Detecção de discurso de ódio em português usando CNN combinada a vetores de palavras

Samuel C. Silva e Adriane B. S. Serapião

Universidade Estadual Paulista (UNESP) - IGCE/DEMAC
sam.kaetano@gmail.com, adriane@rc.unesp.br

Abstract. The current work has proposed to study and to implement a convolutional neural network (CNN) allied to pre-trained (Wang2Vec and GloVe) and trainable word embeddings for hate speech detection in Portuguese. For sake of comparison, the implementation used different gradient descent optimizer functions (RMSprop, Adagrad, Adadelta and Adam), aiming to contrast the performance at each function. For such task, it were used three datasets of comments in Portuguese, annotated as offensive or not offensive. We have concluded that using this proposed approach the results were superior to those from the baseline, achieving higher F-score and accuracy measures.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

Keywords: convolutional neural networks, hate speech, natural language processing

1. INTRODUÇÃO

Aprendizado Profundo de Máquina ou *Deep Learning* (DL) [LeCun et al. 2015] é uma nova área de pesquisa baseada no conceito de redes neurais artificiais, cujas recentes descobertas contribuíram com sua popularização e permitiram que antigos problemas computacionais fossem novamente abordados, tornando possível atingir bons resultados na resolução desses problemas em comparação com outras técnicas tradicionais de Inteligência Artificial. A combinação das técnicas de DL nas tarefas de Processamento de Linguagem Natural (PLN), por exemplo, têm se mostrado relevantes no sentido de melhorar os resultados em tarefas como sumarização de documentos, reconhecimento de fala, análise de sentimento e sistemas de pergunta-resposta.

Em PLN, a detecção de discurso de ódio tem se convertido em um tópico de interesse científico e social recentemente devido à grande audiência e influência que as mídias sociais exercem na sociedade atual [Almeida et al. 2017]. As redes sociais virtuais têm se tornado cada vez mais presentes na sociedade moderna e cada vez mais as pessoas fazem uso dessas plataformas de convívio virtual para se expressarem e comunicarem-se. O excessivo uso dessas plataformas, como principalmente o Facebook, o Instagram, o Twitter e o WhatsApp, permitem análises sociais relevantes para a comunidade científica, permitindo, num certo grau, um retrato bastante preciso da realidade das pessoas. Desse modo, a disseminação de discursos ofensivos dirigidos à minorias e grupos políticos é frequente neste ambiente virtual e difícil de ser tratada ou até mesmo evitada.

Segundo Cohen-Almagor [2011], um discurso de ódio se caracteriza por ser um discurso malicioso, enviesado, hostil e preconceituoso dirigido a grupos específicos por conta de gênero, etnia, religião, nacionalidade, deficiência física ou mental, orientação sexual e condicionamento físico. Uma mensagem com discurso de ódio é definida assim ao possuir palavras de ódio. O discurso de ódio envolve enormes perigos para a sociedade, uma vez que textos *online* altamente raivosos podem ser usados

para ataques pessoais, assédio *online* e comportamentos de *bullying*. Em decorrência, a comunidade científica tem realizado tentativas nos últimos anos de identificar um modelo eficiente para a previsão desses comentários [Georgakopoulos et al. 2018]. No entanto, esses estudos ainda carecem de novas abordagens e estruturas, no sentido de se aumentar a corretude desses modelos na identificação automática de discursos de ódio, isso é ainda mais evidente tratando-se da língua portuguesa.

A quantidade de trabalhos relativos ao idioma português em tarefas de PLN, como detecção de discurso de ódio, é baixa, sendo a maioria das pesquisas realizadas em inglês. Entre os estudos em português, são poucos os pesquisadores que disponibilizam seus conjuntos de dados à comunidade científica. Por isso, é importante que haja contribuição através de *scripts*, algoritmos ou conjuntos de dados públicos em português para que a pesquisa em PLN nesta língua também possa se desenvolver [Almeida et al. 2017].

Neste sentido, o presente trabalho buscou explorar conjuntos de dados públicos em português, com o objetivo de contribuir para a detecção de discurso de ódio neste idioma, que foi o 5º mais utilizado na internet global em 2017[1]. Como abordagem para tratar o problema, utilizou-se um modelo de DL, as *convolutional neural networks* (CNN) [LeCun et al. 2015]. Essas foram primeiramente usadas em aplicações em processamento de imagens devido às suas características inerentes de extrair propriedades estatísticas de estrutura e caracterização de imagens, porém, mais recentemente elas têm sido utilizadas em tarefas de PLN [Lopez and Kalita 2017].

O artigo está organizado como segue. A Seção 2 apresenta alguns trabalhos correlatos sobre o mesmo tema, a Seção 3 indica os procedimentos metodológicos utilizados, a Seção 4 exibe os resultados obtidos e a Seção 5 termina com as considerações finais e conclusões sobre o trabalho.

2. TRABALHOS RELACIONADOS

Entre os trabalhos relacionados à tarefa de detecção de discurso de ódio, poucos são aqueles que trabalham com conjuntos de dados em português e aqueles que assim o fizeram, geraram seus próprios conjuntos de dados. Pelle and Moreira [2017] recolheram comentários do site de notícias brasileiro g1.com.br e realizaram a detecção de discurso de ódio através de métodos de aprendizado supervisionado clássicos como *Support Vector Machines* (SVM) e *Naive Bayes* (NB). O método SVM foi o que permitiu obter maior F-*score*. Fortuna [2017] coletou tuítes em português da plataforma do Twitter a fim de realizar a detecção de discurso de ódio por meio de um sistema de classificação hierárquica, através de SVMs modificadas (*SVMLinear*), com o objetivo de captar melhor as nuances que compõem o discurso discriminatório.

Kim [2014] propõe o uso de técnicas de DL para a tarefa de classificação de sentenças, alcançando resultados consideravelmente bons em conjuntos de dados na língua inglesa. Almeida et al. [2017] realizam a detecção de ódio por meio de técnicas de Teoria da Informação (entropia e divergência). Nobata et al. [2016], a partir da detecção de discurso de ódio, disponibilizaram um *corpus* de palavras abusivas em inglês; segundo os autores, os primeiros a proporem tal estratégia. Zhang et al. [2018] arquitetaram uma CNN somada a uma rede neural recorrente para detecção do discurso de ódio e apesar dos resultados obtidos, este estudo não foi comparado a Zhang et al. [2018] devido às diferenças nos modelos e pelo uso de conjuntos de dados distintos. Pitsilis et al. [2018] aplicaram redes neurais recorrentes à essa tarefa. Malmasi and Zampieri [2017] identificaram discurso de ódio em mídias sociais utilizando SVM. Schmidt and Wiegand [2017] também realizaram estudos sobre técnicas de detecção de discurso de ódio com tuítes.

Nosso trabalho se destaca dos demais por (i) aplicar CNN na classificação de conjuntos de dados em língua portuguesa e (ii) comparar resultados entre diferentes configurações do modelo de CNN utilizado.

[1]Disponível em: `https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/`

3. METODOLOGIA

No presente trabalho aplicou-se dois modos de treinamento da rede neural: (i) CNN com vetores de palavras pré-treinados e (ii) CNN sem vetores de palavras pré-treinados. Os vetores de palavras pré-treinados utilizados tinham dimensões distintas, que contribuíram para a análise da melhor configuração para essa arquitetura. De cada conjunto de dados foram extraídos 10% de seus comentários (sentenças) para o conjunto de teste. O treinamento ocorreu em 10 dobras (*folds*) com validação cruzada, usando o embaralhamento em *mini-batch* e gradiente descendente com diferentes funções de otimização. Os modelos aqui treinados estão disponível em `https://drive.google.com/open?id=17XlVuFSdB-J8-PhYOBKl5mAgbHJLeYrn`.

3.1 Conjuntos de dados

Pelle and Moreira [2017] propuseram dois conjuntos de dados contendo comentários ofensivos (e não ofensivos) de um portal de notícias brasileiros, o OffComBr[2]. O processo de anotação dos comentários se deu através de três juízes humanos que permitiram a geração de dois conjuntos. Apesar do processo de coleta dos comentários ter obtido 10.366 comentários, os autores o limitaram a 1.250 amostras aleatórias. Esses comentários selecionados foram categorizados entre as classes "racismo", "sexismo", "homofobia", "xenofobia", "intolerância religiosa", "xingamento" e "não ofensivo". Embora os autores tenham realizado essa rotulação multi-classe, o formato do conjunto de dados disponibilizado pelos mesmos possui rotulação binária, identificando apenas "ofensivo"e "não ofensivo". A partir dessa análise, gerou-se o OffComBr-2, que contém 1.250 comentários que foram anotados como ofensivos ou não ofensivos por no mínimo dois juízes. Enquanto que o OffComBr-3 contém 1.033 comentários, que foram anotados pelos três juízes. Ao todo, o OffComBr-2 possui 419 comentários identificados como ofensivos, 33,5% do total de seus comentários, e o OffComBr-3 possui 202 comentários identificados como ofensivos, 19,5% do total.

Fortuna [2017] propôs o *Hate Speech Dataset*[3] (HSD), um conjunto de dados anotado hierarquicamente composto por tuítes em português que foram extraídos da plataforma do Twitter através de (i) perfis de usuários específicos e (ii) palavra-chaves. Em (i), foram listados perfis conhecidos por postarem tuítes ofensivos sobre diferentes assuntos. Segundo o autor, esses perfis foram listados através de buscas pelas palavra-chaves "*hate*", "*hate speech*" ou "*offensive*". Em (ii), foram listadas palavra-chaves comumente relacionadas ao discurso de ódio pela literatura, de modo a obter *hashtags*, perfis e outras palavra-chaves que se relacionassem ao discurso de ódio. Obtiveram-se 42.390 tuítes ao final do processo, mas o conjunto de dados foi reduzido para 5.668 tuítes após o pré-processamento. Este conjunto de dados foi anotado por dois juízes humanos. Dos tuítes totais que compõem o HSD, 1.228 são classificados como discurso de ódio, 22% do conjunto de dados. Apesar do HSD ter sido anotado com as classes de ódio ("sexismo", "homofobia", "racismo", entre outros), para facilitar o trabalho e a detecção do discurso de ódio neste estudo, foi adotada uma rotulação binária para todos os tuítes deste conjunto de dados, passando a considerar apenas "ofensivo" ou "não ofensivo", ao invés da classe de ódio específica. Essa abordagem torna a comparação entre o HSD e os OffComBrs similar, de modo a permitir a classificação binária entre todos os conjuntos de dados.

3.2 Arquitetura da CNN

Kim [2014] realizou a implementação de CNN aliada a vetores de palavras pré-treinados (Word2Vec) para classificação de documentos. Seu trabalho classificou diversos conjuntos de dados em inglês e obteve resultados significativos. Yin et al. [2017] realizaram um estudo comparativo entre CNN e modelos de redes neurais recorrentes em tarefas de PLN, seus resultados mostraram que houve

[2]Disponível em: `https://github.com/rogersdepelle/OffComBR`
[3]Disponível em: `https://rdm.inesctec.pt/dataset/cs-2017-008`

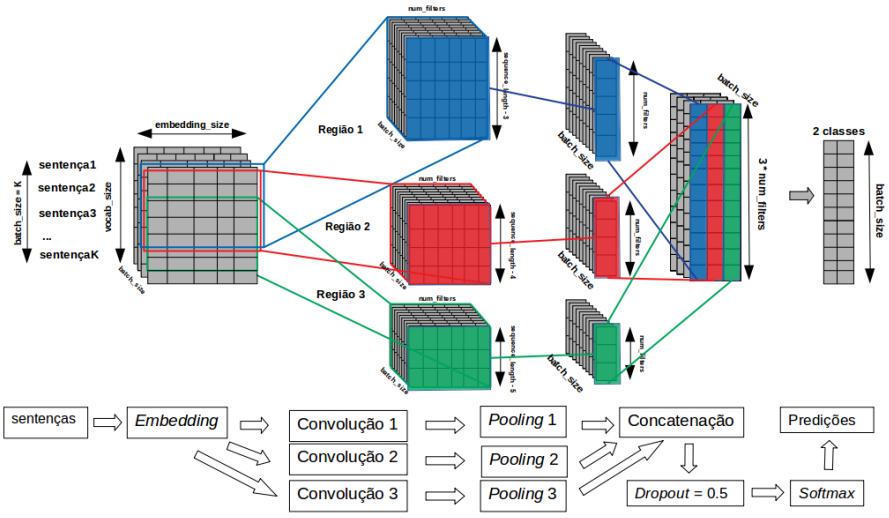

Fig. 1. Modelo de CNN para classificação de sentenças em língua portuguesa.

pouco ganho em tarefas de classificação de texto quando se alterou o modelo de CNN para o de redes neurais recorrentes. Entretanto, [Zhang and Wallace 2015] sinalizaram que esse modelo é sensível aos hiperparâmetros (regularizadores, dimensão de filtros, entre outros).

A Figura 1 ilustra o modelo de CNN aqui utilizado, o qual diferencia-se de Zhang por usar dimensões de filtros e hiperparâmetros diferentes. Seja uma sentença s formada por *tokens*, $s = t_1 \oplus t_2 \oplus t_3 \oplus \cdots \oplus t_n$, a camada de entrada é um texto (sentença) tokenizado e transformado em matriz sentencial, $S_{n,d}$, onde n é a quantidade de *tokens* que compõe essa sentença e d é a dimensionalidade do maior *token* em s. Cada linha em S é um vetor de palavra (*embed*), que é uma representação numérica para cada *token*. Tendo-se S construída é possível lhes aplicar filtros de convoluções, os quais possuem a mesma dimensão que o *embed*, uma vez que devem capturar todo o vetor de palavra de um *token*. Varia-se a "altura" dos filtros convolucionais, de modo a cobrir regiões diferentes da sentença. Usa-se filtros sobrepostos numa mesma região, a fim de se descobrir características complementares no local, uma vez que sentenças são um tipo de dados notadamente sequenciais e muito dependentes das vizinhanças.

As operações de convolução são divididas em três regiões, tendo 100 filtros de "alturas" (3, 4, 5) para suas respectivas regiões. Cada convolução possui a função ReLU como ativadora, sendo aplicada um *pooling* em cada uma das regiões, de modo que sejam extraídas somente as características mais significativas. Por fim, as características extraídas dessas convoluções são concatenadas, formando um vetor de características final; nessa última matriz é aplicada a função *Dropout* = 50% e depois a função *Softmax* para que ocorra a classificação.

3.3 Vetores de palavras

Vetores de palavras (*word embeddings*) são representações numéricas de uma palavra (*token*) dentro de um universo léxico (*corpus*), onde os valores numéricos representam o grau de similaridade do *token* com os demais *tokens* desse *corpus*.

A utilização de vetores de palavras em tarefas de PLN tem sido frequente devido à sua capacidade de representar de uma maneira bastante significativa a similaridade entre *tokens*. Por isso, o uso desses vetores de palavras podem auxiliar significativamente na construção da matrix S, anteriormente citada. Os algoritmos para obtenção de vetores de palavras mais populares devido o desempenho são: Word2Vec, Wang2Vec, Glove e FastText. Segundo Hartmann et al. [2017] esses *embeddings* se caracterizam por:

—Word2Vec: é um método amplamente utilizado em PLN e possui duas estratégias populares: (i) *Continuous Bag-Of-Words* (CBOW), onde ao modelo é oferecido uma sequência de palavras faltando a palavra do meio e o modelo tenta predizer qual a palavra faltante e (ii) *Skip-Gram*, onde ao modelo é dado uma palavra e esse tenta predizer quais as palavras vizinhas.

—Wang2Vec: é uma variação do Word2Vec, que busca suprir a falta de ordem das palavras no Word2Vec.

—Glove: consiste numa matriz , onde cada elemento é a probabilidade de um *token* estar próximo a um outro *token*.

—FastText: consiste em representar os vetores de palavras como *n-grams*, sendo as palavras representadas pela soma dessas representações.

Hartmann et al. [2017] realizaram um trabalho de compilação de 31 vetores de palavras exclusivamente em língua portuguesa. Os autores coletaram *corpus* de diferentes fontes, indiferentes à variedade do português (brasileiro ou europeu), pois a quantidade de elementos num *corpus* influencia em sua robustez. Com isso, geraram *embeddings* com 50, 100, 300, 600 e 1000 dimensões. No presente trabalho optou-se por utilizar 50, 100 e 300 dimensões para os algoritmos GloVe e Wang2Vec em português, pois foram os que obtiveram melhor desempenho nos experimentos dos autores supracitados.

Quando esses vetores pré-treinados não foram utilizados nos experimentos deste trabalho, treinou-se conjuntamente à rede um novo vetor de palavras (*embedding* não pré-treinado), de modo que além dos pesos da rede, os valores numéricos neste *embedding* também fossem aprendidos durante o treinamento. Essa abordagem se justificou na tentativa de verificar se apenas a CNN era capaz de, além de ajustar os pesos, criar um *embedding* para um dado conjunto de dados.

3.4 Funções otimizadoras do gradiente de descida

Como evidenciado por Zhang and Wallace [2015], os hiperparâmetros influenciam sensivelmente o desempenho desde modelo de CNN, com isso, é proposto confrontar o desempenho do modelo aqui apresentado com quatro diferentes métodos de otimização (RMSprop, Adagrad, Adadelta e Adam) [Ruder 2016] no gradiente descendente. Esses métodos são assim descritos:

—RMSprop é um método de taxa de aprendizagem adaptável. Divide a taxa de aprendizagem por uma média de gradientes quadrados exponencialmente descendente.

—Adagrad é baseado em gradiente que adapta a taxa de aprendizado aos parâmetros, executando atualizações maiores para atualizações pouco frequentes e menores para parâmetros frequentes. Por esse motivo, é bem adequado para lidar com dados esparsos.

—Adadelta é uma extensão do Adagrad que procura reduzir sua taxa de aprendizado agressiva e monotônicamente. Em vez de acumular todos os gradientes do passado, este método restringe a janela de gradientes acumulados anteriores a um tamanho fixo w.

—Adam (*Adaptive Moment Estimation*) também calcula as taxas de aprendizagem adaptativa para cada parâmetro. Além de armazenar uma média dos gradientes quadrados exponencialmente descendentes passados, também mantém uma média dos gradientes exponencialmente descendentes passados, semelhante ao momento.

3.5 Protocolo experimental

Todos os experimentos foram executados de acordo com os mesmos hiperparâmetros da Tabela I. Os treinamentos foram implementados no TensorFlow 1.8 e executados em CPU. Esses experimentos foram executados em 84 cenários, cada um com configuração única, combinando-se os conjuntos de dados com os distintos vetores de palavras pré-treinados (ou não) com diferentes tamanhos de palavras e com as diferentes funções de otimização para o treinamento da CNN.

Table I. Hiperparâmetros usados no treinamento da rede.

Hiperparâmetro	Valor
Taxa de aprendizagem (learning_rate)	0,001
Tamanho do mini-batch (batch_size)	50
Número total de épocas (hm_epochs)	50
Número total de dobras (K_epochs)	10
Quantidade de filtros de convolução (num_filters)	100
"Altura"dos filtros para cada camada de convolução (filter_size)	[3, 4, 5]

Table II. Resultados obtidos pelos trabalhos de referência

Autor	Conjunto de dados	Algoritmo	F-*score*	Acurácia
Pelle e Moreira (2017)	OffComBr-2	SVM	0,77	-
Pelle e Moreira (2017)	OffComBr-3	SVM	0,82	-
Fortuna (2017)	HSD	SVMLinear	0,76	78,3%

4. RESULTADOS

O critério utilizado para validar a classificação foi o F-*score*, o qual consiste em verificar a corretude das predições num conjunto desarmonicamente distribuído. Essa medida foi amplamente utilizada nos trabalhos relacionados [Pelle and Moreira 2017], [Pitsilis et al. 2018], [Almeida et al. 2017], [Nobata et al. 2016] e [Fortuna 2017]. Além disso, calculou-se também a acurácia da classificação, que é a habilidade que o modelo possui de predizer corretamente a classe de uma nova instância, a qual foi usada nos trabalhos de [Malmasi and Zampieri 2017] e [Fortuna 2017]. Os cálculos dessas medidas pode ser encontrado em [Aggarwal 2015].

Nos trabalhos de Pelle and Moreira [2017] e Fortuna [2017], usados como referência para o presente estudo, os resultados encontrados com a utilização de SVM para classificação dos conjuntos de dados são expressos na Tabela II. Nas Tabelas III, IV e V é possível visualizar todas as configurações utilizadas para o treinamento e seus respectivos resultados em termos de F-*score* e acurácia para cada um dos conjuntos de dados avaliados. Valores em negrito representam o melhor resultado obtido dentre os experimentos em cada conjunto de dados.

Dos vetores de palavras utilizados, os pré-treinados obtiveram ganhos consideráveis na classificação em relação ao uso de um vetor de palavras treinado durante o treinamento da CNN. A variação nas dimensões desses *embeddings* pré-treinados não necessariamente implicaram num aumento das métricas de avaliação. Entretanto, a variação entre as funções otimizadoras causou impacto sobre o treinamento do modelo, sendo o Wang2Vec o *word embedding* que permitiu as maiores métricas.

Das funções otimizadoras usadas, a RMSprop obteve a maior média de classificação, com F-*score* médio de 0,90, acurácia de teste média de 83,35% e um desvio padrão de 0,038 em 21 cenários de configurações. Já a função Adadelta obteve as mais baixas médias, com F-*score* de 0,63 e acurácia de teste de 59,91%, com um desvio padrão de 0,114 em 21 cenários de configurações. A Figura 2 ilustra graficamente o desempenho dos vetores de palavras de acordo com suas dimensões, exibindo a perda de treinamento nas 50 épocas.

A composição de cada conjunto de dados e a forma como estão anotados os comentários podem influenciar a detecção de discurso de ódio. Os F-*scores* mais altos para o OffComBr-3 indicam que há um padrão mais bem definido entre seus comentários anotados como ofensivos, em comparação com o OffComBr-2, uma vez que o OffComBR-3 é anotado pela unanimidade de seus três juízes. O desempenho na classificação, portanto, também está relacionado ao modo como os dados de treinamento/teste estão anotados, de modo que, aqueles conjuntos de dados que apresentam um padrão mais coeso, tornam o processo de treinamento menos complexo para a CNN. Considerando-se o F-*score*, os ganhos relativos ao trabalho de base (ver Tabela II) para o OffComBr-2 e OffComBr-3 foram de, respectivamente, +15,58% e +17,07%. Para o HSD, o ganho foi de +25,65%.

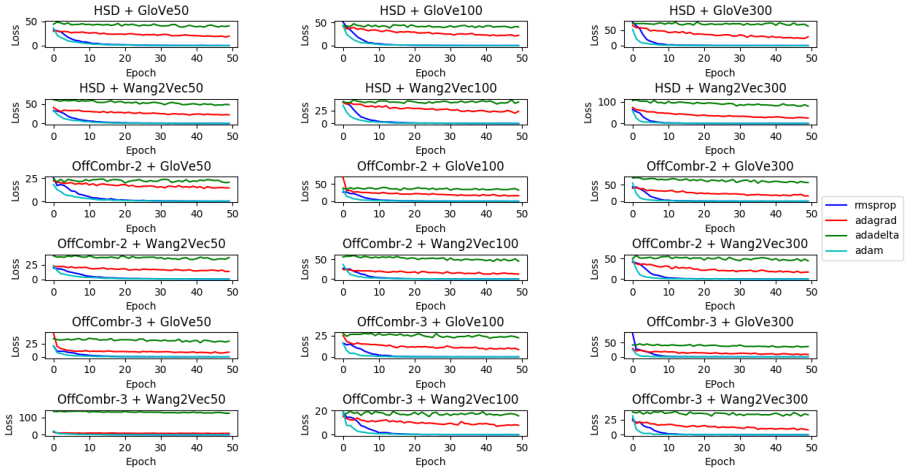

Fig. 2. O desempenho do GloVe e do Wang2Vec de acordo com suas dimensões.

Table III. Resultados do teste da CNN no treinamento do conjunto OffComBr-3 em várias configurações.

OffComBr-3								
Vetor de palavras	RMSprop		Adagrad		Adadelta		Adam	
	F-*score*	Acurácia	F-*score*	Acurácia	F-*score*	Acurácia	F-*score*	Acurácia
Treinável	0,87	77,67%	0,88	78,64%	0,70	66,50%	0,88	78,64%
Wang2Vec 50 dim.	0,91	84,08%	0,92	85,34%	0,49	50,58%	0,92	85,83%
GloVe 50 dim.	0,94	88,93%	0,90	82,43%	0,62	58,35%	0,93	88,45%
Wang2Vec 100 dim.	0,88	81,84%	0,80	67,12%	0,59	56,02%	0,89	82,64%
GloVe 100 dim.	0,88	81,12%	0,84	73,84%	0,76	69,71%	0,83	75,84%
Wang2Vec 300 dim	0,86	78,40%	0,75	64,00%	0,72	66,41%	**0,96**	**92,82%**
GloVe 300 dim	0,92	85,83%	0,93	86,99%	0,68	60,87%	0,94	88,54%

Table IV. Resultados do teste da CNN no treinamento do conjunto OffComBr-2 em várias configurações.

OffComBr-2								
Vetor de palavras	RMSprop		Adagrad		Adadelta		Adam	
	F-*score*	Acurácia	F-*score*	Acurácia	F-*score*	Acurácia	F-*score*	Acurácia
Treinável	0,83	70,96%	0,83	70,40%	0,60	56,56%	0,79	64,80%
Wang2Vec 50 dim.	0,87	80,32%	0,84	72,24%	0,76	68,16%	0,86	79,52%
GloVe 50 dim.	0,84	78,64%	0,77	63,36%	0,41	44,08%	0,86	78,08%
Wang2Vec 100 dim.	0,88	81,84%	0,80	67,12%	0,61	55,84%	**0,89**	**82,64%**
GloVe 100 dim.	0,88	81,12%	0,84	73,84%	0,47	50,16%	0,83	75,84%
Wang2Vec 300 dim	0,86	78,40%	0,75	64,00%	0,57	55,52%	0,88	80,88%
GloVe 300 dim	0,88	82,40%	0,87	78,64%	0,57	55,28%	0,85	78,40%

Table V. Resultados do teste da CNN no treinamento do conjunto HSD em várias configurações.

HSD								
Vetor de palavras	RMSprop		Adagrad		Adadelta		Adam	
	F-*score*	Acurácia	F-*score*	Acurácia	F-*score*	Acurácia	F-*score*	Acurácia
Treinável	0,87	76,88%	0,89	80,11%	0,80	73,76%	0,90	81,72%
Wang2Vec 50 dim.	0,93	88,98%	0,89	80,00%	0,55	53,92%	0,93	88,01%
GloVe 50 dim.	0,95	91,02%	0,91	83,39%	0,70	62,04%	0,93	87,63%
Wang2Vec 100 dim.	**0,96**	**92,74%**	0,86	76,45%	0,54	52,69%	0,95	92,10%
GloVe 100 dim.	0,95	91,99%	0,90	81,99%	0,52	53,60%	0,94	88,92%
Wang2Vec 300 dim	0,94	89,62%	0,91	83,55%	0,61	56,13%	0,95	91,61%
GloVe 300 dim	0,93	87,63%	0,91	83,71%	0,85	76,18%	0,92	87,53%

5. CONCLUSÕES E TRABALHOS FUTUROS

Com base nos resultados obtidos, foi possível verificar empiricamente a capacidade que as técnicas de DL, especificamente as CNN, possuem de permitir ganhos significativos quando comparadas a métodos clássicos nesta tarefa de classificação em PLN. O uso do modelo neural aqui proposto, apesar de simples, foi superior aos modelos utilizados pelos trabalhos de referência e contemplou as expectativas, permitindo tornar público este modelo treinado, onde espera-se que essa seja uma contribuição relevante para a detecção de discursos de ódio em língua portuguesa.

As funções otimizadoras permitiram verificar e reforçar a sensibilidade do modelo. Espera-se que a função Adadelta seja capaz de melhorar seu desempenho através de um número maior de épocas, uma vez que a convergência não foi atingida por ela nos experimentos aqui propostos. Outros hiperparâmetros também são passíveis de alteração, como a quantidade e o tamanho dos filtros de convolução.

Os modelos possuíam uma quantidade de comentários relativamente baixa para os conjuntos de treinamento/validação/teste da CNN, mas espera-se que o desempenho dessa rede continue a ser satisfatório, mesmo com o aumento no tamanho dos conjuntos de dados. Assim, aspira-se que futuramente haja maior contribuição da comunidade científica no sentido de se fornecer publicamente mais conjuntos de dados anotados de grandes volumes em língua portuguesa.

Além disso, dois pontos se tornam relevantes para trabalhos futuros nesta área: a classificação não-binária (*multi-label classification*) do discurso de ódio e o uso de ontologias que permitam enriquecer e atribuir valor semântico para as classificações obtidas através desses modelos de CNN.

REFERENCES

Aggarwal, C. C. *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015.

Almeida, T. G., Nakamura, F. G., and Nakamura, E. F. Uma abordagem para identificar e monitorar haters em redes sociais online, 2017.

Cohen-Almagor, R. Fighting hate and bigotry on the internet. *Policy & Internet* 3 (3): 1–26, 2011.

Fortuna, P. C. T. Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes, 2017.

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., and Plagianakos, V. P. Convolutional neural networks for toxic comment classification. *arXiv preprint arXiv:1802.09957*, 2018.

Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., and Aluisio, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*, 2017.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature* 521 (7553): 436–444, 5, 2015.

Lopez, M. M. and Kalita, J. Deep learning applied to nlp. *CoRR* vol. abs/1703.03091, 2017.

Malmasi, S. and Zampieri, M. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427*, 2017.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, pp. 145–153, 2016.

Pelle, R. and Moreira, V. Offensive comments in the brazilian web: a dataset and baselines results. In *Proc. of the 6th Brazilian Workshop on Social Network Analysis and Mining*. pp. 1–160, 2017.

Pitsilis, G. K., Ramampiaro, H., and Langseth, H. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*, 2018.

Ruder, S. An overview of gradient descent optimisation algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Schmidt, A. and 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*. pp. 1–10, 2017.

Yin, W., Kann, K., Yu, M., and Schütze, H. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

Zhang, Y. and Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

Zhang, Z., Robinson, D., and Tepper, J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*. Springer, pp. 745–760, 2018.