

Universidade de São Paulo

ICMC - Instituto de Ciências Matemáticas e Computação

SCC0633/SCC5908 - Processamento de Linguagem Natural

Prof. Dr. Thiago A.S Pardo

PAE: Germano A.Z Jorge

L.O.V.E.

LANGUAGE OVER VARIOUS EMOTIONS

Gabriela Rodrigues do Prado, 11892917

Gabriel Vicente Rodrigues, 11795377

Laís Piai, 15600960

Maria Júlia Soares De Grandi, 12542501

Paloma Botto de Medeiros Serrão, 12689641

SÃO CARLOS

2024

1. Introdução	2
2. Tópico do trabalho	2
3. Corpus selecionado	2
4. Solução Simbólica	3
4.1. Proposta Inicial	3
4.2. Mudanças Realizadas	3
4.3. Implementação	4
4.4. Avaliação de Métricas e Resultados	4
4.5. Exemplos de Execução	5
5. Solução Estatística/Neural	6
5.1. Proposta Inicial	6
5.2. Mudanças Realizadas	6
5.3. Implementação	6
5.4. Avaliação de Métricas e Resultados	7
5.5. Exemplos de Execução	10
6. Instruções para Execução	11
7. Conclusão	12
Referências	13

1. Introdução

O processamento de linguagem natural (PLN) é uma área da inteligência artificial muito em voga atualmente, que visa possibilitar os computadores a processar a linguagem humana (natural) de forma eficiente. Dentre suas várias aplicações, a análise de sentimentos baseada em aspectos destaca-se por conseguir identificar sentimentos específicos relacionados a diferentes aspectos mencionados em textos. Este relatório aborda a aplicação dessa técnica em resenhas de livros, utilizando um corpus selecionado para garantir a precisão e relevância dos resultados.

2. Tópico do trabalho

Este trabalho tem como objetivo a **Análise de Sentimentos Baseada em Aspectos em Resenhas de Livros**. O foco é identificar e classificar sentimentos expressos acerca de diferentes aspectos, como enredo, personagens, escrita, entre outros, em **resenhas de livros**. Essa abordagem permite uma compreensão mais granular e específica dos sentimentos dos leitores, indo além de uma simples classificação positiva ou negativa sobre o livro como um todo.

3. Corpus selecionado

Para a realização deste trabalho, foi selecionado o Corpus ReLi (Freitas, et. al. 2012):

O ReLi é um repositório rico e diversificado de textos em **língua portuguesa**, o qual conta com uma vasta coleção de resenhas de livros. Ele foi escolhido por sua abrangência e qualidade dos dados, garantindo uma base sólida para a análise de sentimentos; sua utilização possibilita o acesso a uma gama muito vasta de opiniões e estilos de escrita. Nesse sentido, uma análise robusta e representativa dos sentimentos expressos nas resenhas literárias do corpus pode ser realizada de forma mais produtiva.

4. Solução Simbólica

4.1. Proposta Inicial

Inicialmente, a proposta do trabalho envolvia o reconhecimento de entidades nomeadas (EN) e a atribuição de suas polaridades (positiva ou negativa). Esta abordagem seguia os seguintes passos:

- Análise Sintática: Extração das EN e seus respectivos modalizadores.
- Consulta ao Léxico de Sentimentos: Extração das polaridades dos modalizadores.
- Atribuição de Valores: -1 para termos negativos e 1 para termos positivos.
- Soma das Polaridades: Verificação se o valor resultante é maior que 0 (positivo) ou menor que 0 (negativo) e atribuição da polaridade à entidade.

4.2. Mudanças Realizadas

Devido à escolha do corpus ReLi (Freitas, et. al. 2012), a tarefa passou a atribuir sentimento a **alvos da opinião**, um termo mais abrangente que entidades nomeadas. Com isso, as novas etapas para criação do sistema simbólico são as seguintes:

- Divisão do Corpus: Separação dos dados em conjuntos de treino e teste.
- Montagem de Léxico de Alvos: Utilização do dataset de treino para montar um léxico de alvos a ser avaliado no teste.
- Verificação no Teste: Identificação de palavras que são alvos a partir de consultas no léxico e extração da vizinhança (3 palavras anteriores e todas as posteriores até encontrar um novo alvo ou o fim da sentença).
- Atribuição de Sentimento ao Alvo: Primeiramente, análise das palavras anteriores ao alvo e consulta ao léxico de sentimentos para extração da polaridade. Se termos forem encontrados no léxico, verifica-se a presença de "negadores" que podem inverter a polaridade das palavras encontradas.
- Análise dos Termos Posteriores: Caso nenhum termo anterior seja encontrado no léxico, realiza-se a mesma análise para os termos posteriores.

Essa nova proposta é baseada no trabalho de Freitas e Vieira (2015), que utiliza léxico e regras para análise de sentimentos na língua portuguesa.

4.3. Implementação

A implementação foi realizada utilizando a linguagem Python. Os dados são extraídos de arquivos TXT e convertidos para um formato de tabela, facilitando o tratamento. Um array de objetos é montado para cada frase seguindo estas etapas:

- Campo "labels": Indica se as palavras da frase são alvos da opinião ou não, extraindo esta informação diretamente da coluna "alvo" do corpus.
- Campo "opinions": Formado usando a coluna "opinião", que indica termos que atribuem polaridade aos alvos da frase. Para cada alvo, as opiniões relacionadas são somadas para obter um valor positivo ou negativo, atribuindo a polaridade ao alvo.
- Campo "palavras": Contém as palavras da frase.
- Campo "frase": Contém as palavras concatenadas com espaços.

O léxico de sentimentos é extraído do arquivo fonte utilizando regex, enquanto o léxico de opositores é escrito manualmente com base no trabalho de Freitas e Vieira (2015). O léxico de alvos é extraído do próprio corpus, usando apenas a parte de treinamento. O pré-processamento dos textos transforma as palavras para minúsculas, extrai seus radicais e elimina stopwords ou palavras com símbolos indesejados. As stopwords e palavras indesejadas são removidas apenas do léxico de alvos extraído do corpus, enquanto as frases são apenas transformadas para minúsculo e radicalizadas. As regras de extração dos alvos e atribuição de sentimentos são implementadas separadamente, facilitando a análise de métricas como a acurácia. Por fim, uma classe é criada para receber exemplos de frases como entrada e produzir uma saída formatada que indica os alvos preditos e os sentimentos relacionados.

4.4. Avaliação de Métricas e Resultados

A etapa de extração dos alvos da opinião obteve uma acurácia de 64.64% no conjunto de teste, o que é um valor razoavelmente bom para uma abordagem tão simples. Por outro lado, a etapa de predição dos sentimentos obteve 22.9% de acurácia. Nessa etapa usamos apenas os alvos preditos corretamente. Esses resultados indicam que, enquanto a solução proposta é relativamente eficaz na identificação dos alvos da opinião, há margem para melhorias na predição de sentimentos.

4.5. Exemplos de Execução

A seguir são apresentados exemplos de saídas da predição usando a classe *Classifier*, que implementa o pipeline completo da abordagem simbólica:

	frase	alvos	sentimentos
0	Este livro é muito bom	[1, 1, 0, 0, 0]	[0, +, 0, 0, 0]
1	Os personagens são ruins	[1, 1, 0, 0]	[0, 0, 0, 0]
2	O livro é bom mas os personagens são pessimos	[1, 1, 0, 0, 0, 1, 1, 0, 0]	[0, +, 0, 0, 0, +, +, 0, 0]

Nos exemplos, os alvos são identificados corretamente, inclusive o pronome demonstrativo “este” e artigos. No entanto, um problema ocorre quando aplicam-se as regras para predizer os sentimentos associados aos alvos da opinião. Por exemplo, na frase "O livro é bom mas os personagens são péssimos", onde temos dois alvos de opinião: "livro" e "personagens" a falha ocorre quando a opinião associada ao "livro" é predita corretamente, mas a opinião associada aos "personagens" não é. Isso se deve ao fato de que a regra aplicada considera, em um primeiro momento, uma janela de três palavras anteriores para determinar a polaridade dos alvos. No exemplo citado, a palavra "bom" está dentro dessa janela anterior ao alvo "personagens" e nesse caso a regra atribui a polaridade positiva associada a "bom" sem considerar a janela subsequente e encontrar “pessimos”.

5. Solução Estatística/Neural

5.1. Proposta Inicial

A proposta inicial foi dividida em dois modelos: um para predição de alvos da opinião e outro para predição de sentimentos direcionados a esses alvos. Ambos os modelos utilizam o BERTimbau, um modelo baseado na arquitetura BERT, pré-treinado com dados em português, para extração de features textuais. Na camada superior, são utilizadas redes MLP para transformar essas features e gerar distribuições de probabilidade para classificação.

5.2. Mudanças Realizadas

Para a tarefa de predição de sentimentos, a proposta inicial criava duas frases de entrada para o modelo BERT: uma enriquecida com “[SEP] (alvo) | POSITIVO” e outra com “[SEP] (alvo) | NEGATIVO”, sendo (alvo) o alvo da predição. Como duas frases eram criadas, havia duas predições diferentes para cada polaridade, e a polaridade com maior logit de saída era utilizada para indicação da resposta.

No entanto, ao utilizar esta estratégia, percebeu-se que a predição de sentimentos para alvos presentes em sentenças com polaridades opostas era comprometida, com a polaridade mais à direita sendo assinalada para todos os alvos. Para solucionar este problema, optou-se por gerar apenas uma frase enriquecida como entrada para o modelo. O novo enriquecimento é “[SEP] Qual a polaridade da palavra (alvo) na frase anterior?”, onde 0 representa polaridade negativa e 1 polaridade positiva. Além disso, foi utilizado um threshold de 0.4 para predizer que um alvo é positivo. Esse valor foi escolhido através da análise de exemplos de saída obtidos com frases com sentimentos opostos.

5.3. Implementação

Para implementar ambos os modelos de predição de alvo e de sentimento, foi utilizada a biblioteca **transformers**, que oferece uma interface para interação com modelos baseados em BERT, além de classes que já implementam arquiteturas para predição relacionadas a tokens e sentenças por completo.

Foram criadas classes que estendem a classe *Dataset*, do **Pytorch**, e *LightningDataModule*, do **Lightning**, para definir a divisão dos dados e como configurá-los para as etapas de treinamento, teste e validação. Também foram desenvolvidas classes que estendem a classe *LightningModule*, que define como as etapas de validação, treinamento e teste ocorrem, quais são as camadas do modelo e como as métricas são extraídas. Essas classes são passadas para o *Trainer*, também do **Lightning**, que gerencia a divisão dos dados, épocas, batches, atualização de pesos, extração de métricas e salvamento automático de checkpoints dos modelos para reutilização sem necessidade de retreinamento.

Por fim, foram criadas as classes *TargetClassifier* e *SentimentClassifier*, as quais utilizam os modelos treinados para predição de alvos e sentimentos em frases novas, que não possuem anotações. Essas classes implementam o método “classify”, que recebe uma lista de frases como entrada e retorna um DataFrame contendo as predições. A classe *TargetClassifier* implementa uma lógica para transformar a predição dos tokens de saída do BERT em predições para as palavras da frase, visto que o tokenizer quebra a sentença em sub-palavras.

5.4. Avaliação de Métricas e Resultados

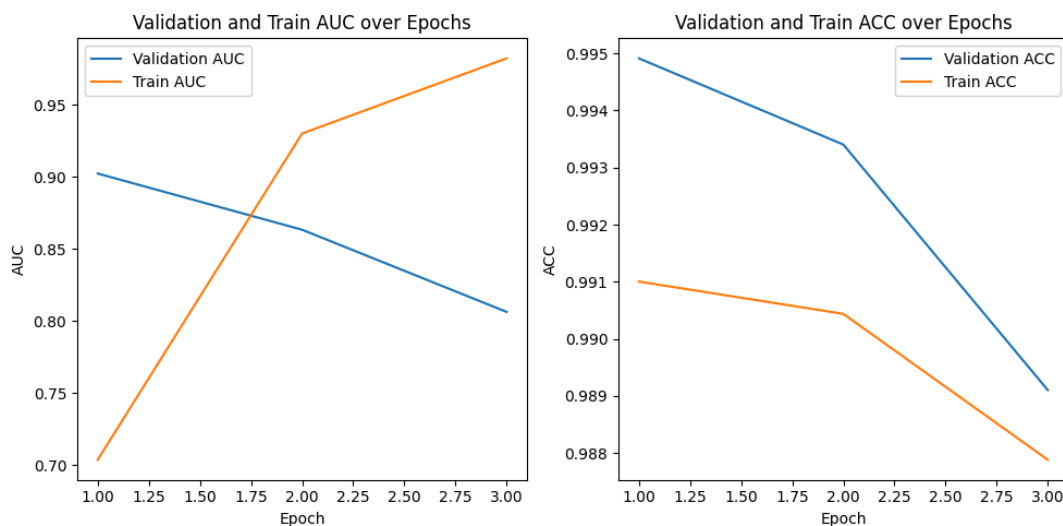
Inicialmente, para uma melhor avaliação das métricas obtidas pelos modelos, é importante considerar que os dados não são balanceados. Portanto, além da clássica ACC (accuracy), estamos utilizando a AUC (Area Under the Curve).

Dito isto, temos dois modelos que foram treinados para este trabalho. O primeiro realiza a predição dos alvos de opinião nas resenhas literárias. Por exemplo, na resenha "Eu adorei o livro", o modelo deve indicar que o alvo é "livro". O segundo modelo realiza a predição de sentimentos atrelados a esse alvo, indicando, por exemplo, que o sentimento em relação ao "livro" é positivo .

Nesse contexto, após o treinamento, coleta de dados e métricas, podemos traçar uma avaliação de desempenho dos modelos:

I. Modelo de Predição de Alvos

Durante os treinamentos, ao decorrer das épocas, foram coletados dados para as métricas AUC e ACC, que podem ser visualizados em formato gráfico:



Neste modelo, nota-se um *overfitting*, desta forma, o modelo gerado após a primeira iteração no dataset foi eleito como o mais apropriado para uso no dataset de teste e verificação de sua robustez.

Levando isto em consideração, temos as métricas obtidas pelo modelo escolhido:

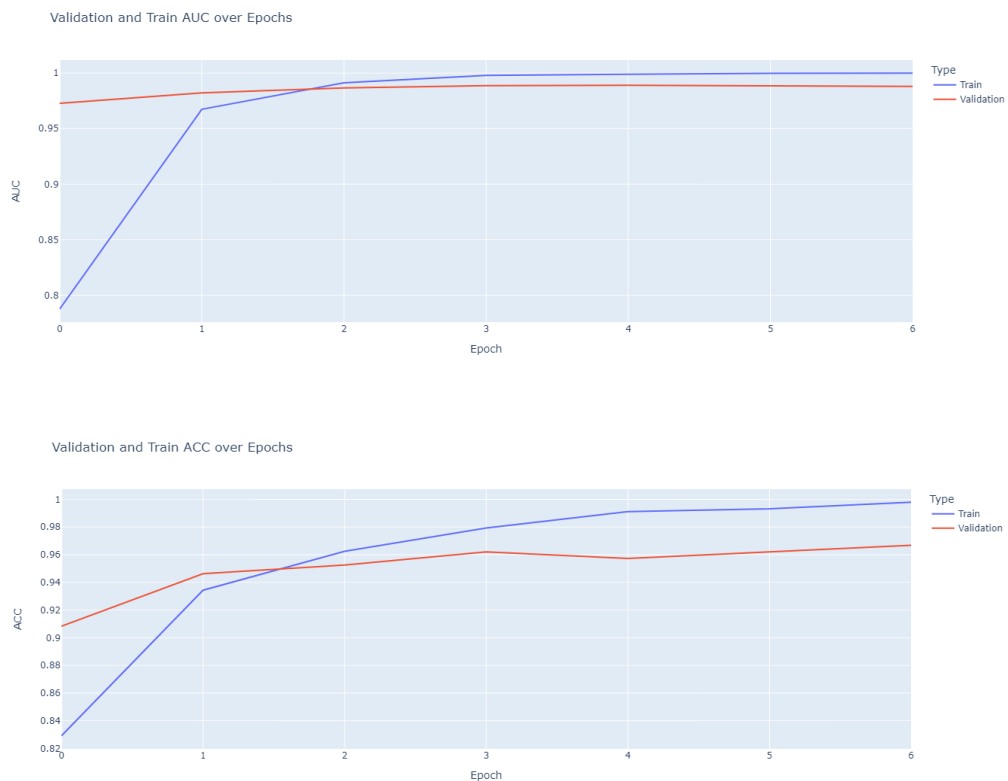
Test metric	DataLoader 0
test_acc	0.9953478574752808
test_auc	0.9106875658035278
test_loss	0.043233152478933334
test_loss_epoch	0.04308860003948212

É observável que o desempenho foi satisfatório, uma vez que as métricas de ACC e AUC foram altas; ACC obteve um valor de 99%, e AUC de cerca de 91%.

Desta forma, podemos concluir que o treinamento do modelo, bem como sua arquitetura e uso foram concluídos de forma satisfatória. O modelo demonstra resultados muito bons na tarefa de prever os alvos de opinião, e, assim, pode ser utilizado em nosso projeto.

II. Modelo de Predição de Sentimentos

Assim como o primeiro modelo, durante o treinamento deste segundo também foram coletados dados para as métricas, que podem ser visualizados nos gráficos a seguir:



É notável que as métricas de ACC e AUC alcançaram níveis muito satisfatórios ao longo das épocas, chegando a 99% no conjunto de treino, e atingindo valores superiores a 95% no conjunto de validação. Os resultados detalhados ao final do treinamento podem ser consultados na tabela abaixo:

Test metric	DataLoader 0
test_acc	0.9542586803436279
test_auc	0.9882827401161194
test_loss	0.17664027214050293
test_loss_epoch	0.1760885864496231

A partir destes dados, pode-se validar que o desempenho obtido com este modelo foi satisfatório, e ele está apto a realizar a tarefa de predição de sentimentos. Desta forma, conclui-se que os resultados são excelentes.

De forma geral, ambos os modelos revelaram um desempenho satisfatório e promissor para este projeto, indicando que ambas as tarefas estão sendo realizadas de maneira excelente e, em conjunto, conferem uma robustez significativa ao sistema de análise de opiniões em resenhas literárias desejado. Com a implementação e treinamento desses modelos, o L.O.V.E. ganha uma ferramenta poderosa para extrair informações mais completas e corretas das resenhas, facilitando a compreensão e a interpretação das opiniões dos leitores sobre os livros analisados.

Dadas as circunstâncias do projeto, os resultados são significativos, e, claro, poderiam ser melhorados com, por exemplo, aumento de poder computacional e mais tempo de treinamento - o limite de tempo de GPU imposto pelo Google Colab mostrou-se uma dificuldade. Mas, em síntese, os resultados encontrados em ambos os modelos foram promissores.

5.5. Exemplos de Execução

	palavras	labels	opinões
0	[O, livro, é, maravilhoso]	[0.0, 1.0, 0.0, 0.0]	[O, +, O, O]
1	[Os, personagens, são, horríveis]	[0.0, 1.0, 0.0, 0.0]	[O, -, O, O]
2	[É, uma, história, cativante]	[0.0, 0.0, 1.0, 0.0]	[O, O, +, O]
3	[Adorei, a, história, mas, a, personagem, prin...]	[0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0]	[O, O, +, O, O, -, -, O, O]
4	[Não, gostei, do, livro, ,, mas, adorei, como,...]	[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...]	[O, O, O, -, O, O, O, O, O, +]

Representação das frases 3 e 4 retiradas da imagem acima:

"Adorei a história mas a personagem principal é fraca"

"Não gostei do livro, mas adorei como os personagens"

Para as execuções dos exemplos acima, foram utilizadas as classes *TargetClassifier* e *SentimentClassifier*. É possível perceber que a coluna de labels determina os alvos a serem analisados, e a coluna de opiniões determina se esses alvos indicados pelas labels são positivos (+) ou negativos (-). No caso de palavras não consideradas alvos, coloca-se 'O' em sua posição.

6. Instruções para Execução

Foi enviado, em conjunto com este relatório, um notebook, de extensão .ipynb, cujo código está escrito em python, é necessário baixar o arquivo .ipynb, e abri-lo. Recomendamos que isto seja feito no ambiente Google Colab.

Agora que ele está aberto, note que há saídas para as células, pois ele foi baixado com os resultados já computados. Caso queira executá-lo novamente siga as instruções abaixo:

Antes da execução, é preciso criar um atalho da seguinte [pasta](#) no seu Drive. Há uma instrução detalhada com print no notebook na primeira célula. Além disso, há uma variável “caminho” que deve condizer com o caminho até o atalho em seu Google Drive.

Após esse passo, há duas possibilidades:

1. Recuperar os dados dos modelos já treinados

Neste caso, na seção “Rodar modelos”, há uma checkbox que seta uma variável booleana. Caso você queira recuperar os dados de checkpoint salvos no drive de execuções passadas, marque esta checkbox. Agora basta executar o notebook inteiro.

Note que, caso não haja arquivos salvos, os treinamentos serão executados ainda que esta checkbox esteja marcada.

2. Treinar os modelos novamente

Já para este caso, inicialmente, recomendamos a conexão ao ambiente da GPU T4 ou superior (caso você disponha de uma assinatura) fornecidos pelo Colab, a fim de diminuir o tempo empreendido na execução.

Após a conexão, na seção “Rodar modelos”, a checkbox supracitada deve estar **desmarcada**. Então, basta executar todas as células do notebook.

7. Conclusão

Durante o desenvolvimento do presente trabalho buscou-se desenvolver a tarefa de Análise de Sentimentos baseada em aspectos em um corpus de resenhas de Livro, ReLi (Freitas, et. al. 2012). Em relação à abordagem simbólica, ela se concentra na previsão de sentimentos através de regras lexicais, porém apresenta limitações significativas. Essa abordagem simplifica o processo ao contar apenas as palavras negativas e positivas nas janelas de contexto anterior e posterior, sem considerar o enquadramento mais amplo das frases, o que pode levar a interpretações erradas.

Além disso, a dependência de léxicos pré-definidos e com poucas entradas é um limitador, pois esses léxicos podem ser incompletos. As regras aplicadas também ignoram a análise sintática e semântica das frases, não considerando a estrutura gramatical nem o significado das palavras no contexto completo. Como possível encaminhamento, a incorporação de técnicas de parsing sintático pode auxiliar na construção de regras que atendam melhor ao objetivo da tarefa. Outra possibilidade é o refinamento das regras de restrição que delimitam os casos aos quais as regras inicialmente propostas se aplicam, aumentando assim a eficácia na identificação dos sentimentos.

Por sua vez, a abordagem neural demonstrou um melhor desempenho na tarefa, obtendo resultados satisfatórios nos dois modelos treinados. É pertinente dizer que o primeiro modelo sofreu *overfitting* e por isso selecionamos parte dele. Também houveram desafios como a dificuldade na extração e preparação dos dados para treinamento, assim como durante o processo de treinamento foi necessário realizar ajustes nos dados disponibilizados ao modelo para corrigir a tendência de receber somente a opinião à direita do alvo. Como futuros encaminhamentos é possível pensar em uma interface mais intuitiva para a disponibilização dos resultados.

Referências

FREITAS, C.; MOTTA, E.; MILIDIÚ, R.; CESAR, J. Vampiro que brilha... rá! Desafios na anotação de opinião em um corpus de resenhas de livros. In: XI ENCONTRO DE LINGUÍSTICA DE CORPUS (ELC 2012), São Paulo, Brasil, 2012.

FREITAS, C. Sobre a construção de um léxico da afetividade para o processamento computacional do português. Revista Brasileira de Linguística Aplicada, [S.L.], v. 13, n. 4, p. 1031-1059, 19 nov. 2013. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1984-63982013005000024>.

FREITAS, L. A. de; VIEIRA, R. Exploring Resources for Sentiment Analysis in Portuguese Language. In: Proceedings of the 2015 Brazilian Conference on Intelligent Systems (BRACIS). Natal: IEEE, 2015. p. 152-156. DOI: 10.1109/BRACIS.2015.52.

XU, H.; LIU, B.; SHU, L.; YU, P. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In: Proceedings of the 2019 Conference of the North American Chapter of The Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota. Association for Computational Linguistics, 2019. p. 2324-2335.