



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Arquitetura LSTM para classificação de discursos de ódio cross-lingual Inglês-PtBR

Dissertação de Mestrado

Thiago Dias Bispo



São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Thiago Dias Bispo

**Arquitetura LSTM para classificação de discursos de ódio
cross-lingual Inglês-PtBR**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Mestrado em Computação Inteligente.

Orientador(a): Hendrik Macedo

São Cristóvão – Sergipe

2018

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE

B622a Bispo, Thiago Dias
Arquitetura LSTM para classificação de discursos de ódio cross-lingual Inglês-PtBR / Thiago Dias Bispo ; orientador Hendrik Macedo. – São Cristóvão, 2018.
72 f. : il.

Dissertação (mestrado em Ciência da Computação) –
Universidade Federal de Sergipe, 2018.

1. Processamento de linguagem natural (Computação). 2. Redes neurais (Computação). 3. Memória de longo prazo. 4. Redes sociais. 5. Discurso de ódio na Internet. I. Macedo, Hendrik Teixeira, orient. II. Título.

CDU 004.8

Thiago Dias Bispo

**Arquitetura LSTM para classificação de discursos de ódio
cross-lingual Inglês-PtBR**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Mestrado em Computação Inteligente.

Hendrik Macedo

Orientador

Convidado

Vlória Celia

Professor

Carlos Alberto Estombelo Montesco

São Cristóvão – Sergipe

2018

Dedico este trabalho à minha mãe que sempre me apoiou em toda a minha vida, inclusive na minha jornada enquanto estudante.

Agradecimentos

Agradeço a todos os meus professores na Universidade, que direta ou indiretamente me inspiraram para chegar onde cheguei.

Em especial, agradeço a meu orientador prof. Hendrik, que, com seu brilhantismo e paciência, me inspirou em toda minha vida acadêmica.

Agradeço à minha namorada pelo apoio incondicional durante todo o trajeto do Mestrado, ela foi meu combustível nos momentos em que nada mais conseguia aliviar o cansaço da rotina de estudos.

Resumo

Uma das consequências da popularização do acesso à Internet é a disseminação de insultos e mensagens discriminatórias, os chamados discursos de ódio (do inglês, *hatespeech*). São comentários que visam discriminar alguém ou um conjunto de pessoas por pertencerem a um certo grupo, normalmente minoritário, ou por possuírem alguma característica também comum a outras pessoas. O combate aos discursos de ódio é uma demanda crescente na vida real e virtual pois eles afetam profundamente a dignidade de suas vítimas.

Detecção de discursos de ódio é uma tarefa difícil porque, além da linguagem natural ser inerentemente ambígua, ela exige certo nível de compreensão de sua estrutura linguística. Em muitos discursos, a discriminação não acontece de forma explícita ou com expressões típicas: é preciso ter conhecimento de mundo para reconhecê-las. Além disso, algumas vezes é necessário entender o contexto da frase para perceber seu teor odioso. O sarcasmo é outro desafio enorme (até para humanos) uma vez que sua presença exige conhecimento da comunidade e potencialmente do usuário responsável pelo comentário para o entendimento de sua intenção.

Diversas abordagens foram propostas para reconhecimento do *hatespeech*. Muitos autores consideram *N-Grams*, dentre os quais aqueles baseados em caracteres mostram-se mais efetivos que aqueles baseados em palavras. Combinadas ou não aos *N-Grams*, *features* léxicas também foram estudadas, como a presença ou não de palavras negativas, classes ou expressões indicativas de insulto, sinais de pontuação, repetições de letras, presença de emojis etc. *Features* linguísticas mostraram-se ineficientes quando utilizadas isoladamente, como as *POS tag*, e a relação entre os termos da árvore de dependência resultante da análise sintática. Recentemente, a abordagem mais bem sucedida usou uma rede neural para criar uma representação distribuída das sentenças presentes em um corpus de discursos de ódio, indicando que o treinamento de *word embeddings* é um caminho promissor para a área.

A língua afeta drasticamente as tarefas de Processamento de Linguagem Natural (PLN), uma vez que a maioria das palavras, se não todas, são diferentes de uma língua para outra, além de sua sintaxe, morfologia e construções linguísticas. Por esta razão, os trabalhos em língua inglesa não são diretamente aplicáveis em *corpora* de língua portuguesa, por exemplo. Além disso, *corpora* em português para discursos de ódio são raros, fazendo com que pesquisadores da área precisem realizar todo o trabalho de construção.

Nessa dissertação, foi estudado o uso de um modelo *deep cross-lingual Long Short-Term Memory* (LSTM), treinado com um *dataset* de discursos de ódio traduzido do Inglês de duas diferentes maneiras, pré-processado e vetorizado com variadas estratégias que foram representadas em 24 cenários. As principais abordagens adotadas consideraram: o treinamento de *embeddings* através

de vetores de índices de palavras (técnica Estado da Arte), vetores TFIDF, vetores *N-Grams*, com ou sem vocabulário *GloVe*, testados com o *dataset* construído e rotulado neste trabalho e com outro disponível em português. O processo invertido também foi experimentado: traduzimos o nosso *corpus* para o inglês e comparamos o desempenho com sua versão original. Com os *embeddings* resultantes do processo de treinamento em cada cenário, usamos uma *Gradient Boosting Decision Tree* (GBDT) como forma de melhorar a classificação e, de fato, os resultados obtidos com a LSTM foram melhorados em muitos cenários.

Alcançamos precisão de até 70% nos experimentos usando o modelo treinado com o *corpus* em Inglês e nosso *dataset* traduzido para esta língua. Em outros, técnicas tradicionais e bem sucedidas como vetores TFIDF associados à uma LSTM não se mostraram suficientes. Duas importantes contribuições deste trabalho são: (i) proposta de uma abordagem de pesquisa alternativa de ataque ao problema baseada na tradução de *corpora* e a (ii) disponibilização de um *dataset* de discursos de ódio em língua portuguesa para a comunidade.

Palavras-chave: Discursos de ódio, Redes sociais, Aprendizagem profunda, LSTM

Abstract

One of the consequences of the popularization of Internet access is the spread of insults and discriminatory messages, the so-called hatespeeches. They are comments that aim to discriminate against someone or a group of people because they belong to a certain group, usually minority, or have some characteristic common to other people. Fighting hatespeech is a growing demand in real and virtual life as it profoundly affects the dignity of its victims.

Detection of hatespeech is a difficult task because, in addition to natural language being inherently ambiguous, it requires a certain level of understanding of its linguistic structure. In many discourses, discrimination does not happen explicitly or with typical expressions: it is necessary world knowledge to recognize them. In addition, sometimes it is necessary to understand the context of the sentence to perceive its hateful content. Sarcasm is another huge challenge (even for humans) since its presence requires knowledge of the community and potentially of the user responsible for the comment for understanding their intent.

Several approaches have been proposed for the hatespeech recognition task. Many authors consider the use of *N-grams*, of which those based on characters are more effective than those based on words. Combined or not with *N-grams*, lexical features were also evaluated, such as the presence or absence of negative words, classes or expressions indicative of insult, punctuation marks, letter repetitions, the presence of emoji, etc. Linguistic *features* were inefficient when used alone, such as *POS tag*, and the relationship between the terms of the dependency tree resulting from the syntax analysis. Recently, the most successful approach has used a neural network to create a distributed representation of the sentences present in a corpus of hatespeech, indicating that word embeddings training is a promising path in the area of hatespeech.

Language drastically affects the tasks of Natural Language Processing (NLP), since most, if not all, words differ from one language to another, as well as their syntax, morphology, and linguistic construction. Thanks to this, works in English are not directly applicable in *corpora* of Portuguese language. In addition, *corpora* in Portuguese for hatespeech are rare, making researchers in the area to do all the construction work.

In this dissertation we studied the use of deep cross-lingual Long Short-Term Memory (LSTM) model, trained with a hatespeech dataset translated from English in two different ways, pre-processed and vectorized with several strategies that were represented in 24 scenarios. The main approaches adopted included the training of embeddings through word index vectors (State of the Art technique), TFIDF vectors, *N-grams* vectors, with or without GloVe vocabulary, tested with the dataset constructed and labeled in this work and with another available in Portuguese. The inverted process was also tried out: we translated our *corpus* into English and compared the

performance with its original version. With the embeddings resulting from the training process in each scenario, we used a Gradient Boosting Decision Tree (GBDT) as a means of improving classification. In fact, the results obtained with LSTM were improved in many scenarios.

We achieved accuracy of up to 70 % in the experiments using the model written with the *corpus* in English and our *dataset* translated into this language. In others, traditional and successful techniques such as TFIDF vectors associated with an LSTM have not proved sufficient. Two important contributions of this work are: (i) proposal of an alternative research approach to attack the problem based on the translation of *corpora* and (ii) provision of a dataset of hatespeech in Portuguese to the community.

Keywords: Hatespeech, Social Networks, Deep Learning, LSTM

Lista de ilustrações

Figura 1 – Unidade RNN	20
Figura 2 – Representação geral de uma RNN	20
Figura 3 – Estrutura de uma célula LSTM	23
Figura 4 – LSTM: <i>Forget Gate</i>	23
Figura 5 – LSTM: <i>Input Gate</i>	24
Figura 6 – LSTM: <i>Output Gate</i>	24
Figura 7 – Sistema de classificação: Tela de boas-vindas	44
Figura 8 – Sistema de classificação: Tela de classificação	45
Figura 9 – Modelo LSTM	53
Figura 10 – Matriz de confusão do Cenário 6 usando o resultado da LSTM	64

Lista de tabelas

Tabela 1 – Exemplos de definições de discursos de ódio	30
Tabela 2 – Discursos de ódio e conceitos relacionados	31
Tabela 3 – Tipos de <i>hatespeech</i> e seus alvos	32
Tabela 4 – Principais abordagens e resultados em termos de Acurácia (Acc), Precisão (P), Cobertura (C) e Medida-F (F), suas <i>features</i> e algoritmos utilizados. . .	36
Tabela 5 – Exemplos de expressões usadas pela CQM para captura dos comentários . .	38
Tabela 6 – Expressões adicionadas à lista da Tabela 5 para captura de <i>tweets</i>	41
Tabela 7 – Quantitativo de páginas e comentários extraídos	42
Tabela 8 – Quantitativo de discursos por fonte de coleta	46
Tabela 9 – Estatísticas do tamanho dos comentários abusivos e limpos	47
Tabela 10 – Configurações dos nós do <i>cluster</i> do LCAD	50
Tabela 11 – Configuração padrão de parâmetros do modelo LSTM	52
Tabela 12 – Lista de <i>datasets</i> usados neste trabalho	54
Tabela 13 – Resumo dos experimentos executados nos Cenários de 1 a 4	56
Tabela 14 – Resultado dos Cenários de 1 a 4	56
Tabela 15 – Resumo dos experimentos executados nos Cenários de 5 a 11	57
Tabela 16 – Resultado dos Cenários de 5 a 11	58
Tabela 17 – Resumo dos experimentos executados nos Cenários de 12 a 15	60
Tabela 18 – Resultado dos Cenários de 12 a 15	60
Tabela 19 – Resumo dos experimentos executados nos Cenários 16,17 e 18	61
Tabela 20 – Resultado dos Cenários de 16, 17 e 18	61
Tabela 21 – Resumo dos experimentos executados nos Cenários de 19 a 24	62
Tabela 22 – Resultado dos Cenários de 19 a 24	62

Lista de abreviaturas e siglas

GBDT	Gradient Boosted Decision Tree
GloVe	Global Vectors for Word Representation
LSTM	Long Short-Term Memory
NLP	Natural Language Processing
STA	Sistema de Tradução Automático
TF-IDF	Term Frequency-Inverse Document Frequency

Sumário

1	Introdução	15
1.1	Objetivos do trabalho	16
1.2	Estrutura do Documento	17
2	Referencial Teórico	19
2.1	Redes Neurais Recorrentes	19
2.1.1	Backpropagation Through Time (BPTT)	21
2.1.2	RNNs bidirecionais	22
2.1.3	Long Short-Term Memory (LSTM)	22
2.1.4	Aplicações das Redes Neurais Recorrentes	24
2.2	<i>Word Embeddings</i>	25
2.2.1	Conceitos gerais	25
2.2.2	Modelos para geração de <i>embeddings</i>	25
2.3	Árvores de decisão	26
2.3.1	Gradient Boosting Decision Tree (GBDT)	27
2.3.1.1	Quando e como uma GBDT funciona?	28
3	Deteção de discursos de ódio em textos	29
3.1	Que são discursos de ódio?	29
3.2	Tipos e termos relacionados a discursos de ódio	31
3.3	Por que a deteção automática de discursos de ódio é uma tarefa difícil?	32
3.4	Trabalhos de deteção de <i>hatespeech</i> relevantes	33
3.5	Dossiê de intolerâncias no mundo digital	36
3.5.1	Importância do Dossiê	37
4	Dataset rotulado com discursos de ódio em Língua Portuguesa	39
4.1	Motivação	39
4.2	Seleção das fontes de coleta	40
4.2.1	Seleção das comunidades do Facebook	40
4.2.2	CrITÉrio de captura de <i>tweets</i>	41
4.3	Coleta	41
4.4	Rotulação	43
4.4.1	Sistema de classificação	43
4.4.2	Elegendo comentários para votação	44
4.4.3	Resultados da rotulação	46
4.4.3.1	Tamanho dos comentários	47

4.4.3.2	Tamanho do vocabulário	47
5	Desempenho de modelos usando o <i>dataset</i> rotulado	48
5.1	Métricas de avaliação adotadas	48
5.2	Pré-processamento	49
5.3	Ambiente de Execução	50
5.4	Modelo LSTM	51
5.4.1	Parâmetros <i>default</i>	52
5.5	Múltiplos <i>datasets</i>	52
5.5.1	Métodos de treinamento	55
5.6	Cenários de experimentação	55
5.7	Análise dos resultados	61
6	Conclusão	67
	Referências	69

1

Introdução

Atualmente as redes sociais são cada vez usadas como forma de comunicar opiniões e trocar experiências. A Internet permitiu a aproximação das pessoas nesse aspecto, ao mesmo tempo em que facilitou a disseminação do *cyber-hate* como uma extensão do ódio e intolerância que existiu desde sempre mas foi então facilitado pelo mundo "sem fronteiras" criado pela revolução digital.

Neste cenário, a detecção automática de comentários com conteúdo abusivo torna-se valiosa, uma vez que eles estão diretamente ligados a crimes de ódio. O Facebook e Twitter vêm adotando medidas que ajudam no controle e combate desse tipo de conteúdo (ALLAN, 2017) (SAFETY, 2017). NOBATA et al. (2016), por exemplo, construiu um modelo que foi adotado pelo Yahoo! como mecanismo de detecção de comentários abusivos.

É notória a quantidade de trabalhos e abordagens em Inglês que lidam com o problema de classificação de discursos de ódio (SCHMIDT; WIEGAND, 2017). Contudo, a maioria não disponibiliza seus dados rotulados publicamente. Em Português, até o presente momento, encontrarmos apenas o trabalho de (FORTUNA, 2017).

Uma vez que as atividades de Processamento de Linguagem Natural são altamente influenciadas pela língua dos *corpora*, pesquisas voltadas à criação de bases de dados e detecção de discursos abusivos em português são de fundamental importância uma vez que a quantidade de pessoas conectadas à Internet que falam esta língua cresce vertiginosamente (ONU, 2017) e, consequentemente, o número de vítimas de discriminação desse tipo sobe anualmente (SOPRANA, 2017).

A tarefa de detecção de discursos de ódio ainda é um desafio para a ciência, por duas principais razões: Primeiro porque ela exige um nível de compreensão avançado da estrutura e semântica dos comentários, envolvendo detecção da intenção do usuário e da presença de ironia/sarcasmo, fatores que englobam, dentre outras coisas, conhecimento de mundo. Ou seja, envolve que o computador tenha representados conceitos comuns que expliquem e delimitem de

forma objetiva elementos do mundo real. Segundo, a língua e os artifícios usados para expressar ou mascarar comentários como discursos de ódio não são estáticos, variando consideravelmente inclusive entre regiões do mesmo país, a exemplos das gírias e vícios de linguagens. Além disso, discursos de ódio podem ser bem construídos e constituídos por inúmeras frases, tornando insuficiente considerar somente os comentários com expressões informais e sendo necessário em muitos casos a inclusão do contexto no qual o comentário está inserido.

Dito isso, o estudo de modelos cross-lingual destaca-se neste cenário por conseguir utilizar *corpora* disponíveis em outras línguas e já bem estabelecidos para representar o conhecimento no treinamento em textos de línguas diferentes de sua original, reaproveitando o trabalho de outros pesquisadores e minimizando a necessidade de realizar a custosa tarefa de criação e rotulação de base de dados, as quais são pré-requisitos para todas as tarefas de aprendizado de máquina.

Apesar da importância, não encontramos trabalho de detecção automática de discursos de ódio usando *datasets* em outros idiomas, ou modelos que se encaixem no perfil cross-lingual para responder de que forma podemos reaproveitar bases de dados existentes em outra língua. Contudo, trabalhos de outras áreas como o de (SILVA et al., 2018) na análise de sentimento em textos traduzidos mostram que traduções são um caminho promissor na detecção de discursos de ódio.

Por esta razão, como hipótese de pesquisa, acreditamos que os modelos LSTM podem ser treinados como modelos cross-lingual e que tais modelos treinados com bases de dados traduzidas automaticamente do inglês são capazes de produzir bons resultados quando testadas em *datasets* originalmente em português, minimizando o impacto nas tarefas de detecção de discursos de ódio que a falta de dados rotulados nesta língua causa, na medida em que reaproveita uma base de discursos em outra língua.

1.1 Objetivos do trabalho

Em posse das informações acima e tendo conhecimento da importância de bases de dados de discursos de ódio rotuladas e do estudo de modelos para detecção automática desse tipo de conteúdo, delimitamos abaixo os objetivos deste trabalho.

O **objetivo geral** é: treinar um modelo cross-lingual LSTM de classificação de discursos de ódio em português usando como base de treinamento o *dataset* com comentários em inglês de forma a utilizar uma base de dados já existente.

Os **objetivos específicos** são:

1. Construir e rotular um *dataset* de discursos de ódio em Língua portuguesa e disponibilizá-lo publicamente para a comunidade a fim de contribuir com os próximos trabalhos para os quais ele possa ser útil.

2. Validar o modelo treinado com o *dataset* criado neste trabalho.
3. Determinar técnicas de pré-processamento e vetorização que são promissoras ou não dentro do contexto de detecção de discursos de ódio com um modelo cross-lingual LSTM usando *dataset* em inglês como base de treinamento.

No primeiro momento, devido à escassez de base de dados em nossa língua alvo (FORTUNA, 2017), focamos na criação de um *dataset* de discursos e rotulamo-os com a ajuda de voluntários que participaram através de um sistema desenvolvido especificamente com esse propósito.

No segundo momento, traduzimos com o auxílio de um Sistema de Tradução Automática (STA) a base de dados de (WASEEM; HOVY, 2016), contendo mais de 16 mil *tweets* rotulados entre sexistas, racistas ou nenhum dos dois.

Em seguida, usando modelo *deep Long Short-Term Memory* (LSTM) (GOODFELLOW; BENGIO; COURVILLE, 2016) e, baseado no trabalho Estado da Arte (BADJATIYA et al., 2017), experimentamos diversas abordagens organizadas em 24 cenários que demonstraram o desempenho do modelo quando configurado e treinado com distintos *datasets* para classificação binária, ou seja, considerando somente duas classes representando a presença ou ausência de discurso de ódio.

Por fim, averiguamos a capacidade cross-lingual do modelo na tarefa de detecção de discursos de ódio usando, em alguns dos cenários criados, o *dataset* construído como base de validação.

Duas importantes contribuições deste trabalho são: (i) proposta de uma abordagem de pesquisa alternativa de ataque ao problema baseada na tradução de *corpora* e a (ii) disponibilização de um *dataset* de discursos de ódio em língua portuguesa para a comunidade. A seguir, definiremos a estrutura deste documento.

1.2 Estrutura do Documento

Para facilitar a navegação e melhor entendimento, este documento está estruturado em capítulos e seções, que são:

- Capítulo 1 - **Introdução**
- Capítulo 2 - **Referencial Teórico**: Neste capítulo apresentaremos os principais conceitos e técnicas relacionados aos experimentos desse trabalho.
- Capítulo 3 - **Detecção de discursos de ódio em textos**: Neste capítulo, nos aprofundamos na definição e nos conceitos relacionados aos discursos de ódio. Fazemos também um levantamento de trabalhos relacionados à detecção de discursos de ódio.

- Capítulo 4 - **Dataset rotulado com discursos de ódio em Língua Portuguesa**: Neste capítulo, comentamos sobre o passo-a-passo na construção do *dataset* criado neste trabalho.
- Capítulo 5 - **Desempenho de modelos usando o dataset rotulado**: Mostramos neste capítulo os experimentos e métodos usados para avaliação do modelo abordado treinado com diferentes *datasets*.
- Capítulo 6 - **Conclusão**: Neste capítulo, apresentamos as considerações finais acerca do resultados obtidos e sugestões para trabalhos futuros visando explorar cenários diferentes para detecção de discursos de ódio.

2

Referencial Teórico

Neste capítulo apresentaremos os principais conceitos e técnicas relacionados aos experimentos desse trabalho. Faremos uma breve introdução às Redes Neurais Recorrentes, inclusive ao modelo LSMT, e algumas de suas aplicações dentro da área de Processamento de Linguagem Natural. Falaremos sobre o conceito e sobre a importância dos *Word Embeddings* e a Gradient Boosting Decision Tree, modelo de árvore de decisão usado neste trabalho.

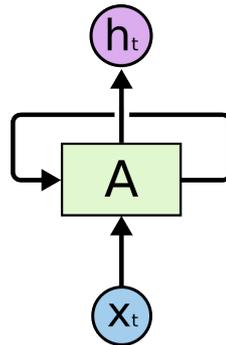
2.1 Redes Neurais Recorrentes

Redes Neurais Recorrentes ou, do Inglês, Recurrent Neural Networks (RNN), são classes de redes neurais especializadas em reconhecimento de cadeias de palavras, ou dados sequenciais, como alguns autores definem. Uma de suas principais vantagens nesta tarefa, é o reconhecimento de longas cadeias de sequências, as quais seriam de difícil reconhecimento em redes não especializadas no reconhecimento deste tipo de dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

As sequências são representadas como um conjunto de entradas $x^{(1)}, \dots, x^{(\tau)}$, no qual a entrada $x^{(t)}$ é o vetor no tempo/ passo t , t variando de 1 a τ . As RNNs compartilham matrizes de pesos ao longo de vários passos no processamento da cadeia de entrada, o que lhe permite aprender padrões que surgem em posições distintas nas sequências. Esse foi um dos avanços em relação às redes neurais tradicionais *feedforward*, nas quais os pesos para cada *feature* da entrada no índice t não são compartilhados entre si, exigindo que identifiquemos as possíveis posições dos padrões desejados nas cadeias de entrada, tarefa que é impraticável.

Na Figura 1, representamos uma porção da RNN para melhor ilustrar sua arquitetura geral. Na imagem, A opera sobre $x^{(t)}$, o vetor de entrada no tempo t , e computa saída h_t da rede após processar sua entrada. Sua saída também é conhecida como o estado da rede no tempo t . O *loop* representa a capacidade da RNN de transferir informações de um passo para o outro.

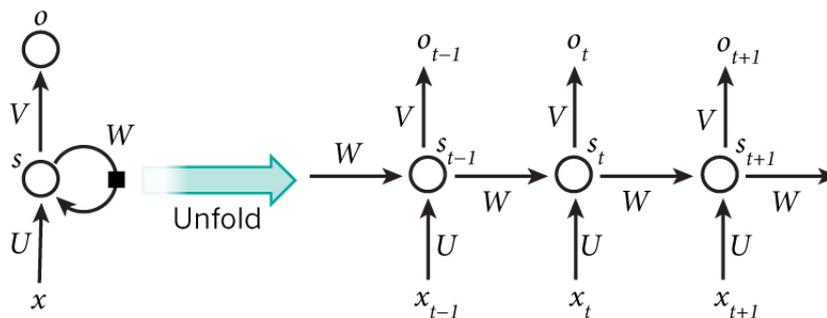
Figura 1 – Unidade RNN



Fonte: retirado do blog de [Christopher Olah](#)

Para uma sequência de tamanho τ , podemos expandir a representação da Figura 1 da maneira ilustrada na Figura 2. Cada entrada no tempo t é mapeada para a entrada $t + 1$ através de um função f que opera sobre o estado s_t .

Figura 2 – Representação geral de uma RNN



Fonte: retirado do site [WildML](#)

Os estados s_t são tratados como a memória da rede, uma vez que ela retém informações sobre o histórico de ocorrências anteriores, ou seja, o histórico de ocorrência dos estados anteriores. Os mesmos parâmetros de f podem ser usados de um passo para outro, como veremos a seguir.

Na ilustração da Figura 2, a operação *Unfold* é simplesmente o desmembramento da RNN em sua forma recorrente (lado esquerdo da imagem) para sua forma completa (lado direito da imagem). A entrada do estado s no tempo t é ponderada pela matriz de pesos U , sua saída é multiplicada pela matriz V para gerar o *output* o . O valor do estado s_t é usado como *input* para o estado s_{t+1} e ponderado pela matriz W .

A Vanilla RNN é um dos modelos mais simples de RNN e sua arquitetura convencional é a mesma descrita na Figura 2. Uma maneira comum de se calcularem os valores de s_t e o é

através das Fórmulas (2.1):

$$\begin{aligned} s_t &= f(Ux_t + Ws_{t-1}) \\ o_t &= Vs_t \\ y_t &= \text{softmax}(o_t) \end{aligned} \quad (2.1)$$

A função "f" é comumente alguma função não linear, como a tangente hiperbólica (*tanh*) ou a *ReLU*. As matrizes U , W e X são compartilhadas entre todos os estados. Nem todos os modelos necessitam que a saída seja gerada para cada estado, e sim para o estado final, por exemplo.

2.1.1 Backpropagation Through Time (BPTT)

O cálculo do gradiente descendente usado para nas *feedforward* sofre modificação para levar em consideração o compartilhamento de parâmetros inerente às RNNs e é conhecido como BPTT. Seja a função de erro cross-entropia definida pela Fórmula (2.2):

$$E_t(y_t, \hat{y}_t) = -y_t \log(\hat{y}_t) \quad (2.2)$$

$$E_t(y, \hat{y}) = \sum_t E_t(y_t, \hat{y}_t) \quad (2.3)$$

$$= -\sum_t y_t \log(\hat{y}_t) \quad (2.4)$$

Na qual y_t representa o valor correto da *feature* no tempo t , e \hat{y}_t é o valor predito pela rede conforme Fórmula (2.1). Dito isto, o erro total consiste da soma de todos os erros em cada estado s_t e, sendo a matriz V um dos parâmetros do modelo, o erro em função de V é calculado como o somatório das derivadas parciais de E_t em relação a V , conforme Equação (2.5).

$$\frac{\partial E}{\partial V} = \sum_t \frac{\partial E_t}{\partial V} \quad (2.5)$$

Expandindo a fórmula acima, podemos perceber que para V , a função de erro depende somente dos valores de y_t , \hat{y}_t e s_t , tornando o resultados de $\frac{\partial E_t}{\partial V}$ calculável a partir de uma única multiplicação matricial, como podemos observar na Equação (2.6).

$$\begin{aligned} \frac{\partial E_t}{\partial V} &= \frac{\partial V}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial V} \\ &= \frac{\partial V}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} \frac{\partial z_t}{\partial V}, z_t = Vs_t \end{aligned} \quad (2.6)$$

Por outro lado, o erro em função de W e U , visto que ambos dependem de s_t (que por sua vez depende de s_{t-1}), envolve a aplicação da regra da cadeia para seu cálculo, uma vez que não podemos considerar s_t como uma constante. Por esta razão, $\frac{\partial E_t}{\partial W}$ é calculado da seguinte maneira:

$$\frac{\partial E_t}{\partial W} = \sum_{k=0}^t \frac{\partial E_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial s_t} \frac{\partial s_t}{\partial s_k} \frac{\partial s_k}{\partial W} \quad (2.7)$$

De outra forma, o gradiente para o cálculo do erro E no tempo t é propagado recursivamente até o tempo $t = 0$, o que propicia o chamado *Vanishing Gradient Problem* (ou, mais raramente, a explosão do valor do gradiente), que consiste na aproximação (degradação) dos valores do gradiente para zero em poucos passos, como consequência das sucessivas multiplicações matriciais envolvidas. Uma vez que o gradiente em cada unidade recorrente tende a zero, ele impulsionará o gradiente das células anteriores para zero também. E quanto maior o valor τ da cadeia de entrada, maiores as chances da rede sofrer com esse problema.

Diversas soluções foram propostas para evitar a aproximação do gradiente a zero, criando versões especializadas da RNN descrita nesta seção. Uma das mais notórias e amplamente utilizadas é aquela conhecida como LSTM, descrita em detalhes na próxima seção.

2.1.2 RNNs bidirecionais

RNNs Bidirecionais (BiRNNs) são redes recorrentes baseadas na concepção de que o estado h_t não depende apenas das informações presentes nos elementos anteriores da sequência de entrada, mas também dos elementos futuros, ou seja, elas levam em consideração tanto o contexto das informações passadas como o contexto das informações que virão a seguir. Normalmente, são compostas por duas RNNs, uma no topo da outra, sendo cada uma responsável por aprender usando o histórico das sequências, e outra usando as sequências futuras. Os resultados de cada uma é combinada para gerar saída da BiRNN.

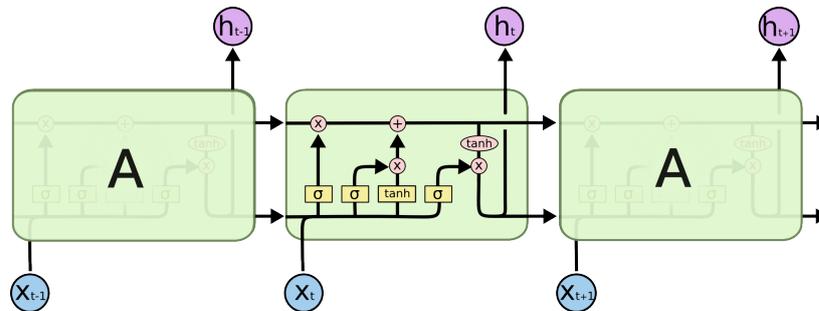
2.1.3 Long Short-Term Memory (LSTM)

As redes LSTM foram inicialmente propostas por [HOCHREITER; SCHMIDHUBER \(1997\)](#) e uma de suas principais características foi a inclusão de unidades especiais conhecidas como portões (*gates*). Essas unidades calculam os pesos que os conectam de forma a evitar a degradação do gradiente descrita na seção anterior através de valores manualmente escolhidos ou parametrizados ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)).

Como toda *gated RNN*, as LSTM têm a capacidade tanto de lembrar quanto de **esquecer** o estado anterior quando essa informação não for mais necessária. Ao logo do tempo de treinamento, a rede tem a capacidade de aprender o que esquecer exatamente, mecanismo que é executado através dos parâmetros do *forget gate*, unidade explicada a seguir.

Dessa forma, os valores do estado anterior, a memória atual e a entrada são combinadas para formar a saída da unidade (ou célula, usando o jargão do modelo LSTM), mecanismo que se mostrou bastante eficiente no aprendizado de longas dependências dos termos de uma sequência. A Figura 3 ilustra os principais elementos que estruturam uma célula LSTM.

Figura 3 – Estrutura de uma célula LSTM

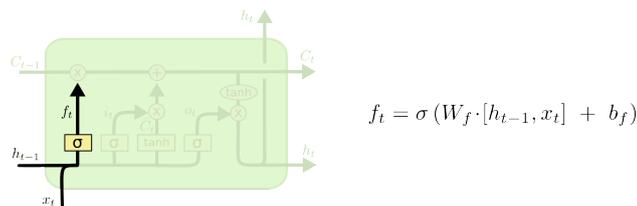


Fonte: retirado do blog de [Christopher Olah](#)

Uma célula LSTM é composta por três portões que controlam diferentes comportamentos: portão de entrada (*input gate*), portão de esquecimento (*forget gate*) e portão de saída (*output gate*). Todos os portões têm uma *sigmoid* (σ) como função de linearidade para controlar o fluxo de informações dentro da célula.

O *forget gate* (Figura (4)) controla a entrada da célula C_t de forma a determinar quais dos valores de índices do vetor de saída da célula anterior C_{t-1} serão mantidos, através de sua função σ , que retorna valores no intervalo entre 0 e 1. W_f e b_f são os pesos e o valor bias para o portão de entrada, respectivamente.

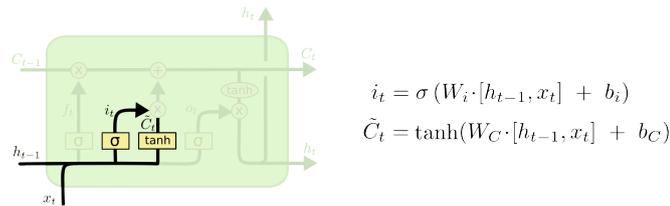
Figura 4 – LSTM: *Forget Gate*



Fonte: retirado do blog de [Christopher Olah](#)

A célula LSTM também tem a habilidade de calcular o que da informação de entrada será armazenado/atualizado. Conforme Figura (5), essa tarefa é dividida em duas etapas: Inicialmente a função σ decide quais das informações de C_{t-1} irá atualizar (no *forget gate* esse passo é responsável por determinar qual informação será esquecida). Em seguida, um novo valor de entrada candidato (\tilde{C}_t) é calculado pela função \tanh , para então ser multiplicado ponto a ponto com o vetor resultante do primeiro passo. Notem que as operações i e \tilde{C} têm seus próprios parâmetros, que também podem ser aprendidos pela rede no processo de treinamento.

Figura 5 – LSTM: *Input Gate*

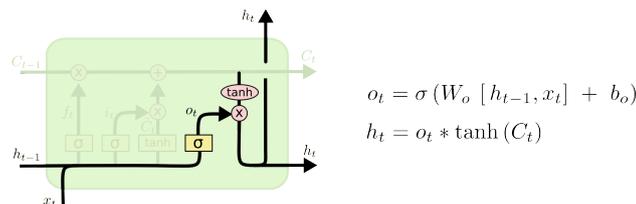


Fonte: retirado do blog de [Christopher Olah](#)

Após essas operações, temos informação suficiente para atualizar o estado da célula C_t . Para isso, ela precisa "esquecer" o que for necessário e atualizar as informações relevantes. Ou seja, o referido estado calculado da seguinte maneira: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$.

Dois passos são necessários para produzir a saída h_t da célula: Primeiro, a função σ determina quais "porções" do vetor/estado C_t farão parte da saída (valor o_t da Figura (6)). A seguir, o estado C_t é submetido a uma função hiperbólica (\tanh) e então multiplicado pelo resultado da operação anterior.

Figura 6 – LSTM: *Output Gate*



Fonte: retirado do blog de [Christopher Olah](#)

2.1.4 Aplicações das Redes Neurais Recorrentes

As RNNs podem ser aplicadas em qualquer problema que exija reconhecimento de sequências de cadeias, como informações em vídeos (que consiste de uma sequência de imagens) e diversas tarefas de Processamento de Linguagem Natural. De fato, áreas como Tradução de Máquina (LIU et al., 2014), Análise de Sentimentos (SINGHAL; BHATTACHARYYA, 2016) e identificação de discurso de ódio (BADJATIYA et al., 2017) se beneficiaram bastante das RNNs e, em particular, dos modelos LSTM.

Um das aplicações mais notórias é a geração de modelos linguagem: Modelos de linguagem tem o objetivo de determinar a probabilidade de ocorrência de uma sequência de palavras: $P(W = w_1, w_2, w_3, \dots, w_n)$.

Os modelos linguagem provêm uma maneira de estimar a vizinhança relativa de diferentes frases presentes em um *corpus*. Normalmente, $P(w)$ é calculado como:

$$P(w_1, w_2 \dots w_i) = P(w_1)P(w_2|w_1) \dots P(w_n|w_1 \dots w_{n-1}) \quad (2.8)$$

Ou seja, a probabilidade P de ocorrência da palavra w é igual à sua probabilidade condicional baseada recursivamente na probabilidade de ocorrência das palavras anteriores, ou seja no **histórico** de palavras da sequência. Por levar em consideração o histórico dos elementos que o precedem, arquiteturas recorrentes são extremamente úteis na geração de modelos de linguagens, os quais, quando criados a partir de redes neuronais, recebem o nome Modelos de Linguagem Neuronal (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para adaptar as sentenças de um *corpus* como entrada de uma LSTM, por exemplo, é preciso representá-las vetorialmente. A maneira mais simples de representação é a codificação one-hot, na qual cada palavra é convertida num vetor do tamanho do vocabulário e apenas o índice do vetor correspondente ao índice da palavra no vocabulário tem valor 1 (os outros índices recebem valor 0). Cada estado h_t representa a palavra predita do modelo de linguagem no tempo t , codificada como one-hot.

2.2 Word Embeddings

2.2.1 Conceitos gerais

Aplicações em Processamento de Linguagem Natural necessitam de uma representação consistente de suas unidades de trabalho, sejam elas palavras, sentenças, parágrafos, documentos ou coleções de informações textuais. Um dos desafios dessa representação é ambiguidade inerente à língua e a outra é a manutenção da coerência dessas representações de forma a preservar relações potencialmente importantes para as diversas tarefas de PLN, como informações semânticas e contextuais.

Os chamados *word embeddings* são a resposta largamente adotada para os desafios citados anteriormente. Eles são representações vetoriais (a princípio de palavras) capazes de manter a relação entre duas palavras semanticamente relacionadas sem perder a habilidade de codificá-las de maneiras distintas (GOODFELLOW; BENGIO; COURVILLE, 2016). No espaço de *embeddings*, palavras que com frequência aparecem em contextos semelhantes estão próximas umas das outras, construindo uma vizinhança de palavras similares. Contudo, ressaltamos que as representações vetoriais desses *embeddings* ainda sofrem com o desafio de representar palavras que têm múltiplos significados ou sentidos (LANDEGHEM, 2016).

2.2.2 Modelos para geração de *embeddings*

Diferentes algoritmos foram desenvolvidos com o propósito de geração de *embeddings*. Eles podem ser divididos em duas famílias de métodos (HARTMANN et al., 2017): Os primeiros são aqueles métodos que trabalham com a matriz de co-ocorrência de palavras, como GloVe (PENNINGTON; SOCHER; MANNING, 2014). E os segundos, são aqueles que trabalham com

modelos preditivos (baseado na vizinha da palavras), como o Word2Vec (MIKOLOV et al., 2013). HARTMANN et al. (2017) resumem alguns dos principais modelos de geração *embeddings*:

- The Global Vectors (**GloVe**) - Algoritmo de aprendizado não supervisionado que computa os vetores através da análise da matriz M de co-ocorrência de palavras construída através das informações contextuais das palavras do *corpus*.
- **Word2vec** - Possui duas diferentes estratégias de treinamento: (i) *Continuous Bag-of-Words (CBOW)*, no qual o modelo tenta prever a palavra do meio suprimida dentro de uma sequência de palavras, e (ii) **Skip-Gram**, o modelo serve para prever a vizinhança de uma da palavra.
- **Wang2Vec**: Modificação do Word2vec cujo objetivo é considerar a ordem das sequências, ao contrário da arquitetura original.
- **FastText**: Nesta arquitetura, *embeddings* são associados N-grams de caracteres, sendo as palavras codificadas como a combinação dessas representações. Portanto, esse método tenta capturar informações morfológicas para construir os seus *word embeddings*.

O Repositório de Word Embeddings do Núcleo Interinstitucional de Linguística Computacional (NILC) ¹ contém diversos vetores de palavras disponíveis publicamente para *download*. Eles foram gerados por meio de um *corpus* em português do Brasil e português Europeu, usando todos os modelos citados acima e em diferentes dimensões. Neste trabalho, para a geração de *embeddings* usando *corpora* em português, adotamos os vetores do NILC construídos através do modelo GloVe de 100 dimensões, denominado daqui em diante de GloVe100.

2.3 Árvores de decisão

Métodos de classificação ou regressão baseados em árvores dividem o espaço de *features* em sub-áreas de forma recursiva, na qual cada passo seleciona a característica mais representativa do dado de entrada no processo de treinamento (MARSLAND, 2014). São chamadas de "árvores de decisão" porque cada sub-área dentro do espaço de features é representada por um nó *root* (que contém a sua *feature* mais relevante) e seus nós folhas, cada um contendo as próximas características relevantes resultantes processo de indução sobre a árvore.

Um das decisões mais críticas na construção e particionamento das árvores de decisão é a escolha de qual *feature* representa melhor os dados no nó que está sendo particionado.

¹ <<http://www.nilc.icmc.usp.br/embeddings>>

Uma das maneiras de avaliar a qualidade do particionamento, consiste do uso dos conceitos de cross-entropia e ganho de informação definidos nas Equações 2.9 e 2.10, respectivamente.

$$Entropia(p) = - \sum_i p'_i \log_2(y_i) \quad (2.9)$$

$$Ganho(S, F) = Entropia(S) - \sum_f \left(\frac{|S_f|}{|S|} \right) Entropia(S_f) \quad (2.10)$$

Onde p é um conjunto de probabilidades de cada classe no conjunto de treinamento, S é o conjunto de exemplos de entrada incluindo seus rótulos e $|S_f|$ é o conjunto dos dados de treinamento que possuem o valor f na *feature* F .

A Entropia é uma medida de desordem dos dados de treinamento que, nos algoritmos de árvores de decisão, mede a homogeneidade do conjunto de treinamento baseado num certo agrupamento de suas características. Já a medida de ganho de informação computa a representatividade em relação ao conjunto S que a inclusão da *feature* F tem no momento de particionamento da árvore no próximo nó. Aquela que obtiver o maior ganho de informação será o próximo *root* da sub-árvore. O processo continua recursivamente até os nós folhas serem gerados.

Árvores de decisões podem ser combinadas para melhorar o poder de predição que cada uma tem isoladamente. Esse método é conhecido como *Tree Ensemble* e existem vários modelos baseados nele. Um dos mais conhecidos é o *Random Forest* (BREIMAN, 2001). Neste trabalho utilizamos um outro modelo chamado Gradient Boosting Decision Tree (GBDT), que é descrito na seção a seguir.

2.3.1 Gradient Boosting Decision Tree (GBDT)

Assim como as *Random Forest* (RF), o GBDT (FRIEDMAN, 2002) se utiliza do particionamento do processo de predição através das *ensemble* para as tarefas de classificação ou regressão linear. Contudo, enquanto as RF constroem várias árvores em paralelo e usam o resultado de cada uma delas para calcular sua própria saída, o GBDT cria um séries de árvores de decisão, em que cada árvore é treinada e a seguinte é construída e treinada de forma a tentar corrigir os erros da anterior.

As árvores criadas nesse processo são *weak learners* (algoritmos de classificação levemente melhores que a classificação aleatória) e à medida que novas árvores são criadas e treinadas, o modelo GBDT tende a cometer cada vez menos erros, conseguindo treinar depois de alguns passos com uma boa acurácia global. Essa técnica de treinamento é conhecida como *boosting* (FRIEDMAN; HASTIE; TIBSHIRANI, 2001).

As funções de erro mais comuns usadas para o treinamento das GBDT são AdaBoost e a regressão logística (FRIEDMAN, 2002). Essa última consiste da função de cross-entropia adaptada para modelos *ensemble*. Já o AdaBoost propõe a criação de pesos para a saída de cada

weak learners. Seu efeito é permitir maior influência aos classificadores com maior acurácia, enfraquecendo a relevância das saídas daqueles que contribuem para diminuir a performance geral do modelo. Uma vez calculado o erro global, ele é propagado para cada classificador, ajustando seus respectivos pesos.

2.3.1.1 Quando e como uma GBDT funciona?

Segundo SUTTON (2005), a técnica de *boosting* costuma beneficiar classificadores cujo método de classificação é instável, diminuindo drasticamente a taxa de erro resultante do método de classificação. Os autores citados justificam que os classificadores instáveis possuem uma variância considerável e o *boosting* a diminui com facilidade, sem aumentar o bias.

Ademais, alguns autores defendem que há uma forte evidência de que as GBDTs são resistentes contra o *overfitting*, possivelmente devido à sua característica de produzir classificadores razoavelmente fortes e descorrelacionados entre si. Em alguns casos, o *boosting* pode piorar o resultado do classificador, mas as razões para isso parecem estar ligadas ao uso de *datasets* pequenos e, portanto, insuficientes para o modelo poder generalizar.

Em resumo, não encontramos consenso sobre a razão pelas quais as GBDTs melhoram os resultados dos classificadores, mas é unânime entre os pesquisadores que há uma melhora significativa na maioria dos casos. Em nossos experimentos, de fato o modelo melhorou significativamente os resultados da rede LSTM usada para treinamento. Mais detalhes podem ser conferidos no Capítulo 5.

3

Detecção de discursos de ódio em textos

Neste capítulo, definiremos e exemplificaremos o que são discursos de ódio (*hatespeech*, no inglês). Usamos diversas fontes para esclarecer seus limites e nuances. Em seguida, listamos o que consideramos ser os principais trabalhos na área de classificação e detecção de discursos de ódio em textos, seus resultados e principais contribuições. Todos eles serviram de embasamento para escolha e execução dos modelos e experimentos abordados neste trabalho. Apresentaremos também um dossiê feito por uma entidade no Brasil com o objetivo de conscientização acerca da problemática ligada à disseminação de discursos de ódio na web e o impacto para algumas de suas vítimas.

3.1 Que são discursos de ódio?

A definição do que são discursos de ódio é por diversas vezes mal interpretada ou confundida com outros conceitos. E, para nos certificarmos de que o computador está classificando corretamente os textos, precisamos esclarecer esse aspecto e garantir que seu resultado é coerente. Na Tabela 1, apresentamos alguns conceitos de discursos de ódio e suas fontes.

Preocupamo-nos em buscar definições de fontes tanto da Internet quanto de autores que se dedicaram a estudar sob outros pontos de vista os discursos de ódio, como é o caso de [MOURA \(2016\)](#) e [SEGUNDO \(2016\)](#). Ambos enfatizam que qualquer tipo de comentário que vise ou tenha a capacidade de instigar a discriminação contra um certo grupo de pessoas é considerado *hatespeech*.

Notem que essa definição é mais abrangente que a do Facebook (vide Tabela 1), o qual permite o uso de conteúdo humorístico e ofensivo, tornando a fronteira do que seria condenável mais difícil de ser estabelecida. Acreditamos que essa definição seja permissiva e favoreça a disseminação de comentários sutis e de violência implícita que são da mesma forma danosos às vítimas, como normalmente as piadas que reforçam os estereótipos (como loiras, gays,

Tabela 1 – Exemplos de definições de discursos de ódio

Fonte	Definição
Facebook ³	"Conteúdos que ataquem pessoas com base em sua raça, etnia, nacionalidade, religião, sexo, gênero ou identidade de gênero, orientação sexual, deficiência ou doença, sejam elas reais ou presumidas, não são permitidos. No entanto, permitimos tentativas claras de piadas ou sátiras que não tenham caráter de ameaças ou ataques. Isso inclui conteúdo que muitas pessoas possam considerar de mau gosto (por exemplo, piadas, comédia stand-up, certas letras de músicas populares etc.)."
Twitter ⁴	"Conduta de propagação de ódio: você não pode promover violência, atacar diretamente ou ameaçar outras pessoas com base em raça, etnia, nacionalidade, orientação sexual, sexo, identidade de gênero, religião, idade, deficiência ou doença grave. Também não permitimos contas cuja finalidade principal seja incitar lesões a outros com base nessas categorias."
(MOURA, 2016)	"O discurso do ódio refere-se a palavras que tendem a insultar, intimidar ou assediar pessoas em virtude de sua raça, cor, etnicidade, nacionalidade, sexo ou religião, ou que têm a capacidade de instigar violência, ódio ou discriminação contra tais pessoas."
(SEGUNDO, 2016)	"O <i>hate speech</i> ou discurso de ódio é aquele que visa disseminar e promover ódio em função da raça, religião, etnia ou nacionalidade [...], muito embora não se limite a tais vetores, podendo se dar também, por exemplo, em função do gênero, da orientação sexual etc."
(FORTUNA, 2017)	" <i>Hate speech</i> is language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humour is used."

gordos e certos aspectos físicos) o são. A repetição de piadas desse tipo, mesmo sem intenção discriminatória, modela a relação entre o grupo daquelas que as proferem e o grupo alvo de vítimas. Em outras palavras, repetir piadas é uma maneira de reforçar atitudes ou pensamentos gordofóbicas, racistas, homofóbicas etc.

Apesar de o foco neste trabalho ser o reconhecimento de discursos de ódio expressados de maneira textual escrita, eles não se limitam a tal forma de comunicação, podendo estar presentes em gestos, panfletos, propagandas, caricaturas etc. Para ser um *hatespeech* é importante que, através do discurso, o alvo ou vítima tenha seus direitos negados pela discriminação por alguma crença ou característica que lhe estejam presentes.

SEGUNDO (2016) afirma que normalmente os ataques partem contra grupos minoritários, acontecendo, algumas vezes, ataques partindo destes contra grupos majoritários, fato que ele denomina de *counter hatespeech*.

No escopo deste trabalho, adotamos o conceito de FORTUNA (2017) por considerá-lo completo na medida em que inclui situações delicadas, inclusive situações humorísticas no geral, como piadas.

3.2 Tipos e termos relacionados a discursos de ódio

Existem muitos conceitos relacionados a discursos de ódio: *hate*, *cyberbullying*, linguagem abusiva, discriminação, profanidade e toxicidade. Alguns deles se confundem entre si pela semelhança. Abaixo, na Tabela 2, reproduzimos resumidamente as distinções elaboradas por FORTUNA (2017) aos referidos conceitos e sua relação com o conceito de discurso de ódio.

Tabela 2 – Discursos de ódio e conceitos relacionados

Conceito	Definição	Diferença de discurso de ódio
<i>Hate</i>	Expressão de hostilidade sem qualquer justificativa declarada.	Discursos de ódio contém ódio voltado contra grupos específicos.
<i>Cyberbullying</i>	Nome dado às agressões e ofensas praticadas de forma eletrônica entre crianças e adolescentes repetidamente e ao longo do tempo, com intenção de humilhar e inferiorizar o outro. ⁶	Discurso de ódio é mais geral, e não necessariamente voltado para uma pessoa específica.
Discriminação	Processo através do qual uma diferença é identificada e usada como base para um tratamento injusto.	Discurso de ódio é uma forma de discriminação que se manifesta verbalmente ou na escrita.
Linguagem abusiva	Refere-se a linguagens danosas e inclui discursos de ódio, profanidade e mensagens depreciativas.	Discursos de ódio são uma forma de linguagem abusiva.
Profanidade	Linguagem ofensiva ou obscena.	Discursos de ódio podem usar linguagem profana, mas não necessariamente.
Linguagem tóxica	Comentários rudes, desrespeitosos e irracionais cujo objetivo é provocar a desistência da pessoa da discussão.	Nem todos os comentários tóxicos contém discursos de ódio. E discursos de ódio podem provocar mais discussões entre as pessoas.

Fonte: Adaptado de FORTUNA (2017)

De acordo com SEGUNDO (2016), é importante destacar que o discurso de ódio deve ter como alvo um grupo de pessoas como um todo e não uma pessoa específica, caso contrário tornar-se-ia mera ofensa pessoal.

Como podemos observar através das definições fornecidas, uma linguagem abusiva não necessariamente contém o tipo de discriminação que caracteriza um discurso de ódio. Contudo, no escopo deste trabalho, por razões de simplicidade, adotamos a definição de que um comentário é abusivo caso contenha algum tipo de discurso de ódio e limpo caso contrário.

A Tabela 3 exhibe os principais tipos de *hatespeech* e alguns dos seus alvos. Potencialmente, todos podemos ser vítimas de discriminação, mas observamos a preponderância de grupos minoritários.

Adicionalmente, SEGUNDO (2016) disserta sobre um tipo de discurso que contém intolerância política e o quanto ele é presente no mundo. Neste cenário, o discurso de ódio costuma se manifestar em comentários com intenção explícita ou não de inferiorizar ou ofender o adversário. No Brasil, a intolerância política é evidente na entre petistas e antipetistas. É comum

Tabela 3 – Tipos de *hatespeech* e seus alvos

Categoria	Alvos do ódio
Raça	peessoas de pele clara, afro-descendentes
Comportamento	peessoas inseguras, peessoas sensíveis
Aparência física	peessoas obesas, peessoas bonitas
Orientação sexual	gays, héteros
Classe social	peessoas pobres, peessoas marginalizadas, peessoas ricas
Gênero	grávidas, feministas
Xenofobia	chineses, indianos, nordestinos
Deficiência	peessoas bipolares, autistas, portadores de necessidades específicas
Religião	peessoas religiosas, islâmicos
Outra	bêbados, peessoas com pouca instrução

Fonte: Adaptado de [SILVA et al. \(2016\)](#)

encontrarmos piadas, sátiras ou expressões com o objetivo de atacar os opositores ideológicos, como a expressão "petralha" que se refere aos petistas.

3.3 Por que a detecção automática de discursos de ódio é uma tarefa difícil?

A identificação de discursos de ódio em comentários é difícil por diferentes razões. A linguagem utilizada no texto, por exemplo, pode ser muito ruidosa e conter erros ortográficos, expressões ou construções linguísticas informais. Além disso, a discriminação pode acontecer de forma velada, exigindo conhecimento da intenção do usuário e análise do contexto para concluir a existência de algum discurso de ódio.

Por exemplo: o comentário "*essi sotaque e muito engraçado*"(sic), postado no Facebook, foi feito em resposta a cantores de rap nordestinos, os quais foram achincalhados por seus pares de outras regiões. Sem esse contexto, é difícil tanto para o computador quanto para um avaliador humano discernir se certo comentário contém ou não conteúdo abusivo. Em posse desta informação, fica claro que a intenção do usuário responsável pelo comentário acima citado é ridicularizar o cantor especificamente pelo sotaque nordestino, usando-o como critério de discriminação. A postagem, portanto, é xenofóbica.

Em outro espectro, o comentário "*sotaque lixo, nordeste lixo, só tem rapadura aí kk*"(sic) (feito no mesmo contexto da mensagem do parágrafo anterior) contém discriminação explícita, tornando mais fácil a sua classificação como abusiva e, especificamente, xenofóbica, uma vez que agride a todos os nordestinos, usando, inclusive, estereótipos (o fato da rapadura ser muito

comum no nordeste) para esta finalidade.

Recentemente, em 2018, algumas declarações racistas ganharam destaque na mídia e, dentre elas, destacamos a seguinte frase: "*sempre quis tocar um cabelo de um negro*"(sic). Identificamos nela uma discriminação sutil, a qual não é determinada pela presença da palavra "negro", e sim pela maneira como o autor fala do negro, colocando-o como alguém diferenciado, exatamente pela cor de sua pele e o tipo de seu cabelo, discriminando-o, e por esta razão praticando racismo. Este comentário ilustra uma outra situação em que é difícil classificar discursos de ódio, considerando somente as palavras da frase e sem reconhecer a intenção do usuário, ou o significado implícito de suas palavras.

No trabalho de [NOBATA et al. \(2016\)](#) os autores listam o que consideram ser as principais razões na tarefa de classificação automática de discursos de ódio, as quais resumimos abaixo:

- **Não basta buscar por palavras-chave:** Uso de expressões regulares no reconhecimento de discursos de ódio pode resultar em falsos positivos, uma vez que comentários com expressões típicas de discriminação nem sempre são discriminatórias. Ademais, a lista de palavras-chave baseadas em *blacklists* costuma variar com o tempo e pode ser ofuscada de diferentes maneiras pelos usuários, tornando seu uso insuficiente para tarefas de classificação.
- **Comentários abusivos podem ser bem construídos:** Nem todos os comentários odiosos são escritos com palavras informais. Alguns podem ser bem fluentes e gramaticalmente impecáveis. Portanto, considerar a presença de ruídos como erros gramaticais não é suficiente para a detecção automática de discursos de ódio.
- **Discursos de ódio podem atravessar frases:** É comum precisarmos considerar mais de uma frase para determinar se um comentário é abusivo ou não. As ideias discriminatórias podem estar manifestadas em sentenças diferentes, inclusive tornando necessário conhecimento de mundo para caracterizá-las.
- **Ironia/Sarcasmo:** Não dificilmente, usuários podem se apoderar de frases irônicas para discriminar seus alvos, tornando o reconhecimento de seu significado mais difícil até mesmo para humanos.

3.4 Trabalhos de detecção de *hatespeech* relevantes

Em [NOBATA et al. \(2016\)](#) os autores usaram três diferentes *datasets* para analisar o impacto do modelo desenvolvido para identificação de mensagens com conteúdo abusivo (os quais incluem discursos de ódio). O chamado "*Primary Data Set*" é o principal, constituído de comentários fornecidos e moderados pelo Yahoo! Finanças e Notícias no período de Outubro de 2012 e Janeiro de 2014. O Segundo, denominado "*Temporal Data Set*", é usado para a análise da

influência da mudança de linguagem na classificação de comentários como abusivos ou limpos; foram fornecidos também pelo Yahoo! Finanças e Notícias e coletados entre Abril de 2014 e Abril de 2015. Por fim, o terceiro *dataset*, nomeado como "WWW2015 Data Set", foi criado por DJURIC et al. (2015) e usado por NOBATA et al. (2016) para fins de comparação com o estado da arte na área à época de finalização do trabalho.

Os autores treinaram um modelo usando o programa de aprendizado de máquina chamado Vowpal Wabbit com 4 diferentes tipos de *features*: *N-Grams*, *features* Linguísticas, Sintáticas e Semânticas. As *features* semânticas incluem modelos de *word embedding* pré-treinados e um modelo de *embedding* criado com o *word2vec* através de um *corpus* de textos de Finanças e Notícias. O trabalho superou o estado da arte em 10AUC (**0.9055** contra 0.8007). Uma das contribuições mais importantes deste trabalho foi comprovar que *n-grams* baseados em caracteres são robustos contra *datasets* com muito ruído, como é o caso daqueles provenientes de redes sociais.

Já BADIYATIYA et al. (2017) executaram diversos experimentos usando arquiteturas *deep learning* distintas para classificação dos tweets coletados e classificados no trabalho de WASEEM; HOVY (2016), usados como *benchmark*: ao todo, foram mais 16k *tweets* rotulados como racistas, sexistas ou nenhum dos dois. O trabalho explorou as técnicas *deep* e usou várias combinações de abordagens como *embeddings* treinados por uma LSTM, *N-Grams* de caracteres, TF-IDF, Bag of Words Vectors (*BoWV*) e Global Vectors for Word Representation (*GloVe*).

Eles usaram tanto uma arquitetura CNN quanto uma LSTM, uma vez que suas características individuais poderiam fazer diferença na detecção dos discursos. De fato, a combinação de uma rede LSTM, os *embeddings* gerados aleatoriamente mais uma Gradient Boosted Decision Trees (GBDT) revelou-se o melhor método, superando o até então Estado da Arte na classificação de discursos de ódio com um valor de Medida-F de **0.93**. Os *embeddings* foram treinados pela LSTM e então submetidos à GBDT.

A arquitetura LSTM (BADIYATIYA et al., 2017) está representada na Figura 9 e uma versão adaptada e amplamente utilizada neste trabalho nos experimentos está descrita na Seção 5.6.

Surpreendentemente, como os próprios autores observaram, o modelo que usou *embeddings* gerados aleatoriamente obteve melhor desempenho que aquele inicializado com o *GloVe*, fato indicativo de que o poder inerente das LSTM de capturar longas séries de dependências de palavras em *tweets* é maximizado quando a representação vetorial dos dados de entrada são inicializados de forma aleatória, pelo menos no *dataset* explorado.

Outra hipótese para justificar esse fato é que o *GloVe* não possui poder de representação semântica suficiente para melhorar o desempenho de um modelo de classificação de discursos de ódio como esperado. Uma outra hipótese, que é a mais provável, é que o método de combinação dos *embeddings* de palavras para representar o *embedding* dos *tweets* pelo *word2vec* não é

suficientemente robusto. Uma abordagem de *paragraph embedding* é provavelmente mais adequada (SCHMIDT; WIEGAND, 2017).

SCHMIDT; WIEGAND (2017) estudaram uma ampla gama de trabalhos em detecção de discursos de ódio com processamento de linguagem natural, destacando a relevância do uso de diversas *features* em vários modelos abordados até então. Algumas das *features* mais relevantes indicadas pelos autores foram: *N-Grams* baseados em caractere, o uso de *paragraph embeddings*, análise de sentimento do comentário, recursos léxicos como a presença de palavras negativas, recursos linguísticos como a classe das palavras (*POS-tag*) e bases de conhecimento com conhecimento especialista sobre padrões de escrita em discurso de ódio, por exemplo.

O trabalho de BADJATIYA et al. (2017) é posterior ao de SCHMIDT; WIEGAND (2017), por esta razão aquele não foi citado. Mas, seu trabalho é importante não somente por ter alcançado altas taxas de precisão na classificação de discursos de ódio, como também por mostrar que é possível obter bons resultados somente com a representação vetorial dos *tweets* baseada em *word embeddings* treinados por uma LSTM. A maioria dos trabalhos até então, como podemos observar no *survey* supracitado, considerou diversas outras *features* que muitas vezes eram de difíceis extração, como é o caso das *POS-tag*.

O único trabalho com *corpora* na língua portuguesa encontrado foi o de (FORTUNA, 2017). A autora realizou um extenso estudo sobre diversas definições que procuram delimitar o que são os discursos de ódio e conceitos relacionados, como o *cyberbullying*. Foi construído um *dataset* de *tweets* manualmente rotuladas, totalizando 5668 mensagens, dentre as quais 22% foram declaradas como um dos 85 tipos e subtipos de discursos de ódio considerados no trabalho.

Para a tarefa categorização dos *tweets* rotulados, foi utilizada a abordagem de classificação hierárquica: técnica que decompõe a tarefa de classificação em um conjunto de problemas menores, os quais podem ser resolvidos de maneira eficiente e combinados para classificar documentos compostos por aqueles (HAO; CHIANG; TU, 2007).

FORTUNA (2017) também conduziu experimentos com a classificação binária (denominada no trabalho como "unimodel") dos discursos e a classificação multi-classe (nomeada de "multimodel"). Em termos de precisão, o melhor algoritmo unimodel e multimodel fora o Rpart (KUHN et al., 2008) com 0,778 e 0,883, respectivamente. O melhor valor de cobertura foi alcançado com o SVM Linear para ambos os modelos, com 0,720 e 0,765, nessa ordem.

PARK; FUNG (2017), por sua vez, usaram redes neurais convolucionais (CNN) baseadas em palavras (representadas em forma de *embedgings* por meio do *word2vec*), caracteres (convertidos para a representação *one-hot*) e numa abordagem híbrida (palavras + caracteres) para a detecção de linguagem abusiva. O *dataset* usado foi o de WASEEM; HOVY (2016) e, primeiro classificando os discursos como abusivos ou não e depois usando essa informação para classificar seu sub-tipo (racista ou sexista), o melhor resultado alcançado foi através da combinação da CNN Híbrida com o algoritmo de regressão logística, obtendo **0,828**, **0,831** e

0,824 de precisão, cobertura e Medida-F, respectivamente.

GAO; HUANG (2017) demonstraram a importância do uso do contexto dos comentários a serem classificados. Com um *dataset* criado por eles mesmos, combinaram o modelo de regressão logística com uma arquitetura LSMT e extraíram N-Grams baseados em caracteres, usando informações semânticas e léxicas para cada palavra do vocabulário. Os melhores resultados em termos de acurácia, precisão, cobertura, Medida-F, AUC foram, respectivamente: **0,779, 0,650, 0,678, 0,600, 0,804**.

Na Tabela 4, listamos as principais referências na área de detecção de discursos de ódio citadas neste trabalho, incluindo seus resultados, *features* e algoritmos utilizados.

Tabela 4 – Principais abordagens e resultados em termos de Acurácia (Acc), Precisão (P), Cobertura (C) e Medida-F (F), suas *features* e algoritmos utilizados.

Ano	Acc	P	C	F	AUC	Features	Algoritmos	Paper
2017	–	0,93	0,93	0,93	–	Word embeddings, BoWV	Logistic Regression, Random Forest, GBDT, SVM, DNN, CNN	(BADJATIYA et al., 2017)
2017	0,78	0,78	0,72	0,764	–	N-Grams de palavras	Logistic Regression, MLP, SVM	(FORTUNA, 2017)
2017	–	0,828	0,831	0,824	–	Caracteres, Word embeddings	Logistic Regression, SVM, CNN	(PARK; FUNG, 2017)
2017	0,78	0,65	0,68	0,60	0,80	N-Grams, Word embeddings, Features semânticas e léxicas	Logistic Regression, LSTM, BiLSTM	(GAO; HUANG, 2017)
2016	–	0,82	0,82	0,82	–	Tamanho de tokens, N-Grams, pontuações, POS-tag	modelo skip-bigram	(NOBATA et al., 2016)

3.5 Dossiê de intolerâncias no mundo digital

O blog Comunica Que Muda (CQM) (CQM, 2016) criou o que ele chamou de Dossiê das Intolerâncias⁷ no mundo digital no Brasil, catalogando os tipos mais evidentes e as expressões e frases mais comuns usados nesse contexto. Em outras palavras, o Dossiê se propôs medir o quanto o brasileiro está intolerante nas redes sociais.

Dez tipos de intolerâncias foram monitoradas durante três meses - de abril a junho de 2016. Os tipos de ódio destacados foram aqueles com relação à aparência das pessoas, às suas classes sociais, às inúmeras deficiências, à homofobia, misoginia, política, idade/geração, racismo, religião, aparência e xenofobia.

Toda vez que alguma palavra ou expressão referente a um desses assuntos era identificada em um post do Facebook, do Twitter, do Instagram, de algum blog ou comentário em sites da

⁷ <http://s18628.pcdn.co/wp-content/themes/comunica/dist/dossie/dossie_intolerancia.pdf>

internet, ela era recolhida e analisada pela equipe do projeto. No total, foram analisadas 542.781 menções.

O método de classificação dos comentários do CQM foi, portanto, baseado em *blacklists*. Essa representa uma fragilidade do método de captura e seleção porque muitos discursos de ódio não contêm expressões presentes em tais listas.

Além disso, comentários intolerantes podem estar estruturados em várias orações de forma a ser necessário levar em consideração frases anteriores para determinar se a outra é ou não abusiva ou carrega indícios de discursos de ódio (NOBATA et al., 2016). Nesses termos, é provável que o filtro usado no projeto tenha deixado de fora uma quantidade representativa de comentários abusivos. A Tabela 5 exibe algumas expressões/menções usadas pelo blog CQM para a captura do conteúdo desejado.

As expressões da Tabela 5 e detalhes do método de captura foram fornecidas prontamente através de contato por e-mail pelos organizadores do projeto. A lista de comentários classificados, contudo, não foi liberada pelo grupo CQM.

Apesar do método de captura ser baseado em *blacklists*, os comentários extraídos e classificados foram relevantes na medida em que deixaram claro os tipo de discursos que são proferidos nas redes sociais, como pode ser evidenciado no documento publicamente disponível do Dossiê. Frases claramente ofensivas ou discretamente intolerantes foram destacadas e mostraram o cenário de pré-conceito percebido e comprovado do grupo.

Por esta razão, usamos essas expressões como ponto de partida para captura de *tweets* e pré-seleção de comentários a serem votados no processo de construção e rotulação do *dataset* apresentado neste trabalho. Os detalhes são discutidos no Capítulo 4.

3.5.1 Importância do Dossiê

Muitos casos de intolerância se manifestam de maneira velada em discursos que consideramos inofensivos ou banais, mas que levam consigo uma enorme carga de pré-conceito atingem de maneira profunda aqueles que são suas vítimas.

O Dossiê, além de denunciar de maneira evidente o quanto os brasileiros são intolerantes, se destaca ao mesmo tempo, por buscar conscientizar de que forma essa intolerância se revela e por que devemos combater os comentários quotidianos nocivos e disfarçados, inclusive, de piada.

Tabela 5 – Exemplos de expressões usadas pela CQM para captura dos comentários

#	Tipo do discurso	Expressões
1	Intolerância contra aparência	“Narigudo”/ “seu” “gordo” / “gordo fazendo gordice” / “cabelo ruim”/ “cabelo de bombril”
2	Intolerância contra classe social	“Bolsa esmola” / “pobraiada” / “parece favelado” / “favelado é foda” / “coisa de favelado”
3	Intolerância contra deficientes	“retardado mental” / “tem down” / “alejado” / “demente” / “leproso” / “aidético”
4	Homofobia - LGBT	“boiola” / “baitola” / “gay” “desperdício” / “cara de travesco” / “voz de travesco”
5	Misoginia	“feminazi” / feminista mal comida / odeio vagabunda / vadia vagabunda / tudo vagabunda / “vai lavar louça” / “mal comida”
6	Intolerância política	/ “comunista safado”/ “coxinha fascista” / “comunista” “ladrão”/ “bolsa esmola” / “bolsa” “compra votos” / “petista vagabundo”
7	Preconceito contra idade/geração	/ “velho asilo”/ “não tenho idade” / “adolescente preguiçoso”/ “adolescente chato” / “adolescente
8	Racismo	“Cabelo ruim”/ “cabelo de bombril” / “não sou tuas nega” / “preto é foda” / “nego é foda”
9	Intolerância Religião	“crente do rabo quente” / “crente do cu quente” / “odeio crente” / “sem Deus no coração” / “muçulmano bomba”
10	Xenofobia	/ “arabe” “bomba” / “muçulmano” “bomba” / “japones é tudo igual” / “volta pra sua terra” / “caiçara folgado”

Fonte: Tabela disponibilizada pelos autores do projeto

4

***Dataset* rotulado com discursos de ódio em Língua Portuguesa**

No presente capítulo, demonstraremos o passo a passo para construção e rotulação do *dataset* criado neste trabalho. Além da motivação, detalharemos os desafios e as estratégias adotadas para maximizar a probabilidade de adquirir textos com discursos de ódio, o sistema desenvolvido e usado para votação online dos comentários candidatos e, por fim, os resultados alcançados e consolidados no *dataset*.

4.1 Motivação

Base de dados rotulados é um pré-requisito para atividades de aprendizagem supervisionada. No começo deste trabalho não encontramos estudo na área de computação relacionado à detecção de discursos de ódio em Língua Portuguesa e, da mesma forma, base de dados com comentários classificados de *hatespeech* disponíveis. Por esta razão, nossa primeira motivação e pré-requisito para testarmos modelos foi criar nosso próprio *dataset*.

O trabalho de [FORTUNA\(2017\)](#), o primeiro divulgado com foco em comentários de Língua Portuguesa, foi publicado após termos começado o processo de coleta (detalhado nas seções seguintes) e rotulação de nosso próprio *dataset*. Ela disponibilizou sua base de comentários publicamente, fato que nos possibilitou validar nossos modelos com mais dados em Português.

Sendo assim, nossas motivações para construção de um *dataset* de discursos de ódio foram:

1. **Disponibilização para a comunidade** - Disponibilizá-lo para a comunidade a fim de contribuir com trabalhos para detecção de discursos de ódio em Língua Portuguesa.
2. **Validação de modelos** - Treinar e validar modelos de aprendizagem de máquina com comentários odioso em Língua Portuguesa.

4.2 Seleção das fontes de coleta

No processo de coleta de comentários, precisamos selecionar as fontes cuidadosamente para maximizar a probabilidade dos textos extraídos conterem algum tipo de discurso de ódio, de maneira que a proporção de textos verdadeiros positivos, ou seja, aqueles que de fato contêm algum discurso de ódio, seja representativa (SCHMIDT; WIEGAND, 2017). Essa estratégia possibilita também direcionar o processo de busca para sub-tópicos específicos e sub-tipos de discursos de ódio desejados.

No escopo deste trabalho, não desejamos classificar comentários com tipos específicos de *hatespeech* e sim obter uma quantidade representativa deles, uma vez que nosso objetivo é determinar a presença ou ausência de conteúdo abusivo nos textos avaliados.

Posto isso, selecionamos alguns sites e tópicos que permitissem a interação de usuários por meio de comentários e cujos temas ou assuntos abordados tivessem uma alta probabilidade de conterem assuntos polêmicos, atraindo *haters* e outras opiniões ou ideias discriminatórias.

No processo de seleção, contamos com a colaboração do grupo de pesquisa Ludiico e outros voluntários para eliciação dos sites mais promissores dentro dos critérios estabelecidos anteriormente. No total, foram listadas 35 URLs, dentre tópicos em sites de notícias, comunidades do Facebook, páginas no YouTube e fóruns.

Os tópicos de sites de notícias agregam a lista de artigos cujo tema está relacionado a ele. Por exemplo: no tópico (listado pelos colaboradores) <<http://g1.globo.com/politica/>>, encontramos todas as notícias do G1 pertinentes ao tema "política". Já no tópico <<https://veja.abril.com.br/noticias-sobre/homofobia/>>, estão os artigos da Veja categorizados como "homofobia". Os comentários nos artigos destes tópicos, portanto, tinham uma alta probabilidade de conterem conteúdo acerca do tema homofobia ou política, sejam eles positivos ou negativos.

4.2.1 Seleção das comunidades do Facebook

As comunidades do Facebook consideradas foram obtidas em sua maior parte através do Mapa de ódio¹ criado pelo Laboratório de Estudos sobre Imagem e Cibercultura (Labic)², que consiste de um mapa de redes de admiradores das Polícias Militares no Facebook. Conforme reportagem³ do CartaCapital sobre o mapa:

"São páginas dedicadas a defender o uso de violência contra o que chamam de "bandidos", "vagabundos", "assaltantes", fazer apologia a linchamentos e ao assassinato, defender policiais, publicar fotos de pessoas "justiçadas" ou mortas violentamente, vender equipamentos bélicos e combater os direitos humanos."

¹ <<https://www.cartacapital.com.br/blogs/outras-palavras/facebook-um-mapa-das-redes-de-odio-327.html/>>

² <<http://www.labic.net/>>

³ <<https://www.cartacapital.com.br/blogs/outras-palavras/facebook-um-mapa-das-redes-de-odio-327.html/>>

Sendo assim, selecionamos as comunidades mais influentes desenhadas no mapa de ódio, por considerarmos que elas conteriam um maior número de publicações com conteúdos ligados a temas polêmicos.

4.2.2 Critério de captura de *tweets*

Utilizamos também o Twitter como fonte de coleta pela relevância que a rede social possui no Brasil. A coleta se deu a partir das palavras-chave listadas na Tabela 5 acrescidas de outras manualmente selecionadas ou copiadas da *blacklist* Hatebase⁴. Abaixo, na Tabela 6, citamos as expressões adicionadas e separadas por categoria.

Tabela 6 – Expressões adicionadas à lista da Tabela 5 para captura de *tweets*

#	Tipo do discurso	Expressões
1	Xenofobia	“nordestino burro”/ "sotaque ridículo"/ "sotaque lixo"/ "nordeste lixo"/ "nada contra o nordeste"/ “nada contra nordestino”/ “nordeste não tem água” / "sotaque de viado"/ "nordestino viado"/ "volta pra sua terra"
2	Racismo	“caboclo” / "mestiço"/ "volta para a senzala"/ "carcamano"

Vale ressaltar que a existência de uma dessas expressões no texto não significa que ele contenha algum discursos de ódio (SCHMIDT; WIEGAND, 2017). E, de maneira inversa, o fato de um comentário não apresentar nenhuma delas não garante que ele seja limpo. Usamos apenas um indício de provável relevância, considerando sua existência como critério de pré-seleção de *tweets*, uma vez que capturar todos eles sem critério seria ineficiente e pouco qualitativo.

Todos os *tweets* capturados pelo método acima foram considerados, independentemente do seu tamanho. Assim, mesmo comentários com apenas 1 palavra poderiam ser votados.

4.3 Coleta

Para a extração de conteúdo do Facebook, usamos o facebook-sdk⁵ e acessamos, para cada comunidade, todos os comentários dos *posts* pelos seus usuários, bem como os metadados disponibilizados pela API.

Usamos a biblioteca Tweepy⁶ e nos conectamos ao *streaming* de *tweets online* que a API do No Twitter disponibiliza, detectando aqueles que continham uma das expressões presentes nas Tabela 5 e 6 e salvando-os em nossa base de dados.

⁴ <<https://www.hatebase.org/>>

⁵ <<http://facebook-sdk.readthedocs.io/en/latest/api.html>>

⁶ <<http://www.tweepy.org/>>

Os fóruns e páginas de notícias não fornecem API para extração de seus conteúdos. Por esta razão, usamos técnicas de *Web scraping* a fim de reconhecer os comentários dos usuários e suas meta-informações disponíveis. Para cada fórum, estudamos as suas *tags* HTML usadas para exibir os comentários e coletamos seu conteúdo textual.

Alguns sites de notícias como o Estadão e a Veja tinham uma estrutura HTML de difícil previsão, por isso não conseguimos coletar suas informações. O processo de extração e *parsing* foram executados pelo *wrapper* do Selenium para python ⁷. Ele foi necessário porque alguns conteúdos das páginas são exibidos somente com interação do usuário através de cliques, por exemplo, e o Selenium possui essa funcionalidade.

Somente na captura dos *tweets* filtramos os comentários por palavras-chave por ser o melhor caminho. Esse tipo de abordagem tem a desvantagem de o *corpus* resultante conter somente discursos com os tipos de discursos de previstos antes da coleta, fazendo com que o modelo aprenda através de padrões já conhecidos pelo pesquisador (FORTUNA, 2017).

Por esta razão, na coleta de outras fontes não descartamos nenhum comentário, o que possibilita a captura de textos abusivos com estrutura textual diferente daquelas previstas por meio de palavras-chave. O quantitativo de páginas e comentários extraídos para cada fonte de coleta é listado na Tabela 7.

Tabela 7 – Quantitativo de páginas e comentários extraídos

Fonte	Páginas	Comentários
G1	11.774	724.997
Facebook	11	658
Youtube	81	74.013
Twitter	–	136.118
Stormfront	129	1.249
TOTAL	11.995	937.035

Abaixo, listamos detalhadamente os passos executados para coleta de nossos comentários após a etapa de seleção das fontes de coleta:

1. Para as fontes sem API (Como as páginas de notícias):

- Analisamos manualmente o código HTML página de cada fonte de coleta para determinar as *tags* que armazenavam os comentários do usuários.
- Por meio de uma ferramenta de *web scraping* ⁷, extraímos e salvamos cada comentário em nossa base de dados.

⁷ <<http://selenium-python.readthedocs.io/>>

- Além dos comentário, salvamos os dados de criação/edição do comentário, bem como o link da página que o contém.

2. Twitter:

- Seleccionamos as expressões de captura definidas nas Tabelas 5 e 6 e inserimos algumas variações linguísticas para melhor se adequarem ao estilo informal de escrita dos usuários. Por exemplo: adicionamos a variação "vc" para a palavra "você", "n" para "não" etc.
- Usamos a API do Twitter para captura *online* de todos os *tweets* que continham pelo menos uma das expressões definidas anteriormente.
- Salvamos a representação serializada de cada *tweet* para eventuais necessidades.

3. Facebook:

- Após selecionarmos as comunidades, usamos a API do Facebook para acessar todos os comentários aos *posts* em cada uma das páginas.
- Além deles, salvamos todo o conteúdo do usuário responsável pelos comentários, bem como os dados do post.

4.4 Rotulação

Para permitir a classificação dos comentários por parte dos voluntários, criamos um sistema web dedicado para esta tarefa. Era necessário que ele fosse amigável e responsivo, permitindo a classificação mesmo em dispositivos móveis.

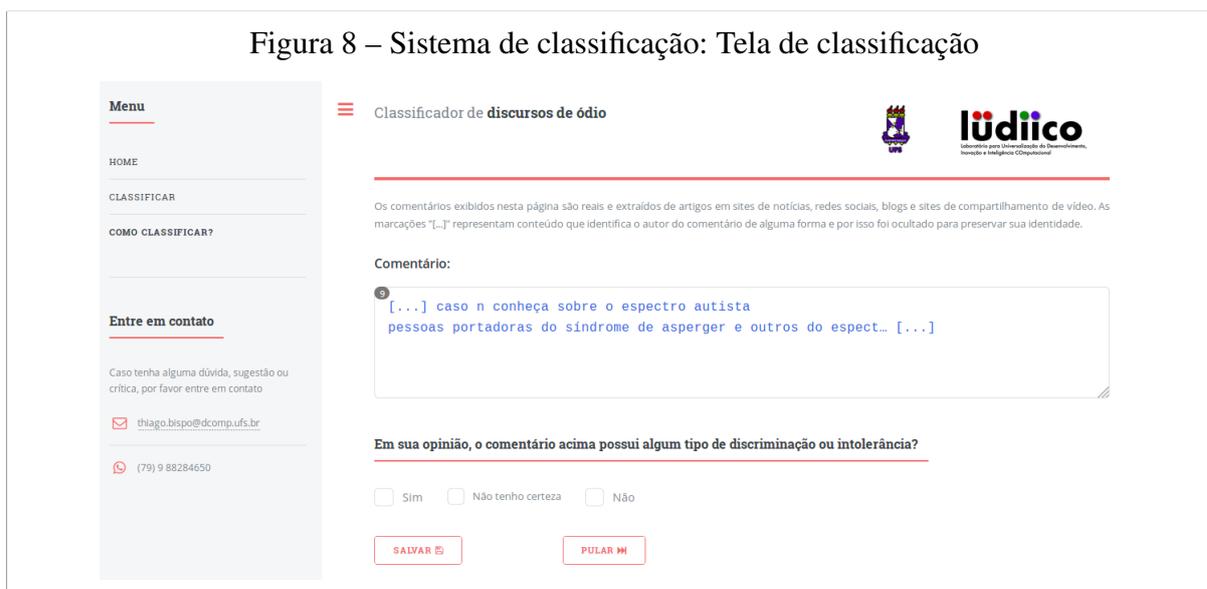
4.4.1 Sistema de classificação

Desenvolvido em Python/Django, o sistema está disponível no endereço ⁸. A Figura 7 exibe sua página de boas-vindas, cuja principal função é incentivar o usuário a colaborar com o projeto. Em todas as páginas, a barra lateral com os menus principais e informações de contato é exibida.

Ainda na Figura 7, podemos notar o menu "COMO CLASSIFICAR?". A função da página para qual esse menu redireciona é auxiliar os usuários na identificação e distinção entre comentários com discursos de ódio e comentários limpos e distinguir outros tipos de ofensas que não são discursos de ódio, aumentando sua probabilidade de tomar a melhor decisão possível no momento da votação.

Após clicar no botão "OK, VAMOS CLASSIFICAR", o sistema é redirecionado para a página de classificação, exibida na Figura 8. Então, o usuário é questionado se o comentário

⁸ <<http://thiagodiasbispo.pythonanywhere.com/>>



1. Comentários com exatamente 2 votos.
2. Comentários com exatamente 1 voto.
3. Comentários votados como indefinido mais de 1 vez, se houver mais que 10 deles.
4. Comentários **contendo ou não** uma das expressões de filtro definidas para os *tweets* e sem qualquer voto.
5. Qualquer comentário já votado.

Os Critérios 1 e 2 foram colocados para maximizar a quantidade de discursos com pelo menos 3 votos, uma vez que sem eles facilmente a maioria dos comentários não seria submetida a uma quantidade suficiente de voluntários dada a quantidade total de comentários presente na base (vide Tabela 7).

O Critério 3, por sua vez, foi usado como forma de minimizar a quantidade de comentários com voto de dúvida. Conforme definimos anteriormente, os próximos 150 comentários não foram repetidos para o mesmo voluntário e, aliado a isso, o Critério 3 faz com que ele tenha a chance votar discursos em que outros voluntários tiveram alguma incerteza.

Usando o Critério 4, objetivamos balancear a quantidade de comentários votados que continham alguma das expressões presentes nas Tabelas 5 e 6 e a quantidade de comentários que não as continham. Dessa forma, mantivemos a chance de aqueles discursos de ódio com discriminações mais sutis serem selecionados.

O Critério 5 garante que sempre exista um comentário candidato para votação, mesmo quando todos já tenham sido votados. Na prática, nunca precisamos usar este critério uma vez que nossa base de dados total é consideravelmente grande e a quantidade de voluntários insuficiente para votar todos os seus comentários.

4.4.3 Resultados da rotulação

Ao longo de período em que os voluntários contribuíram com a rotulação, conseguimos exatamente 7.673 votos no período de 27/09/2017 a 15/05/2018. Os comentários foram selecionados aleatoriamente de forma a aumentar a quantidade de discursos com pelo menos 3 votos, e permitir classificar aqueles comentários já marcados com a opção "Não tenho certeza". Desta forma, sempre que um discurso alcançou pelo menos 3 votos do mesmo tipo, ou seja (3 votos como "abusivos" ou 3 votos como "limpo"), o consideramos classificado.

Percebemos que muitos comentários foram votados diversas vezes como "Não tenho certeza". De acordo com alguns voluntários, eles se deparavam bastante com textos que, sem o contexto ao qual os discursos estavam inseridos, era difícil aferir a presença ou não de conteúdo abusivo.

O total de 1.191 discursos permaneceram com 2 votos e 1.797 com 1 voto, e por essa razão não os incluímos no nosso dataset *final*. Considerando apenas aquelas com pelo menos 3 votos, obtivemos no total 1024 comentários rotulados, dentre os quais 491 foram classificados como abusivos e 533 como limpos.

No total de discursos classificados, detectamos que 299 deles continham alguma das expressão listadas nas Tabelas 5 e 6, representando 24,83% dos 1024 comentários classificados e disponibilizados em nosso *dataset*. Este resultado demonstra que houve uma tendência de nosso método de seleção de escolher mais comentários sem tais expressões, comportamento importante por permitir a exposição em uma proporção maior de comentários com conteúdo abusivo menos comuns aos voluntários.

Na Tabela 8 exibimos o quantitativo de comentários por fonte de coleta. Observamos que todas as fontes das quais houve algum comentário salvo em nossa base de dados foram representadas em nosso *dataset*. Os dados nela sumarizados ilustram que, não por coincidência, a quantidade de *tweets* votados é próxima à quantidade de comentários com alguma das expressões de ódio (299). Esse resultado era esperado, uma vez que os *tweets* foram capturados usando como critério a presença de tais expressões.

Tabela 8 – Quantitativo de discursos por fonte de coleta

Fonte	Quantidade
G1	493
Youtube	132
Twitter	248
Stormfront	92
Facebook	59

4.4.3.1 Tamanho dos comentários

Os comentários apresentaram uma variação considerável de tamanho de sentença, considerando todas as palavras originalmente presentes e desconsiderando sinais de pontuação. Não há diferença significativa entre os comentários abusivos e limpos (Tabela 9). Uma vez que não determinamos a quantidade mínima de palavras do comentário. O tamanho mínimo dos discursos rotulados é 1 para ambas as classes.

Tabela 9 – Estatísticas do tamanho dos comentários abusivos e limpos

	Minimo	Máximo	Mediana	Média
Abusivo	1	119	12	13,12
Limpo	1	88	15	15,75

Ter conhecimento do número de palavras presentes em nosso texto é importante porque assim podemos mensurar o tamanho máximo que os vetores de treinamento precisarão ter para não perdermos informação desnecessariamente.

4.4.3.2 Tamanho do vocabulário

O tamanho do vocabulário foi calculado com base na quantidade total de *unigrams* únicos existentes para a base inteira, para o conjunto de comentários limpos e para os abusivos. Desconsideramos as *hashtags*, links, menções, sinais de pontuações (inclusive *emojis*) e marcas de *retweet* ("RT"). Do total de 1024 mensagens, encontramos 3.607 *unigrams* únicos. Os discursos abusivos têm vocabulário de tamanho 2022. Já os discursos limpos, possuem 2.342 *unigrams* únicos, 13,66% a mais que o vocabulário dos discursos de ódio.

5

Desempenho de modelos usando o *dataset* rotulado

Neste capítulo detalharemos o método adotado para avaliação do *dataset* anotado através de diferentes cenários. Cada cenário consiste de uma combinação de um dos *datasets* usados para treinamento ou teste do modelo LSTM aqui abordado usando distintas formas de pré-processamento. Explicaremos as razões para construção de cada cenário, o desempenho dos modelos usando as métricas adotadas (detalhadas logo em seguida) e análise dos resultados obtidos.

5.1 Métricas de avaliação adotadas

Considerando que **TP** é a quantidade de exemplos positivos corretamente classificados, **FP** é quantidade de exemplos negativos classificadas como positivos e **FN** representa a quantidade de exemplos positivos classificados como negativos, as métricas propostas para avaliar o método apresentado são as seguintes ([ALPAYDIN, 2014](#)):

- **Precisão** - Definida como:

$$Precisao = \frac{TP}{FP + TP} \quad (5.1)$$

Intuitivamente, a precisão mede a capacidade do modelo de não classificar exemplos negativos como positivos. Quanto maior o seu valor, maior a quantidade de exemplos corretamente classificados como positivos.

- **Cobertura** - Definida como:

$$Cobertura = \frac{TP}{TP + FN} \quad (5.2)$$

Em outras palavras, a cobertura mede a eficiência do modelo em "encontrar" todas os exemplos positivos presentes no *dataset*.

- **Medida-F:** Consiste da média harmônica de precisão e revocação. Esta medida é aproximadamente a média de ambas quando seus valores estão próximos.

$$\text{Medida-f} = 2 * \frac{\text{Precisao} * \text{Cobertura}}{\text{Precisao} + \text{Cobertura}} \quad (5.3)$$

Um modelo de alta cobertura e baixa precisão é capaz de classificar muitos exemplos como positivos, porém poucos deles serão de fato positivos (FP). Já um modelo com baixa cobertura e alta precisão, é capaz de classificar poucos exemplos como positivos, mas em contrapartida há uma alta probabilidade dos rótulos positivos estarem corretos (TP).

Num sistema de classificação ideal, altas taxas de precisão e cobertura resultam na maioria dos exemplos positivos sendo classificados corretamente. Quando ambas as métricas são igualmente importantes ou queremos avaliar o balanço entre as duas, a Medida-f é bastante adequada.

Neste trabalho, adotamos como positivos os discursos classificados como abusivos e negativos aqueles que classificados como limpos, ou seja, em que não foi reconhecido nenhum conteúdo discriminatório.

Consideramos que, no mundo real, quanto maior taxa de acertos que um classificador de discursos de ódio tiver acerca do teor abusivo dos comentários que classifica como positivos, melhor. Por esta razão, entendemos que a Precisão é a métrica ideal para avaliar modelos que se proponham a analisar *hatespeech*.

5.2 Pré-processamento

Em alguns trabalhos em PLN, a fase de pré-processamento inclui tanto as etapas de limpeza e tokenização quanto a etapa de vetorização dos dados quando a tarefa exige algum atividade de aprendizado de máquina.

A etapa de limpeza pode incluir diversas subtarefas, dentre as quais: remoção de conteúdo desnecessário como as *stopwords*, correção gramatical, conversão do texto para letras minúsculas, remoção de caracteres ruidosos como aquelas com problema de codificação etc.

A radicalização pode também ser aplicada na etapa de pré-processamento. Com ela, as palavras são reduzidas às suas bases morfológicas ou inflexionadas. Esta tarefa tem por consequência a redução do tamanho do vocabulário do *dataset*. O que é particularmente importante quando precisamos representar vetorialmente os textos como entrada para algoritmos de aprendizado de máquina.

A tokenização corresponde à identificação e separação das partes importantes dos dados de entrada conhecidas em PLN como *tokens*, os quais podem ser palavras, sentenças, caracteres, ou quaisquer informações extraídas do texto como as classes gramaticais das palavras (*POS Tag*).

A vetorização consiste na representação dos textos dentro de um espaço vetorial. É tipicamente realizada como etapa final antes possibilitar uso dos dados nos algoritmos de aprendizagem de máquina.

Neste trabalho, consideramos o pré-processamento como sendo o conjunto de tarefas executadas na preparação dos dados para o processo de vetorização.

O pré-processamento que adotamos foi adaptado de [BADJATIYA et al. \(2017\)](#), que limpam os textos substituindo *tokens* específicos por expressões que os representam. Por exemplo: URLs foram substituídas por '<url>', menções a usuários por '<user>', *hashtags* por '<hashtag>', números por '<number>' e alguns emoticons por seus significados.

Essas representações são interessantes porque preservam a ocorrência de conteúdos que podem ser determinantes em tarefas de categorização de textos como a classificação de discursos de ódio. Configuramos o pré-processamento para fazer substituições de acordo com a língua do *dataset* em que estávamos trabalhando. Assim, em textos de língua portuguesa, menções a usuários, por exemplo, são substituídos por '<usuário>', números por '<números>' etc.

Após essa etapa, os textos são tokenizados com tokenizador de *tweets* do NLTK ¹ as *stopwords* da língua em questão e os sinais de pontuação removidos.

5.3 Ambiente de Execução

Todos os experimentos apresentados foram executados em um cluster ([CCET-UFS, 2017](#)) com arquitetura composta por 5 nós com GPUs, 22 nós sem GPUs e um *master node*. As configurações dos nós com e sem GPU são apresentadas na Tabela 10.

Tabela 10 – Configurações dos nós do *cluster* do LCAD

Nó sem GPU	Nó com GPU
20 Cores em 2 sockets Intel Xeon Ten-Core E5-2660v2 de 2.2-GHz, com 25MB de cache	20 Cores em 2 sockets Intel Xeon Ten-Core E5-2660v2 de 2.2-GHz, com 25MB de cache
Cache, 8 GT/s	Cache, 8 GT/s
64-GB de memória DDR3 1866 MHz	64-GB de memória DDR3 1866 MHz
1 disco de 160-GB SSD	1 disco de 160-GB SSD
1 porta Infiniband QDR 4x 40 Gbps	1 porta Infiniband QDR 4x 40 Gbps
	2 placas NVIDIA Tesla K20

¹ <<http://www.nltk.org/api/nltk.tokenize.html#nltk.tokenize.casual.TweetTokenizer>>

Ademais, a arquitetura aqui proposta foi implementada na linguagem de programação Python3.4, usando o Keras² como ferramenta de prototipagem de alto nível para redes neurais e o TensorFlow executando no *backend* configurado para uso de GPUs e scikit-learn³ como ferramenta auxiliar nos algoritmos de Aprendizado de Máquina tradicional como o GBDT, e a validação cruzada.

5.4 Modelo LSTM

O modelo abordado neste trabalho é adaptado de [BADJATIYA et al. \(2017\)](#). Este é o trabalho Estado da Arte em termos de previsão, acurácia e medida-f na área de classificação de discursos de ódio e mostrou-se bastante adequado para tarefa envolvendo *tweets*. Uma vez que nosso principal dataset (conforme detalhado na Seção 5.5) de treinamento é aquele usado pelos autores supracitados, preferimos adotar o mesmo método de pré-processamento, adaptando-o, contudo, para considerar a língua do texto em questão, visto a base de dados construída neste trabalho é em português.

Na Figura 9, ilustramos a arquitetura do modelo LSMT utilizado. O fluxo de treinamento dos comentários, desde a camada de entrada até a sua efetiva classificação, segue a ordem dos passos discriminados a seguir:

1. **Pré-processamento e vetorização:** Os dados são pré-processados conforme Seção 5.2 e vetorizados de acordo com a necessidade de cada cenário mas, em todos eles, os vetores resultantes são criados de forma a respeitar o tamanho máximo dentre todos os dados de treinamento, tamanho esse representado pela variável *max_sentence_length*.
2. **Camada de entrada:** Uma vez vetorizados, os dados são submetidos em *batches* de tamanho *batch_size*. A função dessa camada é criar uma representação de *embeddings*. Os valores de suas dimensões são calculados através de um distribuição uniforme e a quantidade delas é controlada através do parâmetro *embedding_dim*.
3. **Camada LSTM:** A função desta camada é aprender uma representação adequada para cada comentário submetido, gerando, portanto, *embeddings* específicos voltados ao domínio de discursos de ódio ([BADJATIYA et al., 2017](#)). Essas representações são usadas então para classificação do modelo, conforme detalharemos a seguir. Os parâmetros da LSTM podem ser conferidos na Tabela 11.
4. **Camada densa:** Em modelos *deep*, é comum adicionarmos uma camada totalmente conectada para fins de classificação dos dados de saída da camada neural que a precede. Em nosso modelo, a camada densa consiste de um Multi-Layer Perceptron, o qual é

² <<https://keras.io/>>

³ <http://scikit-learn.org/stable/supervised_learning.html#supervised-learning>

responsável por receber o vetor de dimensão *units* da camada LSTM e determinar sua classe. O erro referente à classificação nesta camada é então propagado para as camadas anteriores e a atualização de seus parâmetros realizada conforme o algoritmo de otimização escolhido. A saída desta camada é submetida à função *softmax*, responsável por calcular a probabilidade para cada classe e com isso, predizer a classe para os comentários recebidos.

Conforme definimos anteriormente, em nossos experimentos usamos a LSTM e a combinação dela com um GBDT. Salvamos o resultado da classificação usando a rede neural e a camada densa MLP (resultado identificado com a palavra "LSTM" após o número do cenário) para então, utilizando o mesmo modelo treinado, extraírmos os *embeddings* aprendidos e os submetemos à GBDT, executando, portanto, uma nova classificação (resultado identificado com a palavra "GBDT" após o número do cenário). Em outras palavras, quando combinamos o modelo LSTM com o GBDT, ignoramos o resultado da classificação da rede neural usando a camada densa e executamos um novo treinamento com a GBDT.

5.4.1 Parâmetros *default*

Os parâmetros *default* do modelo são definidos na Tabela 11.

Tabela 11 – Configuração padrão de parâmetros do modelo LSTM

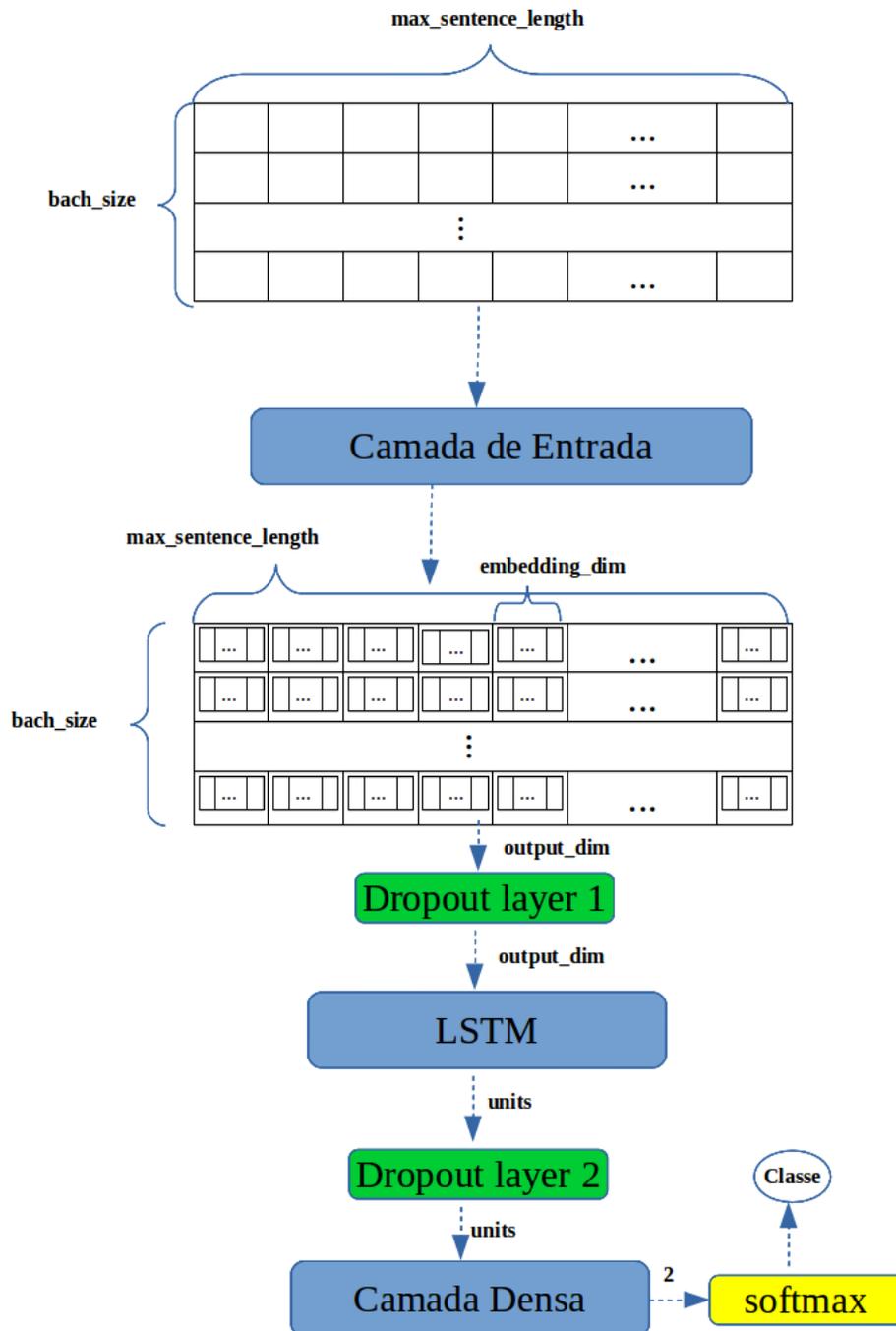
Parâmetro	Descrição	Valor
input_dim	Valor numérico máximo esperado na camada de entrada	10000
output_dim	Tamanho do <i>embedding</i> de palavra a ser gerado	200
input_length	Tamanho do vetor de sentença de entrada	Variável
units	Quantidade de células na camada LSTM	100
weights	Pesos de inicialização da camada de entrada	[]
dropout_rate1	Taxa de <i>dropout</i> da camada de entrada	0,25
dropout_rate2	Taxa de <i>dropout</i> da camada LSTM	0,50
optimizer	Otimizador da função e perda	"rmsprop"
loss_fun	Função de erro	"categorical_crossentropy"
batch_size	Tamanho do <i>mini batch</i>	128

O tamanho máximo adotado para cada sentença vetorizada (*input_length*) foi definido em cada cenário e calculado de forma a respeitar o tamanho máximo dentre os comentários de treinamento e teste.

5.5 Múltiplos *datasets*

Quando executamos experimentos com algoritmos de aprendizado supervisionado, precisamos tomar cuidado com vários aspectos que influenciam em seu desempenho. Um deles

Figura 9 – Modelo LSTM



Fonte: Autoria própria

é o processo de escolha dos dados de treinamento de teste. Para o treinamento, poucos dados pode causar *underfitting*, ou seja, o modelo não terá informações e exemplos suficientes para generalizar na predição.

O oposto disso, o *overfitting*, ocorre quando o modelo se super-especializa nos dados de entrada, tornando-se especializado em prever os exemplos já vistos, porém, provavelmente, será ruim na classificação de exemplos não presentes na base de treinamento.

A fim de executar os processos de treinamento e teste com o modelo LSTM aqui abordado, usamos os *datasets* de WASEEM; HOVY (2016) e FORTUNA (2017) porque foram os únicos publicamente disponibilizados e encontrados. A Tabela 12 lista todas as bases de dados utilizadas no trabalho. A coluna "Nome" refere-se ao nome usado para referenciar o *dataset* neste trabalho. A coluna "Tamanho", exibe quantidade e comentários presentes em cada *dataset*, mostrando o quantitativo de positivos e negativos.

Tabela 12 – Lista de *datasets* usados neste trabalho

Nome	Descrição	Tamanho
discursos_votado	Dataset construído no mestrado	491 neg + 533 pos = 1024
discursos_votados_en	Dataset "discursos_votados" traduzido para o inglês	491 neg + 533 pos = 1024
NAACL_SRW_2016	Dataset de <i>tweets</i> criado por (WASEEM; HOVY, 2016)	11034 neg + 5047 pos = 16081*
NAACL_SRW_2016_pt	Dataset "NAACL_SRW_2016" traduzido para o português	11034 neg + 5047 pos = 16081*
NAACL_SRW_2016_cleaned_pt	Dataset "NAACL_SRW_2016" pré-processado e traduzido, nessa ordem	11034 neg + 5047 pos = 16081*
dataset_portugues	Dataset em português criado por (FORTUNA, 2017)	1977 neg + 547 pos = 2524*

Nota: **neg** = Discurso limpo, **pos**=Discurso abusivo

O NAACL_SRW_2016 é um conjunto de *tweets* rotulados dentre uma de três classes distintas: "sexism"(Sexismo/Misoginia), "racism"(Racismo) e "none"(Nenhum dos dois). Assumimos que os rótulos correspondentes a "none" referem-se a comentários que não contêm discursos de ódio, ou seja, usando a expressão adotada neste trabalho, são comentários limpos. Os comentários rotulados como "racism", ou "sexism" são considerados portanto como comentários abusivos por conterem algum tipo de *hatespeech*. Dessa forma, convertemos a base originalmente criada com 3 tipos de classes diferentes para uma base de rótulos binários.

Originalmente, o NAACL_SRW_2016 possui 16.914 *tweets* rotulados, dentre os quais 3.383 são sexistas, 1.972 racistas e 11.559 não são nenhum dos dois. Contudo, como *tweets* só podem disponibilizados através de seus IDs de acordo com a política de privacidade do Twitter⁴, conseguimos fazer *download* de apenas 16.131 deles. Detectamos então que 25 deles eram duplicados e com rótulos divergentes e os removemos junto com suas duplicatas, uma vez que dados redundantes pode causar *overfitting* e rótulos divergentes dificulta o processo de aprendizagem do modelo.

O *dataset* NAACL_SRW_2016_pt corresponde à base "NAACL_SRW_2016" traduzida para a Língua Portuguesa com o auxílio da API do *Google Translate*. O uso de Sistemas de Tradução automática (STA) como esse não é novidade na tarefa de categorização textual. SILVA et al. (2018) usaram essa abordagem para treinar um modelo CNN baseado em caracteres em um *dataset* de análise de sentimentos traduzido. Os resultados foram equiparáveis aos resultados do mesmo modelo treinado com os dados na Língua Inglesa original.

⁴ <<https://developer.twitter.com/en/developer-terms/agreement-and-policy>>

O dataset_portugues originalmente possui 5.668 *tweets* anotados em diversos tipos e subtipos de *hatespeech*. Somente 2.524 deles estavam disponíveis para download.

5.5.1 Métodos de treinamento

Os cenários aqui descritos foram submetidos a dos métodos de treinamento discriminados abaixo, conforme a necessidade de cada cenário:

- **Método 1** - Validação cruzada: Quando o *dataset* é treinado sobre ele mesmo, ou seja, não há um conjunto de teste independente, o modelo é submetido a uma validação cruzada com 10 *folds* e as métricas calculadas através da média de seus valores em cada *fold*. Portanto, os modelos treinados por esse método podem apresentar valor de Medida-f fora do intervalo definido pela precisão e cobertura apresentados, uma vez que aquela não fora calculada diretamente a partir destes, e sim da média de seu histórico.
- **Método 2** - Quando os dados de treinamento e teste estão pré-definidos: O modelo é treinado com a base de treinamento completa e testado com os dados de teste, calculando-se, assim, as métricas de avaliação.

Em ambos os métodos, o treinamento ocorre em *mini batches* de tamanho *batch_size* (Tabela 11).

5.6 Cenários de experimentação

Separamos os experimentos em cenários para que a evolução deles fique de mais fácil compreensão. Cada cenário corresponde a uma tentativa de aprimoramento dos resultados do cenário anterior ou uma nova abordagem para avaliar o desempenho dos *datasets* da Seção 12 pré-processados, vetorizados ou combinados de maneira específica. Ao todo, foram 24 cenários que serão discriminados a partir de agora.

Na Tabela 13, resumimos os experimentos executados nos cenários de 1 a 4. Inicialmente, testamos o desempenho do modelo LSTM + GBDT na tarefa de classificação dos discursos de ódio presentes em NAACL_SRW_2016 considerando os seus rótulos originais ("racism", "sexism" ou "none") e usando o Método 1 de treinamento, que é executado no Cenário 1, apresentado na Tabela 14, juntamente com o Cenário 2, descrito a seguir. Exibimos tanto o resultado do modelo utilizando o classificador LSTM quanto o GBDT. O desempenho das redes é exibido em termos das métricas descritas na Seção 5.1 e representa o nosso *baseline*.

Nosso objetivo nos Cenários de 1 a 4 foi validar a hipótese de pesquisa acerca da performance da LSTM como modelo cross-lingual usando tão somente o dataset traduzido, sem dados originalmente em Português.

Tabela 13 – Resumo dos experimentos executados nos Cenários de 1 a 4

Cenário	Experimento
1	LSTM treinada com NAACL_SRW_2016 usando o próprio vocabulário para classificação ternária
2	LSTM treinada com NAACL_SRW_2016_pt usando o próprio vocabulário para classificação ternária
3	LSTM treinada com NAACL_SRW_2016 usando o próprio vocabulário para classificação binária
4	LSTM treinada com NAACL_SRW_2016_pt usando o próprio vocabulário para a classificação binária

Tabela 14 – Resultado dos Cenários de 1 a 4

Cenário	Precisão	Cobertura	Medida-F
1 (LSTM)	0,818 (+/- 0,0095)	0,807 (+/- 0,0219)	0,807 (+/- 0,0176)
1 (GBDT)	0,913 (+/- 0,0097)	0,913 (+/- 0,0096)	0,913 (+/- 0,0096)
2 (LSTM)	0,813 (+/- 0,0065)	0,815 (+/- 0,0080)	0,812 (+/- 0,0070)
2 (GBDT)	0,918 (+/- 0,0063)	0,918 (+/- 0,0061)	0,918 (+/- 0,0063)
3 (LSTM)	0,739 (+/- 0,0457)	0,695 (+/- 0,0714)	0,712 (+/- 0,0178)
3 (GBDT)	0,867 (+/- 0,0104)	0,843 (+/- 0,0161)	0,855 (+/- 0,0097)
4 (LSTM)	0,732 (+/- 0,0159)	0,663 (+/- 0,0345)	0,695 (+/- 0,0173)
4 (GBDT)	0,873 (+/- 0,0125)	0,847 (+/- 0,0157)	0,860 (+/- 0,0097)

No Cenário 2, treinamos o modelo com o dataset **NAACL_SRW_2016_pt** a fim de validar sua qualidade para a tarefa de predição de discursos de ódio, também com rótulos ternários. Os resultados equivalem às médias dos valores calculados ao longo da validação cruzada, conforme elucidamos anteriormente. Entre parênteses, constam os desvios padrão do conjunto das métricas calculadas.

Os valores das métricas para cada *fold* foi macro ponderada, de acordo com a quantidade de rótulos para cada classe, como é comum ser feito em categorização multi-classe.

Os Cenários 3 e 4, ainda na Tabela 14, consistem do mesmo experimento executado nos Cenários 1 e 2, respectivamente, porém, desta vez, considerando somente rótulos binários: treinamos o modelo LSTM com os *datasets* **NAACL_SRW_2016** e **NAACL_SRW_2016_pt** para classificar seus comentários como abusivos ou limpos.

Uma vez comprovado o bom desempenho da LSTM treinada com **NAACL_SRW_2016_pt**, executamos nos Cenários de 5 a 11 experimentos usando a mesma rede treinada de diferentes formas com ele e testada com o nosso *dataset*. Os Cenários são resumidos na Tabela 15 e seus resultados listados na Tabela 16. Desta vez, buscamos validar a hipótese de pesquisa de uma maneira diferente, considerando dados originalmente em Português.

Nos Cenários 5 e 6, o modelo LSTM foi treinado com **NAACL_SRW_2016_pt** e testado

com "discursos_votados", conforme o Método 2 de treinamento. Somente as palavras presentes no *dataset* de treinamento foram consideradas no Cenário 5. Em outras palavras, o vocabulário do modelo fora o vocabulário de NAACL_SRW_2016_pt após o pré-processamento e tokenização discutidos na Seção 5.2.

No Cenário 6, o vocabulário considerado fora o do GloVe100: somente as palavras presentes neste modelo foram adicionados ao vetor resultante do processo de vetorização para cada sentença. A hipótese que justifica essa abordagem é a seguinte: usando um vocabulário maior que o do Cenário 6, menor a quantidade de palavras fora do vocabulário no momento da vetorização e assim mais informação um modelo cross-lingual teria para generalizar.

Tabela 15 – Resumo dos experimentos executados nos Cenários de 5 a 11

Cenário	Experimento
5	LSTM treinada com NAACL_SRW_2016_pt, seu próprio vocabulário) e testada com discursos_votados
6	LSTM treinada com NAACL_SRW_2016_pt, vocabulário GloVe e testada com discursos_votados
7	LSTM treinada com NAACL_SRW_2016_cleaned_pt seu próprio vocabulário e testada com discursos_votados
8	LSTM treinada com NAACL_SRW_2016_cleaned_pt, vocabulário Glove e testada com discursos_votados
9	LSTM treinada com NAACL_SRW_2016_cleaned_pt + dataset_portugues , com o vocabulário resultante e testada com discursos_votados
10	LSTM treinada com NAACL_SRW_2016_cleaned_pt + dataset_portugues , com vocabulário GloVe e testada com discursos_votados
11	LSTM treinada com NAACL_SRW_2016, vocabulário próprio e testada com discursos_votados_en

Uma vez que o *dataset* NAACL_SRW_2016 é composto por *tweets* que contém, naturalmente, expressões informais como *hashtags*, menções a usuários, URLs, gírias etc, a sua tradução (NAACL_SRW_2016_pt) contém muitas palavras ainda em inglês desconhecidas.

Por esta razão, criamos o NAACL_SRW_2016_clenaned_pt e usamo-os no Cenário 7. Este *dataset* consiste do NAACL_SRW_2016 pré-processado seguindo o método descrito na Seção 5.2 e então, traduzido. Conforme comentamos anteriormente, o pré-processamento adotado tem a vantagem de não eliminar determinados conteúdos do texto, transformando-os em expressões "bem comportadas".

Para ficar clara a motivação por trás da criação do NAACL_SRW_2016_clenaned_pt, observe o comentário abaixo pertencente a NAACL_SRW_2016 e rotulado como sexista:

Borrowed time #CuntAndArsehole cant wait for you to get_from_file blown away by

Tabela 16 – Resultado dos Cenários de 5 a 11

Cenário	Precisão	Cobertura	Medida-F
5 (LSTM)	0,606	0,274	0,377
5 (GBDT)	0,648	0,280	0,391
6 (LSTM)	0,720	0,126	0,214
6 (GBDT)	0,653	0,176	0,278
7 (LSTM)	0,627	0,283	0,390
7 (GBDT)	0,661	0,300	0,413
8 (LSTM)	0,703	0,156	0,255
8 (GBDT)	0,665	0,223	0,334
9 (LSTM)	0,659	0,171	0,271
9 (GBDT)	0,653	0,240	0,351
10 (LSTM)	0,679	0,167	0,268
10 (GBDT)	0,619	0,161	0,256
11 (LSTM)	0,558	0,281	0,374
11 (GBDT)	0,553	0,285	0,376

the decent teams. #FirstElimination #BeatItDogs #MKR #KatAndAndre (sic)

A tradução do *Google Translator* para o comentário acima é:

O tempo de empréstimo #CuntAndArsehole não pode esperar por você ficar impressionado com as equipes decentes. #FirstElimination #BeatItDogs #MKR #KatAndAndre

O símbolo "#" será eliminado no pré-processamento e as palavras em inglês que o sucedem, permanecerão. Como não são palavras da Língua Portuguesa, no momento da vetorização serão ignoradas e substituídas por um *token* especial que representa uma palavra desconhecida no vocabulário.

Já na frase pré-processada abaixo (ainda em Inglês), notem a substituição das *hashtags* pela palavra "hashtag", fato que por um lado faz com que seu conteúdo (e possível valor semântico) seja eliminado, mas possibilita a inclusão de informação genérica representativa.

borrowed time hashtag cant wait for you to get_from_file blown away by the decent teams. hashtag hashtag hashtag hashtag

A tradução automática do comentário anterior é:

hashtag tempo emprestado não pode esperar por você para ser surpreendido pelas equipes decentes. hashtag hashtag hashtag hashtag

Desta vez, todas as palavras são reconhecidas e representadas por valores específicos no momento da vetorização, retornando uma representação mais significativa que sua versão traduzida sem pré-processamento e enriquecendo o vetor resultante com informações de vizinhança que essas expressões mantidas representam.

Por exemplo: no vetor de índices de vocabulário calculado da expressão acima, após o índice da palavra "descentes", será adicionado o índice da palavra "hashtag". Já com o comentário traduzido da versão original em inglês, as *hashtags* mantidas seriam transformadas num índice especial que representa uma palavra desconhecida, como dissemos anteriormente.

Após esse experimento, no Cenário 8, avaliamos o desempenho da LSTM treinada com NAACL_SRW_2016_cleaned_pt usando o vocabulário do GloVe100, aumentando, assim, o vocabulário do treinamento, da mesma forma que no Cenário 6. Em seguida, experimentamos unir o *dataset* anterior com o *dataset_portugues* e treinar usando o vocabulário resultante (Cenário 9) e o vocabulário GloVe100 (Cenário 10).

Nosso objetivo foi introduzir no conjunto de treinamento discursos abusivos e limpos originalmente em português para atenuar possíveis efeitos negativos que a tradução automática do NAACL_SRW_2016 possa causar no processo de treinamento e, assim, melhorar o desempenho no teste do modelo usando nosso *dataset*, visto que seus comentários são em Língua Portuguesa.

No Cenário 11, experimentamos o inverso do que testamos nos Cenários de 5 a 10: treinamos a LSTM com o NAACL_SRW_2016 e testamos com o nosso *dataset* traduzido automaticamente para o Inglês. Desta vez, o objetivo foi avaliar se as traduções do Português para o Inglês causam menos ruídos do que as traduções do Inglês para o Português, melhorando o desempenho do modelo treinado.

Até o Cenário 11, vetorizamos os comentários através de seus vetores de índices de vocabulário calculados, convertendo através do processo de treinamento esses vetores em *embeddings* dirigidos aos *corpora* de discursos de ódio usados em cada cenário.

Experimentamos também vetorizar os comentários usando duas técnicas tradicionais de vetorização em Aprendizado de Máquina: vetores de frequências de N-grams e vetores TFIDF. Ambas são técnicas *Bag of Words* que permitem calcular a frequência dos top K N-grams que estão presentes em cada exemplo de treinamento e teste. O primeiro representa os comentários usando as frequências em si, ou seja as contagens de ocorrências ou, a depender do caso, a presença ou não de cada sequência relevante para cada comentário. Já o segundo, constrói vetores de forma a representar a importância dos N-gram para cada sentença em relação a todas as outras.

Posto isso, nos Cenários de 12 a 15 (resumidos na Tabela 17), vetorizamos os comentários usando os métodos de contagem de ocorrências de caracteres e vetores TFIDF discutidos anteriormente. Em todos eles, treinamos o modelo LSTM com o *dataset* NAACL_SRW_2016_cleaned_pt e testamos com discursos_votados. Seus resultados encontram-se

na Tabela 18.

No Cenário 12, treinamos com vetores TFIDF usando o próprio vocabulário do *dataset* de treinamento, enquanto que no Cenário 13 usamos vetores de frequências. A seguir, no Cenário 14, experimentamos **somar** os vetores de índices de vocabulário aos vetores de frequência de cada comentário. Já no Cenário 15, concatenamos esses dois vetores para cada exemplo de treinamento e teste. A Tabela 17 resume os experimentos executados para cada um destes cenários.

Tabela 17 – Resumo dos experimentos executados nos Cenários de 12 a 15

Cenário	Experimento
12	LSTM treinada com vetores TF-IDF de <i>N-grams</i> de caracteres de NAACL_SRW_2016_cleaned_pt e testada com discursos_votados
13	LSTM treinada com a frequência dos <i>N-grams</i> de caracteres de NAACL_SRW_2016_cleaned_pt e testada com discursos_votados
14	LSTM treinada com a frequência de <i>N-grams</i> somadas aos vetores de índices de NAACL_SRW_2016_cleaned_pt e testada com discursos_votados
15	LSTM treinada com a frequência de <i>N-grams</i> concatenadas aos vetores de índices de NAACL_SRW_2016_cleaned_pt e testada com discursos_votados

Tabela 18 – Resultado dos Cenários de 12 a 15

Cenário	Precisão	Cobertura	Medida-F
12 (LSTM)	0,0	0,0	0,0
12 (GBDT)	0,0	0,0	0,0
13 (LSTM)	0,0	0,0	0,0
13 (GBDT)	0,417	0,0	0,0
14 (LSTM)	0,530	0,381	0,443
14 (GBDT)	0,563	0,443	0,496
15 (LSTM)	0,0	0,0	0,0
15 (GBDT)	0,0	0,0	0,0

A seguir, na Tabela 19 descrevemos os resultados do *dataset* NAACL_SRW_2016_cleaned_pt usado para treinamento da LSTM e testado com discursos_votados, sem e com o vocabulário GloVe100, nos Cenários 16 e 17, respectivamente. Treinamos também o modelo com o dataset_portugues e testamos com discursos_votados (Cenário 18). Os resultados podem ser conferidos na Tabela 20. Objetivamos nestes cenários validar nossa hipótese de pesquisa treinando ou testando o modelo com outro *dataset* em português que não fosse o nosso.

Tabela 19 – Resumo dos experimentos executados nos Cenários 16,17 e 18

Cenário	Experimento
16	LSTM treinada com <code>NAACL_SRW_2016_cleaned_pt</code> , seu próprio vocabulário e testada com <code>dataset_portugues</code>
17	LSTM treinada com <code>NAACL_SRW_2016_cleaned_pt</code> , vocabulário GloVe e testada com <code>dataset_portugues</code>
18	LSTM treinada com <code>dataset_portugues</code> , seu próprio vocabulário e testada com <code>discursos_votados</code>

Tabela 20 – Resultado dos Cenários de 16, 17 e 18

Cenário	Precisão	Cobertura	Medida-F
16 (LSTM)	0,275	0,402	0,326
16 (GBDT)	0,274	0,349	0,307
17 (LSTM)	0,284	0,325	0,303
17 (GBDT)	0,279	0,369	0,318
18 (LSTM)	0,586	0,146	0,234
18 (GBDT)	0,596	0,191	0,290

A fim de simplificarmos o vocabulário do *dataset* `NAACL_SRW_2016_cleaned_pt`, experimentamos adicionar ao seu pré-processamento a etapa de lematização (MARTIN; JURAFSKY, 2009). Por hipótese, alguns dos resultados anteriores poderiam ser aprimorados se a quantidade de palavras em comum entre o conjunto de treinamento e o conjunto de teste aumentasse. Como a lematização reduz as palavras às suas unidades lexicográficas mais básicas, ela mostrou-se ideal para tentarmos validar nossa hipótese. Os experimentos foram executados nos Cenários de 19 a 23 (Tabela 21) e seus resultados são listados na Tabela 22.

No cenário 24, usamos uma BiLSTM (respeitando a mesma arquitetura da LSTM aqui abordada) para avaliar como ela se comportava no treinamento com o `NAACL_SRW_2016_cleaned_pt` e teste com `discursos_votados` enquanto modelo cross-lingual.

5.7 Análise dos resultados

Nosso objetivo nos Cenários 1 e 2 foi reproduzir parcialmente o experimento usado no artigo Estado da Arte e validar a qualidade da tradução automática de seu *dataset* usando o mesmo modelo LSTM combinado ou não com a GBDT para classificação ternária. Os resultados usando o Cenário 1 semelhantes aos valores alcançados por BADJATIYA et al. (2017) ao usar os modelos combinados: **0,913** em precisão, cobertura e medida-f contra **0,93** do artigo. Somente usando a LSTM inicializada com pesos randômicos, nossos resultados foram sutilmente melhores que os dele: **0,818**, **0,807** e **0,807** contra **0,805**, **0,804**, **0,804** de precisão, cobertura e medida-f,

Tabela 21 – Resumo dos experimentos executados nos Cenários de 19 a 24

Cenário	Experimento
19	LSTM treinada com NAACL_SRW_2016_cleaned_pt lematizada e vocabulário GloVe e testada com dataset_portugues
20	LSTM treinada com o vetores de N-Grams concatenados aos vetores de índices de NAACL_SRW_2016_cleaned_pt lematizada e testada com discursos_votados .
21	LSTM treinada com NAACL_SRW_2016_cleaned_pt lematizado, seu próprio vocabulário e testada com dataset_portugues + discurso_votado lematizado .
22	LSTM treinada com NAACL_SRW_2016_cleaned_pt lematizado, seu próprio vocabulário e testada com dataset_portugues + discurso_votado lematizado .
23	LSTM treinada com NAACL_SRW_2016_cleaned_pt lematizado, seu próprio vocabulário e testada com discursos_votados lematizado.
24	LSTM bidirecional treinada com NAACL_SRW_2016_cleaned_pt , seu próprio vocabulário e testada com discursos_votados .

Tabela 22 – Resultado dos Cenários de 19 a 24

Cenário	Precisão	Cobertura	Medida-F
19 (LSTM)	0,219	0,322	0,261
19 (GBDT)	0,206	0,360	0,262
20 (LSTM)	0,0	0,0	0,0
20 (GBDT)	0,0	0,0	0,0
21 (LSTM)	0,0	0,0	0,0
21 (GBDT)	0,676	0,047	0,088
22 (LSTM)	0,346	0,231	0,277
22 (GBDT)	0,329	0,250	0,284
23 (LSTM)	0,562	0,161	0,251
23 (GBDT)	0,560	0,296	0,388
24 (LSTM)	0,703	0,049	0,091
24 (GBDT)	0,586	0,077	0,136

respectivamente.

A diferença discreta no resultado bem é justificável pelo *dataset* usado neste trabalho não possuir a mesma quantidade de exemplos de sua versão original, visto que alguns *tweets* encontravam-se indisponíveis no momento do seu download e exemplos duplicados foram removidos, conforme explicado anteriormente.

O Cenário 2 usou **NAACL_SRW_2016_pt** para a tarefa de classificação ternária e alcançou resultados superiores ao mesmo modelo treinado o **dataset** original em inglês: **0,918**

contra **0,913** para as métricas adotadas, citando somente o resultado do modelo combinado com a árvore de decisão. O resultado é promissor porque valida a hipótese de que o uso *dataset* traduzido para treinamento do mesmo modelo usado com os dados em inglês garante resultados equiparáveis.

O fato do modelo ter bom desempenho em ambos os *datasets* na classificação em 3 classes distintas, não garante que ele alcance taxas de acerto semelhantes ou mesmo razoáveis na classificação com 2 duas classes. Por esta razão, nos Cenários 3 e 4, executamos os mesmos experimentos dos Cenários 1 e 2 alterando-os apenas para classificação binária. O melhor resultado foi em termos de precisão: **0,867** e **0,873** nos Cenários 3 e 4, respectivamente.

Os resultados deles são inferiores às suas versões executadas com as 3 classes. Acreditamos que a diferença ocorra devido a padrões dos discursos limpos ou abusivos que são melhores reconhecidos quando agrupadas em suas classes originais (sexista, racista ou none), preservando assim as features específicas que as definem. Apesar de diferença, os valores ainda assim são bons, principalmente porque um dos *datasets* é resultado da tradução automática.

Vale ressaltar que o uso das GBDTs melhorou drasticamente todos os resultados da rede LSTM nos cenários comentados anteriormente, demonstrando que sua técnica de *boosting* é eficaz também ao lidar com redes treinadas em *datasets* traduzidos como em nosso caso.

O modelo LSTM e sua combinação a GBDT produziram, portanto, bons resultados, fato que valida nossa hipótese de pesquisa: a mesma arquitetura alcançou bons resultados tanto para os comentários originais em Inglês quanto para sua versão traduzida, caracterizando como cross-lingual. O próximo passo é determinar a validade da hipótese quando o mesmo modelo é avaliado com comentários em Português, como é o caso de nosso *dataset*.

Terminados os experimentos com os *datasets* principais de treinamento, do Cenário 5 ao 11, avaliamos o desempenho do mesmo modelo validado nos Cenários 1 e 2, testando-o agora com nosso *dataset*. Como podemos observar, alcançamos resultados expressivos em vários cenários em termos de Precisão, dentro os quais o melhor dos resultados foi o do Cenário 6, com **0,720** nesta métrica, validando mais uma vez, por conseguinte, nossa hipótese de pesquisa acerca do caráter cross-lingual da arquitetura LSMT.

Outro Cenário de destaque foi o Cenário 8, cujo processo de vetorização considerou somente as palavras do vocabulário GloVe100, fazendo com que *tokens* não presentes neles fossem descartados, mas fornecendo a vantagem de ser um vocabulário muito maior e potencialmente conter mais palavras em comum presentes nos dados de treinamento e teste.

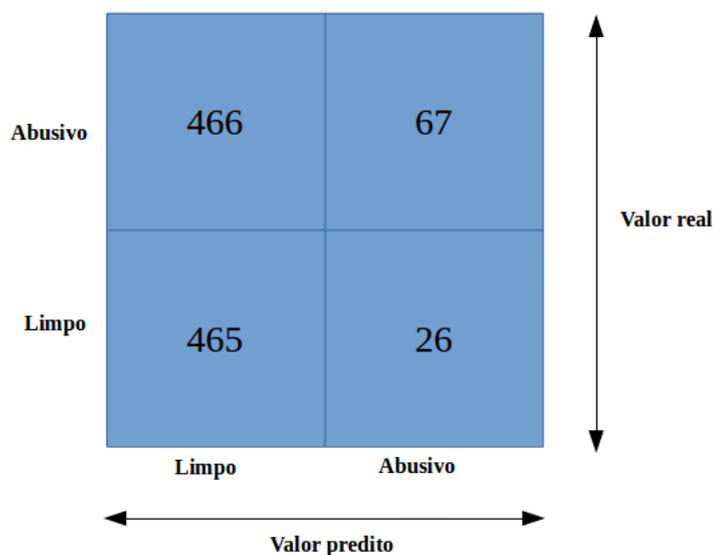
Notemos que o resultado entre os Cenários 6 e 8 são próximos: **0,720** e **0,703**, respectivamente. No caso do Cenário 8, usamos o NAACL_SRW_2016_cleanned_pt com o objetivo de diminuirmos a quantidade de palavras fora do vocabulário no momento da vetorização. Essa abordagem, contudo, não produziu diferenças significativas para justificar o uso da estratégia, contrariando a hipótese estabelecida anteriormente.

O uso da GBDT nem sempre resultou em melhora de desempenho da LSTM, como podemos constatar nos Cenários 6, 8, 9, 10 E 11 cujos valores de precisão decaíram com o uso da Árvore de Decisão. Uma das hipóteses para esse fato é a base de dados não ter sido de tamanho suficiente, nos cenários listados, para a GBDT poder melhorar e estabilizar a classificação da LSTM (Mas detalhes, na Seção 2.3.1).

Diferentemente da precisão, da qual conseguimos resultados razoáveis, os resultados da cobertura em nenhum dos Cenários entre o 5 e o 11 foram bons, permanecendo todos abaixo de 0,5. A medida-F, uma vez que é a média harmônica entre os valores de precisão e cobertura, fora impulsionada para valores também abaixo de 0,5, como consequência do comportamento da cobertura em todos os cenários.

Observando a Equação da cobertura (5.2), podemos notar que a razão de seus valores baixos é normalmente altas taxas de Falsos Negativos (FN). Quando maior for seu valor, maior será o resultado do denominador e menor o valor final da fração. Tomando como exemplo o Cenário 6, podemos facilmente identificar este comportamento através de sua matriz de confusão (Figura 10) calculada a partir dos valores preditos pela rede LSTM (sem a GBDT):

Figura 10 – Matriz de confusão do Cenário 6 usando o resultado da LSTM



Por convenção, na matriz de confusão as linhas representam valores verdadeiros positivos (comentários abusivos) e verdadeiros negativos (comentários limpos). Já as colunas representam os valores preditos pela rede. Cruzando linhas e colunas, estabelecemos as relações de que precisamos para calcular as métricas que usamos neste trabalho. 466 comentários foram classificados como limpos quando na verdade eles são abusivos (Falsos negativos).

Contudo, conforme defendemos anteriormente, consideramos a precisão como sendo a métrica mais importante para modelos de detecção de discursos de ódio. Neste sentido, os valores de precisão dos Cenários 6 e 7 são indicativos da capacidade do modelo de reconhecer de fato discursos abusivos.

Nos Cenários de 12 a 15, experimentamos algumas *features* clássicas de aprendizado de máquina associados à LSTM. Nosso objetivo foi avaliar se sequências de ocorrências de N-Grams de caracteres (Cenários 13, 14 e 15) treinadas por uma rede recorrente poderia causar o aprendizado das sequências dessas ocorrências, associando, assim, o modelo BoW com um modelo cuja ordem das sequências de entrada é importante. No Cenário 12, testamos inicialmente o comportamento da rede usando vetores TFIDF. Nenhum dos cenários apresentou resultado satisfatório, conforme podemos observar na Figura 17.

Podemos concluir que vetores de ocorrências de sequências caracteres, isoladamente, não agregam informações o suficiente para permitir o tipo de aprendizado que a LSMT exige. Notemos que, no Cenário 14, houve algum ganho de informação em relação aos outros, por mais que irrisório. Somente nele, levamos em consideração os vetores de índices de palavras, através de sua soma aos vetores de frequências de N-Grams. Provavelmente por esta razão, a rede conseguiu obter algum aprendizado usando as porções do vetor de entrada da rede que continha os índices de palavras do vocabulário.

Nos Cenários 16 e 17, experimentamos testar o modelo LSTM com o dataset_portugues para o avaliar o seu desempenho usando-o como dados de teste. Os resultados foram menores até em termos de precisão, a qual ficou abaixo de 0,3. Mesmo a LSTM treinada como o dataset_portugues e testada com nosso *dataset* (Cenário 18) teve resultados insatisfatórios, ficando abaixo do 0,6.

Esse resultado era esperado, uma vez que a quantidade de dados treinamento é muito pequena em relação à quantidade de parâmetros a serem treinados (mesmo não treinando os *embeddings* na LSTM), que somam neste Cenário 80.602. Quando esse tipo de problema ocorre, nos deparamos com maldição da dimensionalidade (CHRISTOPHER, 2016).

Nos Cenários 19 a 23, avaliamos o desempenho do modelo LSTM usando em alguns casos a técnica de lematização e no Cenário 24 aplicamos o *dataset* NAACL_SRW_2016_cleaned_pt num modelo BiLSTM. Os melhores resultados alcançados em termos de precisão foram nos Cenários 21 e 24. No primeiro, testamos a rede unindo nosso dataset lematizado como dataset_portugues usando o próprio vocabulário dos dados de treinamento e alcançamos precisão de **0,676**. No segundo, a rede LSTM alcançou **0,703**. Neste cenário, a aplicação da GBDT resultou na piora da performance da rede. Já no Cenário 21, a rede saiu de precisão 0,0 para **0,676** somente com o uso da árvore de decisão.

No cenário 20, notem que, novamente, a rede não conseguiu aprender com o uso de sequências de N-Grams mesmo com os dados de entrada lematizados, reforçando o indício de esse tipo de representação de dados não é adequada para modelos recorrentes.

O resultado do Cenário 24 sugere que o modelo BiLSTM comporta-se também como modelo cross-lingual, confirmando nossa hipótese de pesquisa mais uma vez, assim como a LSTM tradicional. Sua precisão é equiparável ao nosso melhor Cenário (Cenário 6), apesar do

uso do GBDT associado a ele não ter causado melhora no desempenho.

Os cenários de 1 a 4 são promissores: eles demonstram que o modelo LSMT abordado é robusto no reconhecimento de discursos de ódio mesmo em *datasets* traduzidos automaticamente, confirmando que ele pode ser usado tanto para a língua portuguesa, quanto para a inglesa, sem nenhum tipo de configuração especial.

Contudo, o desempenho nos cenários seguintes não foram tão bons quanto esses 4, sendo notória a tendência de alguns modelos e técnicas de pré-processamento de alcançarem alta precisão porém baixa cobertura de maneira geral. Fato que confirma que os modelos estão classificando corretamente o que de fato considera discursos de ódio.

Os resultados também sugerem que trabalhar em uma base de comentários com múltiplos tipos de discursos de ódio mas tratados como se fossem todos discursos limpo ou abusivo, pode ser desafiador. Os diversos padrões inerentes a cada tipo provavelmente exige mais dados de treinamento e teste, e de preferência dados com tipos ou subtipos de discursos compatíveis entre si.

É provável que se nossa base de dados contivesse somente discursos sexistas, racistas ou limpos, alguns dos cenários alcançassem resultados de precisão e cobertura superiores quando treinados com a base NAACL_SRW_2016_pt.

Em resumo, constatamos os seguintes itens depois dos experimentos realizados:

1. **Bases homogêneas:** Baseado nos Cenário 1, 2, 3 e 4: Bases de treinamento e teste com os mesmos tipos de discursos tendem a alcançar melhores resultados.
2. **As GBDTs são robustas:** Sua técnica de *boosting* permite estabilizar e melhorar classificações mesmo em cenários difíceis como os que envolvem *datasets* de línguas diferentes, na maioria das vezes.
3. **Vocabulário externo:** O uso de vocabulário externo, apesar do custo computacional quando ele é muito grande, é decisivo para vetorizar os dados de entrada, principalmente na diminuição da quantidade de palavras não reconhecidas na vetorização.
4. **Vetores de ocorrência de sequências de palavras:** Isoladamente, eles não representam boas *features* para modelos recorrentes como a LSTM.

6

Conclusão

Neste trabalho, exploramos alguns modelos com o objetivo de classificar comentários de usuários recolhidos na Internet como sendo discursos de ódio ou não. Usamos uma base de dados em inglês para nos auxiliar neste objetivo e exploramos em diversos cenários diferentes maneiras de pré-processamento e vetorização dos dados a fim de alcançar nosso objetivo.

Treinamos em diversos experimentos modelos cross-lingual para detecção de discursos de ódio (objetivo geral de nosso trabalho). Devido à falta de *datasets* de discursos de ódio rotulados em Português, criamos e rotulamos nosso próprio com a ajuda de voluntários, alcançando nosso primeiro objetivo específico. Dividimos os experimentos em cenários e em cada um deles avaliamos o desempenho dos modelos quando submetidos aos diferentes dados de entrada usados, principalmente o *dataset* de discursos de ódio criado neste trabalho, tarefa que consistia em nosso segundo objetivo específico, alcançando uma precisão de **0,720**.

Consideramos este trabalho importante na medida em que explora técnicas para automatizar e auxiliar a detecção de discursos de ódio, conteúdo cuja presença nas redes sociais é cada vez mais comum, ofendendo e discriminando suas vítimas. Usamos a técnica Estado da Arte, originalmente para comentários em Inglês, e demonstramos de que forma ela pode ser adaptada para trabalhar com textos em Português, além de demonstrarmos caminhos pouco promissores nesse quesito, ou seja, caminhos cujos resultados não são bons.

Para tanto, exploramos diversas formas de pré-processamento e vetorização dos *datasets* usados e os submetidos ao modelo *deep* LSTM e sua variante, a BiLSMT, detectando quais dessas abordagens representadas pelos 24 cenários criados eram as mais promissoras dentro do escopo de nosso trabalho, servindo como referência para outros pesquisadores e, com isso, concluindo nosso terceiro e último objetivo específico.

Em resumo, nossas contribuições foram: (i) treinamento de modelos cross-lingual através do uso de uma base de dados em inglês para classificação automática de discursos de ódio em português, atenuando o impacto que a falta de *datasets* nessa língua causa na área, (ii) criação e

rotulação de base de dados de discursos de ódio em português (iii), determinação das técnicas de pré-processamento e vetorização mais promissoras usando *dataset* em inglês, servindo como referência a pesquisadores na área detecção de discursos de ódio interessados em modelos cross-lingual.

Como trabalho futuro, consideramos a investigação de *features* de natureza semântica como viés para classificação de discursos de ódio. Alguns artigos encontrados usam informações semânticas para iterativamente encontrar mais sentenças semelhantes às já encontradas com base em palavras enriquecidas por informações do WordNet, por exemplo.

Ademais, uma vez que *corpus* rotulados de discursos de ódio são extremamente raros, trabalhar com aprendizagem não supervisionada ou semi-supervisionada como em [XU et al. \(2017\)](#) parece bastante adequado para pré-identificação de comentários abusivos. Outra área bastante promissora são os Frames semânticos ([BARREIRA; PINHEIRO; FURTADO, 2017](#)). Eles são ideais para problemas como identificação de discursos de ódio, tanto por não exigir bases de dados enormes quanto por identificar facilmente novos padrões de discriminação e ofensa, característica apreciada uma vez que a variação na língua e nas expressões de ódio variam constantemente no mundo digital.

Referências

- ALLAN, t. B. R. *Hard Questions: Hate Speech*. 2017. Disponível em: <<https://newsroom.fb.com/news/2017/06/hard-questions-hate-speech/>>. Citado na página 15.
- ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2014. Citado na página 48.
- BADJATIYA, P. et al. Deep learning for hate speech detection in tweets. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 26th International Conference on World Wide Web Companion*. [S.l.], 2017. p. 759–760. Citado 8 vezes nas páginas 17, 24, 34, 35, 36, 50, 51 e 61.
- BARREIRA, R.; PINHEIRO, V.; FURTADO, V. Framefor—uma base de conhecimento de frames semânticos para perícias de informática (framefor—a knowledge base of semantic frames for digital forensics)[in portuguese]. In: *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*. [S.l.: s.n.], 2017. p. 171–180. Citado na página 68.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 27.
- CCET-UFS. Laboratório de computação alto desempenho. In: . Universidade Federal de Sergipe, 2017. Disponível em: <<http://www2.lcad.ufs.br/>>. Citado na página 50.
- CHRISTOPHER, M. B. *PATTERN RECOGNITION AND MACHINE LEARNING*. [S.l.]: Springer-Verlag New York, 2016. Citado na página 65.
- CQM. *Quando a intolerância chega às redes*. 2016. Disponível em: <<http://www.comunicaquemuda.com.br/dossie/quando-intolerancia-chega-as-redes/>>. Citado na página 36.
- DJURIC, N. et al. Hate speech detection with comment embeddings. In: ACM. *Proceedings of the 24th International Conference on World Wide Web*. [S.l.], 2015. p. 29–30. Citado na página 34.
- FORTUNA, P. C. T. Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes. 2017. Citado 9 vezes nas páginas 15, 17, 30, 31, 35, 36, 39, 42 e 54.
- FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. *The elements of statistical learning*. [S.l.]: Springer series in statistics New York, 2001. v. 1. Citado na página 27.
- FRIEDMAN, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, Elsevier, v. 38, n. 4, p. 367–378, 2002. Citado na página 27.
- GAO, L.; HUANG, R. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*, 2017. Citado na página 36.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado 4 vezes nas páginas 17, 19, 22 e 25.

- HAO, P.-Y.; CHIANG, J.-H.; TU, Y.-K. Hierarchically svm classification based on support vector clustering method and its application to document categorization. *Expert Systems with applications*, Elsevier, v. 33, n. 3, p. 627–635, 2007. Citado na página 35.
- HARTMANN, N. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*, 2017. Citado 2 vezes nas páginas 25 e 26.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 22.
- KUHN, M. et al. Caret package. *Journal of statistical software*, v. 28, n. 5, p. 1–26, 2008. Citado na página 35.
- LANDEGHEM, J. V. A survey of word embedding literature. 2016. Citado na página 25.
- LIU, S. et al. A recursive recurrent neural network for statistical machine translation. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [S.l.: s.n.], 2014. v. 1, p. 1491–1500. Citado na página 24.
- MARSLAND, S. *Machine Learning: An Algorithmic Perspective*. [S.l.]: CRC Press, 2014. Citado na página 26.
- MARTIN, J. H.; JURAFSKY, D. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. [S.l.]: Pearson/Prentice Hall, 2009. Citado na página 61.
- MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Citado na página 26.
- MOURA, M. A. *O Discurso do Ódio em Redes Sociais*. [S.l.]: Lura Editorial (Lura Editoração Eletrônica LTDA-ME), 2016. Citado 2 vezes nas páginas 29 e 30.
- NOBATA, C. et al. Abusive language detection in online user content. In: INTERNATIONAL WORLD WIDE WEB CONFERENCES STEERING COMMITTEE. *Proceedings of the 25th International Conference on World Wide Web*. [S.l.], 2016. p. 145–153. Citado 5 vezes nas páginas 15, 33, 34, 36 e 37.
- ONU. *INFORMATION ECONOMY REPORT: Digitalization, trade and development*. [S.l.], 2017. 130 p. Citado na página 15.
- PARK, J. H.; FUNG, P. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*, 2017. Citado 2 vezes nas páginas 35 e 36.
- PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543. Citado na página 25.
- SAFETY, T. *A Calendar of Our Safety Work*. 2017. Disponível em: <https://blog.twitter.com/official/en_us/topics/company/2017/safetycalendar.html>. Citado na página 15.
- SCHMIDT, A.; WIEGAND, M. A survey on hate speech detection using natural language processing. In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics, Valencia, Spain*. [S.l.: s.n.], 2017. p. 1–10. Citado 4 vezes nas páginas 15, 35, 40 e 41.

- SEGUNDO, A. d. H. C. *Questão de Opinião?* [S.l.]: Lumen Juris, 2016. Citado 3 vezes nas páginas 29, 30 e 31.
- SILVA, L. A. et al. Analyzing the targets of hate in online social media. In: *ICWSM*. [S.l.: s.n.], 2016. p. 687–690. Citado na página 32.
- SILVA, R. P. da et al. Cross-language approach for sentiment classification in brazilian portuguese with convnets. In: LATIFI, S. (Ed.). *Information Technology - New Generations*. Cham: Springer International Publishing, 2018. p. 311–316. ISBN 978-3-319-77028-4. Citado 2 vezes nas páginas 16 e 54.
- SINGHAL, P.; BHATTACHARYYA, P. *Sentiment analysis and deep learning: a survey*. 2016. Citado na página 24.
- SOPRANA, P. *Há um aumento sistemático de discurso de ódio na rede, diz diretor do SaferNet*. 2017. Disponível em: <<https://epoca.globo.com/tecnologia/experiencias-digitais/noticia/2017/02/ha-um-aumento-sistematico-de-discurso-de-odio-na-rede-diz-diretor-do-safernet.html>>. Citado na página 15.
- SUTTON, C. D. Classification and regression trees, bagging, and boosting. *Handbook of statistics*, Elsevier, v. 24, p. 303–329, 2005. Citado na página 28.
- WASEEM, Z.; HOVY, D. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: *Proceedings of the NAACL Student Research Workshop*. San Diego, California: Association for Computational Linguistics, 2016. p. 88–93. Disponível em: <<http://www.aclweb.org/anthology/N16-2013>>. Citado 4 vezes nas páginas 17, 34, 35 e 54.
- XU, Z. et al. Semi-supervised learning in large scale text categorization. *Journal of Shanghai Jiaotong University (Science)*, Springer, v. 22, n. 3, p. 291–302, 2017. Citado na página 68.



UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do
Curso de Mestrado em Ciência da Computação-UFS.
Candidato: THIAGO DIAS BISPO

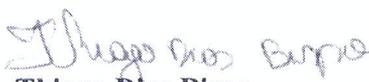
Em oito dias do mês de junho do ano de dois mil e dezoito, com início às 09h00min, realizou-se na Sala de Seminários do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Thiago Dias Bispo**, que desenvolveu o trabalho intitulado: “*Arquitetura LSTM para classificação de discursos de ódio cross-lingual Inglês-PtBR*”, sob a orientação do Prof. Dr. Hendrik Teixeira Macedo. A Sessão foi presidida pelo Prof. Dr. Hendrik Teixeira Macedo (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. Carlos Alberto Estombelo Montesco (PROCC/UFS) e, em seguida, a Prof^a. Dr^a. Vlândia Célia Monteiro Pinheiro (UNIFOR). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando APROVADO “(aprovado/reprovado)” COM “(com/sem)” ressalvas. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEP), e da Resolução nº 25/2014/CONEP que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária “Prof. José Aloísio de Campos”, 08 de Junho de 2018.


Prof. Dr. Hendrik Teixeira Macedo
(PROCC/UFS)
Presidente


Prof. Dr. Carlos Alberto Estombelo Montesco
(PROCC/UFS)
Examinador Interno


Prof. Dr. Vlândia Célia Monteiro Pinheiro
(UNIFOR)
Examinador Externo


Thiago Dias Bispo
Candidato

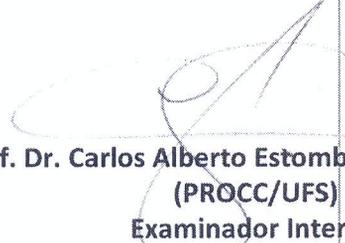


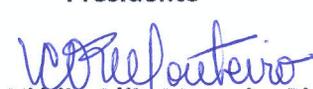
UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Observações (em caso de aprovação com ressalvas):

- ① REVISÃO DA CORRETEDE LINGUÍSTICA E USO APROPRIADO DE CONSTRUÇÕES
- ② INCLUSÃO DE QUESTÕES/HIPÓTESES DE PESQUISA
- ③ CRIAÇÃO E DETALHAMENTO DE QUADRO COMPARATIVO DOS TRAB. RELACIONADOS
- ④ FORMALIZAR MÉTODO DE COLETA E ROTULAÇÃO
- ⑤ QUALIFICAR O DATASET
- ⑥ AMARRAR QUESTÕES/HIPÓTESES DE PESQUISA NOS CENÁRIOS
- ⑦ CORRIGIR ARQUITETURAS LSTM-MLP vs. LSTM-GBDT
- ⑧ CONTRIBUIÇÕES NA CONCLUSÃO


Prof. Dr. Hendrik Teixeira Macedo
(PROCC/UFS)
Presidente


Prof. Dr. Carlos Alberto Estombelo Montesco
(PROCC/UFS)
Examinador Interno


Prof.ª. Dr.ª. Vlândia Célia Monteiro Pinheiro
(UNIFOR)
Examinador Externo


Thiago Dias Bispo
Candidato