Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo

Introdução ao Processamento de Língua Natural

Atividade 3

Classificação de Inferência Textual

Felipe Oliveira Da Silva - 14728582
Giovana Meloni Craveiro - 9791264
João Matheus Siqueira Souza - 10309100
Vinícius Ferreira da Silva - 9745101
Vinícius João de Barros Vanzin - 14696775

Professores:
Profª. Dra. Maria das Graças Volpe Nunes
Prof. Dr. Thiago A. S. Pardo

Maio de 2023

1 Introdução

Este trabalho abordou o entendimento de técnicas, implementação, realização de experimentos e comparação de resultados de diferentes abordagens para o problema de classificação de inferência textual. No problema citado, temos um par de sentenças e o objetivo é construir um modelo capaz de estimar se a segunda sentença é implicada pela primeira.

Para realizar o trabalho, tivemos acesso ao *córpus* ASSIN-2, previamente anotado e já dividido em subconjuntos de treinamento, validação e teste [7]. Para o treinamento, temos um conjunto balanceado de treinamento com 6500 pares de sentença, um de validação com 500 pares e um de teste com 2448 pares. Como temos uma base de dados previamente anotada, o problema é abordado como um problema de classificação.

O restante do documento está organizado da seguinte forma: na Seção 2 temos a apresentação da abordagem de *Bag of Words* (BOW) e suas variações, bem como apresentação dos resultados obtidos. Na Seção 3 temos a apresentação das abordagens e dos resultados de *Word Embeddings*. Na Seção 4, temos a apresentação da abordagem simbólica proposta bem como seus resultados e, finalmente, nas Seções 5 e 6 temos a comparação de resultados entre as diferentes abordagens e a conclusão do trabalho, respectivamente.

2 Abordagem *Bag of Words*

Nesta seção detalhamos os experimentos realizados e os resultados obtidos com técnicas de representação *Bag of Words*.

2.1 Estratégias de Representação

Para realizar os experimentos de *Bag of Words* da atividade, consideramos três diferentes técnicas, aplicando-as isoladamente e também em combinações. As técnicas são:

1. A primeira categoria de técnica é a de representação do texto em formato numérico. Consideramos duas diferentes formas de representações do texto que se valem do formato de uma tabela, possibilitando a utilização de algoritmos de Aprendizado de Máquina. A primeira técnica é a *Bag of Words* tradicional, que transforma textos em dados numéricos (ou tabelas atributo-valor/formato tabular) calculando a frequência com que as palavras ocorrem nos textos. Essa técnica cria uma matriz única para todo o conjunto de dados, na qual em cada linha i temos a contagem do número de vezes que cada palavra ocorreu para a i-ésima sentença da base de dados. A segunda estratégia de representação considerada foi a *Term Frequency-Inverse Document Frequency* (TF-IDF), que calcula a frequência dos termos em um documento e normaliza os valores de frequência pela presença geral daquele termo no corpus, beneficiando termos raros e entendendo-os como importantes;

2. A segunda técnica é o *Principal Component Analysis* (PCA), que se trata de uma técnica bastante utilizada na literatura de Aprendizado de Máquina para redução de dimensionalidade dos conjuntos de treinamento e adequada também para outras utilidades. No cenário do desenvolvimento desta tarefa, em que criamos uma representação matricial na qual cada coluna representa uma possível palavra do *córpus*, a alta dimensionalidade é inexorável. Dessa forma, testamos o PCA como técnica para reduzir o número de variáveis nos conjuntos de dados, simplificando o treinamento dos modelos preditivos;

3. A terceira técnica considerada é uma generalização da forma BoW citada anteriormente, conhecida como estratégia n-gramas. A BoW tradicional é um n-grama em que temos $n = 1$. É possível aumentar o valor de n para que, ao invés de realizarmos a contagem de ocorrências de uma palavra por vez, consideremos um combinado sequencial de mais palavras, podendo assim adicionar um pouco mais de contexto à representação criada. Embora enriqueça o conjunto de dados, tal técnica aumenta significativamente a dimensionalidade do conjunto.

Para construir modelos para a solução do problema de inferência textual, conduzimos experimentos combinando as três técnicas detalhadas acima e utilizando algoritmos de classificação através de Aprendizado Supervisionado. Na Subseção 2.2 detalhamos as configurações dos experimentos realizados. Na Subseção 2.3 apresentamos e discutimos os resultados.

2.2 Configuração dos experimentos

Para todos os experimentos gerados, os algoritmos de Aprendizado de Máquina e as configurações de hiperparâmetros foram mantidos fixos, para que diferenças entre espaços de busca não influenciassem os resultados. Utilizamos três algoritmos de Aprendizado de Máquina para problemas de classificação nos experimentos: Regressão Logística, Random Forests e eXtreme Gradient Boosting (XGBoost). Dessa forma, temos um algoritmo de classificação linear e dois não-lineares, com diferentes vieses de treinamento.

Além dos algoritmos, testamos 10 combinações de hiperparâmetros para geração de modelos nos algoritmos. Para o algoritmo de Regressão Logística, variamos apenas a forma de regularização da regressão, variando entre regularização L1 e L2. Nos algoritmos baseados em *ensembles* de árvores de decisão, *Random Forest* e *XGBoost*, variamos os hiperparâmetros de números de estimadores e altura máxima das árvores de decisão. Todos os parâmetros testados envolvem o controle de possíveis sobre ajustes (*oversamplings*) que poderiam ocorrer devido à alta dimensionalidade do conjunto de dados em suas representações numéricas.

Como detalhado anteriormente na Subseção 2.1 de abordagens, testamos a implementação do algoritmo de *Principal Component Analysis* (PCA) em algumas das abordagens. O PCA contém, por si só, alguns hiperparâmetros que podem ser ajustados. Para nossos testes, consideramos variar apenas o número de componentes no PCA. Portanto, ao configurar o PCA com o número de componentes igual a 3, por exemplo, a matriz de atributos é projetada para um espaço $\mathbb{R}^3$. Quanto menor o número de componentes, maior a probabilidade de mais informação ser perdida, em contrapartida, reduzimos o número de dimensões. A redução de dimensionalidade pode ser útil para o problema, visto que ficamos com 2305 dimensões para o BoW tradicional e 33516 dimensões para a abordagem com *n-grams* de 1 a 3. Testamos seis combinações diferentes de número de componentes para o PCA, sendo $NC = \{3, 4, 5, 10, 100, 500\}$.

Finalmente, para cada combinação de hiperparâmetros fizemos uma validação cruzada com $k = 5$, para que a seleção de melhor modelo sofresse menos com vieses de amostragem. A medida de *F1* foi utilizada para selecionar o melhor modelo dentre as combinações de hiperparâmetros testados para todos os diferentes algoritmos. Tal métrica foi escolhida pois assumimos que não temos pesos diferentes entre erros Falsos Positivos (FP) e Falsos Negativos (FN). Dessa forma, uma métrica que equaliza os pesos para ambos os pesos e penaliza desbalanceamentos de erros entre os dois tipos é adequada como medida de otimização de desempenho. Para a primeira parte da atividade, de implementação de BoW, foram feitos os testes mostrados na Tabela 1.

Foram ajustados ao todo 750 modelos preditivos. Considerando que cada ajuste foi feito com uma validação-cruzada com $k = 5$, totalizaram-se 150 configurações diferentes de modelos, com

abordagens distintas de representação e configurações de hiperparâmetros.

Tabela 1: Características utilizadas em cada estratégia de análise usando *Bag of Words*

Estratégia	Representação Utilizada	PCA?	n-Gramas	Modelos Ajustados
BoW	BoW	Não	Não	50
BoW-PCA	BoW	Sim	Não	300
TF-IDF	TF-IDF	Não	Não	50
BoW-NGram	BoW	Não	$\{1, (1-2), (1-3)\}$	50
BoW-NGram-PCA	BoW	Sim	$\{1, (1-2), (1-3)\}$	300

2.3 Resultados das abordagens *Bag of Words*

A Tabela 2 apresenta os resultados obtidos pelos experimentos no conjunto de testes para todas as abordagens descritas. As linhas destacadas em negrito representam as melhores abordagens para cada um dos conjuntos.

Tabela 2: Resultados obtidos aplicando estratégias baseadas em *Bag of Words*

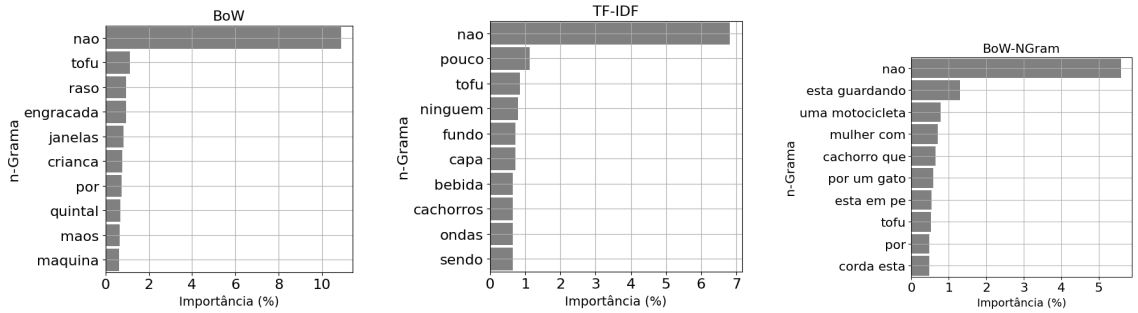
Conjunto	**Abordagem**	**Métricas**			
		F1	**Precisão**	**Revocação**	**Acurácia**
Treinamento	BoW	0.94	0.92	0.95	0.93
	BoW-PCA	0.94	0.95	0.94	0.94
	TF-IDF	0.94	0.93	0.96	0.94
	BoW-NGram	0.94	0.93	0.95	0.94
	BoW-NGram-PCA	**0.98**	**0.99**	**0.98**	**0.98**
Validação	BoW	0.88	0.87	0.89	0.88
	BoW-PCA	**0.90**	**0.92**	**0.88**	**0.90**
	TF-IDF	0.89	0.89	0.89	0.89
	BoW-NGram	0.89	0.88	0.91	0.89
	BoW-NGram-PCA	0.89	0.91	0.87	0.89
Teste	**BoW**	**0.77**	**0.68**	**0.88**	**0.73**
	BoW-PCA	0.69	0.57	0.89	0.61
	TF-IDF	0.74	0.65	0.87	0.70
	BoW-NGram	0.76	0.66	0.88	0.72
	BoW-NGram-PCA	0.68	0.57	0.85	0.61

Seguindo a metodologia de seleção de modelos recomendada pela literatura de Aprendizado de Máquina, teríamos a estratégia escolhida como BoW-PCA utilizando como critério o conjunto de validação e a métrica F1. Entretanto, ao observarmos os resultados nos conjuntos de treinamento e de testes, observamos que o modelo escolhido a partir do conjunto de validação não tem o melhor desempenho nos dois outros conjuntos, sendo apenas o segundo pior desempenho em termos de F1. Entretanto, apresenta o melhor desempenho em termos de Revocação no conjunto de teste.

As abordagens que utilizam PCA como redução de dimensionalidade apresentaram os piores resultados nos conjuntos de teste, embora não tenham tido um desempenho ruim no conjunto de validação. No conjunto de teste, o melhor resultado foi obtido através da aplicação de um BoW tradicional, sem nenhum outro tipo de tratamento.

Para as estratégias que não utilizaram PCA para redução de dimensionalidade, é possível visualizarmos quais n-gramas foram considerados mais importantes para o modelo. Para os modelos lineares, consideramos como variáveis mais importantes aquelas cujos coeficientes relacionados pos-

suem maior valor absoluto. Para os modelos de *ensembles* de árvores de decisão, são tidas como variáveis de maior importância as que possuem, na média, o maior ganho de informação entre todos os estimadores que compõem o *ensemble*. Considerando os dez n-gramas mais importantes para as estratégias, obtemos os gráficos de importância percentual das variáveis preditivas, para cada uma das abordagens, que podem ser visualizados nas Figuras 1.



(a) BoW (b) TF-IDF (c) BoW-NGram

Figura 1: Importância percentual das variáveis preditivas para cada uma das abordagens.

É possível observar que, independente da estratégia utilizada, a palavra "não" é sempre tida como a palavra mais importante. Inclusive com importâncias próximas de 10% em cenários com milhares de variáveis preditivas, o que indica que a presença da palavra "não" é um forte indicativo da presença ou ausência de inferência textual entre as duas sentenças. A palavra "tofu"também aparece como sendo de bastante importância, inclusive na abordagem de n-gramas, com importâncias próximas de 1% em todas as abordagens. Novamente, é bastante notável, visto que são milhares de palavras presentes no corpus e "tofu" é um termo bem menos frequente do que "não" nas sentenças da Língua Portuguesa de forma geral.

Concluindo, testamos também duas formas de pré-processamento dos pares de textos para diminuir a dimensionalidade antes mesmo da modelagem. Primeiramente, tentamos padronizar as palavras, removendo acentos de todas as palavras. O objetivo desse tratamento era que casos como 'automóvel' e 'automovel' não fossem tratados como palavras diferentes no conjunto. A segunda estratégia foi a remoção das *stopwords* de Língua Portuguesa nos textos. Entretanto, em todas as abordagens, a aplicação dessas duas estratégias de pré-processamento ocasionaram numa piora dos resultados para todas as métricas, portanto foram desconsideradas.

3 Abordagem *Word Embeddings*

Nesta Seção, detalhamos os experimentos realizados e os resultados obtidos com técnicas utilizando a representação por *word embeddings*.

3.1 Técnicas

Nesta abordagem com *word embeddings*, realizamos três experimentos com as seguintes técnicas:

1. A primeira estratégia foi utilizar o algoritmo CatBoost [6], um algoritmo da categoria de *gradient boosting* da área de aprendizado de máquina, que permite receber texto em sua representação original como entrada. Como pré-processamento dos dados, aplicamos a normalização

dos textos das colunas de premissa e hipótese para palavras minúsculas e, ao contrário da abordagem BoW, não removemos *stopwords* e concatenamos as duas sentenças separando-as por um token especial "[SEP]". No formato tabular, depois de concatenar as sentenças, criamos uma coluna de *embeddings* que extrai da sentença a sua representação por *word embedding*. A sentença foi representada pela soma dos vetores de *embeddings* individuais de cada palavra. Os *embeddings* que usamos foram pré-treinados por [3] e testamos usando diferentes estratégias como CBow e Skip-gram com diferentes dimensões.

2. A segunda técnica empregou uma rede neural recorrente bidirecional (BRNN) para classificação. As redes bidirecionais estendem as redes recorrentes tradicionais no sentido em que conectam camadas ocultas em direções opostas em relação ao tempo à camada anterior, permitindo acesso a informações dos estados passados e futuros simultaneamente [10]. As BRNN são aplicadas tradicionalmente a tarefas que requerem conhecimento do contexto da entrada, como o processamento de dados sequenciais. No processamento de língua, esta arquitetura têm sido utilizada, por exemplo, nas tarefas de classificação [10], tradução [13], análise de dependência [5] e extração de entidades nomeadas [2].

3. A terceira técnica considerada é de modelos de língua pré-treinados. Utilizamos o modelo BERTimbau [11], um modelo de língua do português brasileiro proveniente da arquitetura de *Transformers* [14] que foi treinado no córpus brWaC [15]. Nossa estratégia foi aplicar uma técnica conhecida na literatura como ajuste fino (*fine-tuning*) [4] que tem como objetivo ajustar o modelo pré-treinado para uma outra tarefa, apropriando-se do conhecimento prévio que o modelo adquiriu no pré-treinamento. O BERTimbau possui duas versões de modelos pré-treinados: a primeira possui 12 camadas de *encoders* e 110 milhões de parâmetros; a segunda versão possui 24 camadas de *encoders* e 335 milhões de parâmetros. Realizamos o ajuste fino nas duas versões e os hiperparâmetros que utilizamos para ambas as versões foram os mesmos.

3.2 Configuração dos experimentos

Para os testes realizados com o algoritmo CatBoost, os modelos foram treinados com os hiperparâmetros padrão, variando-se o tipo de *word embedding* utilizado. Testamos dois modelos, o primeiro com os *embeddings* de [3], fornecendo o texto e os *embeddings* em formato tabular como entrada. Nesse caso, o tipo de *embedding* utilizado foi o *word2vec CBoW* de 100 dimensões. No segundo modelo, a entrada permanece a mesma com texto e *embeddings*, porém utilizando os *embeddings Glove Skip-gram* de 300 dimensões. A escolha dos *embeddings* permite comparar os resultados de uma técnica que captura um conhecimento semântico local (*word2vec*) e global (*Glove*) [3].

Para os experimentos com o modelo BRNN, a arquitetura e hiperparâmetros da rede foram fixados, variando-se somente a dimensão da camada de *embeddings* de acordo com a dimensão do modelo de *embedding* utilizado, a fim de investigar o efeito da escolha de *embeddings* no desempenho do modelo. A arquitetura da rede consiste em três camadas ocultas de células LSTM bidirecionais de, respectivamente, 100, 200 e 100 neurônios cada. Para treinamento do modelo, as premissas e hipóteses foram concatenadas para cada amostra, separadas somente por espaço. Cada sentença foi tratada com as mesmas técnicas de pré-processamento de [3] para maximizar a quantidade de palavras para as quais havia um *embedding* válido. Para a camada de entrada, a dimensão 64 foi escolhida para comportar todas as amostras do conjunto. A camada de *embedding* consiste na matriz de palavras-vetor de *embedding* construída de acordo com o modelo de *embedding* selecionado. Uma vez que a quantidade de palavras únicas do conjunto não era significativa (inferior a três mil tipos),

a matriz de *embeddings* inclui todas as palavras do conjunto. Aplicamos a técnica de *dropout* de 20% dos neurônios para cada camada oculta da rede e treinamento com parada antecipada caso a métrica de acurácia no conjunto de validação não demonstrasse crescimento em 5 épocas. A camada de saída é uma camada densa de dois neurônios (um para cada classe do atributo alvo) com ativação *softmax*. Treinamos o modelo com o algoritmo de otimização *Adam* por, no máximo, 25 épocas com tamanho de *batch* de 128 amostras. No total, foram treinados 21 modelos variando-se o modelo de *embedding* por técnica (*word2vec skip-gram, word2vec CBoW, wang2vec skip-gram, wang2vec CBoW, FastText skip-gram, FastText CBoW e Glove*) e por número de dimensões (50, 300 e 1000).

No modelo com BERTimbau, aplicamos o ajuste fino nas duas versões anteriormente citadas: a versão *base* com 110 milhões de parâmetros e a versão *large* com 335 milhões de parâmetros. Os *embeddings* da versão base possuem 768 dimensões e a versão *large* possui 1024 dimensões. Foi utilizado o *tokenizador* do próprio modelo pré-treinado, com as configurações de sequência máxima de 128 *tokens*, truncando sentenças maiores que esse valor. Configuramos o número máximo de épocas de treinamento como quatro, porém observamos que com o *fine-tuning* convergia em somente três épocas. O tamanho do *batch* foi de 16 para treinamento e 64 para validação. Os demais hiperparâmetros do modelo não foram alterados.

Tabela 3: Características utilizadas em cada estratégia de análise usando *Word Embeddings*

Estratégia	Representação	Embeddings NILC?	Técnica Embedding	Dimensões Embedding
CatBoost	Texto + Embeddings	Sim	Word2Vec - CBow	100
CatBoost	Texto + Embeddings	Sim	Glove - SkipGram	300
LSTM bidirecional	Embeddings	Sim	word2vec, wang2vec, FastText e Glove	50, 300 e 1000
BERTimbau-base	Embeddings	Não	BERT	768
BERTimbau-large	Embeddings	Não	BERT	1024

Ao todo, foram ajustados 2 modelos de CatBoost, 21 modelos de LSTM bidirecional e 2 modelos BERT, com abordagens distintas na utilização de *word embeddings*, como mostrado na Tabela 3.

3.3 Resultados das abordagens com *word embeddings*

A Tabela 4 apresenta os resultados obtidos para cada técnica usando *embeddings*. No caso da abordagem com BRNN, são apresentados somente os dois modelos com o melhor desempenho em termos das métricas F1 e acurácia. As linhas destacadas em negrito representam as melhores abordagens para cada um dos conjuntos.

Os modelos treinados com a técnica CatBoost obtiveram resultados inferiores às demais abordagens (BRNN e BERT). Verificamos que o modelo treinado com os *embeddings* Glove de 300 dimensões obteve resultados melhores que os *embeddings* word2vec de 100 dimensões. Nos conjuntos de validação e teste, os modelos CatBoost demonstraram resultados consideravelmente inferiores ao conjunto de treinamento. Ambos os modelos treinados com *embeddings* word2vec e Glove exibiram um queda nas métricas em aproximadamente 50%. Os modelos não conseguiram generalizar o que aprenderam

Tabela 4: Resultados obtidos aplicando estratégias baseadas em *Word Embeddings*

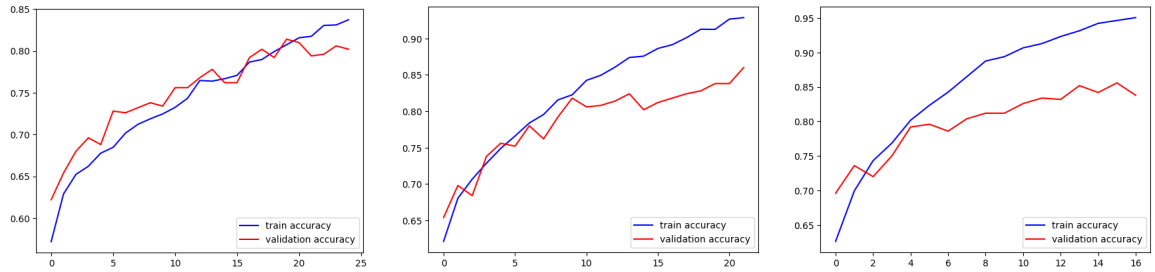
Conjunto	**Abordagem**	**Métricas**			
		F1	**Precisão**	**Revocação**	**Acurácia**
Treinamento	CatBoost (word2vec)	0.95	0.93	0.96	0.95
	CatBoost (Glove)	0.95	0.96	0.93	0.96
	BRNN (word2vec 1000 dim)	0.94	0.94	0.95	0.95
	BRNN (FastText 300 dim)	0.93	0.92	0.94	0.93
	BERTimbau-base	**0.98**	**0.98**	**0.98**	**0.98**
	BERTimbau-large	0.96	0.97	0.96	0.96
Validação	CatBoost (word2vec)	0.49	0.54	0.46	0.63
	CatBoost (Glove)	0.58	0.64	0.53	0.67
	BRNN (word2vec 1000 dim)	0.85	0.77	0.94	0.83
	BRNN (FastText 300 dim)	0.84	0.79	0.90	0.84
	BERTimbau-base	0.94	0.94	0.94	0.94
	BERTimbau-large	**0.96**	**0.96**	**0.96**	**0.96**
Teste	CatBoost (word2vec)	0.47	0.55	0.45	0.49
	CatBoost (Glove)	0.60	0.62	0.55	0.56
	BRNN (word2vec 1000 dim)	0.70	0.63	0.94	0.76
	BRNN (FastText 300 dim)	0.71	0.65	0.90	0.75
	BERTimbau-base	0.88	0.89	0.88	0.88
	BERTimbau-large	**0.89**	**0.90**	**0.89**	**0.89**

com os dados de treinamento para os conjuntos de validação e de teste. A acurácia considerável no conjunto de treinamento indica que houve *overfitting* do modelo às características desse conjunto.

No caso da abordagem BRNN, observamos desempenho semelhante dos modelos treinados para todas as variações de *word embeddings* em diferentes dimensões dos vetores de palavras. Para o conjunto de teste, a acurácia dos modelos manteve-se na faixa 0,73 a 0,76 com medida F1 entre 0,62 e 0,65. O comportamento de alta acurácia nos conjuntos de treinamento e validação, seguida de queda de acurácia no conjunto de teste repetiu-se para todas as combinações de *embeddings*. Em particular, verificamos que a diferença de desempenho entre os conjuntos foi mais acentuada para os modelos com *embeddings* de alta dimensionalidade (300 e 1000). Possivelmente, os efeitos do tamanho reduzido do córpus (milhares de amostras), o treinamento das camadas ocultas do modelo desde o início (sem pré-treino) e a representatividade insuficiente nas amostras (maneiras diferentes de como a implicação pode ser expressada) contribuíram para esse resultado no desempenho dos modelos.

Constatamos que a utilização de um *word embedding* de maior dimensão, no caso dos modelos BRNN, não se traduz diretamente em um aumento nas métricas de desempenho: em somente dois dos sete tipos de *embedding* houve ganho de acurácia com vetores de alta dimensionalidade. Esse resultado é característico de *overfitting*, já que os modelos com vetores de *embedding* maiores têm mais parâmetros a serem aprendidos e, portanto, requerem mais dados de treinamento para evitar que o modelo adapte-se somente às particularidades do conjunto de treinamento e para generalizar adequadamente a dados novos. Para a maioria dos modelos com *embeddings* acima de 50 dimensões, o treinamento foi interrompido pelo mecanismo de parada antecipada, porque a acurácia no conjunto de validação permaneceu inalterada por várias épocas. Essa diferença no processo de treinamento dos modelos é exemplificada na Figura 2, que compara o impacto da dimensão dos vetores word2vec Skip-Gram. É possível notar como a acurácia em validação distancia-se menos da acurácia em treinamento para dimensões menores, isto é, o fenômeno de *overfitting* é menos pronunciado. Além

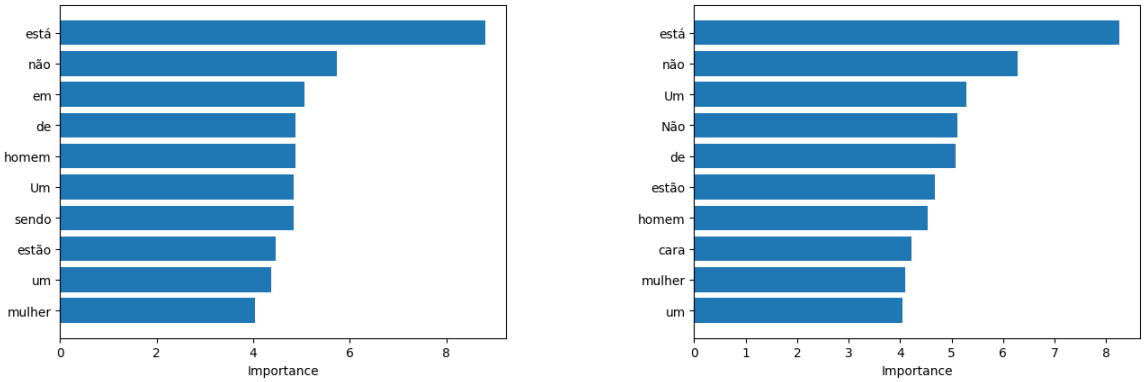
disso, no critério de revocação, os modelos BRNN mostraram alto desempenho, em vezes superando o dos modelos BERT, ou seja, identificam corretamente as amostras positivas (baixa taxa de falsos negativos). É possível que as amostras da classe positiva exibam características em comum na sua distribuição que facilitem a sua classificação, mas que a generalização do modelo não seja suficiente para evitar falsos positivos.



(a) 50 dimensões (b) 300 dimensões (c) 1000 dimensões

Figura 2: Acurácia do modelo BRNN com *embedding* word2vec Skip-Gram durante as épocas de treinamento para os conjuntos de treinamento e de teste.

A importância das palavras para os modelos BRNN foi analisada através da técnica LIME [8], que busca explicar modelos pouco interpretáveis como redes neurais artificiais aprendendo um modelo local interpretável localmente próximo à predição. Para o cálculo do vetor de importâncias, aplicamos o método a mil amostras do conjunto de teste e computamos a importância total de cada palavra como a raiz quadrada da soma dos valores absolutos das importâncias de cada uma das amostras consideradas, conforme [8]. Essa métrica permite avaliar como cada palavra contribui para a classificação do modelo considerando uma parcela considerável do conjunto de dados. A Figura 3 apresenta as dez palavras com maior importância pela métrica LIME para os dois modelos BRNN de maior desempenho na métrica de acurácia (word2vec de 1000 dimensões) e de medida F1 (FastText de 300 dimensões), respectivamente.



(a) word2vec 1000 dimensões (b) FastText 300 dimensões

Figura 3: Importância das 10 palavras mais relevantes nos modelos BRNN com a técnica LIME.

Verificamos que, para ambos os modelos, os dois termos de maior importância para a classificação foram "está"e "não". Com a aplicação da técnica LIME para amostras isoladas, observamos também que os modelos tendem a ter maior confiança (maior probabilidade na classe predita) conforme a quantidade de palavras repetidas na premissa e na hipótese aumentam. Quando as duas sentenças têm baixa intersecção no conjunto de palavras usadas, a confiança dos modelos na predição foi

reduzida. Isto sugere que os modelos BRNN consideram a repetição das palavras na premissa e na hipótese um indicativo da implicação, o que é corroborado pela importância dada ao termo "está", que aparece em uma fração considerável dos pares de frases nas locuções verbais. O termo "não", por outro lado, tem a função oposta, como indicativo da falta de implicação entre as duas sentenças.

No conjunto de treinamento, o modelo que obteve o melhor resultado foi o BERTimbau-base, com 0.98 nas quatro métricas de avaliação. Os resultados foram superiores à versão *large* que tem três vezes mais parâmetros treináveis. Em contrapartida, o modelo BERTimbau-large manteve resultados quase idênticos entre os conjuntos de treinamento e de validação, demonstrando generalização adequada para dados novos. No conjunto de teste, o BERTimbau-large predominou como o melhor modelo, atingindo uma medida F1 de 0.89, que supera os resultados do BERT-multilingual e RoBERTa [9] na tarefa, e fica a um ponto percentual do atual estado da arte na tarefa de [7].

Além dos testes de desempenho, investigamos os resultados dos modelos BERT, na tentativa de atribuir interpretações aos resultados desses modelos de aprendizado profundo que são considerados caixas pretas [1]. Para os modelos BERT, selecionamos amostras aleatórias do conjunto de teste e avaliamos novos exemplos para entender quais palavras do texto tem uma importância maior para o modelo na classificação. Aplicamos o método *integrated gradients* (gradientes integrados) [12], que faz uso dos gradientes das redes neurais para, dado uma entrada de texto, atribuir na camada final qual palavra da sentença tem maior contribuição para a decisão. O objetivo foi entender para quais palavras de uma sentença o modelo atribuiu um peso maior para a classificação. O modelo testado foi a versão *large*. A seguir, apresentamos exemplos de fenômenos observados com a técnica de explicabilidade.

- Premissa: As pessoas não estão andando na estrada ao lado de uma bela cachoeira.
- Hipótese: Uma cachoeira está fluindo em uma piscina rasa.
- Classe real: Sem implicação (*None*)
- Classe predita: Sem implicação (*None*)

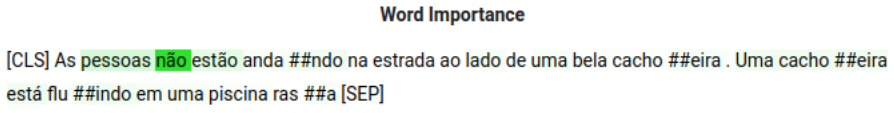


Figura 4: Importância das palavras de negação em uma sentença de acordo com o modelo BERT

No exemplo acima, o modelo atribui um peso maior para a palavra "não". Assim como nos resultados das demais técnicas, o termo "não"foi o que teve maior importância para o modelo classificar um determinado par de sentenças como *None* (Sem implicação). Grande parte dos dados que são classificados como "Sem implicação"possuem uma frase que apresenta negação e o modelo foi capaz de capturar esse padrão.

- Premissa: O garoto está sendo treinado em artes marciais.
- Hipótese: O menino está praticando artes marciais.
- Classe real: Tem implicação (Entailment)

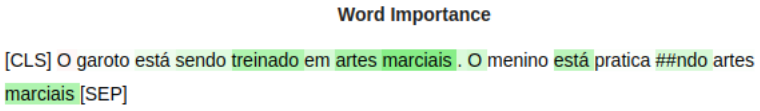


Figura 5

- Classe predita: Tem implicação (Entailment)

Para os casos de que a classificação indicou implicação (*Entailment*), há uma certa variância para quais palavras possuem uma importância maior. Ao contrário do caso de frases de negação como no exemplo anterior, verificamos que o modelo atribui importância a palavras de múltiplas classes. No exemplo o modelo atribuiu maior contribuição para as palavras "treinado"(verbo), "artes"(substantivo) e "marciais"(adjetivo). É possível que o modelo esteja considerando, assim como o modelo BRNN, termos que aparecem repetidamente entre as frases de premissa e hipótese.

4 Abordagem Simbólica

Nesta seção, detalha-se a construção da abordagem simbólica, para a qual ambicionou-se utilizar elementos do raciocínio humano para implementar regras capazes de executar a tarefa de definir se há relação de inferência ou não entre as sentenças.

Partiu-se das diretrizes de anotação do ASSIN-2 [7], o banco de dados utilizado. Tais diretrizes não especificam detalhadamente o raciocínio que deve ser utilizado pela pessoa anotadora, mas fornece certas explicações úteis para elucidá-lo.

O documento utiliza os termos "senhora" e "mulher" e instrui que "senhora" implica necessariamente em uma "mulher". Tal exemplo indica que o subconjunto de "senhoras"está contido no conjunto de "mulheres", ou seja, em uma frase que é inferida por outra, os referentes podem ser indicados com termos mais gerais que na frase original. Para considerar esse critério na abordagem simbólica para casos em que tal grau de especificidade esteja contido na própria palavra utilizada, seria necessário considerar a semântica de cada termo, o que não foi ambicionado nesta abordagem, para sua simplificação. Contudo, para abranger este critério de que a frase inferida pode ser mais geral do que a frase de referência, considera-se que a frase de referência pode conter mais palavras que especifiquem o cenário. Exemplo: "Um cachorro corre alegremente pela areia da praia em um lindo dia de sol" pode inferir "Um cachorro corre na praia". Esta regra não é desprovida de falhas, pois expressões de diferentes tamanhos podem possuir significados semelhantes, por exemplo: "ao ar livre" pode inferir "do lado de fora" e vice-versa, sem que o tamanho das frases impacte no grau de especificidade. Contudo, supôs-se que na maioria dos casos, esta regra funcionaria e, portanto, implementou-se como critério para a classificação de acarretamento que a sentença A deve ter comprimento maior do que a sentença B.

Outro elemento crucial para que uma sentença acarrete outra é que a mensagem trazida por cada uma delas refira-se aos mesmos elementos extra-linguísticos. Como, nesta abordagem, não há formas de consultar a semântica dos termos utilizados, a única maneira possível de verificar tal característica é comparando os termos das duas sentenças em busca de palavras iguais ou mesmo palavras que contenham muitos caracteres iguais. Isto porque as palavras em si podem diferir morfologicamente sem que a relação de inferência textual deixe de existir. Assim, o segundo critério para a classificação de acarretamento é que a taxa de semelhança das sentenças atinja no mínimo cinquenta por cento.

Para realizar este cálculo, pensou-se que a sentença A pode ser mais específica que a sentença B, e poderia portanto conter termos que não estão presentes e não precisam estar presentes na sentença B para que haja uma relação de inferência textual de A para B. Assim, faz-se uma lista com as palavras comuns entre as sentenças A e B. A seguir, calcula-se o comprimento desta lista dividido pelo comprimento da sentença B e o valor resultante é considerado a taxa de similaridade, que deve ser igual ou superior a 0.5.

Estes dois únicos critérios foram as regras iniciais da abordagem. Contudo, analisando as importâncias aprendidas pelas abordagens que utilizam aprendizado de máquina, percebeu-se que a presença do termo "não" nas sentenças era relevante para determinação da relação de inferência textual entre elas. Tal aspecto também seria observado pela mente humana ao executar a classificação e, portanto, foi abrangido na última regra da abordagem. Esta verifica se a palavra "não" está presente na sentença A, ao mesmo tempo que está ausente na sentença B, ou vice-versa. Se for o caso, o algoritmo determina que não há relação de inferência textual. Isto porque a palavra "não" indica negação. Assim, se as duas sentenças forem muito parecidas e somente uma das sentenças contém o termo "não", e portanto nega sua mensagem, esta significará o oposto da outra sentença em algum grau. Por exemplo, a sentença "Não há um cachorro correndo na praia" possui o termo "não" em posição onde nega a sentença toda e na sentença "Um cachorro está correndo, mas não na praia" o termo "não" está posicionado de maneira que nega somente o local, mas de qualquer forma não pode haver relação de inferência textual entre nenhuma das duas com a sentença "Um cachorro está correndo na praia".

Experimentou-se também adicionar uma regra análoga à utilizada para o termo "não", mas com os termos "nenhum"e "ninguém", mas tais testes não melhoraram a performance do algoritmo e portanto, a regra foi removida.

A Tabela 5 apresenta os resultados obtidos pelos experimentos no conjunto de testes para a abordagem simbólica com as regras descritas.

Tabela 5: Resultados obtidos aplicando uma estratégia simbólica

Conjunto	**Abordagem**	**Métricas**			
		F1	**Precisão**	**Revocação**	**Acurácia**
Treinamento	Simbólica	0.70	0.75	0.67	0.72
Validação	Simbólica	0.72	0.75	0.69	0.73
Teste	Simbólica	0.69	0.74	0.65	0.71

Sob a perspectiva quantitativa, considera-se que os resultados, todos entre 0.65 e 0.75, foram muito positivos, principalmente dada a simplicidade das regras aplicadas. Contudo, é essencial ter cuidado na interpretação de tais resultados, pois refletem somente a performance das regras no conjunto de dados utilizado. É evidente que esta abordagem tem muitas limitações, principalmente por não considerar semântica. Este método é incapaz de identificar relações de sinonímia e hiperonímia entre termos, características cruciais para que a mente humana defina se há acarretamento entre duas sentenças.

A partir de uma análise qualitativa, notou-se que a abordagem poderia beneficiar-se da técnica de lematização, pois houve casos em que havia relação de acarretamento e o algoritmo simbólico não foi capaz de identificá-la pois os termos continham diferenças morfológicas. Por exemplo, classificou-se que "batatas estão sendo fatiadas por um homem" não acarreta "O homem está fatiando a batata". Para a resolução deste tipo de caso, também é possível experimentar com diferentes formas de

calcular a taxa de similaridade entre as duas sentenças, com ou sem lematização. Seria interessante também que tais técnicas considerassem a ordem das palavras nas sentenças.

Por fim, para uma abordagem simbólica que almeje assemelhar-se de fato ao raciocínio humano, é essencial que considere a semântica dos termos e verifique se as palavras utilizadas nas sentenças possuem relação de sinonímia ou de hiperonímia.

5 Comparações entre Abordagens

Analisando os resultados de cada uma das abordagens propostas na atividade, podemos identificar diferentes características que beneficiam a escolha de uma ou outra em função da aplicação de interesse. Na Tabela 6, sumarizamos os resultados do método campeão considerando o parâmetro **F1** no conjunto de **Teste** para cada uma das abordagens (*Bag of Words*, *Word Embeddings* e Solução Simbólica).

Tabela 6: Resultados obtidos aplicando estratégias baseadas em *Bag of Words*

Conjunto	**Abordagem**	**Métricas**			
		F1	**Precisão**	**Revocação**	**Acurácia**
Treinamento	BOW	0.94	0.92	0.95	0.93
	BERTimbau-large	0.96	0.97	0.96	0.96
	Simbólica	0.70	0.75	0.67	0.72
Validação	BOW	0.88	0.87	0.89	0.88
	BERTimbau-large	0.96	0.96	0.96	0.96
	Simbólica	0.72	0.75	0.69	0.73
Teste	BOW	0.77	0.68	0.88	0.73
	BERTimbau-large	0.89	0.90	0.89	0.89
	Simbólica	0.69	0.74	0.65	0.71

Pode-se observar que o modelo *BERTimbau-large* apresentou os melhores resultados nas métricas consideradas, $F1 = 0.89$ para o conjunto de Teste, o que era esperado em função das características desse modelo. No entanto, o *BERTimbau-large* possui 335 milhões de parâmetros, como comentado em 3. Embora consiga gerar resultados superiores, demanda uma infraestrutura de treinamento bastante complexa e inacessível para a maioria dos desenvolvedores/pesquisadores da área, além de exigir recursos computacionais às vezes não coerentes com uma aplicação de interesse. Em contrapartida, o modelo baseado em *Bag of Words* - sem nenhuma sofisticação adicional (como o TF-IDF ou redução de dimensionalidade por PCA), apresentou o indicador $F1 = 0.77$ para o conjunto de Teste. Comparando a *Revocação*, o *BERTimbau-large* ultrapassa o BoW por apenas um centésimo, porém com um custo computacional imensamente maior para possibilitar esse "ganho". Olhando para a Solução Simbólica, embora apresente valores consideravelmente menores nas métricas de performance, também exige consideravelmente menos recursos para funcionar. Além disso, comparando os indicadores de Precisão e Acurácia para o conjunto de Teste, a solução Simbólica apresenta pouca distância do BoW.

Além das comparações através das métricas de performance, realizamos uma breve análise sobre casos de erros de cada uma das abordagens, identificando possíveis padrões de 'entendimento' de cada técnica. Abaixo, exemplificamos 4 casos de erro, dois que não apresentavam *entailments* e outros dois que apresentavam.

- **Caso sem *entailment* 1**

  [t]: *um cachorro está nadando atrás da bola*

  [h]: *um cachorro está latindo para uma bola*

  - *Bag of Words:* ENTAILMENT

  - *BERT:* NONE

  - Simbólica: ENTAILMENT

  Neste caso, observa-se uma estrutura da premissa e hipótese muito similar, sendo que a não ocorrência de *entailment* se dá exclusivamente pelo significado de *nadando* e *latindo*. Apenas a técnica usando BERT foi capaz de acertar este caso.

- **Caso sem *entailment* 2**

  [t]: *um grande cachorro branco está correndo através do gramado*

  [h]: *um pequenino cachorro branco está correndo através do gramado*

  - *Bag of Words:* ENTAILMENT

  - *BERT:* ENTAILMENT

  - Simbólica: ENTAILMENT

  Assim como no caso anterior, esse par de sentenças possui uma estrutura similar, diferindo apenas por uma parte da caracterização do cachorro (grande e pequenino). Todos os métodos classificaram como *entailment*, não conseguindo capturar que o tamanho do cachorro destoa entre as sentenças.

- **Caso com *entailment* 1**

  [t]: *um palhaço está cantando no palco e pessoas estão dançando*

  [h]: *uma pessoa fantasiada de palhaço está cantando*

  - *Bag of Words:* NONE

  - *BERT:* NONE

  - Simbólica: NONE

  Diferentemente dos casos anteriores, esse par de sentenças não possui uma estrutura similar entre si. Não há oposição de sentidos entre as frases, porém parte da informação da premissa é omitida na hipótese. Nenhuma das 3 abordagens conseguiu classificar de maneira correta este exemplo.

- **Caso com *entailment* 2**

  [t]: *o cachorro preto está carregando uma bola azul e branca na boca*

  [h]: *um animal está levando um brinquedo colorido em sua boca*

  - *Bag of Words:* NONE

  - *BERT:* ENTAILMENT

  - Simbólica: NONE

  Neste caso, a hipótese substitui algumas palavras por hiperônimos, embora mantenha a estrutura sintática similar. Apenas o modelo BERT foi capaz de capturar as similaridades dos significados entre os pares de hiperônimos/hipônimos.

Esses exemplos citados mostram alguns casos em que o significado das palavras era fundamental para a classificação do *entailment*. Nesses casos, era esperado que a solução simbólica e o *Bag of Words* tivessem resultados inferiores, uma vez que essas abordagens não levam em consideração possíveis extrações de significado. O BERT também apresentou falhas, tanto no caso sem *entailment* 2 - no qual o modelo não conseguiu dar o devido peso à informação de tamanho (grande e pequenino); quanto no caso com *entailment* 1 - onde havia variação na quantidade de informação entre as sentenças.

Dado esses exemplos e também as demais discussões e análises geradas no trabalho, conclui-se que, assim como em inúmeras áreas da ciência, não existe um modelo campeão sempre. Cada aplicação terá *trade-offs* que devem ser avaliados individualmente.

O problema proposto na atividade, por exemplo, é bastante complexo, podendo ter aplicações nas mais variadas áreas de atividade humana. Para o escopo da atividade proposta, considerando a base de dados ASSIN2, a utilização de um derivado do modelo *BERT* apresentou bons resultados. Para outras aplicações, com outros requisitos e referências, outras análises deveriam ser realizadas para avaliar soluções pertinentes.

6 Conclusão

Neste trabalho foi possível explorar um pouco da área de inferência textual, analisando e comparando diferentes métodos para a compreensão e processamento de texto. Analisamos as técnicas baseadas em *Bag of Words* e *Word Embeddings*, e observamos, primeiramente, resultados muito interessantes que podem ser aplicados em inúmeras outras tarefas de processamento de língua natural, mas também as vantagens e limitações de cada um dos métodos.

Em relação ao *BOW*, trata-se de um método mais simples, porém eficiente. O funcionamento do método baseia-se na frequência de ocorrência de palavras. Por isso, este método tende a não ser tão eficaz na extração da semântica de um texto, pois não leva em consideração a relação entre as palavras dentro da frase. Embora existam técnicas que potencializem o método, como o uso de n-gramas (que utilizamos em nossas análises), o *BOW* tem como ponto forte principal a eficiência e rapidez para implementação.

Quanto às técnicas envolvendo *Word Embeddings*, nota-se mais recursos envolvendo semântica. A própria similaridade entre os vetores de palavras de significados próximos já é um indicativo deste fato. Ou seja, trata-se de uma representação mais robusta da informação, o que permite que os algoritmos de aprendizado de máquina consigam obter melhor performance na tarefa. No entanto, a quantidade adicionada de informação nos *Embeddings* também traz requisitos computacionais mais elevados e, principalmente, uma demanda por mais dados para que a representação da informação seja significativa.

Sob a análise das importâncias utilizadas, palavras como "tofu"e "está"se destacam, de modo que as abordagens de aprendizado de máquina revelam que os critérios utilizados não são coerentes com a proposta de identificar relação de inferência textual entre duas sentenças e sim de prever a presença de relações de acarretamento especificamente para a base de dados na qual foram treinadas.

Já a abordagem simbólica foi baseada em três regras condicionais e tem complexidade baixíssima. Apresentou uma performance relativamente próxima às demais, e possui critérios mais relevantes para sua decisão. É possível que explorá-la mais a fundo, realizando um ajuste fino sobre seus parâmetros, incluindo semântica e adicionando mais regras, seja um caminho de chegar aos resultados quantitativos observados nas outras abordagens, ao passo que mantém o ganho de simplicidade.

Portanto, o que fica claro é que a escolha de um método adequado depende dos requisitos

e motivação da tarefa proposta. Com as abordagens recentes, como o caso do modelo BERT, empregado neste trabalho, ou mesmo o ChatGPT, que tem ganhado cada vez mais visibilidade nos últimos tempos, pode ser tentador optar por resolver todas as tarefas com o que há de mais moderno. No entanto, não podemos esquecer de pesar as vantagens e desvantagens de cada método e considerar também as restrições de demandas específicas de cada projeto.

Referências

[1] Vanessa Buhrmester, David Münch e Michael Arens. *Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey*. 2019. arXiv: 1911.12116 [cs.AI].

[2] Franck Dernoncourt, Ji Young Lee e Peter Szolovits. «NeuroNER: an easy-to-use program for named-entity recognition based on neural networks». Em: *arXiv preprint arXiv:1705.05487* (2017).

[3] Nathan S. Hartmann et al. «Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks». Em: *arXiv preprint arXiv:1708.06025* (2017). URL: https://arxiv.org/abs/1708.06025.

[4] Jeremy Howard e Sebastian Ruder. «Universal Language Model Fine-tuning for Text Classification». Em: *arXiv preprint arXiv:1801.06146* (2018).

[5] Eliyahu Kiperwasser e Yoav Goldberg. «Simple and accurate dependency parsing using bidirectional LSTM feature representations». Em: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 313–327.

[6] Liudmila Prokhorenkova et al. *CatBoost: unbiased boosting with categorical features*. 2019. arXiv: 1706.09516 [cs.LG].

[7] Livy Real, Erick Fonseca e Hugo Gonçalo Oliveira. «The ASSIN 2 Shared Task: A Quick Overview». Em: fev. de 2020, pp. 406–412. ISBN: 978-3-030-41504-4. DOI: 10.1007/978-3-030-41505-1_39.

[8] Marco Tulio Ribeiro, Sameer Singh e Carlos Guestrin. «"Why should i trust you?"Explaining the predictions of any classifier». Em: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[9] Ruan Rodrigues et al. «Multilingual Transformer Ensembles for Portuguese Natural Language Tasks». Em: mar. de 2020.

[10] Mike Schuster e Kuldip K Paliwal. «Bidirectional recurrent neural networks». Em: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[11] Fábio Souza, Rodrigo Nogueira e Roberto Lotufo. «BERTimbau: Pretrained BERT Models for Brazilian Portuguese». Em: *Intelligent Systems*. Ed. por Ricardo Cerri e Ronaldo C. Prati. Cham: Springer International Publishing, 2020, pp. 403–417. ISBN: 978-3-030-61377-8.

[12] Mukund Sundararajan, Ankur Taly e Qiqi Yan. *Axiomatic Attribution for Deep Networks*. 2017. arXiv: 1703.01365 [cs.LG].

[13] Martin Sundermeyer et al. «Translation modeling with bidirectional recurrent neural networks». Em: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 14–25.

[14] Ashish Vaswani et al. «Attention Is All You Need». Em: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. 2017, pp. 5998–6008.

[15] Jorge A Wagner Filho et al. «The brwac corpus: A new open resource for brazilian portuguese». Em: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.