

Compreendendo Análise de Sentimentos

Aplicada a Processamento de Linguagem Natural

Pedro Koziel Diniz



UFG

UNIVERSIDADE
FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

PEDRO KOZIEL DINIZ

COMPREENDENDO ANÁLISE DE SENTIMENTOS
Aplicada a Processamento de Linguagem Natural

Goiânia
2024



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): **PEDRO KOZIEL DINIZ**

Título do trabalho:

COMPREENDENDO ANÁLISE DE SENTIMENTOS

Aplicada a Processamento de Linguagem Natural

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [] NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro.

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



Documento assinado eletronicamente por **Ana Clara Goncalves Melo Da Silva Limeira, Assistente em Administração**, em 15/02/2024, às 19:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Pedro Koziel Diniz, Discendente**, em 15/02/2024, às 19:34, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Fernando Marques Federson, Professor do Magistério Superior**, em 12/09/2024, às 11:07, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site

[https://sei.ufg.br/sei/controlador_externo.php?](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **4383424** e o código CRC **7BC0E9B9**.

PEDRO KOZIEL DINIZ

COMPREENDENDO ANÁLISE DE SENTIMENTOS

Aplicada a Processamento de Linguagem Natural

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2024

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

DINIZ, PEDRO KOZIEL
COMPREENDENDO ANÁLISE DE SENTIMENTOS [manuscrito] :
Aplicada a Processamento de Linguagem Natural / PEDRO KOZIEL
DINIZ. - 2024.
73 f.

Orientador: Prof. Dr. FERNANDO MARQUES FEDERSON.
Trabalho de Conclusão de Curso (Graduação) - Universidade
Federal de Goiás, Instituto de Informática (INF), Inteligência
Artificial, Goiânia, 2024.

1. inteligência artificial. 2. análise de sentimentos. 3.
processamento de linguagem natural. I. FEDERSON, FERNANDO
MARQUES, orient. II. Título.

CDU 004

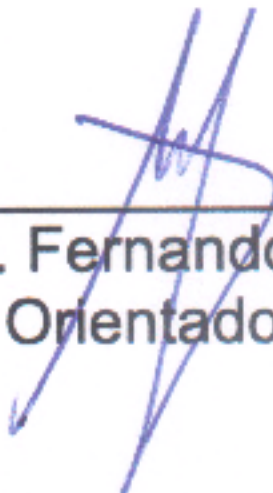
PEDRO KOZIEL DINIZ

COMPREENDENDO ANÁLISE DE SENTIMENTOS

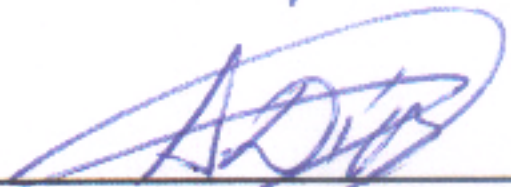
Aplicada a Processamento de Linguagem Natural

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

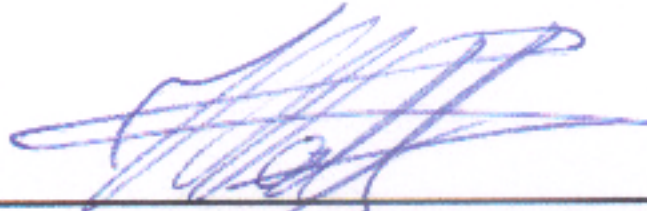
Data da Aprovação: 08 de fevereiro de 2024.



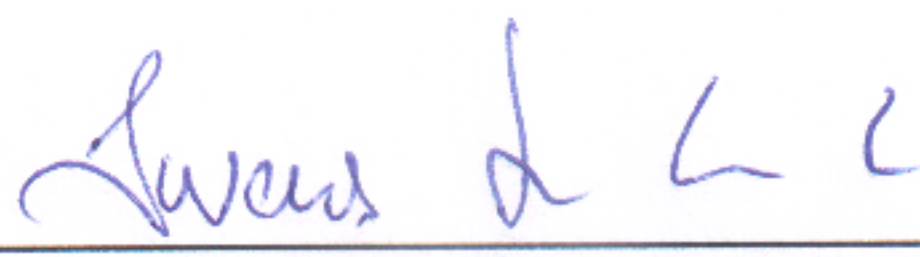
Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)



Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)



Prof. Dr. Vinícius Sebba Patto
Coordenador do BIA (INF-UFG)



Prof. Dr. Ivens Gervasio Sene Junior
(INF-UFG)

PEDRO KOZIEL DINIZ

COMPREENDENDO ANÁLISE DE SENTIMENTOS

Aplicada a Processamento de Linguagem Natural

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Análise de Sentimentos**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, análise de sentimentos, processamento de linguagem natural.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Sentiment Analysis**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, sentiment analysis, natural language processing.

Goiânia

2024

Minha Jornada

Pedro Koziel Diniz

Especialista em:
Análise de Sentimentos



Template baseado em [Slidesgo](#) e [Freepik](#)

MINHA JORNADA

Nome: Pedro Koziel Diniz

Especialidade: Análise de Sentimentos

Objetivo deste documento

Durante o processo da disciplina Residência em IA¹, foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Durante a primeira **Semana** da minha jornada, ainda não havia definido em específico qual área de conhecimento seria o objeto de estudo para a minha especialização. Porém, já havia decidido que seria dentro de Processamento de Linguagem Natural (NLP), visto que é a área da Inteligência Artificial que mais me interessa e possui diversos assuntos e tarefas, como por exemplo Análise de Sentimentos, *Chatbot*, Extração de Entidade Nomeadas (NER) e, mais recentemente, *Large Language Model* (LLM). Comecei analisando a *Conference on Computational Science and Computational Intelligence* (CSCI'23), uma conferência na qual são divulgados trabalhos científicos sobre diversas áreas envolvendo tecnologia e computação, entre elas, a Inteligência Artificial e o NLP. Assim, para a **Semana 1**, analisei os temas mencionados na CSCI'23 e selecionei alguns artigos que considerarei importantes para obter uma visão geral do assunto, como principais termos, usos e tarefas. Para a segunda **Semana**, ainda mantive meus estudos na

¹ Dez semanas, entre setembro de 2023 e janeiro de 2024.

área de NLP de modo geral, fazendo uma revisão dos principais conceitos que, independente de qual tarefa ou sub-área eu escolhesse para me especializar, esse conhecimento seria útil. Maiores detalhes sobre o que foi realizado nessas duas primeiras **Semanas** podem ser encontrados no **Apêndice 1**.

Já na terceira **Semana**, com base nos estudos realizados anteriormente, foi possível tomar a decisão de me especializar em Análise de Sentimentos, uma área de aplicação de NLP que, mesmo já sendo mais antiga quando comparada a tecnologias como os LLMs, ainda é muito explorada tanto no meio científico quanto no mercado, sendo utilizada em diversos contextos. Ao decorrer dessa **Semana 3**, meu foco foi estudar sobre esse campo de aplicação utilizando principalmente artigos científicos e *surveys*. Com base nesse estudo, anotei o que considerei serem os principais pontos que um especialista na área deveria conhecer. Esse estudo englobou uma pesquisa sobre quais as diferentes aplicações possíveis para Análise de Sentimentos em textos (como avaliações de produtos ou serviços, e uso em redes sociais), os principais métodos, como lidar com *emojicons* e as diversas abordagens (desde as mais clássicas até o estado da arte), e como avaliar o desempenho de aplicações dessa área (as principais métricas utilizadas, como são calculadas e quando é mais adequado o uso de cada uma). Gostaria de destacar o trabalho “*A survey on sentiment analysis methods, applications, and challenges*” (2022), de Mayur Wankhadem, Annavarapu Chandra Sekhara Rao e Chaitanya Kulkarni, no qual os autores discorrem de forma completa e detalhada sobre Análise de Sentimentos em textos e fornecem desde uma visão inicial, sobre o que é Análise de Sentimentos e seus usos, até as principais técnicas e os desafios. O detalhamento desse estudo pode ser encontrado no **Apêndice 2**.

Na **Semana 4**, já pensando em como realizar as aplicações e treinamento de modelos para Análise de Sentimentos, realizei um levantamento dos principais frameworks e bibliotecas utilizados na área, utilizando como base os artigos e aplicações vistas anteriormente, assim como a própria documentação desses

frameworks. O objetivo era que, assim que fosse dado início à parte de experimentações, dependendo da abordagem que fosse seguida, já seria possível começar os experimentos com o framework adequado. Enquanto na quarta **Semana** realizei um levantamento dos frameworks, na **Semana 5** e, em parte da sexta, realizei uma busca por diferentes abordagens utilizadas para Análise de Sentimentos, visando encontrar os métodos e representações mais utilizados. Além disso, foi uma forma de validação do levantamento de frameworks realizado anteriormente, visto que nessa busca pelas diferentes abordagens também analisei as ferramentas que foram utilizadas em cada uma e observei se estavam dentro dos frameworks anotados. Resumindo, no decorrer das **Semanas 4 e 5**, realizei um estudo sobre as principais ferramentas e abordagens existentes ao se trabalhar com Análise de Sentimentos. Mais informações sobre este trabalho e seus resultados estão no **Apêndice 3**.

Na **Semana 6**, finalizei a busca por abordagens, e iniciei a etapa de testes, com o intuito de explorar e experimentar as principais abordagens para Análise de Sentimentos em NLP. Primeiramente, testei uma abordagem mais simples, baseada em léxico. Nessa aplicação é utilizado um dicionário no qual é atribuída uma polaridade a cada palavra para indicar se o sentimento é positivo, negativo ou neutro, e então, a partir de um conjunto de regras, é calculada a polaridade sentimental do texto em questão. É uma técnica simples e, quando comparada às próximas abordagens, pode gerar uma assertividade menor, mas apresenta baixo custo computacional e tempo de inferência mais rápido. Todos os testes foram realizados no mesmo conjunto de dados B2W-Reviews01 e as métricas analisadas foram acurácia, *F1-Score weighted* e tempo de inferência. Na sétima **Semana** explorei técnicas mais complexas do que a abordagem de léxico, porém ainda consideradas clássicas. Os primeiros experimentos foram com a representação TF-IDF e os classificadores *Random Forest* e *Logistic Regression*, e em seguida utilizei a representação Word2Vec com os mesmos classificadores, e também com

uma rede neural utilizando a LSTM (*Long Short-Term Memory*). As métricas analisadas foram as mesmas daquelas do primeiro teste, porém também considerando agora, além do tempo de inferência, o tempo de treinamento dos modelos. O intuito ao utilizar tais abordagens era analisar o desempenho de cada uma no mesmo conjunto de dados (B2W-Reviews01), considerando tanto assertividade quanto tempo de treinamento e inferência, de modo a conhecer melhor os pontos fortes e fracos de cada abordagem. Os resultados obtidos e algumas conclusões estão disponíveis no **Apêndice 4**.

Durante as **Semanas 8 e 9**, dando sequência à experimentação de diferentes abordagens, dediquei meus estudos e testes a modelos baseados em arquitetura Transformers. Comparado às abordagens anteriores, esses modelos possuem uma melhor compreensão de contexto e em diversos *benchmarks* alcançam os melhores resultados, porém para isso é necessário treiná-los com altas quantidades de dados. Como esse treinamento é demorado e requer alto poder computacional, a abordagem utilizada foi realizar o *fine-tune* de um modelo já pré-treinado, reduzindo significativamente o tempo e recursos computacionais necessários. Também existem modelos já treinados em textos voltados para Análise de Sentimentos, inclusive com tokenização de emojis, porém em português esses modelos ainda são escassos. Dessa forma, durante a **Semana 8**, o foco foi realizar o *fine-tune* de um modelo BERT para Análise de Sentimentos, e, na **Semana 9**, foram pesquisadas algumas formas de melhorar os resultados desses modelos. A principal forma encontrada foi a otimização de hiperparâmetros, utilizando o framework Optuna. Como o intuito era obter conhecimento sobre as diferentes abordagens para Análise de Sentimentos, e o processo de otimização de hiperparâmetros em modelos Transformers é demorado e computacionalmente custoso, foi feito um *notebook* de exemplo para realizar essa otimização, porém a sua utilização e desenvolvimento ficaram limitados. Maiores detalhes sobre o *fine-tune* realizado e a otimização de hiperparâmetros podem ser encontrados no **Apêndice 5**.

Já na **Semana 10**, decidi buscar algumas abordagens mais recentes e encontrei alguns artigos e trabalhos fazendo uso de *Large Language Models* (LLMs). O principal trabalho encontrado foi “*SentimentGPT: Exploiting GPT for Advanced Sentiment Analysis and its Departure from Current Machine Learning*” (2023), de Kiana Kheiri e Hamid Karimi. Neste trabalho, os autores realizam alguns testes utilizando o modelo GPT. Tendo esse trabalho como inspiração, realizei uma busca por LLMs *open source* em português ou multilingual. Os testes iniciais nestes LLMs não apresentaram bons resultados, então optei por utilizar um modelo mais testado, o GPT 3.5 da OpenAI, mesmo modelo utilizado no artigo mencionado. O objetivo era analisar se esse modelo, por ser treinado em grandes bases de dados, é capaz de lidar com tarefas de Análise de Sentimentos de forma *zero-shot*, ou seja, sem a necessidade de um *fine-tune*. Os resultados do experimento podem ser encontrados no **Apêndice 6**.

Ao final desse processo acredito que fui capaz de me especializar e compreender melhor a área de Análise de Sentimentos, sua importância, e principais abordagens existentes. Porém, não considero que, com o término da Disciplina Residência em IA, esse processo tenha sido encerrado, tendo em vista que constantemente são feitas novas descobertas de abordagens, métodos e aplicações na área. Dessa forma, pretendo dar seguimento a essa especialização e aprofundar nos estudos sobre LLMs, visto que o uso desses modelos vêm crescendo cada vez mais. Durante a disciplina, utilizei esses modelos apenas para realizar inferência, porém o uso de LLMs inclui outras abordagens que podem melhorar consideravelmente os resultados, como a engenharia de prompt e a realização de *fine-tune* dos modelos. Com novos modelos e técnicas surgindo frequentemente, conhecer melhor os LLMs se torna, para mim, um alvo de estudo interessante.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 19 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Héber Júnior, Pedro Koziel

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Definição do tema: Classificação de textos de redes sociais para detecção de discurso de ódio e explicabilidade desse modelo.

Para essa entrega foi realizado o estudo de artigos da área de NLP, em específico classificação de textos, para possibilitar um melhor entendimento sobre o que é o estado da arte nessa área e, assim, também buscar matérias abordando essas técnicas e representações. Foi identificado o uso de arquiteturas Transformer como estado da arte em grande parte dos materiais, porém também foram analisados artigos utilizando técnicas clássicas. Os artigos analisados podem ser encontrados em: [Gate 19/10/2023](#)

Além do uso dos Transformers, alguns desses artigos também apresentam outras etapas do pipeline de classificação que são importantes e serão exploradas ao decorrer deste TCC, como: análise dos dados, pré-processamento dos textos e avaliação dos modelos. Dessa forma, serão exploradas as etapas que levam ao treinamento de um modelo de classificação de texto e como explorar sua explicabilidade, fazendo uso tanto de estudos teóricos quanto aplicações práticas.

Como solicitado, em relação aos termos que aparecem no Conference on Computational Science and Computational Intelligence (CSCI'23), esse trabalho se enquadra nos temas: Natural language processing, Neural networks and applications, Supervised Learning, Aspects of natural language processing, Knowledge discovery, Applications.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Buscar materiais sobre classificação de texto em situações similares àquela que será analisada nesse projeto (em redes sociais), e também materiais que abordem explicabilidade de modelos.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

LUANA GUEDES BARROS MARTINS: Go! ▾

[Documento em anexo ao Termo de Aceite de Entrega de 19/10/2023]

Principais materiais revisados

1- A Survey on Text Classification: From Traditional to Deep Learning

(<https://arxiv.org/pdf/2008.00364.pdf>)

Neste trabalho é feito um estudo sobre classificação de textos, abordando desde as abordagens tradicionais, como TF-IDF e Bag of Words (BOW) até as consideradas Deep Learning, como Transformers. Ao fornecer esse histórico os autores evidenciam a importância de também se estudar as técnicas e abordagens clássicas ao invés de começar os estudos e testes apenas com os modelos considerados “estado da arte”, visto que cada abordagem diferente possui seus pontos positivos e negativos. Além disso, também são apresentadas algumas formas de pré-processamento dos textos, datasets conhecidos e métricas para avaliar seus desempenhos, assim como algumas dificuldades atualmente encontradas na área, sendo um trabalho que fornece uma ampla visão de como lidar com classificação de textos.

2- How to Fine-Tune BERT for Text Classification?

(<https://arxiv.org/pdf/1905.05583v3.pdf>)

Neste trabalho o foco é em explicar melhor como realizar o fine tune de um modelo BERT. Isso se deve ao fato que os modelos Transformers são treinados em grandes quantidades de dados, porém isso faz com que muitas vezes sejam modelos generalistas, não performando tão bem em cenários específicos. Dessa forma, uma possível abordagem é realizar o fine tune desses modelos em dados específicos, para que assim seja necessária uma quantidade menor de dados do que caso fosse treinar um modelo do zero e sejam obtidos melhores resultados.

3- An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation (<https://arxiv.org/ftp/arxiv/papers/1806/1806.06407.pdf>)

Neste trabalho são mostradas três abordagens para uma classificação de texto em análise de sentimentos, porém utilizando apenas abordagens clássicas com as representações Bag of Words e TF-IDF. É um artigo interessante por mostrar a viabilidade dessas representações ao se trabalhar com classificação de texto, nesse caso em específico análise de sentimentos.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 26 de out. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel, Héber Júnior e Pedro Augusto

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa semana e após os encontros da semana passada, compreendemos melhor o que se espera de nós dessa residência. Assim, realizamos a busca por surveys de NLP, tanto no aspecto geral quanto específico da parte de classificação, a fim de ter um histórico de técnicas e métodos da área, e também buscar quais dessas técnicas e métodos são os que possuem melhores resultados atualmente. Os surveys podem ser encontrado em:

- A Survey on Text Classification: From Traditional to Deep Learning (<https://arxiv.org/pdf/2008.00364.pdf>)
- A Survey on Text Classification Algorithms: From Text to Predictions (<https://www.mdpi.com/2078-2489/13/2/83>)
- Efficient Methods for Natural Language Processing: A Survey (<https://arxiv.org/pdf/2209.00099.pdf>)

Também foi montado um repositório compartilhado de artigos entre todos da turma que estão trabalhando com NLP e/ou LLM: https://drive.google.com/drive/folders/11igIGGITPdB_qAXi71--XOu71Rxx3cWH

Junto com esse estudo, também fizemos um compilado de termos que julgamos ser mais importante para essa etapa inicial da residência, que constitui os fundamentos de NLP. Para isso foi utilizado de base tanto os surveys acima quanto os artigos que foram mencionados na entrega da semana passada. Esse compilado pode ser acessado em: [gate 26/10/2023](#)

Além disso, eu defini qual parte de NLP em específico irei me especializar, e optei pela parte de análise de sentimentos.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega, pretendo:

- Aprofundar no estudo sobre análise de sentimentos
- Busca de surveys e artigos específicos sobre análise de sentimentos

- Busca por datasets e benchmarks em análise de sentimentos

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

LUANA GUEDES BARROS MARTINS: Go! ▾

[Documento em anexo ao Termo de Aceite de Entrega de 26/10/2023]

Principais tópicos e termos gerais em NLP

Usando os surveys e artigos que foram citados como referência nos termos de entrega da semana passada e essa, separamos os principais tópicos sobre representações de palavras e modelos de NLP.

1. Representação de Palavras

A representação de palavras é fundamental para a classificação de textos, pois transforma o texto em um formato que os modelos de aprendizado de máquina podem entender e, conseqüentemente, possibilitar seu processamento e realização de diversas tasks, como classificação e sumarização. Existem várias abordagens, incluindo:

a. Bag of Words (BoW): Nessa abordagem, um texto é representado como um conjunto de palavras, ignorando a ordem das palavras. Cada palavra é codificada como um vetor de recursos. As frequências das palavras são contadas e usadas como valores nos vetores.

b. TF-IDF (Term Frequency-Inverse Document Frequency): O TF-IDF atribui valores a palavras com base em sua frequência em um documento específico e sua importância geral em todo o corpus de documentos, o que ajuda a destacar palavras importantes.

c. Word Embeddings Estáticos (word2vec): Além das abordagens tradicionais, é importante mencionar o uso de word embeddings estáticos, como word2vec. Esses embeddings representam palavras como vetores em um espaço de alta dimensão, capturando relações semânticas e contextualmente relevantes entre palavras. O word2vec é treinado em grandes conjuntos de texto e é uma maneira eficaz de capturar o significado das palavras para uso em tarefas de processamento de linguagem natural.

2. Classificadores Clássicos:

Os classificadores clássicos podem ser usados com representações de palavras como BoW ou TF-IDF. Alguns exemplos de classificadores clássicos incluem:

a. Naive Bayes: É um classificador probabilístico que assume independência entre as palavras.

b. Regressão Logística: Uma técnica que modela a probabilidade de uma instância pertencer a uma classe em função das variáveis explicativas (representação de palavras).

c. SVM (Support Vector Machine): mapeia os vetores de palavras para um espaço de alta dimensão e encontra hiperplanos de separação entre classes.

3. Métodos Baseados em Deep Learning:

Os métodos baseados em deep learning têm se destacado na classificação de textos, especialmente quando combinados com representações de palavras mais avançadas. Alguns métodos incluem:

a. Redes Neurais Artificiais (MLP): Redes neurais de feedforward, como perceptrons multicamadas (MLP), podem ser usadas para classificação de textos. Eles podem ser alimentados com representações vetoriais de palavras.

b. Redes Neurais Recorrentes (RNN): As RNNs são capazes de lidar com sequências de palavras e podem capturar dependências temporais. No entanto, elas têm dificuldade em lidar com sequências muito longas e sofrem de desvanecimento de gradientes (vanishing gradients).

c. Redes Neurais com Mecanismo de Atenção (Attention): As redes com atenção, como o mecanismo de atenção de seq2seq, são capazes de dar mais peso a certas partes do texto durante a classificação, o que é útil para entender o contexto.

d. Transformers: Os modelos baseados em Transformers revolucionaram o campo do processamento de linguagem natural (NLP) e a classificação de textos. Eles introduziram uma arquitetura de rede neural altamente paralelizável que se destaca em uma variedade de tarefas NLP, incluindo classificação de textos.

4. Arquitetura dos Transformers:

A principal inovação dos Transformers é a arquitetura de atenção, que permite que o modelo considere todas as palavras do contexto ao mesmo tempo. Isso é fundamental para a compreensão do contexto e a captura de relações de longo alcance. Cada camada do Transformer contém duas partes principais:

a. Mecanismo de Auto-Atenção (Self-Attention): É a espinha dorsal do Transformer. Ele permite que o modelo avalie a importância de todas as palavras na sequência em relação a uma palavra de entrada específica. Isso é feito por meio de pesos de atenção que indicam a importância relativa de cada palavra para a palavra de referência.

b. Redes de Feedforward (Feedforward Networks): Após a auto-atenção, uma camada de redes neurais de feedforward é aplicada para combinar informações e gerar saídas finais.

Os modelos Transformers são altamente adaptáveis para tarefas de classificação de textos. Para usar um modelo Transformer para classificação, a camada de saída pode ser personalizada para se ajustar ao número de classes do problema de classificação. Geralmente, é adicionada uma camada de saída softmax no topo da arquitetura para calcular as probabilidades de pertencer a cada classe.

Uma das vantagens dos modelos Transformers é a capacidade de pré-treinamento em grandes quantidades de dados e, em seguida, ajustar o modelo para tarefas específicas por meio do fine-tuning. Para classificação de textos, é possível pré-treinar um modelo Transformer em um grande corpus de texto e, em seguida, ajustá-lo em dados de treinamento de classificação específicos.

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 9 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Definição do escopo e aprofundamento teórico no mesmo: Análise de sentimentos em textos (NLP).
Estudo de surveys e outros materiais sobre análise de sentimentos em textos, com anotação dos principais pontos observados, o que pode ser encontrado em: [Gate 9/11/2023](#)

O estudo englobou diversos tópicos, como por exemplo os motivos para criação de modelos de análise de sentimentos e suas aplicações, como os dados para esse tipo de task são coletados e utilizados, e quais as principais tecnologias/metodologias utilizadas, seja em abordagens clássicas seja em métodos atuais, e as métricas utilizadas para avaliar um modelo.

Também foi visto a possibilidade de aplicação da análise de sentimentos em áreas além de NLP, como visão computacional e áudio, porém optei por focar na parte de NLP.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Levantamento das principais ferramentas utilizadas nas etapas de desenvolvimento de um modelo de análise de sentimentos, e também quais fundamentos vistos até o momento são empregados em conjunto a essas ferramentas.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

LUANA GUEDES BARROS MARTINS: [Go!](#)

[Documento em anexo ao Termo de Aceite de Entrega de 09/11/2023]

[Documento em anexo ao Termo de Aceite de Entrega de 09/11/2023]

Principais tópicos e termos para análise de sentimentos

Ao longo dessas duas semanas, assim como mencionado no gate anterior, comecei a estudar a parte de análise de sentimentos. Dei foco maior a trabalhos mais recentes, em especial surveys, visto que meu objetivo é compreender melhor a área como um todo, desde os motivos por trás de seu estudo até quais os métodos mais eficazes na atualidade. Também deixarei anotado nesse documento os pontos que considerarei mais relevantes.

Os principais artigos e sites que me baseei foram:

- A survey on sentiment analysis methods, applications, and challenges (<https://link.springer.com/article/10.1007/s10462-022-10144-1>)
- A comprehensive survey on sentiment analysis: Approaches, challenges and trends (<https://sci-hub.se/downloads/2021-06-20/4c/birjali2021.pdf>)
- Sentiment Analysis Based on Deep Learning: A Comparative Study (<https://www.mdpi.com/2079-9292/9/3/483>)
- <https://www.youtube.com/watch?v=n4L5hHFcGVk>
- <https://www.youtube.com/watch?v=nBZUcBrq6oA>
- <https://www.cio.com/article/189218/what-is-sentiment-analysis-using-nlp-and-ml-to-extract-meaning.html>

Análise de sentimentos: subárea da Inteligência Artificial que avalia e classifica a polaridade emocional em determinado tipo de dado. Pode ser binária (emoção positiva e negativa), também pode incluir o sentimento neutro, ou então incluir uma determinada quantidade múltipla de sentimentos. Pode ser explorada em diversas áreas da IA e tipos de dados, como textos, imagens, vídeos e áudios, porém irei focar na parte de textos. É utilizada para compreender a emoção ou opinião expressa pelo autor do texto, tendo aplicações em diversos ramos empresariais, como análise de textos de redes sociais, reviews de produtos e suporte ao cliente.

Pode ser aplicada em diferentes níveis de granularidade, ou seja, tanto em documentos inteiros, quanto em sentenças ou até em cada frase, a depender do objetivo da aplicação. No caso de documentos inteiros há algumas dificuldades, visto que a própria anotação do dado pode ser mais complexa, e também pelo fato

que os modelos estado da arte baseados em transformers possuem um limite máximo de tokens.

Os dados podem ser coletados de diversas fontes, como redes sociais, sites de reviews ou vindos de uma empresa específica. O mais comum é a abordagem supervisionada, portanto há a necessidade de coleta dos dados e posterior anotação, sendo que em alguns casos essa anotação deve ser feita por um profissional. Caso a anotação não possua uma boa qualidade, é praticamente impossível criar modelos que funcionem como o esperado.

Já no processamento desses textos, é necessário realizar a extração de features, ou seja, os textos devem ser passados para uma representação numérica, de modo a tentar capturar as características mais relevantes do texto. Isso pode ser feito de algumas formas:

1. **Word Embeddings:** vetores numéricos que representam palavras em um espaço vetorial contínuo, capturando significado semântico e contextual.
2. **TF-IDF (Term Frequency-Inverse Document Frequency):** O TF-IDF é uma técnica que atribui valores numéricos a palavras com base em sua importância em documentos individuais ou em um conjunto de documentos. Isso cria representações numéricas para as palavras em relação a documentos específicos.
3. **Bag of Words:** Essa técnica cria representações baseadas na frequência das palavras em um documento ou corpus, ignorando a ordem das palavras. Cada palavra é representada por um valor numérico correspondente à sua frequência.
4. **N-gramas:** A extração de n-gramas envolve a criação de representações com base em sequências de palavras (bi-gramas, tri-gramas, etc.), permitindo capturar informações de contexto local.
5. **Modelos de linguagem pré-treinados:** Modelos como o BERT, GPT-3 e outros podem ser usados para extrair representações complexas de texto por meio de embeddings contextualizados. A extração de características desempenha um papel fundamental em tarefas de NLP, pois permite que algoritmos de aprendizado de máquina processem e entendam o conteúdo textual, tornando possível a realização de tarefas como classificação, análise de sentimentos, respostas a perguntas, tradução automática e muito mais.

Um ponto importante a se considerar é o uso de emoticons para análise de sentimentos, principalmente em redes sociais, que podem ser abordados de diferentes formas, algumas mais simples, como apenas retirá-los do texto, transcrevê-los para texto, ou usar representações com capacidade de detectar e atribuir alguma representação a esses emoticons.

Uma abordagem mais clássica é a Lexicon Based Approach, que faz uso de dicionários ou léxicos predefinidos para analisar e interpretar textos. Nesse método, um léxico é uma lista de palavras ou termos com informações associadas a eles, como a polaridade de sentimento. Ou seja, um léxico de sentimentos é usado para determinar a polaridade (positiva, negativa ou neutra) de um texto, de modo que as palavras no texto são comparadas com as palavras no léxico e a polaridade do texto é calculada com base na soma ou média das polaridades das palavras. É uma abordagem mais simples, e caso já se tenha o léxico não é necessário nenhum treinamento, porém possui desvantagens como considerar apenas as palavras e não o contexto da frase (dependendo do contexto certas palavras podem ser positivas ou negativas), e a questão do léxico ser algo limitado e não tão flexível quanto outras abordagens, como os modelos baseados em transformers.

Já nas abordagens utilizando aprendizado de máquina, é possível utilizar representações como Bag of Words ou TF-IDF, e classificadores clássicos, como Naive Bayes, SVM ou classificadores baseados em árvores. Embora sejam mais simples do que modelos que utilizam redes neurais, podem obter resultados interessantes e o tempo de treinamento é consideravelmente menor, além da maior viabilidade do treinamento em CPU ao invés de GPU.

A metodologia que, via de regra, alcança os melhores resultados é o transfer learning. Modelos baseados em transformers, como BERT, são treinados em quantidade massivas de dados e requerem tempo e capacidade computacional elevada (pode levar semanas ou até meses de treinamento em GPUs de alta qualidade). A grande maioria dos indivíduos e empresas não possuem meios para treinar um modelo desse tamanho, por isso é feito uso do transfer learning.

No transfer learning é utilizado um modelo pré-treinado (por exemplo BERT) e, a partir desse modelo, é feito o fine tuning em outro dataset menor. Dessa forma, como o modelo original já aprendeu diversas coisas no pré-treinamento, ao realizar o fine tuning é possível alcançar bons resultados em bem menos tempo e com menor poder computacional.

Para avaliar o desempenho de um modelo, existem diversas métricas que podem ser utilizadas, com as principais sendo:

1. True Positive (TP): O número de avaliações positivas que foram corretamente classificadas como positivas.
2. True Negative (TN): O número de avaliações negativas corretamente classificadas como negativas.
3. False Positive (FP): O número de avaliações positivas incorretamente classificadas.
4. False Negative (FN): O número de avaliações negativas incorretamente classificadas.
5. Acurácia (Accuracy): A acurácia mede a proporção de previsões corretas em relação ao total de previsões. É uma métrica simples e amplamente utilizada, mas pode não ser adequada em casos de desequilíbrio de classe, onde uma classe é muito mais frequente do que a outra.
6. Precisão (Precision): A precisão mede a proporção de verdadeiros positivos (TP) em relação ao total de previsões positivas. É útil quando é importante minimizar falsos positivos.
7. Revocação (Recall): A revocação mede a proporção de verdadeiros positivos em relação ao total de exemplos da classe de interesse no conjunto de dados. É útil quando é importante minimizar falsos negativos.
8. F1-Score: O F1-Score é a média harmônica da precisão e da revocação e é útil quando se deseja equilibrar o compromisso entre essas duas métricas, sendo melhor adaptado do que a acurácia para casos de dados desbalanceados.
9. Matriz de Confusão (Confusion Matrix): A matriz de confusão fornece uma visão detalhada do desempenho do modelo, mostrando o número de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. Exemplo de matriz de confusão:

ACTUAL	Positive	True Positive	False Negative
	Negative	False Positive	True Negative
		Positive	Negative

PREDICTED

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 16 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel, Héber Júnior e Pedro Augusto

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Ao longo desta semana, conforme mencionado no planejamento anterior, desenvolvemos um levantamento com as principais ferramentas para a realização da tarefa de classificação de textos (e, no meu caso em específico, para análise de sentimentos). O objetivo desse levantamento é que, como especialistas, ao nos depararmos com algum problema nesta área, já sejamos capazes de, antes mesmo de iniciar a programação, identificar quais frameworks são mais adequados ao problema em questão. Além disso, ao fazer esse levantamento também pudemos explorar abordagens e técnicas de NLP, visto que a documentação das bibliotecas trazem esse conteúdo. Este levantamento pode ser encontrado em:

☰ Levantamento de Frameworks

A fim de fechar mais o escopo da aplicação, também decidi explorar a parte de análise de sentimentos em textos de avaliações de produtos e/ou serviços, de modo a ter como objetivo treinar um modelo que consiga classificar as avaliações de usuários como positivas ou negativas, ou até em um escopo maior de classes (por exemplo muito negativo, negativo, neutro, positivo e muito positivo).

Análise de sentimentos é uma forma de classificação de texto, e boa parte das ferramentas utilizadas são iguais, por isso optamos por realizar essa semana em grupo.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega, pretendo:

- Procurar datasets de análise de sentimentos sobre reviews de produtos e/ou serviços (de preferência em português);
- Definindo os dados a serem utilizados, definir o pipeline a ser seguido na implementação prática e os frameworks a serem utilizados (utilizando de referência o levantamento realizado essa semana).

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go! ▾](#)

LUANA GUEDES BARROS MARTINS: [Go! ▾](#)

[Documento em anexo ao Termo de Aceite de Entrega de 16/11/2023]

Principais frameworks para análise de sentimentos

Esse documento contém um levantamento dos principais frameworks para a tarefa de classificação de textos. Foram consideradas as etapas listadas abaixo de desenvolvimento de um modelo de classificação de textos. Dependendo da implementação pode ser que algumas etapas não sejam adotadas ou que outras não listadas sejam, porém de forma geral as etapas a serem consideradas são as descritas abaixo:

- Importação dos dados
- Pré-processamento / feature engineering
- Feature extraction
- Treino do modelo
- Obtenção de métricas
- Otimização de hiperparâmetros

Vale lembrar que todos os frameworks comentados abaixo são da linguagem Python.

Pandas: Oferece estruturas de dados poderosas, como DataFrames, que simplificam a manipulação e transformação de conjuntos de dados. Especificamente, no contexto de processamento de linguagem natural (NLP), o Pandas é frequentemente utilizado para a importação, pré-processamento e manipulação de dados textuais. Suas funcionalidades incluem operações eficientes para lidar com texto, como remoção de duplicatas, seleção de features e engenharia de características (feature engineering). Apesar de ser uma ferramenta de alto desempenho em tarefas de manipulação de dados, é importante notar que para volumes muito grandes de dados, o Pandas pode exigir uma quantidade considerável de memória. <https://pandas.pydata.org/>

PySpark: Desenvolvido como uma interface Python para o Apache Spark, o PySpark é uma biblioteca que oferece poderosas capacidades de processamento distribuído. Projetado para lidar com grandes volumes de dados, é especialmente valioso em ambientes de big data, onde o Spark é frequentemente utilizado. No contexto do processamento de linguagem natural (NLP), o PySpark é aplicado para tarefas de importação de dados, pré-processamento e análise em larga escala. Sua arquitetura distribuída permite a execução eficiente de operações em clusters,

tornando-o adequado para lidar com conjuntos de dados extensos e complexos em projetos de NLP em grande escala. Possui funcionalidade similares ao Pandas, porém ao contrário dele, que é mais eficiente para operações em conjuntos de dados que cabem na memória, o PySpark é projetado para processamento distribuído, tornando-o uma escolha poderosa em ambientes onde o dimensionamento é essencial.

<https://spark.apache.org/docs/latest/api/python/index.html>

NLTK (Natural Language Toolkit): Capaz de auxiliar em diversas tarefas de pré-processamento textual, oferece uma ampla variedade de módulos para tokenização, lematização, análise sintática, entre outras operações. Sua versatilidade é evidente na capacidade de lidar com uma variedade de tarefas, desde simples manipulações de texto até análises linguísticas mais complexas. Além disso, o NLTK inclui recursos como corpora e modelos pré-treinados, facilitando a implementação de soluções em projetos de NLP. <https://www.nltk.org/>

VADER: Valence Aware Dictionary and sEntiment Reasoner (VADER) é uma biblioteca especificamente projetada para análise de sentimento em mídias sociais e inclui uma abordagem baseada em léxico (lexicon based) que é ajustada para a linguagem das mídias sociais. Inclui um léxico de sentimento pré-construído com medidas de intensidade para sentimento positivo e negativo, e incorpora regras para lidar com intensificadores de sentimento, emojis e outros recursos específicos que, via de regra, são utilizados em mídias sociais. <https://github.com/cjhutto/vaderSentiment>. Possui uma adaptação para português: <https://github.com/rafjaa/LeIA>

SpaCy: Oferece uma ampla gama de funcionalidades, desde tokenização e lematização até análise sintática e reconhecimento de entidades nomeadas. No âmbito do processamento de linguagem natural, o SpaCy é frequentemente utilizado para realizar diversas etapas do pré-processamento textual. Ele se destaca por sua capacidade de processar grandes volumes de texto de forma rápida e eficiente. Além disso, o SpaCy possui modelos pré-treinados para várias línguas, o que facilita a incorporação de recursos linguísticos em projetos de NLP. Uma característica notável do SpaCy é a sua extensibilidade, permitindo que os usuários adicionem componentes personalizados para atender a requisitos específicos. Com documentação abrangente e uma comunidade ativa, o SpaCy é uma escolha

popular em projetos que envolvem análise de texto e processamento de linguagem natural. <https://spacy.io/>

TextBlob: É uma biblioteca em Python que oferece uma gama de ferramentas para tarefas de pré-processamento e análise de texto. Embora possua menos recursos em comparação com outros frameworks, é mais recomendado para projetos que buscam soluções mais simples e eficazes. Facilita tarefas comuns de NLP, como tokenização, lematização e análise de sentimentos, possuindo uma interface mais simples que permite a extração rápida de informações relevantes de texto, enquanto a funcionalidade de análise de sentimentos incorporada oferece uma solução direta para avaliar a polaridade e subjetividade de expressões. Embora não seja tão extensivo quanto alguns frameworks especializados em NLP, o TextBlob é uma escolha prática para projetos nos quais a facilidade de implementação é valorizada. <https://textblob.readthedocs.io/en/dev/>

Scikit-learn: É uma ferramenta essencial para tarefas de aprendizado de máquina em Python, destacando-se por sua robustez e variedade de algoritmos para classificação, regressão, clustering e, especialmente, e pelo suporte sólido às tarefas de Processamento de Linguagem Natural (NLP). Ao integrar funcionalidades avançadas, o Scikit-learn oferece não apenas algoritmos de aprendizado de máquina, mas também ferramentas essenciais para pré-processamento de dados textuais. O Scikit-learn se destaca por sua capacidade de realizar extração de características, incluindo técnicas como Bag-of-Words (BoW) e Term Frequency-Inverse Document Frequency (TF-IDF). Além disso, o framework facilita a implementação de pipelines completos para tarefas complexas de NLP, desde a tokenização até a criação de modelos preditivos, e possui um design modular e documentação abrangente. Também possui funcionalidades para avaliar o desempenho de modelos, como F1 e acurácia. <https://scikit-learn.org/stable/>

FastText: É uma biblioteca que se destaca pela capacidade de treinar modelos de embeddings de palavras e realizar classificação de texto de forma rápida e eficaz. Desenvolvido pelo Facebook, este framework aprende representações vetoriais de palavras e também leva em consideração a estrutura de subpalavra (subword) das palavras. É projetado para lidar com grandes conjuntos de dados textuais e oferece uma interface fácil de usar para tarefas comuns de NLP, como classificação de texto e análise de sentimentos. Além disso, sua funcionalidade de embeddings de palavras pré-treinados permite incorporar

facilmente conhecimento linguístico em seus modelos. Dessa forma, o FastText se destaca como uma escolha sólida para projetos de NLP que demandam velocidade e desempenho. <https://fasttext.cc/>

Gensim: É especialmente destacado por sua eficiência na modelagem de tópicos e na criação de representações semânticas de documentos. Projetado para lidar com grandes conjuntos de dados textuais, o Gensim oferece uma implementação eficaz de algoritmos de modelagem de tópicos, como o Latent Dirichlet Allocation (LDA), permitindo a descoberta de padrões latentes em grandes coleções de documentos. Uma característica distintiva do Gensim é sua capacidade de treinar modelos de embeddings de palavras, como o Word2Vec e Glove. Além disso, o Gensim oferece funcionalidades para similaridade de documentos, indexação eficiente e manipulação de grandes quantidades de texto de maneira escalável. Ou seja, é um framework recomendado para fazer o uso de embeddings estáticos, como Word2Vec. <https://radimrehurek.com/gensim/>

Hugging Face: Reconhecido como um hub central para inovações em processamento de linguagem natural (NLP), também é uma plataforma que oferece uma ampla gama de modelos pré-treinados, datasets e ferramentas para tarefas diversas em NLP. Destaca-se por sua comunidade ativa e pela capacidade de compartilhar, descobrir e implementar modelos de última geração. Uma de suas características marcantes é a biblioteca Transformers, que facilita o acesso e o uso de uma variedade de modelos pré-treinados de última geração, como BERT, GPT-3 e muitos outros. Essa abordagem simplifica significativamente a implementação de modelos de NLP avançados, estimulando a pesquisa e o desenvolvimento em larga escala. Assim, é a principal ferramenta para fazer uso de modelos baseados em Transformers, além de possuir funções para treino, fine-tuning e avaliação dos modelos. <https://huggingface.co/>

TensorFlow: Como uma das principais bibliotecas para aprendizado de máquina e processamento de linguagem natural (NLP), o TensorFlow se destaca por sua versatilidade e eficiência computacional. Desenvolvido pela Google, o TensorFlow oferece uma infraestrutura robusta para a criação e treinamento de modelos de aprendizado profundo, sendo amplamente adotado em projetos de NLP. Para tarefas específicas de NLP, o TensorFlow disponibiliza módulos e ferramentas, incluindo o TensorFlow Text, que oferece funcionalidades avançadas para processamento de texto, como tokenização, embeddings de palavras e camadas

especializadas para tarefas como classificação de texto e tradução. A integração natural do TensorFlow com unidades de processamento gráfico (GPU) e suas capacidades de computação distribuída fazem dele uma escolha poderosa para treinamento de modelos em larga escala. Além disso, a comunidade ativa e a extensa documentação do TensorFlow contribuem para sua popularidade entre pesquisadores e desenvolvedores de NLP. <https://www.tensorflow.org/?hl=pt-br>

PyTorch: Similar ao TensorFlow, destaca-se como um dos principais frameworks para aprendizado de máquina e, em particular, para processamento de linguagem natural (NLP), o PyTorch é reconhecido por sua flexibilidade e interface intuitiva. Desenvolvido pelo Facebook, o PyTorch oferece uma abordagem dinâmica e orientada para o usuário, o que facilita a construção e experimentação rápida com modelos complexos. Para tarefas específicas de NLP, o PyTorch conta com o pacote torchtext, que simplifica o pré-processamento de dados textuais e fornece utilitários para carregamento eficiente de conjuntos de dados. Além disso, o PyTorch é a escolha preferida para muitas pesquisas e implementações de modelos de linguagem, especialmente em contextos nos quais a exploração de arquiteturas personalizadas é crucial. Sua integração natural com a computação em GPU e o foco na experiência do usuário fazem dele uma ferramenta valiosa para projetos que demandam flexibilidade e eficácia no desenvolvimento de modelos de linguagem. <https://pytorch.org/>

MXNet: Destacando-se como uma biblioteca de aprendizado profundo escalável e eficiente, é especialmente reconhecido por sua arquitetura flexível e suporte eficaz para treinamento distribuído. Desenvolvido pela Apache Software Foundation, oferece uma plataforma robusta para projetos de processamento de linguagem natural (NLP) e outras tarefas de aprendizado de máquina. Com interfaces para Python e outras linguagens, o MXNet é acessível e versátil, sendo possível ao usuário montar suas próprias redes neurais. Além disso, é conhecido por sua eficiência em treinamento de modelos em ambientes distribuídos, tornando-o adequado para lidar com grandes volumes de dados. Seja para experimentação em pequena escala ou implementações em larga escala, o MXNet oferece uma variedade de recursos para modelagem e treinamento de redes neurais em projetos que envolvem NLP e outras tarefas complexas de aprendizado de máquina. <https://mxnet.apache.org/>

FastAI: Reconhecido por tornar o aprendizado profundo mais acessível, o FastAI é uma biblioteca construída sobre o PyTorch, que fornece uma interface de alto nível que simplifica o processo de criação e treinamento de modelos complexos. O FastAI oferece abstrações poderosas para tarefas comuns em aprendizado profundo, incluindo processamento de linguagem natural (NLP). Para tarefas específicas de NLP, ele conta com o módulo `fastai.text`, que simplifica o pré-processamento de texto, o carregamento de dados e a criação de modelos de linguagem. Além disso, a biblioteca inclui funcionalidades avançadas, como transferência de aprendizado e métodos de treinamento eficazes, que são especialmente úteis para experimentação rápida e desenvolvimento de modelos de NLP com desempenho superior. <https://github.com/fastai/fastai>

Ray Tune: Especializado em otimização de hiperparâmetros, é uma biblioteca eficiente para busca e ajuste sistemático de configurações ideais de modelos. Desenvolvido sobre o ecossistema Ray, oferece uma interface intuitiva para experimentação escalável e distribuída. Ideal para projetos de processamento de linguagem natural (NLP) e aprendizado de máquina. Além disso, simplifica a seleção de hiperparâmetros, acelerando a descoberta de configurações otimizadas e melhorando o desempenho dos modelos. <https://docs.ray.io/en/latest/tune/>

Optuna: Similar ao Ray Tune, é focado em otimização de hiperparâmetros eficiente e automática, e oferece uma abordagem flexível para encontrar as melhores configurações de modelos. Faz uso de algoritmos de busca eficazes para explorar o espaço de hiperparâmetros, acelerando a descoberta de configurações otimizadas, além de possuir suporte para integração em diversas bibliotecas de aprendizado de máquina, como TensorFlow, PyTorch e Scikit-learn. <https://optuna.readthedocs.io/en/stable/>

WandB (Weights & Biases): Reconhecido como uma plataforma abrangente para rastreamento, visualização e colaboração em projetos de aprendizado de máquina, oferece uma solução completa para experimentação e monitoramento de modelos. Com suporte para várias bibliotecas, como TensorFlow e PyTorch, ele simplifica a análise e compartilhamento de resultados. Para projetos de processamento de linguagem natural (NLP) e outras tarefas, o WandB permite o registro de métricas, gráficos interativos e visualização de embeddings, fornecendo uma compreensão aprofundada do desempenho do modelo. Além disso, sua integração com fluxos de trabalho de aprendizado profundo facilita a compreensão e

otimização contínua dos modelos. Ou seja, permite uma análise detalhada e compartilhamento transparente de experimentos. <https://wandb.ai/>.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 23 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Ao longo desta semana, conforme mencionado no planejamento anterior, realizei uma busca para definir o pipeline a ser seguido para utilizar e/ou treinar um modelo para análise de sentimentos. Durante essa busca, percebi a necessidade de procurar implementações da área para, assim, identificar o pipeline que geralmente é utilizado. Porém, durante essa busca, encontrei algumas formas de implementações que possuem diferenças em seus pipelines. Assim, optei por dividir a definição do pipeline em duas semanas, sendo que para esta entrega foquei em analisar diferentes implementações de diversas fontes (artigos científicos, repositórios, blogs, etc) e analisar os principais aspectos de cada uma. Alguns dos materiais ainda não foram lidos por completo, então podem estar listados mas sem descrições, estes casos serão analisados até a próxima entrega. Esse levantamento pode ser encontrado em: [Gate 23/11/2023](#)

A busca por datasets foi adiada para a próxima semana, visto que durante essa semana o foco foi encontrar códigos e analisar os pipelines utilizados.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para a próxima entrega, pretendo:

- Finalizar a busca por implementações e identificação dos principais pipelines a serem seguidos para a parte prática da residência, considerando a viabilidade de cada um;
- Procurar datasets de análise de sentimentos sobre reviews de produtos e/ou serviços para implementar os pipelines encontrados (de preferência em português), e buscar como os dados foram coletados e anotados.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#) ▾

LUANA GUEDES BARROS MARTINS: [Go!](#) ▾

[Documento em anexo ao Termo de Aceite de Entrega de 23/11/2023]

Principais abordagens para análise de sentimentos

Durante essa semana optei por pesquisar diversas aplicações de análise de sentimentos para identificar o pipeline a ser seguido. A seguir encontra-se uma listagem dos principais materiais utilizados:

1- Twitter discussions on breastfeeding during the COVID-19 pandemic:

<https://link.springer.com/content/pdf/10.1186/s13006-023-00593-x.pdf> .

Nessa abordagem inicialmente os autores realizam a coleta dos tweets utilizando o “Tweepy” utilizando palavras-chave relacionadas a COVID e “breastfeeding”. Em seguida realizam o pré-processamento dos textos (principalmente com NLTK e Spacy), e utilizam o framework SentiStrength para classificar a polaridade emocional desses textos. Por fim, utilizam a biblioteca gensim para agrupar os textos em clusters. Dessa forma, nessa abordagem foi utilizado um classificador baseado em léxicos para realizar a análise de sentimentos dos textos.

2- Sentiment Analysis of COVID-19 Tweets: From Model Training to Hugging Face Deployment:

<https://medium.com/@rasmowanyama/sentiment-analysis-of-covid-19-tweets-from-model-training-to-docker-deployment-e73ba2a7aebf>

Esse artigo mostra o fine-tuning de um modelo pré-treinado do Hugging Face (RoBERTa e BERT) para realizar análise de sentimentos, e também como fazer seu deploy utilizando o Gradio. Porém, meu foco é no desenvolvimento do modelo. Para isso é utilizado como pipeline: importação dos dados, pré-processamento destes, definição dos conjuntos de dados para treino e para teste, definir as configurações de treino, treinar o modelo e, por fim, avaliar o desempenho.

3- The Most Favorable Pre-trained Sentiment Classifiers in Python:

<https://towardsdatascience.com/the-most-favorable-pre-trained-sentiment-classifiers-in-python-9107c06442c6>

Apresenta os principais classificadores pré-treinados para análise de sentimentos, e mostra as principais características de cada um. O código pode ser encontrado em

<https://github.com/PetrKorab/The-Most-Favorable-Pre-trained-Sentiment-Classifiers-i>

[n-Python](#) , e a partir dele é possível visualizar o pipeline seguido para cada implementação.

4- Use ChatGPT API for Sentiment Analysis in Python:

https://medium.com/@financial_python/use-chatgpt-api-for-sentiment-analysis-in-python-5a152ddb3238

Nessa implementação é mostrado como utilizar o ChatGPT para realizar a análise de sentimentos. É explorado tanto como utilizar o ChatGPT em um código python para inferência, quanto os parâmetros que podem ser necessários para realizar essa inferência de forma adequada (por exemplo temperature e quantidade máxima de tokens).

5- Deploy BERT for Sentiment Analysis with FastAPI:

<https://github.com/curiously/Deploy-BERT-for-Sentiment-Analysis-with-FastAPI>

Nesse repositório o foco é como realizar o deploy do modelo, porém também contém os arquivos mostrando o desenvolvimento do modelo para análise de sentimentos utilizando modelos pré-treinados baseados em arquitetura Transformers. Dessa forma, é possível analisar as etapas seguidas para realizar o fine-tune do modelo.

6- Como fazer uma análise de sentimentos com Vader:

<https://medium.com/turing-talks/como-fazer-uma-an%C3%A1lise-de-sentimentos-com-vader-21bbe3f3e38d>

Nessa abordagem, inicialmente é comentado sobre o que é análise de sentimentos, e em seguida uma explicação sobre os dados utilizados. Na implementação em si, primeiro é realizado o pré-processamento, em especial com BeautifulSoup, e logo em seguida é utilizada a biblioteca VADER para realizar a análise de sentimento. Maiores detalhes sobre o funcionamento do VADER e suas regras podem ser encontrados nos seguintes links:

<https://akladyous.medium.com/sentiment-analysis-using-vader-c56bcffe6f24>

<https://github.com/cjhutto/vaderSentiment>

7- Sentiment Analysis on Amazon Reviews using TF-IDF Approach:

<https://medium.com/analytics-vidhya/sentiment-analysis-on-amazon-reviews-using-tf-idf-approach-c5ab4c36e7a1>

Nessa abordagem, primeiro é apresentado o dataset e sua importação. Em seguida é realizado seu processamento, e então mostrado como realizar a extração de features utilizando o TF-IDF. Após isso, são utilizados diferentes classificadores e

é analisada a eficácia de cada um. Ou seja, mostra como obter um classificador de análise de sentimentos utilizando abordagens clássicas.

8-Twitter Sentiment Analysis:

<https://www.kaggle.com/code/paoloripamonti/twitter-sentiment-analysis>

Nessa abordagem são utilizados embeddings estáticos (word2vec) para representar as palavras, e após essa feature extraction é utilizada uma rede neural tipo LSTM para treinar um modelo de análise de sentimentos.

Principais pipelines analisados

Analisando as abordagens listadas e algumas outras que não foram listadas por seguirem processos semelhantes, percebi 4 abordagens principais:

- Realizar o fine-tuning de um modelo baseado em Transformers pré-treinado. Nesse caso o pipeline, de modo geral, seria:
importação dos dados -> pré-processamento -> feature extraction com um Tokenizador pré-treinado -> fine-tune do modelo -> avaliação de métricas
- Utilizar algum método clássico para a extração de features (por exemplo TF-IDF ou Word2Vec) e, então, utilizar algum algoritmo para classificação (por exemplo SVM) ou montar uma rede neural (por exemplo LSTM) e realizar o treino com essa rede. Nesse caso o pipeline, de modo geral, seria:
importação dos dados -> pré-processamento -> feature extraction -> treino de um classificador -> avaliação de métricas
Ou então:
importação dos dados -> pré-processamento -> feature extraction -> criação de uma rede neural -> treinamento da rede -> avaliação de métricas
- Utilizar um classificador baseado em léxico (por exemplo o VADER). Nesse caso o pipeline, de modo geral, seria:
importação dos dados -> pré-processamento -> aplicação do classificador -> avaliação das métricas

Cada uma possui suas vantagens e desvantagens, assim como detalhes específicos, que serão melhor abordados quando forem realizadas suas implementações.

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 30 de nov. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Ao decorrer dessa semana, assim como mencionado no planejamento do termo de entrega anterior, finalizei a análise das implementações que foi iniciada semana passada, selecionando e descrevendo as principais, o que pode ser conferido em: [Gate 23/11/2023 - 30/11/2023](#)

Além disso, no termo anterior também foi mencionado a busca por um dataset para realizar experimentos, e optei por utilizar o B2W Reviews01. Para me aproximar mais de um especialista e considerando o tempo restante de residência, optei por começar a explorar diferentes aplicações envolvendo análise de sentimentos. Nessa semana explorei a abordagem baseada em léxico, utilizando o repositório LeIA (um fork do VADER para português). Mais informações tanto do dataset quando da aplicação, assim como o código, podem ser encontrados em : [Gate 30/11/2023](#)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Pretendo seguir na linha de explorar diferentes abordagens para análise de sentimentos em textos, indo atrás de aplicações e buscando implementá-las, de preferência utilizando o mesmo dataset que foi utilizado nesta semana (caso necessário, será buscado algum outro dataset).

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

LUANA GUEDES BARROS MARTINS: Em análise! ▾

[Documento em anexo ao Termo de Aceite de Entrega de 30/11/2023]

Especificações dos dados e da abordagem baseada em léxico

1- Descrição do que foi feito

A fim de me tornar um especialista em análise de sentimentos em NLP, e tendo como base o que foi feito até o momento, a partir de agora irei focar minha especialização em explorar diferentes abordagens para análise de sentimentos. Para essa semana, finalizei o documento da semana passada pesquisando diferentes métodos para análise de sentimentos, e realizei a busca por datasets que possam ser utilizados para essa área. Inicialmente, será utilizado o dataset B2W Reviews01, que pode ser encontrado em: <https://github.com/americanas-tech/b2w-reviews01>.

Como o foco é aprofundar nas abordagens para análise de sentimentos, e não tanto em etapas como pré-processamento e seleção de dados, irei utilizar os dados já disponibilizados no artigo <https://arxiv.org/pdf/2112.05459.pdf>, que podem ser encontrados em <https://www.kaggle.com/datasets/fredericods/ptbr-sentiment-analysis-datasets>.

Nesse repositório já estão presentes 5 datasets distintos (entre eles o B2W-Reviews01) e uma versão concatenada desses dados. Já foram adicionadas colunas com os dados pré-processados, tokenizados, e a divisão em folds para separação em treino, teste e validação, tanto para abordagem binária (positivo ou negativo) quanto multi classe (1 a 5 estrelas).

Dessa forma, irei passar a focar em implementar e analisar diferentes abordagens para análise de sentimentos, começando essa semana pela abordagem baseada em léxico e com dados binários.

2- Descrição dos dados

No dataset mencionado estão presentes 8 colunas:

- `original_index`: indica o index da linha em questão em relação ao dataset original;
- `review_text`: texto original da avaliação;
- `review_text_processed`: texto da avaliação após passar pelo pré-processamento (remoção de stopwords e caracteres especiais, todo o texto passado para minúsculo, e conversão para letras em inglês, como por exemplo remoção de acentos);

- `review_text_tokenized`: texto da coluna `review_text_processed` já tokenizado (cada palavra separada);
- `polarity`: polaridade do sentimento para uma abordagem binário (avaliações de nota 1 ou 2 viram 0.0, com nota 4 ou 5 viram 1.0, e com nota 3 são descartadas);
- `rating`: nota original do autor da review. Um valor inteiro de 1 a 5;
- `kfold_polarity`: em qual subgrupo do dataset aquele dado se encontra, considerando um modelo binário;
- `kfold_rating`: em qual subgrupo do dataset aquele dado se encontra, considerando um modelo multi classe.

3- Descrição das implementações

Inicialmente, optei por uma abordagem binária, apenas com duas classes: negativo (0, com avaliações de 1 ou 2 estrelas) e positivo (1, com avaliações de 4 ou 5 estrelas). Nesse experimento, as avaliações de 3 estrelas foram descartadas. O processador utilizado em ambos os casos foi Intel(R) Xeon(R) CPU @ 2.20GHz.

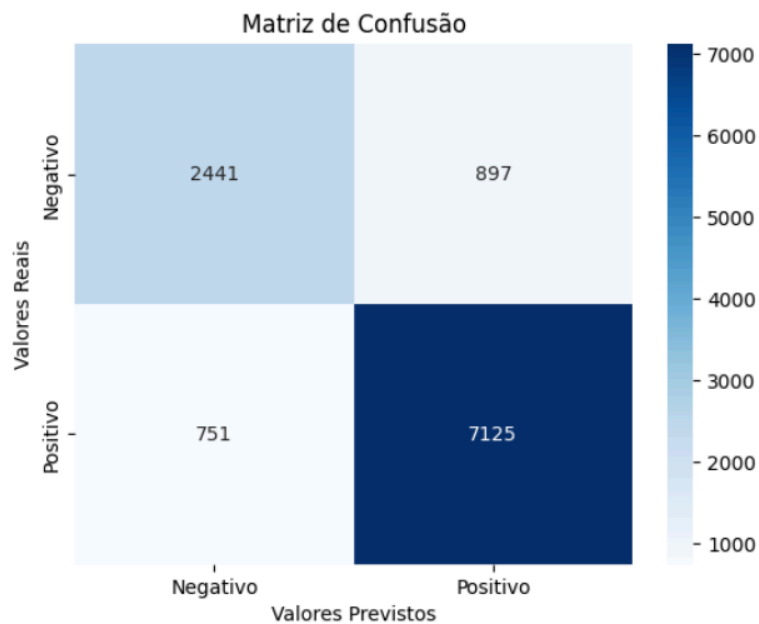
Para testar a abordagem baseada em léxico, primeiramente importei os dados do dataset utilizando o Pandas. Em seguida, o pré-processamento dos textos já havia sido realizado, porém foi necessário verificar a existência de linhas duplicadas e excluí-las. Assim, de 132.373 dados originais restaram 129.863.

Os dados já estavam divididos em folds para facilitar a divisão em treino, teste e validação. Como para essa abordagem não é necessário um conjunto de treino, utilizei apenas os dados do fold número 10 para realizar o teste.

Em seguida, utilizei o framework LelA (versão em português do VADER) para realizar inferência nos dados de teste. A inferência retorna um valor “compound” de -1 a 1, no qual considerei valores de 0 a 1 como label positiva, e valores negativos como label negativa.

Por fim, utilizei os labels originais do dataset em comparação com a inferência do LelA para obter as métricas relevantes. Dessa forma, foram obtidos os seguintes resultados:

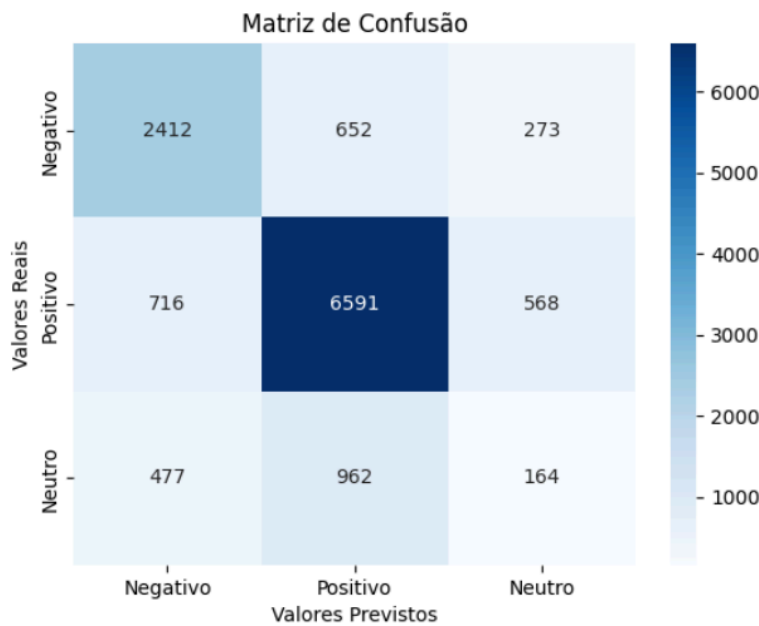
- Acurácia: 0.8530
- F1-score weighted: 0.8521
- Tempo gasto na inferência: 174.7931 segundos
- Matriz de confusão:



Foi possível perceber algumas limitações dessa abordagem, como por exemplo a existência de frases com palavras que não estão no dicionário do repositório LeIA, o que afeta diretamente as inferências do modelo.

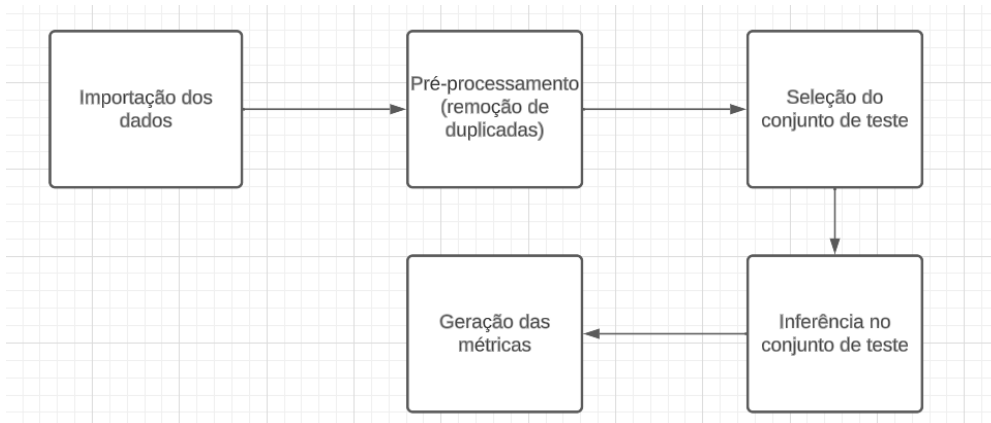
Também foi experimentado adicionar as avaliações com nota 3 como uma nova classe (neutro), de modo que se o compound fosse 0 seria considerado neutro, se for maior que 0 é positivo e menor que 0 é negativo. Dessa forma, foram obtidos os seguintes resultados:

- Acurácia: 0.7153
- F1-score weighted: 0.7004
- Tempo gasto na inferência: 198.3104 segundos
- Matriz de confusão:



Observa-se que no caso binário os resultados obtidos foram superiores, o que é justificável levando em consideração que a adição de uma classe aumenta a complexidade do problema.

De modo geral, em ambos os casos o pipeline seguido foi:



Link para o repositório do LeIA: <https://github.com/rafjaa/LeIA>

O código feito pode ser encontrado em:  Gate 30/11/2023.ipynb

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 7 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa semana dei prosseguimento à experimentação de diferentes abordagens utilizadas para análise de sentimentos. Assim, foram experimentadas algumas abordagens de machine learning mais clássicas, fazendo uso das representações TF-IDF e Word2Vec, e também tanto de classificadores clássicos, como Random Forest, quanto de redes neurais LSTM. O objetivo de testar essas abordagens é para que, como um especialista em análise de sentimentos, eu seja capaz de julgar quais abordagens melhor se encaixam em cada problema, visto que todas possuem suas vantagens e desvantagens. As principais métricas consideradas, assim como na semana passada, foram a acurácia, f1-weighted e tempo gasto. Mais detalhes de cada implementação, assim como o link para o notebook utilizado, podem ser encontrados em: [Gate 07/12/2023](#)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

O próximo passo será experimentar abordagens com arquiteturas Transformer, que possuem redes neurais mais robustas e exigem maior tempo e recursos computacionais para serem implementados.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

Neste gate, o Professor Aldo André Díaz Salazar esteve na banca avaliadora substituindo a Professora Luana.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

LUANA GUEDES BARROS MARTINS: Em análise! ▾

[Documento em anexo ao Termo de Aceite de Entrega de 07/12/2023]

Especificações das abordagens clássicas e com redes neurais

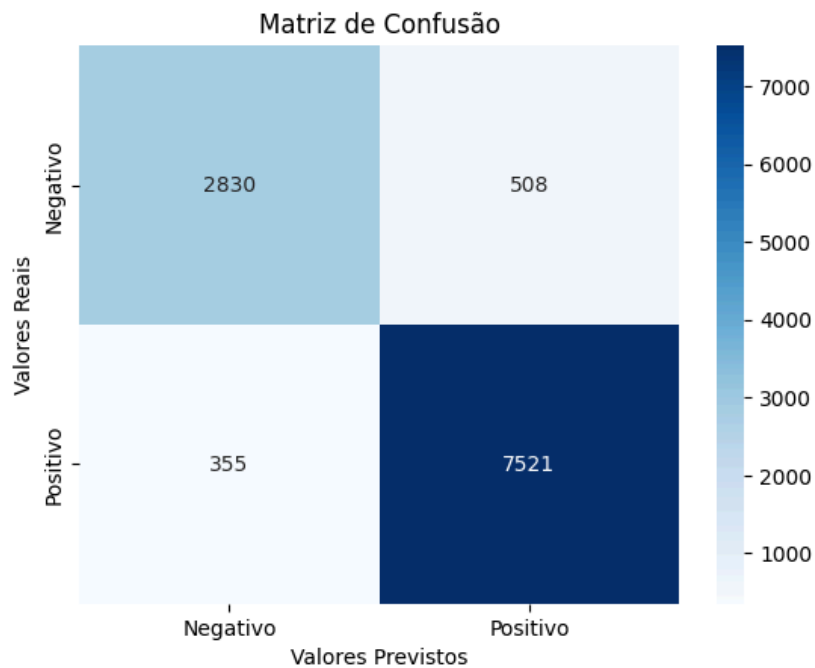
Seguindo o objetivo de testar diferentes abordagens para análise de sentimentos, durante essa semana foram testadas algumas abordagens mais clássicas, fazendo uso das representações TF-IDF e Word2Vec, e também tanto de classificadores clássicos, como Random Forest, quanto de redes neurais LSTM. Foram utilizados o mesmo conjunto de dados (B2W Reviews01) [1] e o mesmo processador (Intel(R) Xeon(R) CPU @ 2.20GHz) da semana anterior. Foi utilizada a abordagem binária das classes do dataset.

1- Abordagem com TF-IDF: Random Forest e Logistic Regression

Os dados foram separados em treino (primeiros 9 folds) e teste (fold 10), para posteriormente utilizar a representação TF-IDF com a biblioteca sklearn. Em seguida, foram utilizados os classificadores Random Forest e Logistic Regression, para assim compreender melhor essa abordagem.

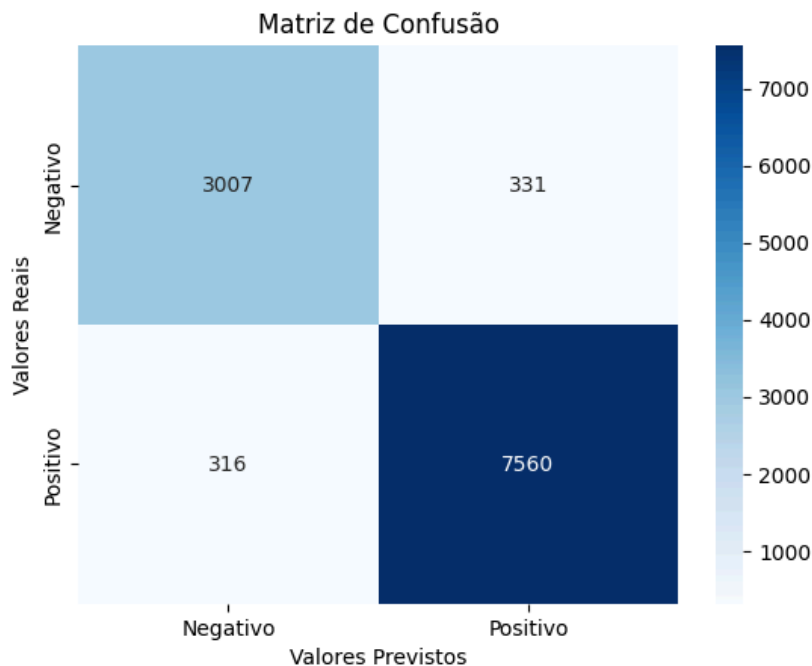
Com o Random Forest os resultados obtidos foram:

- Acurácia: 0.9230
- F1-score weighted: 0.9225
- Tempo gasto no treino e inferência: 311.7701 segundos
- Matriz de confusão:



Com o Logistic Regression os resultados obtidos foram:

- Acurácia: 0.9423
- F1-score weighted: 0.9423
- Tempo gasto no treino e inferência: 3.5027 segundos
- Matriz de confusão:



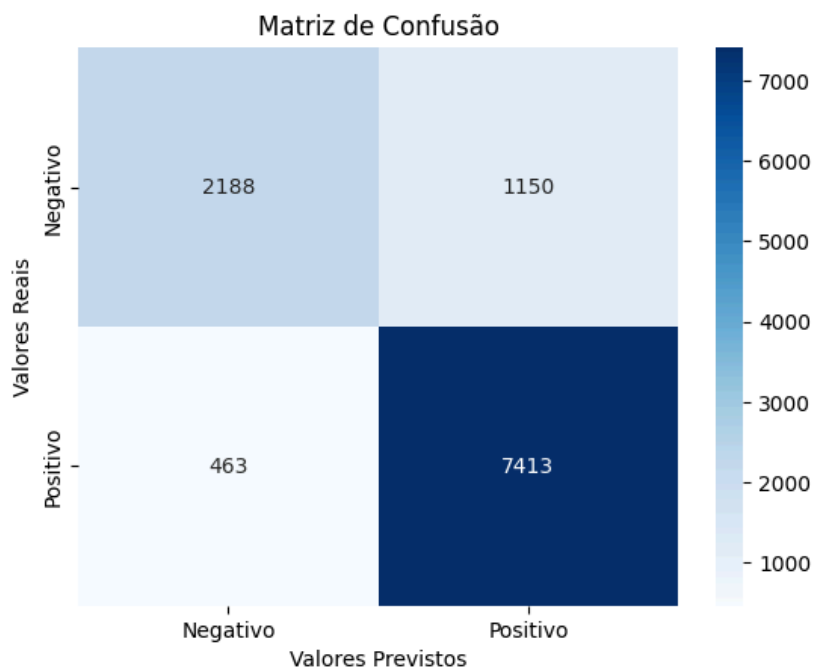
2- Abordagem com Word2Vec: Random Forest e Logistic Regression

Primeiramente, utilizando a mesma divisão de dados que foi utilizada com o TF-IDF, utilizei uma representação skip-gram de 300 dimensões já treinada, fornecido pelo NILC [2], visto que o ideal é que o treinamento de um modelo Word2Vec seja feito em grandes conjuntos de dados [3]. Como o objetivo é explorar as possibilidades de abordagens para análise de sentimentos, sem necessariamente focar em uma em específico, foi optado utilizar um modelo já treinado e avaliar seus resultados.

Após passar os dados para a representação Word2Vec, foram utilizados os classificadores Random Forest e Logistic Regression para realizar a análise de sentimentos.

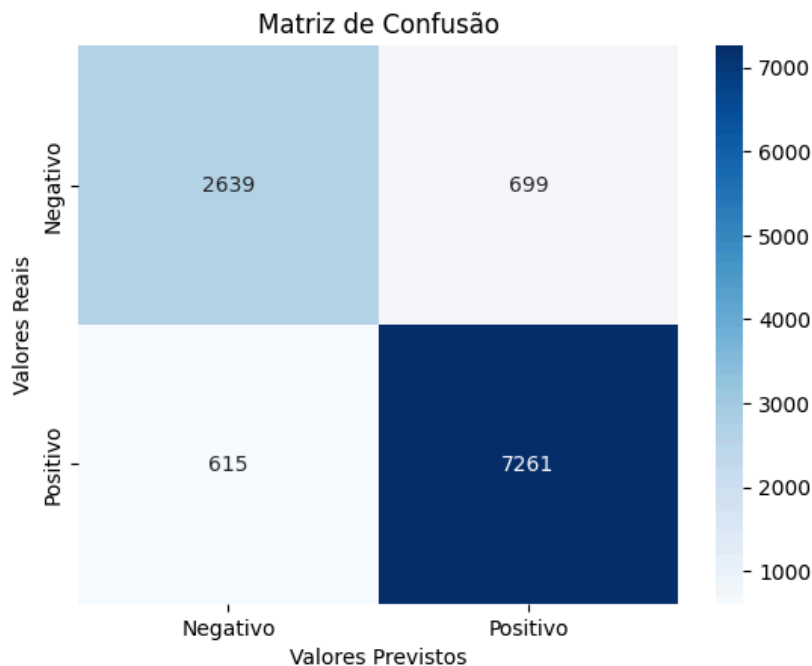
Com o Random Forest os resultados obtidos foram:

- Acurácia: 0.8562
- F1-score weighted: 0.8509
- Tempo gasto no treino e inferência: 297.0116 segundos
- Matriz de confusão:



Com o Logistic Regression os resultados obtidos foram:

- Acurácia: 0.8828
- F1-score weighted: 0.8824
- Tempo gasto no treino e inferência: 9.2007 segundos
- Matriz de confusão:

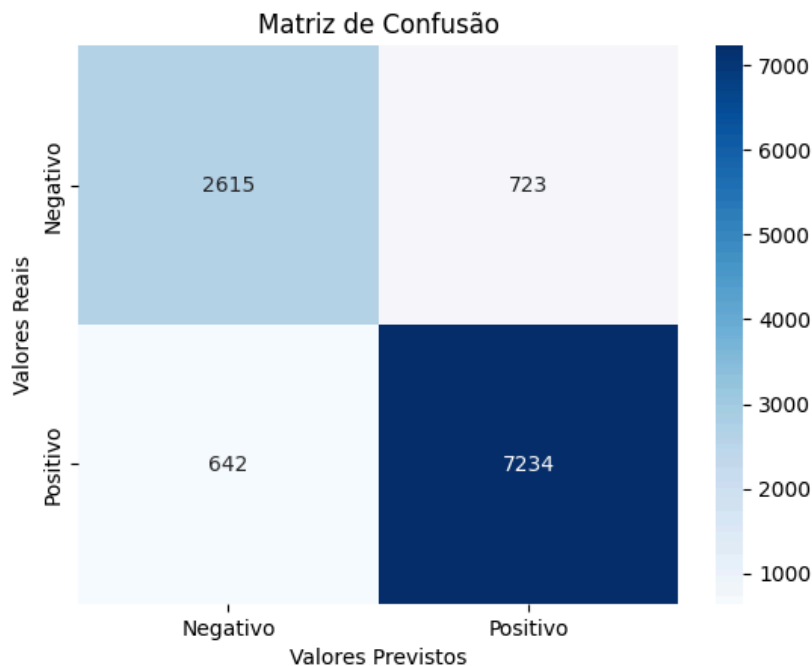


3- Abordagem com Word2Vec: LSTM

Assim como no tópico 2, foi mantida a representação Word2Vec utilizada, porém para realizar a análise de sentimentos foi utilizada uma arquitetura de rede neural, a LSTM (Long Short-Term Memory) [4]. Para isso, inicialmente foi utilizada uma camada de incorporação (Embedding), seguida por uma camada LSTM com 100 unidades, e por fim, uma camada Dense com função de ativação Softmax. Durante o treinamento foram utilizadas 5 épocas, com batch size de 1024 e learning rate de $1e-3$. Mais detalhes podem ser conferidos no notebook mencionado ao final desta seção.

Os resultados obtidos com essa implementação foram:

- Acurácia: 0.8783
- F1-score weighted: 0.9138
- Tempo gasto no treino e inferência: 7469.9533 segundos (aproximadamente 2 horas)
- Matriz de confusão:



Percebe-se que com o aumento da complexidade dos modelos o tempo de execução aumenta cada vez mais, indicando que para futuras abordagens será necessário o uso de máquinas mais potentes, como GPUs. Além disso, observa-se que não necessariamente um aumento da complexidade da representação ou do classificar gera métricas melhores, visto que dentre esses experimentos o que gerou melhores resultados foi o TF-IDF em conjunto ao Logistic Regression.

O notebook com todas as implementações descritas pode ser encontrado em:

 Gate 7/12/2023 .ipynb

Referências

- [1] Frederico Dias Souza, Joao Baptista de Oliveira e Souza Filho. 2021. Sentiment Analysis on Brazilian Portuguese User Reviews.
- [2] “Repositório de Word Embeddings do NILC”. Disponível em: <<http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>>.
- [3] Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space.
- [4] Sepp Hochreiter, Jürgen Schmidhuber. 1997. Long Short-Term Memory.

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 14 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa semana dei sequência à exploração de diferentes abordagens para análise de sentimentos em NLP, e assim como mencionado no planejamento do último gate, explorei abordagens baseadas em Transformers. Utilizei a biblioteca Hugging Face para compreender melhor como realizar o fine-tune de modelos pré-treinados, visto que o treinamento de um modelo próprio não é viável no momento devido à alta necessidade de tempo, dados e recursos computacionais.

Inicialmente vi que o próprio Hugging Face possui implementações para realizar esse fine-tune, porém são implementações de alto nível, ou seja, em uma função já são realizadas diversas etapas, o que dificulta o entendimento do processo. Dessa forma, além de utilizar essa abordagem também explorei uma aplicação utilizando o Pytorch, a fim de entender melhor as peculiaridades de cada abordagem, assim como seus pontos positivos e negativos.

Uma melhor descrição das implementações realizadas, assim como o notebook feito, podem ser encontrado em: [Gate 14/12/2023](#)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Explorar mais as abordagens baseadas em Transformers, buscando o que geralmente é feito para extrair o máximo dessa ferramenta.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

LUANA GUEDES BARROS MARTINS: [Go! ▾](#)

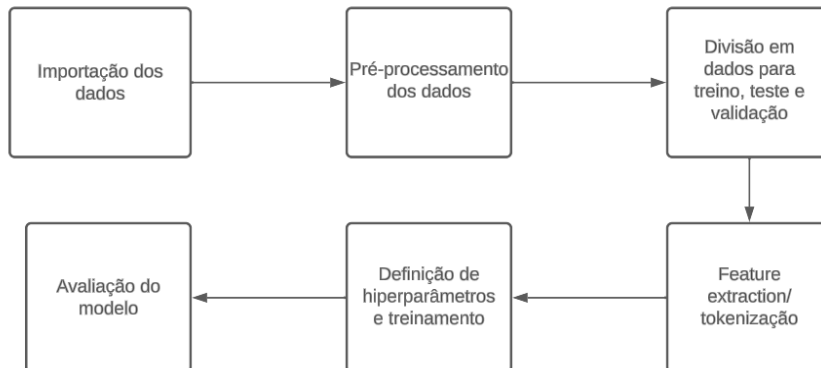
[Documento em anexo ao Termo de Aceite de Entrega de 14/12/2023]

Especificações da abordagem com Transformers

Seguindo o objetivo de testar diferentes abordagens para análise de sentimentos, durante essa semana foram testadas abordagens utilizando arquitetura Transformers, fazendo uso do mesmo conjunto de dados (B2W Reviews01) [1] das semanas anteriores. Foi utilizada a biblioteca Hugging Face, que contém diversos modelos pré-treinados para serem utilizados em fine-tuning, visto que o treinamento de um modelo Transformers exige altas quantidade de dados e recursos computacionais. O próprio Hugging Face disponibiliza uma função Trainer para realizar o fine-tuning, porém também é possível desenvolver as funções para essa tarefa. A fim de explorar melhor as duas abordagens irei desenvolver modelos utilizando ambas, visando observar possíveis diferenças entre elas, assim como suas vantagens e desvantagens.

Em ambos os experimentos foi utilizado o mesmo modelo pré-treinado e tokenizador Bertimbau [2], assim como mesma quantidade de épocas, hiperparâmetros (como seed, learning rate e batch size) e divisão dos dados. Em ambos os casos foi utilizada uma GPU Tesla T4. Maiores detalhes podem ser conferidos no notebook mencionado ao final deste documento. De modo geral, o pipeline utilizado foi:

1. Importação dos dados e pré-processamento (no dataset utilizada o pré-processamento dos textos já tinham sido feitos);
 2. Separação dos dados para treino, teste e validação;
 3. Tokenização
 4. Definição dos hiperparâmetros e treinamento (fine-tune do modelo pré-treinado);
 5. Avaliação do modelo no conjunto de teste
- Isso pode ser visto no diagrama abaixo:



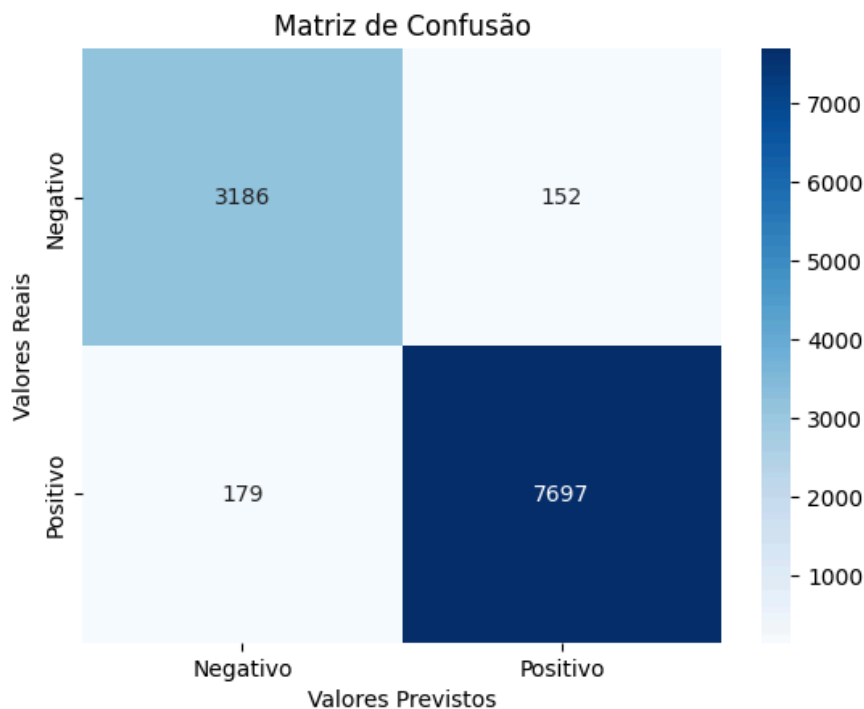
1- Fine-tuning usando Trainer do Hugging Face

Inicialmente foi necessário colocar o dataset em formato DatasetDict, visto que o Trainer da biblioteca requer os dados nesse formato. Assim, passei o dataset de .csv para DatasetDict, já dividindo os conjuntos de treino, teste e validação. Em seguida, os textos foram tokenizados, adicionando ao dataset as features input_ids, token_type_ids e attention_mask, que serão utilizadas pela arquitetura Transformers.

Em seguida, foi necessário criar uma função para definir as métricas que seriam utilizadas, definir os hiperparâmetros com TrainingArguments, definir os parâmetros da função Trainer e, por fim, realizar o fine-tune do modelo Bertimbau para o dataset em questão. Ao final, foi feita a inferência no conjunto de teste, visto que as métricas obtidas durante o treino são referentes ao conjunto de validação.

Foram obtidos os seguinte resultados no conjunto de teste:

- Acurácia: 0.9705
- F1-score weighted: 0.9705
- Tempo gasto no treino e inferência no modelo de validação: 7230.7482 segundos
- Matriz de confusão:



2- Fine-tuning com Pytorch

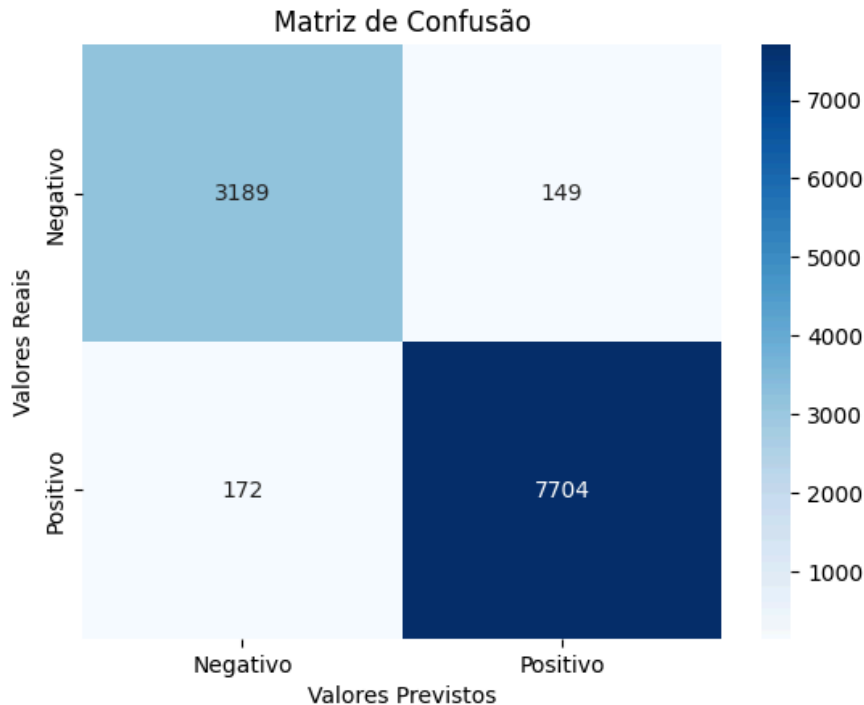
Para essa abordagem foi utilizado o mesmo modelo pré-treinado do Bertimbau, tanto para a tokenização quanto para o modelo. A divisão dos dados em treino, teste e validação foi a mesma, porém como foi utilizado o Pytorch ao invés da função pronta do Hugging Face foi necessário realizar etapas extras. Primeiramente, foi necessário realizar a tokenização de cada subset de dados em separado, e depois passar cada um para formato de TensorDataset e criar o dataloader com o Pytorch.

Depois foi definido o otimizador e o scheduler, e também hiperparâmetros como quantidade de épocas e learning rate. Em seguida, foi necessário criar uma função para realizar o fine-tune do modelo, e iterar pela quantidade de épocas definida, obtendo ao final os valores da loss e F1-score para os conjuntos de treino e validação. Por fim, com o modelo treinado foi realizada a inferência no conjunto de teste.

Foram obtidos os seguinte resultados no conjunto de teste:

- Acurácia: 0.9714
- F1-score weighted: 0.9714
- Tempo gasto no treino e inferência no modelo de validação: 11389.2370 segundos

- Matriz de confusão:




3- Observações

Percebe-se que mesmo utilizando os mesmos dados e hiperparâmetros ainda há uma pequena diferença nos resultados, provavelmente devido ao fato da implementação do Hugging Face ser em alto nível, ou seja, grande parte das etapas estão contidas dentro das funções já implementadas, havendo menor controle sobre elas.

Também é nítida a diferença no tempo para treinamento de cada modelo, visto que a implementação do Hugging Face otimiza diversas etapas do treinamento, o que não foi feito na implementação com pytorch.

O notebook com todos os experimentos descritos pode ser encontrado em

 [Gate 14/12/2023.ipynb](#)

Referências

- [1] Frederico Dias Souza, Joao Baptista de Oliveira e Souza Filho. 2021. Sentiment Analysis on Brazilian Portuguese User Reviews.
- [2] NEURALMIND. BERTimbau Base. Disponível em: <https://huggingface.co/neuralmind/bert-base-portuguese-cased>. Acesso em: 10 dez. 2023.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 21 de dez. de 2023

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa semana foi dada sequência à exploração do uso de modelos baseados em Transformers para análise de sentimentos, buscando conhecer melhor as possíveis abordagens e métodos a serem utilizados. Primeiramente, explorei mais o repositório de modelos do Hugging Face, buscando algum modelo que fosse pré-treinado especificamente para análise de sentimentos em português (por exemplo em avaliações de produtos ou redes sociais), porém foram encontrados apenas 3 modelos distintos e algumas variações destes que se encaixam nessa categoria, e todos com arquivos faltantes no repositório, impossibilitando seu uso (os modelos podem ser visualizados em:

https://huggingface.co/models?pipeline_tag=text-classification&sort=trending&search=sentiment+pt).

Existem modelos em outras línguas (especialmente em inglês) que foram pré-treinados para análise de sentimentos, e seu uso é similar ao que experimentei até o momento, porém em português optei por manter utilizando o modelo BERTimbau, o mesmo da semana passada, visto que o objetivo principal é explorar as diferentes abordagens e possibilidades de implementação.

Além disso, também fui atrás de formas para realizar a otimização de hiperparâmetros, visto que um dos pontos cruciais para realizar o fine tune de um modelo é a escolha de hiperparâmetros. Para isso foi utilizado o framework Optuna, que já havia sido visto por alto no levantamento de frameworks realizado para o gate 4. Essa otimização foi realizada em menor escala, visto que é algo que demanda tempo e poder computacional (uso de GPU), porém compreendendo o processo de realizar essa otimização, mesmo que em menor escala, se necessário acredito que serei capaz de realizar uma otimização em maior escala, seja otimizando mais hiperparâmetros ou aumentando a quantidade de testes. Maiores detalhes desta implementação e dos testes realizados podem ser encontrados em: [Gate 21/12/2023](#)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para o próximo gate pretendo:

- Explorar o uso de LLMs para realizar análise de sentimentos de forma zero-shot (sem fornecer dados de treino).

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go! ▾

LUANA GUEDES BARROS MARTINS: Go! ▾

[Documento em anexo ao Termo de Aceite de Entrega de 21/12/2023]

Especificações da otimização de hiperparâmetros

Para essa semana o foco foi explorar formas de melhorar o desempenho do fine tune de modelos baseados em Transformers para análise de sentimentos. Para isso, primeiro foi feita uma busca por modelos pré-treinados em conjuntos de dados para análise de sentimentos. Em português foram encontrados poucos modelos específicos para análise de sentimentos ([1] e [2]), e eles possuem arquivos faltantes, então não foi possível utilizá-los. Foram testados alguns modelos que foram treinados em inglês [3], mas os resultados foram inferiores àqueles obtidos na semana anterior utilizando o Bertimbau [4].

Além disso, fui atrás de como realizar a otimização de hiperparâmetros, uma abordagem extremamente útil para melhorar o desempenho de modelos, visto que simples mudanças de hiperparâmetros podem diferenciar um bom modelo de um modelo que não aprende nada [5]. Assim, optei pelo framework Optuna [6] para realizar essa otimização de hiperparâmetro.

Na semana anterior foram exploradas duas abordagens para fine tune de um modelo pré-treinado, uma utilizando o Trainer do Hugging Face e outra com Pytorch. Para realizar a otimização irei utilizar a abordagem com Pytorch, visto que nela a definição dos hiperparâmetros é mais clara e melhor adaptada para o formato do Optuna.

O principal desafio dessa otimização é que o fine tune de um modelo Transformers envolve diversos hiperparâmetros, e pode ser necessário realizar vários testes, o que consome muito tempo. Como o objetivo é compreender a abordagem, limitei a quantidade de épocas a apenas 2, trunquei os textos em apenas 128 tokens, realizei a otimização apenas na learning rate e no weight decay, e utilizei uma quantidade de trials (quantidade de vezes que o modelo será treinado) em apenas 3, de modo que fosse possível compreender o processo e conseguir executar tudo no tempo proposto. Assim, não foram obtidas grandes melhorias no modelo, mas compreendendo como realizar essa abordagem será útil para cenários futuros em que, havendo maior disponibilidade de tempo e recursos computacionais, seja possível obter melhorias significativas.

Foi possível perceber que a mudança de hiperparâmetros afeta o resultado final do modelo, podendo ser o diferencial entre um modelo bom e um ruim, porém é um processo demorado e custoso do ponto de vista computacional, sendo

necessário levar esses fatores em consideração ao buscar utilizar uma otimização de hiperparâmetros.

O notebook com a implementação do Optuna mencionado pode ser encontrado em: [Gate 21/12/2023.ipynb](#) .

Referências

- [1] CardiffNLP. XLM-Roberta-base-tweet-sentiment-pt. Disponível em: <https://huggingface.co/cardiffnlp/xlm-roberta-base-tweet-sentiment-pt>. Acesso em: 18 dez. 2023
- [2] CardiffNLP. xlm-v-base-tweet-sentiment-pt. Disponível em: <https://huggingface.co/cardiffnlp/xlm-roberta-base-tweet-sentiment-pt>. Acesso em: 18 dez. 2023
- [3] CardiffNLP. twitter-roberta-base-sentiment. Disponível em: <https://huggingface.co/cardiffnlp/xlm-roberta-base-tweet-sentiment-pt>. Acesso em: 18 dez. 2023
- [4] NEURALMIND. BERTimbau Base. Disponível em: <https://huggingface.co/neuralmind/bert-base-portuguese-cased>. Acesso em: 10 dez. 2023.
- [5] LIU, Xueqing; ZHANG, Zihang; ZHENG, Minghui; SUN, Yuting; YANG, Yi. An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models.
- [6] OPTUNA. Disponível em: <https://optuna.org/>. Acesso em: 18 dez. 2023.

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião (“gate”) de aprovação: 11 de jan. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Pedro Koziel Diniz

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante essa última semana explorei mais uma abordagem para análise de sentimentos, fazendo uso de Large Language Model (LLM). Utilizei o modelo GPT 3.5 turbo e a API da OpenAI, um LLM multilingual treinado em diversos tipos diferentes de dados textuais.

Mesmo não sendo seu foco, parte dos dados de treino do GPT é constituída com informações sobre análise de sentimentos, fazendo com que seja uma abordagem mais recente interessante de ser investigada para essa tarefa, já havendo trabalhos feitos envolvendo essa abordagem.

Optei por realizar essa análise de sentimentos de forma zero shot, ou seja, sem realizar um fine-tune do modelo, utilizando um prompt simples solicitando que o modelo faça a análise de sentimentos. Existem outros usos com LLM que poderiam ser utilizados, como realizar um fine-tune para análise de sentimentos ou então organizar um prompt que direcione melhor o modelo para essa tarefa, e que serão alvos de estudos futuros.

Maiores detalhes podem ser encontrados no seguinte documento: [Gate 11/01/2024](#)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Organizar as entregas passadas e adequar para o formato do TCC.

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: [Go!](#)

LUANA GUEDES BARROS MARTINS: [Go!](#)

[Documento em anexo ao Termo de Aceite de Entrega de 11/01/2024]

Especificações da abordagem com LLM

Para essa última semana, foi explorado o uso de Large Language Model (LLM) para análise de sentimentos, sem realizar o treino ou fine-tuning do modelo, ou seja, de forma zero-shot. Para isso, utilizei o mesmo conjunto de dados das semanas anteriores, porém apenas a partição de teste, visto que não utilizei dados para treinar nem validar um modelo, resultando em um total de 11605 dados. Diferentemente dos testes anteriores, não foi utilizado nenhum pré-processamento nos dados.

Inicialmente procurei modelos de código aberto, porém como a quantidade de dados é alta encontrei dificuldades devido tanto a limite de requisições quanto linguagem, visto que muitos desses modelos são apenas para a língua inglesa, o que me levou a optar por utilizar a API da OpenAI com o modelo “GPT 3.5 turbo”. Mesmo sendo um modelo pago ainda há um limite de requisições diárias, fazendo com que fosse necessário realizar as predições dos dados em dias diferentes e houvesse um delay devido ao tempo de requisição, por isso para esse caso não analisei o tempo.

Essa abordagem foi inspirada no artigo SentimentGPT [1], que explora o uso do modelo GPT para análise de sentimentos na língua inglesa, com uma das abordagens utilizadas sendo apenas com o uso de prompt, ou seja, não foi realizado fine-tune do modelo específico para a tarefa de análise de sentimentos.

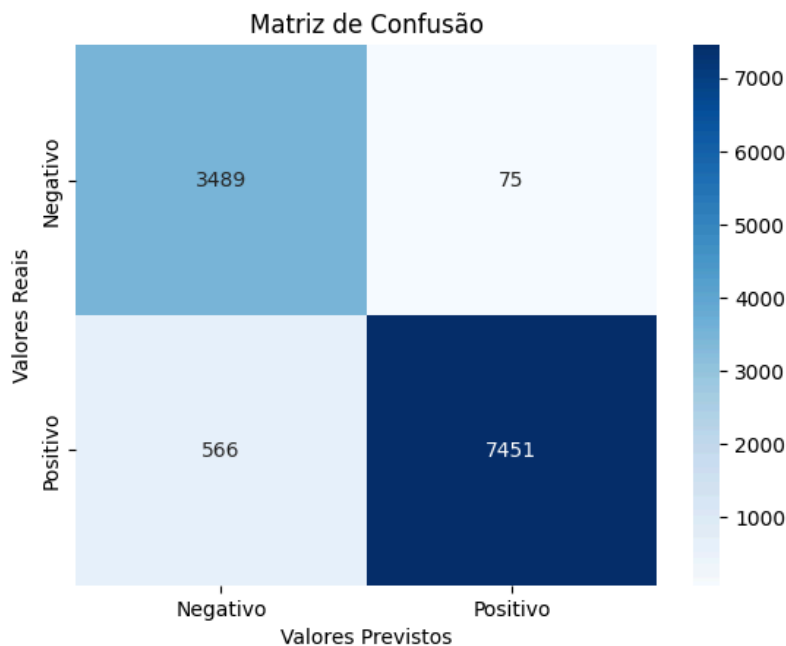
Inicialmente foram feitos testes pelo próprio site da OpenAI para determinar um modelo de prompt em que o GPT retornasse apenas 0, no caso de sentimento negativo, ou 1, no caso de sentimento positivo. Ainda houve alguns poucos casos em que a resposta do modelo não seguiu exatamente esse formato, sendo necessário realizar um pós-processamento dessas respostas para se adequar ao formato das outras. Esse desenvolvimento pode ser conferido em maiores detalhes no notebook referenciado ao final deste documento.

O conjunto de dados de teste em questão possui apenas comentários de produtos positivos ou negativos, porém com o uso do GPT foi possível detectar alguns dados que não deveriam estar nesse conjunto, visto que mesmo sendo comentários na seção dos produtos não tinham nenhuma relação com estes, como por exemplo divulgação de canais do youtube, dúvidas de clientes ou simplesmente textos sem nenhum nexos. Isso mostra a possibilidade de uso dos LLMs tanto para realizar análise de sentimentos, quanto como uma filtragem dos dados, devido à sua

capacidade de compreensão semântica e detecção de dados que não se enquadram em uma análise de sentimentos.

Foram obtidos os seguinte resultados no conjunto de teste:


- Acurácia: 0.9447
- F1-score weighted: 0.9456
- Matriz de confusão:



Observa-se que os resultados obtidos são ligeiramente inferiores aos obtidos realizando o fine-tune do BERTimbau realizado em semanas anteriores, que alcançou uma acurácia e F1 de 0.97, mas vale ressaltar que foi utilizado o GPT 3.5 ao invés do GPT 4.0, o que poderia gerar resultados melhores, e não foi realizada nenhuma forma de fine-tune, indicando que ainda há espaço para melhorias e aprimoramentos no uso de LLMs para análise de sentimentos.

Dessa forma, conclui-se que os LLMs ainda são uma tecnologia em desenvolvimento, com novos modelos sendo publicados frequentemente (por exemplo o Bode [2], LLM em português que foi publicado no dia 5 de janeiro de 2024 e por isso não tive tempo para testá-lo). Ou seja, é possível utilizar esses modelos para análise de sentimentos com diferentes abordagens, como a forma zero shot, construindo prompts específicos, realizando fine-tune ou com outros métodos como o uso de RAG (Retrieval-Augmented Generation), o que pode melhorar ainda mais o desempenho desses modelos, porém ainda podem ser complicados de se realizar devido ao alto custo computacional necessário.

O notebook com a implementação descrita pode ser encontrado em:

 Gate 11/01/2024.ipynb

Referências

- [1] Kheiri, K., & Karimi, H. (2023). SentimentGPT: Exploiting GPT for Advanced Sentiment Analysis and its Departure from Current Machine Learning.
- [2] Recogna-NLP. Bode 7B Alpaca PT-BR. Hugging Face, 2023. Disponível em: <https://huggingface.co/recogna-nlp/bode-7b-alpaca-pt-br>. Acesso em: 11 de janeiro de 2024.