Instructions for Authors of SBC Conferences Papers and Abstracts

1

Abstract. *This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese ("resumo"). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.*

Resumo. *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

1. Introdução

2. Trabalhos Relacionados

3. Material e Métodos

A metodologia fundamenta-se na premissa de que o viés ideológico se manifesta em padrões discursivos e semântico recorrentes. A abordagem proposta processa o conteúdo textual de notícias utilizando modelos pré-treinados para a geração de *embeddings*, os quais são otimizados via *fine-tuning* com as funções de perda *Contrastive Loss* e *Triplet Loss*. Esse refinamento visa maximizar orientações políticas, servindo como entrada para um classificador de aprendizado de máquina.

Conforme ilustrado na Figura 1, o fluxo de trabalho compreende quatro etapas principais:

- **Conjunto de treino e teste**: Coleta baseada em conjunto de dados da literatura correlara;
- **Geração de *embeddings***: *Fine-tuning* de modelos pré-treinados com aprendizagem métrica (*Contrastive* e *Triplet Loss*);
- **Classificação**: Treinamento do modelo de classificação sobre os vetores otimizados;
- **Avaliação da abordagem**: análise do desempenho do sistema e dos resultados obtidos.

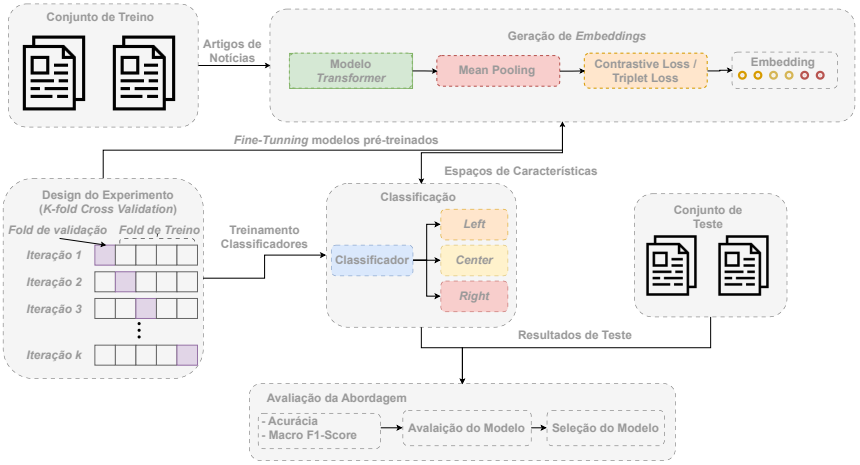

Figura 1. Visão geral do método de detecção de viés ideológico por meio do conteúdo textual de artigos de notícias.

3.1. Dados Experimentais

Para o desenvolvimento deste estudo, utilizou-se o conjunto de dados *Article Bias Prediction* (ABP), proposto por Baly et al, 2020. O *corpus* compreende 30.246 artigos de notícias em língua inglesa, classificados de forma supervisionada nas categorias de viés ideológico: *left*, *center* e *right*. A confiabilidade dos rótulos advém da plataforma AllSides[1], que emprega um processo rigoroso de auditoria – incluindo revisões de terceiros e *feedback* da comunidade – garantindo que anotações reflitam com precisão a orientação política dos textos.

O ABP apresenta elevada diversidade temática, abrangendo desde processos eleitorais até questões sociais complexas. Visando assegurar que os modelos de aprendizado de máquina capturam a ideologia expressa linguisticamente, e não apenas identifiquem a fonte de notícia, os autores realizaram um pré-processamento para remover marcadores explícitos, como nomes de autores e de portais. Essa preocupação é fundamental para garantir a generalização do modelo e a integridade da análise discursiva, evitando que o classificador aprendizada vieses específicos de veículo de imprensa em vez de padrões semânticos.

A robustez da avaliação foi garantida através de dois métodos de particionamento: o *random split* e o *media-bias split*. Na Tabela 1, observa-se a configuração do *media-bias split*, no qual as fontes são segregadas para garantir que o modelo seja testado em veículos não epostos durante o treinamento, mitigando o vazamento de dados. em contraste, o *random-split* permite a sobreposição de fontes entre as partições, mantendo a consitência na distribuição de classes entre treino e validação, como apresentado na Tabela 2. Neste trabalho, a totalidade das amostras do ABP foi empregada em todas as etapas metodológicas, desde a geração de *embeddings* até o treinamento e avaliação final.

3.2. Tarefa de Geração de *Embeddings*

Para a execução da tarefa, a Figura 2 ilustra o fluxo de processamento adotado. O processo inicia-se com a *tokenização* dos artigos de notícias, seguida pela mineração de exem-

[1] `https://www.allsides.com/media-bias/media-bias-rating-methods`

Tabela 1. Estatísticas da partição *media-bias split*.

Treino			Validação			Teste		
Viés	*Total*	*%*	*Viés*	*Total*	*%*	*Viés*	*Total*	*%*
Left	8.861	33,32%	Left	1.640	69,60%	Left	402	30,92%
Center	7.488	28,16%	Center	618	26,23%	Center	299	23,00%
Right	10.241	38,51%	Right	98	4,15%	Right	599	46,07%

Tabela 2. Estatísticas da partição *random split*.

Treino			Validação			Teste		
Viés	*Total*	*%*	*Viés*	*Total*	*%*	*Viés*	*Total*	*%*
Left	9.750	34,84%	Left	2.438	34,84%	Left	402	30,92%
Center	7.988	28,55%	Center	1.998	28,55%	Center	299	23,00%
Right	10.240	36,60%	Right	2.560	36,59%	Right	599	46,07%

plos. Esta etapa é fundamental para selecionar amostras informativas que otimizam a convergência e o aprendizado do modelo.

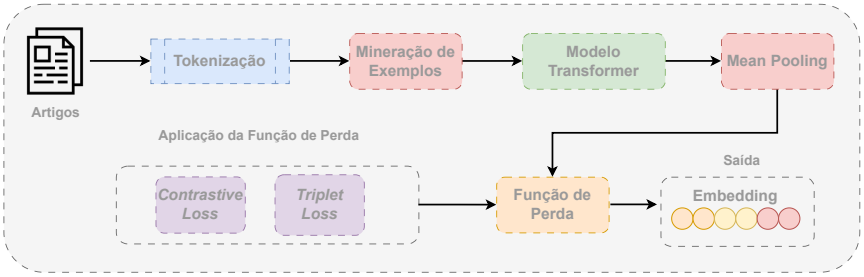

Figura 2. Fluxo de treinamento e geração de características para artigos de notícias.

Conforme delineado na arquitetura apresentada, empregaram-se dois modelos fundamentados em *Bidirectional Encoder Representations from Transformers (BERT)* [Devlin2018], reconhecidos pela eficácia na modelagem de dependências de longo alcance e na extração de relações semânticas granulares [Gao2021]. A seleção recaiu sobre o DistilBERT [Sanh2019] e o DistilRoBERTa [Liu2019], variantes destiladas que preservam a robustez das arquiteturas originais, contudo, apresentam reduções substanciais no custo computacional e nos requisitos de memória.

O modelo *Transformer* processa as sequências de entrada, seguido por uma camada de *Mean Pooling* que consolida as representações em um vetor único. O (*fine-tuning*) é regido por estratégias de aprendizagem métrica[2], utilizando as funções de perda *Contrastive Loss* ou *Triplet Loss*. Tal abordagem assegura que os *embeddings* gerados na saída posicionem instâncias contextualmente similares em regiões próximas do espaço de representação, otimizando a discriminação entre as classes.

No que se refere ao pré-processamento, as *stopwords* foram preservadas, visto

[2]Aprendizagem métrica (ou *metric learning*) refere-se ao uso de algoritmos para aprender uma função de distância que capture a similaridade entre dados.

que a arquitetura BERT demonstra eficácia na extração de nuances contextuais a partir desses elementos. Por fim, o treinamento foi estabelecido com um limite de 100 épocas, utilizando o otimizador Adam com taxa de aprendizado de $0,0001$ e *batch size* de 16. Para mitigar o *overfitting* e assegurar a capacidade de generalização dos modelos, aplicou-se a técnica de *Early Stopping* com paciência de 30 ciclos, monitorando-se a convergência da função de perda no conjunto de validação.

3.2.1. Mecanismos de Aproximação e Distanciamento

Nesta abordagem, os codificadores (DistilBERT e DistilRoBERTa) ajustam os pesos de suas camadas para otimizar a qualidade dos *embeddings* via *Contrastive Loss* e *Triplet Loss*. O objetivo é o aprendizado de representações vetoriais onde instâncias semanticamente similares convirjam no espaço representação, enquanto exemplos dissimilaridades sejam repelidos.

A *Contrastive Loss* é aplicada utilizando a distância Euclidiana sobre pares de exemplos, conforme definido na Equação 1:

$$L = \frac{1}{2}(1-y)D^2 + \frac{1}{2}y\{\max(0, m-D)\}^2 \tag{1}$$

Onde y representa o rótulo binário (0 para similar, 1 para dissimilar), D denota a distância entre as representações e m é a margem de separação.

Complementarmente, a *Triplet Loss* utiliza triplas compostas por uma âncora (a), um exemplo positivo (p) e um negativo (n). O objetivo, expresso na Equação 2, assegura que a distância entre a âncora e o positivo seja inferior à distância entre a âncora e o negativo por uma margem m:

$$L = \max(0, D(a,p) - D(a,n) + m) \tag{2}$$

Para otimizar o aprendizado, empregou-se o *mining* de negativos *semi-hard*. Esses exemplos, que satisfazem a condição $D(a,p) < D(a,n) + m$, fornecem gradientes mais informativos e mitigam o *overfitting* em comparação a negativos *hard* [kertez,2021]. Esse processo refina a capacidade discriminatória do modelo, permitindo que os *embeddings* capturem relações semânticas profundas, como a ideologia de uma notícia, independentemente da fonte de publicação.

3.3. Tarefa de Classificação: Modelos e Parametrizaçao

Após o mapeamento dos *embeddings*, onde a proximidade entre os veotres reflete a similaridade ideológica das notícias. A classificação dos artigos foi realizada por meio de três algoritmos: *K-Nearest Neighbors (KNN)*, *K-Means* e *Multilayer Perceptron (MLP)*. O *KNN* e o *K-Means* foram utilizados para explorar a organização dos dados por vizinhaça e agrupamento, respectivamente. Para o *KNN*, aplicou-se um *grid search* sistemático para otimização de hiperparâmetros, variando o número de vizinhos (k) entre $5, 10, 15, 20, 25$ e 30. Já o *K-Means* foi configurado com o número de *clusters* equivalente às classes presentes no conjunto de dados ABP.

A rede *MLP* foi estruturada com duas camadas densas (512 e 256 neurônios). Adotou-se a função de ativação *ReLU* para garantir um treinamento mais rápido e estável [REFERENCIA], enquanto a camada de saída utilizou a *softmax* para a classificação final. O modelo otimizado com o algoritmo *Adam* [REFERENCIA] e a função de perda *Categorical Cross-Entropy*, escolhas consolidadas na literatura para problemas multiclasse [REFERENCIA GOODFELLOW].

A confiabilidade do experimento foi assegurada pela validacruzada estritificada (5-fold). Esse procedimento garante que a proporção das classes seja mantida em todas as etapas, evitando resultados enviesados e permitindo medir com precisão a capacidade do modelo em classificar novos dados [REFERENCIA]

3.4. Avaliação de Desempenho

O desempenho dos modelos de classificação será avaliado pelas métricas de Acurácia e *Macro F1-score*. A Acurácia (Equação 3) fornece uma medida geral da taxa de acerto para o conjunto de classes C. Complementarmente, o *Macro F1-score* (Equação 4) permite uma avaliação equilibrada entre as classes, mitigando distorções causadas por eventuais desbalanceamento no conjunto de dados.

As métricas são formalmente definidas conforme segue:

$$\text{Acurácia} = \frac{1}{|C|} \sum_{c \in C} \left(\frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c} \right) \tag{3}$$

$$\text{Macro } F1 = \frac{1}{C} \sum_{c \in C} F1_c \tag{4}$$

Em que $F1_c$ representa a média harmônica entre a Precisão (P_c) e a Revocação (R_c) para cada classe:

$$P_c = \frac{TP_c}{TP_c + FP_c}, \quad R_c = \frac{TP_c}{TP_c + FN_c} \tag{5}$$

$$F1_c = 2 \times \frac{P_c \times R_c}{P_c + R_c} \tag{6}$$

Neste contexto, TP, TN, FP e FN representam, respectivamente, os verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos.

A validação da hipótese de pesquisa — de que o discurso textual reflete o viés ideológico — dar-se-á mediante a obtenção de altos índices em ambas as métricas. Espera-se que valores elevados de *Macro F1-score* confirmem a capacidade discriminatória do modelo entre as diferentes vertentes ideológicas, assegurando que o desempenho não seja reflexo de uma classe majoritária no conjunto de dados.

4. Resultados e Discussão

A linguagem Python, com as bibliotecas Numpy, Pandas, Scikit-Learn e PyTorch, foi a ferramenta primária para implementação e avaliação dos modelos. Os experimentos ocorreram em um servidor com processador Intel Xeon W-2235, 128 GB de

RAM e GPU NVIDIA RTX 8000 (48 GB VRAM), visando a aceleração em hardware. Para garantir a reprodutibilidade, o código-fonte, hiperparâmetros e scripts de pré-processamento estão disponíveis em: `https://github.com/jailsonpj/detecting-ideological-bias`.

5. Considerações Finais

6. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991], and [Smith and Jones 1999].

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

Referências

Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd.

Knuth, D. E. (1984). *The TEX Book*. Addison-Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.