar do critério mais restritivo estabelecido para esta categoria[9], somente o erro destacado no terceiro exemplo

[9]Os *tokens* só foram anotados como funções se estivessem acompanhando o nome da pessoa desempenhando o papel correspondente no processo.

da figura se enquadrou em uma predição equivocada do modelo em um cenário que fugiu deste critério.

Exemplo 7.16: *Exemplos de erros de classificação de entidades do tipo FUNCAO. Ao lado de cada* ***token*** *classificado de forma errada está a classificação feita pelo modelo treinado, e o rótulo definido no* ***corpus*** *de teste. Foram destacadas as classificações somente dos erros de classificação. Termos em* ***negrito*** *destacam as funções corretamente identificadas.*

*"Aos 26 dias do mês de abril do ano de dois mil e dezoito , às 11h13min , na sala de audiências desta 1ª Vara do Trabalho de Maceió / AL - TRT / 19ª Região , na presença da MM . **Juíza do Trabalho Substituta** Kellen Yoko Nakao , foram apregoados os litigantes (O,U-FUNCAO) , JOCELINO MOREIRA BARROS , **parte autora** , e RETOQUE RÁPIDO PINTURA AUTOMOTIVA , parte (O,B-FUNCAO) ré (O,L-FUNCAO) ."*

"E , para constar , eu , Rafaela Ribeiro Ramos , técnico (O,B-FUNCAO) judiciário (O,L-FUNCAO) , digitei a presente , que vai assinada , na forma da lei ."

"O demandante , por seu turno , pretende a majoração para R $ 1.000.000,00 (relatório do Exmº Des (B-FUNCAO,O) . (I-FUNCAO,O) Relator (L-FUNCAO,O))."

CAPÍTULO 8

Conclusão

Neste trabalho, foram avaliados modelos de representações de palavras e Reconhecimento de Entidades Nomeadas baseados em arquiteturas de Redes Neurais Profundas, tanto para o domínio geral da língua portuguesa, quanto para o domínio da Justiça do Trabalho do Brasil. Adotou-se o ELMo [75] como sendo o modelo de linguagem para representação contextual de palavras a ser utilizado na tarefa de REN, na hipótese de que ele melhoraria a acurácia desta tarefa para a língua portuguesa, em relação aos modelos encontrados na literatura para este idioma. A arquitetura do modelo de REN foi a rede *LSTM-CRF*, amplamente adotada na literatura. Também foram avaliadas diferentes formas de representação de palavras dentro dos modelos de REN treinados no *framework* do *AllenNLP* [30], com o objetivo de identificar a combinação de representações que traria o melhor desempenho para esta tarefa.

Também foi desenvolvido neste trabalho um *corpus* da justiça trabalhista do Brasil, anotado com entidades tradicionais da tarefa de REN, tais como *Pessoa*, *Organização* e *Local*, e também foram anotadas entidades específicas do domínio em questão, tais como *Fundamento*, *Vara* e *Valor de Condenação*. Neste *corpus* foram conduzidos outros estudos visando a obtenção de um modelo eficiente de extração de entidades para este domínio.

8.1 Sumário das Principais Contribuições

Duas hipóteses foram apresentadas no início deste trabalho: a de que os modelos propostos iriam melhorar o desempenho da tarefa de REN para a língua portuguesa, e que seria possível criar um modelo de REN para o domínio trabalhista com um desempenho de pelo menos 80%. Para comprovação destas hipóteses, vários objetivos foram estabelecidos e cumpridos, conforme a lista a seguir:

- Foi treinado um modelo de REN para o domínio da Justiça do Trabalho do Brasil com o desempenho de 93.81% no *F-Score*, utilizando como representação de

palavras um ELMo pré-treinado em acervo de documentos trabalhistas e um *GloVe* obtido a partir de [71].

- Foi treinado um modelo de REN para a língua portuguesa que elevou o estado da arte da tarefa no *benchmark* do HAREM[1] de 71.23%, no cenário seletivo, para 83.22%; e de 69.14%, no cenário total, para 78.04%, mostrando a eficiência da representação de palavras do ELMo.
- Após um estudo comparativo extenso, avaliando diferentes tipos de representações de palavras e vetores estáticos, concluiu-se que o melhor desempenho para a tarefa de REN no domínio geral do Português foi *ELMo+CNN+Wang2Vec*, e que o melhor vetor estático foi o *Wang2Vec*, no modelo *Structured Skip-Gram*.
- Concluiu-se, também, que apesar de ser um modelo de linguagem com uma perplexidade quase duas vezes maior, o ELMo treinado no brWaC tem um desempenho superior na tarefa de REN do domínio geral do Português, em relação ao ELMo treinado no Wikipedia. O tamanho do *corpus* de treino pode implicar em uma maior qualidade da representação das palavras pelo ELMo, apesar da métrica de qualidade do modelo de linguagem não ser necessariamente melhor.
- Utilizando a arquitetura *biLM* [75], foram treinados modelos de linguagem para os domínios geral do Português e do direito trabalhista, que se mostraram indispensáveis para a otimização do desempenho do modelo de REN nestes domínios.
- Para o domínio específico do direito, também foram treinados e avaliados vetores estáticos de palavras em acervo jurídico. Mesmo tendo sido submetidos ao mesmo pré-processamento, e mesmo tendo sido pré-treinados em um *corpus* quase cinco vezes maior que os vetores de [71], não houve um ganho de desempenho na tarefa de REN do domínio específico, talvez porque o ELMo se torne predominante neste cenário em relação aos vetores estáticos, conforme se observou nos resultados apresentados.
- Finalmente, foi construído um *corpus* trabalhista baseado em documentos públicos de processos trabalhistas, que foi utilizado para o treino do modelo de REN do domínio jurídico deste trabalho.

8.2 Publicações Geradas e Artigos em Andamento

As seguintes contribuições foram realizadas durante o desenvolvimento deste trabalho:

- QUINTA DE CASTRO, P. V.; FÉLIX FELIPE DA SILVA, N.; DA SILVA SOARES, A. **Portuguese Named Entity Recognition using LSTM-CRF**. In: Villavi-

[1] HAREM I para treino e MiniHAREM para teste.

cencio, A.; Moreira,V.; Abad, A.; Caseli, H.; Gamallo, P.; Ramisch, C.; Gonçalo Oliveira, H.; Paetzold, G. H., editors, Computational Processing of the Portuguese Language, p.83–92, Cham, 2018. Springer International Publishing.

- QUINTA DE CASTRO, P. V.; FÉLIX FELIPE DA SILVA, N.; DA SILVA SOARES, A. **Contextual Representations and Semi-Supervised Named Entity Recognition for Portuguese Language**. In: Iberian Languages Evaluation Forum (IberLEF 2019), 2019, Bilbao, Spain. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019) co-located with 35th Conference of the Spanish Society for Natural Language Processing (SEPLN 2019), 2019. v. 2421. p.411–420.
- Modelo de linguagem pré-treinado do ELMo para a língua portuguesa, disponível publicamente em [2].
- Encontra-se em andamento a escrita de um artigo sobre Reconhecimento de Entidades Nomeadas para Português a ser publicado em periódico indexado.
- Pretende-se também submeter um artigo para o periódico indexado *Artificial Intelligence and Law*[2] com os resultados obtidos para o domínio jurídico.

8.3 Limitações e Perspectivas Futuras

Considerando que os vetores estáticos de palavras que foram pré-treinados no domínio jurídico não trouxeram ganho de desempenho para o modelo de REN neste domínio[3], serão avaliadas outras formas de pré-processamento do acervo jurídico para obtenção de uma representação mais significativa para este domínio.

Como houve um ganho de desempenho no modelo de REN do domínio geral aumentando o tamanho do *corpus* de treino do ELMo, também será experimentado aumentar o tamanho do *corpus* do domínio jurídico. Considerando os indícios de que não há uma relação clara entre a perplexidade de um modelo de linguagem, e o desempenho do mesmo na tarefa de REN, será avaliado aplicar o mesmo critério de seleção do vocabulário que foi feito no domínio geral. Serão mantidas no vocabulário somente palavras de acordo com a frequência das mesmas no *corpus* de treino do ELMo, ao invés de descartar valores monetários, números de processos e identificadores de documentos.

As avaliações dos estudos comparativos do domínio jurídico serão melhoradas por meio da aplicação de validação cruzada, tal qual foi feita no domínio geral. Estima-se aproximadamente 60 dias para a realização de todos os treinos envolvidos na comparação de dois modelos propostos.

[2] `https://link.springer.com/journal/10506`

[3] Em comparação com os vetores estáticos obtidos em [71].

Em relação ao *corpus* de REN trabalhista que foi criado para este trabalho, serão revisados mais documentos dentre os 1.161 que foram anotados automaticamente. Ainda em relação ao *corpus* criado, é considerada uma limitação deste trabalho que ele tenha sido anotado e revisado por somente uma pessoa. Tem-se como objetivo melhorar a qualidade do trabalho de anotação adicionando mais anotadores, de forma que seja possível fazer uma avaliação de concordância entre eles.

Referências Bibliográficas

[1] AKBIK, A.; BLYTHE, D.; VOLLGRAF, R. **Contextual string embeddings for sequence labeling**. In: *COLING 2018, 27th International Conference on Computational Linguistics*, p. 1638–1649, 2018. 27, 43, 44, 49, 53, 54, 77, 80, 87

[2] ALLENNLP. **Elmo: Deep contextualized word representations**. URL: https://allennlp.org/elmo. 115

[3] ANGELIDIS, I.; CHALKIDIS, I.; KOUBARAKIS, M. **Named Entity Recognition, Linking and Generation for Greek Legislation**. In: *JURIX*, p. 1–10. IOS Press, 2018. 23, 57, 77

[4] BADJI, I.; CORCHO, O.; RODRÍGUEZ-DONCEL, V. **Legal Entity Extraction with NER Systems**. Master's thesis, Universidad Politécnica de Madrid, 2018. 23, 57

[5] BASILE, C. R. O. **Processo do trabalho: justiça do trabalho e dissídios trabalhistas**. Saraiva, 2012. 60, 61

[6] BENGIO, Y.; SIMARD, P.; FRASCONI, P. **Learning long-term dependencies with gradient descent is difficult**. *Trans. Neur. Netw.*, 5(2):157–166, Mar. 1994. 43

[7] BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. **Enriching word vectors with subword information**. *Transactions of the Association for Computational Linguistics*, 5:135–146, Dec. 2017. 44, 46, 79, 86, 88, 89

[8] CARDELLINO, C.; TERUEL, M.; ALEMANY, L. A.; VILLATA, S. **A low-cost, high-coverage legal named entity recognizer, classifier and linker**. In: *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, p. 9–18, New York, NY, USA, 2017. ACM. 23, 57

[9] CAVAIONI, M. **Deeplearning series: Convolutional neural networks**. URL: https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524. 15, 41

[10] **Constituição da república federativa do brasil**. URL: http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm, 1988. 61

[11] CHALKIDIS, I.; ANDROUTSOPOULOS, I. **A Deep Learning Approach to Contract Element Extraction**. In: *JURIX*, p. 155–164, 2017. 57

[12] CHALKIDIS, I.; ANDROUTSOPOULOS, I.; MICHOS, A. **Extracting contract elements**. In: *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, p. 19–28, New York, NY, USA, 2017. ACM. 57

[13] CHAURASIA, M. **Implement your own word2vec(skip-gram) model in python**. URL: https://www.geeksforgeeks.org/implement-your-own-word2vecskip-gram-model-in-python/. 15, 45

[14] CHELBA, C.; MIKOLOV, T.; SCHUSTER, M.; GE, Q.; BRANTS, T.; KOEHN, P.; ROBINSON, T. **One billion word benchmark for measuring progress in statistical language modeling**. Technical report, Google, 2013. 80, 81

[15] CHIU, J. P.; NICHOLS, E. **Named entity recognition with bidirectional LSTM-CNNs**. *Transactions of the Association for Computational Linguistics*, 4:357–370, Dec. 2016. 27, 39, 43, 50, 53, 54, 77, 87

[16] **Consolidação das leis do trabalho**. URL: http://www.planalto.gov.br/ccivil_03/decreto-lei/del5452.htm, 1943. 61

[17] **Justiça em números**. http://www.cnj.jus.br/pesquisas-judiciarias/justicaemnumeros/2016-10-21-13-13-04/pj-justica-em-numeros, 2018. Último acesso: 01/05/2019. 23, 25, 63

[18] COLLOBERT, R.; WESTON, J.; BOTTOU, L.; KARLEN, M.; KAVUKCUOGLU, K.; KUKSA, P. **Natural language processing (almost) from scratch**. *J. Mach. Learn. Res.*, 12:2493–2537, Nov. 2011. 39, 52, 53, 54, 55, 77

[19] COLLOVINI, S.; BONAMIGO, T. L.; VIEIRA, R. **A review on relation extraction with an eye on portuguese**. *Journal of the Brazilian Computer Society*, 19:553–571, 2013. 22, 23

[20] **Código de processo civil**. URL: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13105.htm, 2015. 60, 61

[21] DA COSTA, P.; PAETZOLD, G. H. **Effective sequence labeling with hybrid neural-crf models**. In: *International Conference on Computational Processing of the Portuguese Language*, p. 490–498. Springer, 2018. 27, 33, 55, 56, 77, 87, 97

[22] DANG, T. H.; LE, H.-Q.; NGUYEN, T. M.; VU, S. T. **D3ner: biomedical named entity recognition using crf-bilstm improved with fine-tuned embeddings of various linguistic information**. *Bioinformatics*, 34(20):3539–3546, 2018. 22

[23] DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. **Bert: Pre-training of deep bidirectional transformers for language understanding**. *arXiv preprint arXiv:1810.04805*, 2018. 27, 44, 49, 53, 54, 77, 80

[24] DO AMARAL, D. O. F.; VIEIRA, R. **O reconhecimento de entidades nomeadas por meio de conditional random fields para a língua portuguesa (named entity recognition with conditional random fields for the Portuguese language) [in Portuguese]**. In: *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, 2013. 27, 55, 56, 97

[25] DO AMARAL, D. O. F. **Reconhecimento de entidades nomeadas na área da geologia: bacias sedimentares brasileiras**. PhD thesis, Pontifícia Universidade Católica do Rio Grande do Sul, 2017. 22, 27, 33, 55

[26] DOS SANTOS, C.; GUIMARÃES, V. **Boosting named entity recognition with neural character embeddings**. In: *Proceedings of the Fifth Named Entity Workshop*, p. 25–33, Beijing, China, July 2015. Association for Computational Linguistics. 27, 33, 39, 49, 50, 52, 53, 55, 56, 77, 80, 87, 97

[27] DOS SANTOS, C. N.; MILIDIÚ, R. L. **Entropy Guided Transformation Learning - Algorithms and Applications**. Springer Briefs in Computer Science. Springer, 2012. 27, 55, 56, 87, 97

[28] DOZIER, C.; REUTERS, T.; KONDADADI, R.; LIGHT, M.; VACHHER, A.; VEERAMACHANENI, S.; WUDALI, R. **Named entity recognition and resolution in legal text**. In: *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, p. 27–43, 01 2010. 23, 57

[29] FILHO, J. A. W.; WILKENS, R.; IDIART, M.; VILLAVICENCIO, A. **The brWaC Corpus: A New Open Resource for Brazilian Portuguese**. In: chair), N. C. C.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Hasida, K.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; Piperidis, S.; Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). 83

[30] FOR ARTIFICIAL INTELLIGENCE, A. I. **Allennlp**. URL: `https://allennlp.org/`. 78, 113

[31] FOR ARTIFICIAL INTELLIGENCE, A. I. **Tensorflow implementation for pretraining bilm**. URL: https://github.com/allenai/bilm-tf/. 86

[32] FUKUSHIMA, K. **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position**. *Biological Cybernetics*, 36(4):193–202, Apr 1980. 39

[33] GENSIM. **models.fasttext – fasttext model**. URL: https://radimrehurek.com/gensim/models/fasttext.html. 80, 89

[34] GENSIM. **models.word2vec – word2vec embeddings**. URL: https://radimrehurek.com/gensim/models/word2vec.html. 80, 89

[35] GOLDBERG, Y. **A primer on neural network models for natural language processing**. *J. Artif. Int. Res.*, 57(1):345–420, Sept. 2016. 47, 48, 49

[36] GOLDBERG, Y.; HIRST, G. **Neural Network Methods in Natural Language Processing**. Morgan & Claypool Publishers, 2017. 47

[37] GRAVES, A.; RAHMAN MOHAMED, A.; HINTON, G. E. **Speech recognition with deep recurrent neural networks**. *CoRR*, abs/1303.5778, 2013. 39, 43

[38] GRUBER, T. **Ontology**. In: *Encyclopedia of Database Systems (2nd ed.)*. Springer, 2018. 59

[39] HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; SILVA, J.; ALUÍSIO, S. **Portuguese word embeddings: Evaluating on word analogies and natural language tasks**. In: *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*, p. 122–131, Uberlândia, Brazil, Oct. 2017. Sociedade Brasileira de Computação. 58, 80

[40] HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998. 37

[41] HOCHREITER, S.; SCHMIDHUBER, J. **Long short-term memory**. *Neural Comput.*, 9(8):1735–1780, Nov. 1997. 43

[42] HOVY, E.; MARCUS, M.; PALMER, M.; RAMSHAW, L.; WEISCHEDEL, R. **Ontonotes: The 90% solution**. In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, p. 57–60, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. 33

[43] HOWARD, J.; RUDER, S. **Universal language model fine-tuning for text classification**. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, p. 328–339, 2018. 43, 44, 49

[44] HUANG, Z.; XU, W.; YU, K. **Bidirectional lstm-crf models for sequence tagging**. *CoRR*, abs/1508.01991, 2015. 27, 39, 43, 53, 54, 77

[45] JAEGER ZABALA, F.; SILVEIRA, F. F. **Jurimetria: Estatística aplicada ao direito**. *Revista Direito e Liberdade*, 16(1):87–103, 2014. 62, 63

[46] JARRETT, K.; KAVUKCUOGLU, K.; RANZATO, M.; LECUN, Y. **What is the best multi-stage architecture for object recognition?** In: *2009 IEEE 12th International Conference on Computer Vision*, p. 2146–2153, Sep. 2009. 39, 79

[47] JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. **Bag of tricks for efficient text classification**. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, p. 427–431, Valencia, Spain, Apr. 2017. Association for Computational Linguistics. 44, 46, 79, 86, 88, 89

[48] JÚNIOR, C. M.; MACEDO, H.; BISPO, T.; SANTOS, F.; SILVA, N.; BARBOSA, L. **Paramopama: a Brazilian-Portuguese Corpus for Named Entity Recognition**. Technical report, Universidade Federal de Sergipe, 2015. 33

[49] KARN, U. **An intuitive explanation of convolutional neural networks**. URL: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/. 15, 41

[50] KIM, Y.; JERNITE, Y.; SONTAG, D.; RUSH, A. M. **Character-aware neural language models**. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2741–2749. AAAI Press, 2016. 50

[51] KRISHNAN, V.; GANAPATHY, V. **Named entity recognition**, 2005. 34

[52] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **Imagenet classification with deep convolutional neural networks**. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, p. 1097–1105, USA, 2012. Curran Associates Inc. 39

[53] LAFFERTY, J. D.; MCCALLUM, A.; PEREIRA, F. C. N. **Conditional random fields: Probabilistic models for segmenting and labeling sequence data**. In: *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, p. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. 35, 78

[54] LAMPLE, G.; BALLESTEROS, M.; SUBRAMANIAN, S.; KAWAKAMI, K.; DYER, C. **Neural architectures for named entity recognition**. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, p. 260–270, San Diego, California, June 2016. Association for Computational Linguistics. 16, 27, 39, 43, 49, 52, 53, 54, 56, 58, 77, 78, 80, 87

[55] LE CUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. **Handwritten digit recognition with a back-propagation network**. In: *Proceedings of the 2Nd International Conference on Neural Information Processing Systems*, NIPS'89, p. 396–404. MIT Press, Cambridge, MA, USA, 1989. 39

[56] LI, J.; SUN, A.; HAN, J.; LI, C. **A survey on deep learning for named entity recognition**. *CoRR*, abs/1812.09449, 2018. 20, 30, 31, 33

[57] LING, W.; DYER, C.; BLACK, A.; TRANCOSO, I. **Extension of the original word2vec using different architectures**. URL: https://github.com/wlin12/wang2vec. 80, 89, 90

[58] LING, W.; DYER, C.; BLACK, A. W.; TRANCOSO, I. **Two/too simple adaptations of Word2Vec for syntax problems**. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1299–1304, Denver, Colorado, May–June 2015. Association for Computational Linguistics. 44, 45, 79, 86, 89, 90

[59] **Coleção dourada do harem i e miniharem**. URL: https://www.linguateca.pt/primeiroHAREM/harem_coleccaodourada.html. 87

[60] LUZ DE ARAUJO, P. H.; DE CAMPOS, T. E.; DE OLIVEIRA, R. R. R.; STAUFFER, M.; COUTO, S.; BERMEJO, P. **Lener-br: a dataset for named entity recognition in brazilian legal text**. In: *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), p. 313–323, Canela, RS, Brazil, September 24-26 2018. Springer. 23, 26, 33, 57, 77

[61] MA, X.; HOVY, E. **End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF**. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 1064–1074, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. 27, 36, 39, 43, 50, 53, 54, 77, 87

[62] MARRERO, M.; URBANO, J.; SÁNCHEZ-CUADRADO, S.; MORATO, J.; GÓMEZ-BERBÍS, J. M. **Named entity recognition: fallacies, challenges and opportunities**. *Computer Standards & Interfaces*, 35(5):482–489, 2013. 22, 56

[63] MAYNARD, D.; BONTCHEVA, K.; AUGENSTEIN, I. **Natural language processing for the semantic web**. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 6(2):1–194, 2016. 20, 22, 30

[64] MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. **Efficient estimation of word representations in vector space**. *CoRR*, 2013. 44, 45, 53, 55, 57, 79, 86, 88, 89

[65] MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. **Distributed representations of words and phrases and their compositionality**. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, p. 3111–3119, 2013. 44

[66] MITRA, B. **Neural text embeddings for information retrieval**. URL: https://www.slideshare.net/BhaskarMitra3/neural-text-embeddings-for-information-retrieval-wsdm-2017. 15, 47

[67] MOREIRA, S. **Rede neural perceptron multicamadas**. URL: https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9. 15, 38

[68] Mota, C.; Santos, D., editors. **Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM**. Linguateca, 2008. ISBN: 978-989-20-1656-6. 34, 56, 101

[69] NADEAU, D.; SEKINE, S. **A survey of named entity recognition and classification**. *Lingvisticae Investigationes*, 30(1):3–26, 2007. 22

[70] NAIR, V.; HINTON, G. E. **Rectified linear units improve restricted boltzmann machines**. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, p. 807–814, USA, 2010. Omnipress. 39, 79

[71] NILC. **Repositório de word embeddings do nilc**. URL: http://www.nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc. 79, 80, 87, 88, 89, 90, 102, 114, 115

[72] NOTHMAN, J.; RINGLAND, N.; RADFORD, W.; MURPHY, T.; CURRAN, J. R. **Learning multilingual named entity recognition from wikipedia**. *Artificial Intelligence*, 194:151–175, 2013. 33

[73] NUNO CARDOSO. **Harem e miniharem: Uma análise comparativa**. URL: http://www.linguateca.pt/documentos/encontroHAREM_cardoso.pdf, 7 2006. 34, 56, 87, 93

[74] PENNINGTON, J.; SOCHER, R.; MANNING, C. **Glove: Global vectors for word representation**. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. 44, 46, 50, 53, 79, 80, 86, 89, 90, 95, 105

[75] PETERS, M.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. **Deep contextualized word representations**. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, p. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. 17, 18, 27, 43, 44, 49, 50, 53, 54, 77, 78, 79, 80, 81, 84, 86, 87, 88, 89, 90, 92, 113, 114

[76] PIRES, A. R. O. **Named entity extraction from Portuguese web text**. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2017. 87

[77] PIROVANI, J. P. C. **CRF+LG: uma abordagem híbrida para o reconhecimento de entidades nomeadas em português**. PhD thesis, Universidade Federal do Espírito Santo, 2019. 27, 33, 55, 56, 87, 97

[78] **Processo judicial eletrônico (pje)**. http://www.pje.jus.br/wiki/index.php/Página_principal. "Acessado em 02/09/2018". 25, 64

[79] QU, X.; YANG, J.; WU, B.; XIN, H. **A news event detection algorithm based on key elements recognition**. In: *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, p. 394–399, June 2016. 22

[80] QUARESMA, P.; GONÇALVES, T. **Using Linguistic Information and Machine Learning Techniques to Identify Entities from Juridical Documents**, p. 44–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 23, 56

[81] RADFORD, A.; NARASIMHAN, K.; SALIMANS, T.; SUTSKEVER, I. **Improving language understanding by generative pre-training**. Artigo disponibilizado em https://openai.com/blog/language-unsupervised/, 2018. 44, 49

[82] RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I. **Language models are unsupervised multitask learners**. Artigo disponibilizado em https://openai.com/blog/better-language-models/, 2019. 49

[83] REALE, M. **Lições Preliminares de Direito. 2.ª tiragem**. Saraiva, 2001. 59

[84] ROSENBLATT, F. **The perceptron: A probabilistic model for information storage and organization in the brain**. *Psychological Review*, p. 65–386, 1958. 37

[85] SANG, E. F.; VEENSTRA, J. **Representing text chunks**. In: *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, p. 173–179. Association for Computational Linguistics, 1999. 34, 36, 53, 88

[86] SANJEEVI, M. **Recurrent neural networks with math**. URL: https://medium.com/deep-math-machine-learning-ai/chapter-10-deepnlp-recurrent-neural-networks-with-math-c4a6846a50a2. 15, 42

[87] SANTOS, D.; CARDOSO, N. **Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área**. Linguateca, November 2007. ISBN: 978-989-20-0731-1. 34, 56, 87, 97, 98, 101

[88] ŠAVELKA, J.; ASHLEY, K. D. **Detecting agent mentions in U.S. court decisions**. In: *Frontiers in Artificial Intelligence and Applications*, volume 302, p. 39–48. IOS Press, 2017. 23, 57

[89] SAVELKA, J.; ASHLEY, K. D. **Segmenting U.S. Court Decisions into Functional and Issue Specific Parts**. In: *JURIX*, p. 111–120. IOS Press, 2018. 57

[90] STANFORD. **Glove: Global vectors for word representation**. URL: https://github.com/stanfordnlp/GloVe. 80, 89, 90

[91] TJONG KIM SANG, E. F.; DE MEULDER, F. **Introduction to the conll-2003 shared task: Language-independent named entity recognition**. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, p. 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. 33, 54, 56, 81, 87, 90, 93, 102

[92] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L. U.; POLOSUKHIN, I. **Attention is all you need**. In: Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, p. 5998–6008. Curran Associates, Inc., 2017. 54

[93] WENG, J.; AHUJA, N.; HUANG, T. **Learning recognition and segmentation of 3-d objects from 2-d images**. In: *1993 IEEE 4th International Conference on Computer Vision*, p. 121–127. Publ by IEEE, 1993. 40