An Analysis of Convolutional Neural Networks for Sentence Classification

João Paulo Albuquerque Vieira, Raimundo Santos Moura
Departamento de Computação
Universidade Federal do Piauí
Teresina, Piauí
Email: joaopauloalbu@gmail.com,
rsm@ufpi.edu.br

***Abstract*—Over the past few years, neural networks have re-emerged as powerful machine-learning models, yielding state-of-the-art results in fields such as image recognition and speech processing. More recently, neural network models started to be applied also to textual natural language signals, again with very promising results. This paper show a series of experiments with Convolutional Neural Networks for sentence-level classification tasks with different hyperparameter settings and how sensitive model performance is to changes in these configurations.**

***Index Terms*—Natural language processing, Sentiment analysis, Deep neural network.**

I. Introdução

Análise de Sentimentos (AS) é o campo de estudo que analisa as opiniões das pessoas, sentimentos, avaliações, atitudes, e emoções com respeito as entidades e seus atributos expressos em texto escrito [1]. Com o rápido crescimento das redes sociais, muitas pessoas têm compartilhado suas visões e opiniões na Internet, através de *reviews*, fóruns de discussões, blogs, notícias e comentários diversos.

A oportunidade de capturar a opinião de um público geral tem levantado o crescente interesse da comunidade científica (por causa dos excitantes desafios abertos) e da comunidade de negócios (por causa dos notáveis benefícios para o *marketing* e as previsões do mercado financeiro). Hoje, pesquisadores tem suas aplicações nos mais diferentes cenários. Há um bom número de companhias, de grande e pequena escala, que focam na análise de opiniões e sentimentos como parte das suas missões [2]. Assim sendo, esse fascinante problema tem ficado cada vez mais importante na sociedade e nos negócios.

Recentemente, as Redes Neurais Convolucionais (do inglês *Convolutional Neural Network* - CNN) obtiveram impressionantes resultados na importante tarefa de classificação de sentenças [3]–[8]. No entanto, o espaço de possíveis configurações para esse modelo é vasto, o treinamento do mesmo é relativamente lento e ajustar esses parâmetros é bastante custoso, especialmente porque a estimação dos parâmetros é computacionalmente intensivo, mesmo usando GPUs[1].

[1]*Graphics Processing Unit*, é o nome dado a um tipo de microprocessador especializado em processar gráficos.

000-0-0000-0000-0/00/$00.00

Por isso, neste trabalho reportamos resultados de um grande número de experimentos explorando diferentes configurações de um modelo CNN com arquitetura fixa para classificação de sentimentos em nível de sentença. Nossos objetivos são identificar empiricamente as configurações que os profissionais normalmente gastam esforços e prover modelos com variações dos parâmetros.

Assim sendo, foram feitos experimentos sobre CNNs de apenas uma camada para exclusão de modelos mais complexos e devido à sua simplicidade comparativa e forte desempenho empírico, em um conjunto de dados reais coletados do Mercado Livre[2], reportando acurácia, precisão, cobertura e medida-F, que são métricas que comparam os resultados obtidos contra os resultados esperados determinados pelo rótulo das classes, para explorar os efeitos dos componentes na performasse do modelo.

O restante desse artigo é organizado do seguinte modo. Seção II revisa os trabalhos relacionados. Na Seção III, nos apresentamos o modelo da CNN. Seção IV discute os experimentos e os estudos comparativos. Finalmente, na Seção V, nos mostramos as conclusões e os trabalhos futuros.

II. Trabalhos Relacionados

As abordagens dominantes na análise de sentimento são impulsionadas por métodos de aprendizagem de máquina [9], [10]. A abordagem mais comum consistem no modelo *Bag of Words* (BOW), onde cada documento é transformado em um vetor de características que é então alimentado para um algoritmo de classificação. Outros tipos de técnicas são usualmente usadas, tal como *Part of Speech* (POS) *tagging*, que é um modelo elementar de análise sintática [11]. A abordagem estatística para a representação de documentos, conhecida como TF-IDF, onde as palavras são ponderadas dependendo das suas frequências no *Córpus* também faz parte da literatura [12].

Além disso, muitos conceitos da análise de sentimentos envolvem o uso de um léxico de sentimentos como fonte de informações subjetivas [13]. Porém, abordagens baseadas em léxicos tem muitas desvantagens: a necessidade de dados etiquetados que sejam confiáveis e consistentes, as expressões

[2]http://mercadolivre.com.br/

linguísticas que são dependentes do domínio e o fato que o léxico não pode ser traduzido automaticamente para uso multilíngue [14]. Também, extrair características não-simples de texto e descobrir quais deles são relevantes é um questão fundamental nas técnicas que conduzem o aprendizado de máquina [15].

A abordagem de [16] utilizou a estrutura sintática das sentenças usando POS *tagging* para identificar as características e seus respectivos qualificadores. O autor considerou os verbos como palavras opinativas, além dos adjetivos e advérbios e utilizou padrões linguísticos pré-definidos por [17] e algumas extensões para satisfazer o domínio de produtos nos quais ele trabalhou. Também fez o uso do léxico de sentimentos Sentilex-PT como método para inferência da orientação semântica das expressões.

Alternativamente, técnicas de aprendizado profundo tem mostrado desempenho promissor em muitas tarefas do Processamento de Linguagem Natural (PLN), incluindo análise de sentimentos [18]. Um uso comum de aprendizado profundo é aprender características complexas dos dados com o mínimo de contribuições através de redes neurais profundas [19]. As representações continuas de palavras como vetores, também conhecidas como *Word Embeddings* (WE) tem sido usadas para análise de sentimentos [20].

[3] desenvolveu uma simples CNN para classificação de sentenças em sete *benchmarks*. O autor usou como entrada um vetor de palavras pré-treinadas com 100 bilhões de palavras do Google News[3] para CNN. O vetor tinha 300 dimensões cada e estava treinado com a arquitetura *Continuous Bag of Words* (CBOW) [21]. Para todos os conjuntos de dados, o autor usou os seguintes parâmetros: $filter\_lenght$ com 3, 4, 5 com 100 nb_filter cada, taxa de *dropout* de 0.5, restrição $l2$ de 3 e $batch\_size$ de tamanho 50. Para essa configuração, ele melhorou os estados da arte em quatro *benchmarks*.

[22] propõe uma variante semi-supervisionada da CNN que primeiro aprende *embeddings* de pequenas regiões do texto a partir de dados não-rotulados, e então integra-os em uma CNN supervisionada. Enfatizando que, se os dados de treinamento forem abundantes, a aprendizagem de uma *embedding* do zero pode ser, de fato, melhor.

Na análise de sentimentos, diferentes níveis de granularidade de analise foram propostos, cada um tendo suas próprias vantagens e desvantagens [23]. Opiniões e sentimentos expressos em revisões de textos geralmente podem ser analisados em nível de documento, sentença ou aspecto. Uma das principais direções da área é a análise de sentimento em nível de sentença. Grande parte das pesquisas existentes sobre este tópico focou-se na identificação da polaridade de uma frase (por exemplo, positiva, negativa, neutra) com base nas "pistas"extraídas do seu conteúdo textual [17], [24], [25].

III. Modelo Proposto

Neste trabalho nos desenvolvemos uma CNN semelhante ao modelo primeiramente proposto por [3] representado na Figura 1.

[3]https://news.google.com/

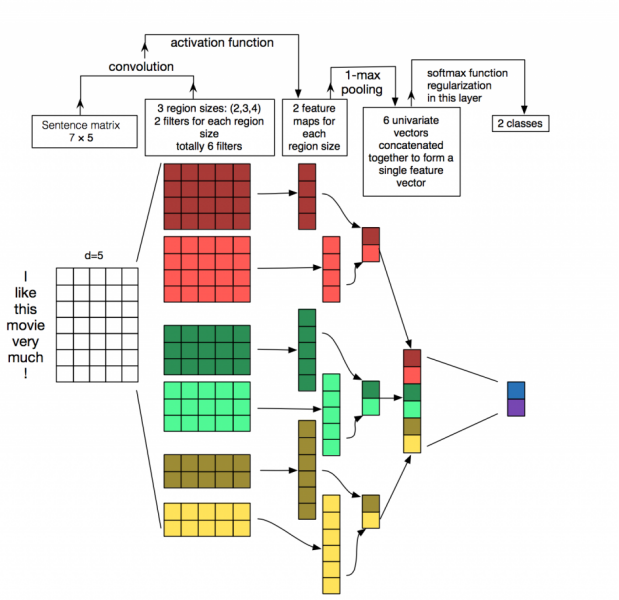

Figura 1. Modelo CNN para classificação de sentenças [26]

A camada de *embedding* recebe uma sentença tokenizada que nos convertemos em uma matriz de sentenças M. Antes do treinamento, são geradas WE para cada palavra em um glossário de todas as entradas da sentença, a quantidade de palavras selecionadas para o glossário é definida pelo parâmetro $max\_features$. As WE das palavras que correspondem a sentença atual são montadas em M, onde as linhas são representações vetoriais das palavras de cada *token*.

Nos denotamos a dimensionalidade do vetor da palavra por d. Se o comprimento de uma dada sentença é s, então a dimensionalidade da matriz da sentença é $s \times d$. O comprimento máximo das sentenças que a rede manipula é definido como um parâmetro, que chamamos por $maxlen$. As sentenças maiores que $maxlen$ são truncadas e as menores são preenchidas com vetores de zero.

WE são uma família de técnicas de PLN visando o mapeamento do significado semântico em um espaço geométrico. Isso é feito associando um vetor numérico a cada palavra em um dicionário, tal como a distância (por exemplo, L2 ou mais comumente cosseno) entre quaisquer dois vetores podendo capturar parte das relações semânticas entre as duas palavras associadas. WE é calculada aplicando técnicas de dimensionalidade reduzidas para conjuntos de dados de co-ocorrência estatísticas entre palavras em um *Córpus* de texto. Isso pode ser feito via redes neurais ou matriz de fatoração. Estes podem ser, por exemplo, saídas de modelos treinados, tal como *word2vec* [21] ou *GloVe* [27]. Em nosso trabalho seguimos [22] e treinamos as WE do zero porque o número de dimensões que ela contém também é um dos parâmetros avaliados neste trabalho, definido por we_dims.

Na camada de convolução nos tratamos a matriz da sentença como uma "imagem", e executamos convoluções sobre ela via filtros lineares. As linhas representam as palavras, esta é a

razão de usarmos filtros com largura igual a dimensionalidade do vetor de palavras. Assim nos podemos simplesmente variar a "altura"do filtro, definido por $filter\_lenght$, e executar convoluções de uma dimensão sobre a matriz usando múltiplos filtros com diferentes tamanhos de janela. À medida que os filtros se movem, por várias sequências, eles capturam as características sintáticas e semânticas geradas no *n-grama* filtrado.

Muitas sequencias de características são combinadas em um mapa de características. Na camada de *pooling*, uma operação de *maxpooling* é aplicada para capturar a característica local mais importante do mapa de características [18]. Em seguida, é aplicada *dropout* para controlar o *overfitting* do modelo.

As funções de ativação são adicionadas para incorporar o *element-wise* de não-linearidade. As saídas dos múltiplos filtros são concatenadas em uma camada de *merge* para um vetor de características univariada. Em seguida, a última camada recebe esse vetor de característica como entrada e classifica a sentença sobre os rótulos de várias classes. Neste trabalho assumimos a classificação binária: positiva e negativa.

A. Córpus

O *Córpus* foi coletado do site Mercado Livre em um total de 43318 comentários sobre produtos de escolha arbitrária. O conjunto de dados foi dividido em sentenças, que foram etiquetadas automaticamente através das *tags Prós* e *Contras*, encontradas em sua descrição textual, em positivos e negativos respectivamente. Ao final, o *Córpus* gerado possui um total de 75098 sentenças etiquetadas sendo 39204 positivas e 35894 negativas. A distribuição de frequência pelo comprimento da sentença pode ser vista na Figura 2. Também calculamos a dispersão do *Córpus* pelo coeficiente de variação de Pearson, encontrando o valor 1,21. Outros informações sobre o *Córpus* que consideramos importantes são: 203 é o comprimento da maior sentença; 36207 é a quantidade de *tokens* únicos.

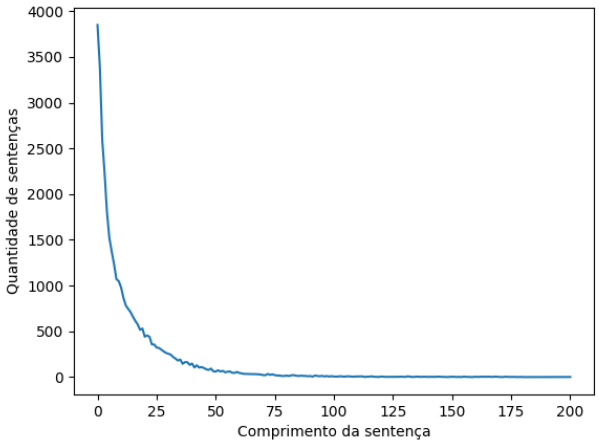

Figura 2. Distribuição de frequência x comprimento das sentenças

IV. Resultados e Discussões

Nesta seção nos apresentamos os resultados dos experimentos conduzidos para avaliar o desempenho das diferentes configurações de uma CNN para a tarefa de classificação de sentenças. Nestes experimentos exploramos os parâmetros: $maxlen$, $max\_feature$, we_dims, act_func, $filter\_len$, nb_filter e $batch\_size$. Para a avaliação dos resultados das variações dos parâmetros no desempenho do modelo calculou-se a taxa de erro, bem como as medidas de acurácia (A), precisão (P), revocação (R) e medida-F (F) que são definidas como:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F = 2 \times \frac{P \times R}{P + R} \tag{4}$$

onde TP, TN, FP, e FN referem-se a verdadeiro positivo, verdadeiro negativo, falso positivo, e falso negativo, respectivamente.

Os experimentos foram conduzidos inicialmente com os parâmetros atribuídos de forma empírica. Depois escolhemos parâmetro por parâmetro e os testamos com diversos valores que podem ser vistos nas tabelas abaixo. Depois selecionamos o melhor valor da medida-F, por ser uma medica harmônica entre a precisão e a revocação. Nos demais experimentos mantemos os parâmetros ja testados como fixo e variamos os demais ate montarmos uma rede na qual todos os parâmetros já foram testados, ao final teremos a rede com todos os parâmetros que obtiveram os melhores resultados.

Na Tabela I podemos notar que a medida-F foi crescendo conforme aumentamos o comprimento da sentença, mas chegando nas sentenças maiores que 50, onde não são tão representativas, como pode ser observado na Figura 2, a medida-F não melhora com tanta expressividade. Uma observação interessante é que o parâmetro $maxlen$ obteve melhor resultado quando foi definido com tamanho 200 que é o valor que engloba aproximadamente o comprimento de todas as sentenças.

Tabela I
Sensibilidade em relação ao tamanho da sentença

$maxlen$	Erro	Acurácia	Precisão	Revocação	Medida-F
10	0.42	82.56%	86.03%	79.62%	82.23%
20	0.38	84.51%	86.45%	83.29%	84.43%
40	0.35	86.07%	87.14%	85.43%	85.90%
60	0.36	85.88%	88.79%	83.64%	85.75%
80	0.36	86.09%	89.41%	83.32%	85.85%
100	0.36	86.04%	88.81%	83.70%	85.79%
200	0.36	86.04%	87.89%	84.82%	**85.94%**
300	0.36	85.49%	87.41%	84.66%	85.59%

A Tabela II mostra os resultados referentes ao parâmetro $max\_feature$, que determina a quantidade dos *tokens* mais frequentes que serão usados na rede. Apesar do *Córpus* conter 36207 *tokens* únicos, o melhor resultado foi encontrado com apenas 20000 deles, significando que os outros não são importantes ou são ruídos (por exemplo, pontuação).

Tabela II
SENSIBILIDADE EM RELAÇÃO A QUANTIDADE DE CARACTERÍSTICAS

$max\_feature$	Erro	Acurácia	Precisão	Revocação	Medida-F
100	0.48	78.93%	79.45%	80.31%	79.36%
200	0.44	80.90%	81.98%	81.03%	81.09%
400	0.40	83.25%	84.38%	82.63%	83.07%
600	0.39	84.13%	85.91%	83.55%	84.31%
800	0.37	85.15%	87.55%	83.46%	85.03%
1000	0.37	84.99%	86.90%	83.63%	84.83%
2000	0.36	86.22%	88.03%	84.97%	86.09%
5000	0.35	86.04%	88.36%	84.54%	86.00%
10000	0.35	86.06%	88.13%	84.97%	86.14%
20000	0.35	86.83%	88.78%	85.73%	**86.86%**
40000	0.35	86.46%	88.87%	84.89%	86.48%
50000	0.34	86.39%	89.31%	84.08%	86.26%

Olhando para a Tabela III podemos notar que não ocorrem variações significativas do desempenho ao aumentar a dimensionalidade do vetor de representação da palavra. Isso significa que um vetor com menos dimensões já consegue representar o *Córpus*, ou seja, a quantidade de dados usados para treinar a WE é insuficiente; No trabalho de [3] por exemplo são usadas 100 bilhões de palavras para a formação da WE utilizada. Infelizmente não temos conhecimento de um recurso semelhante para a língua portuguesa do Brasil.

Tabela III
SENSIBILIDADE EM RELAÇÃO A DIMENSIONALIDADE DA EMBEDDING

we_dims	Erro	Acurácia	Precisão	Revocação	Medida-F
50	0.33	86.87%	88.20%	86.34%	86.88%
100	0.33	86.62%	89.76%	83.80%	86.26%
150	0.31	87.30%	89.47%	85.16%	**86.95%**
200	0.33	86.72%	88.57%	85.85%	86.81%
250	0.33	86.64%	87.32%	87.15%	86.85%
300	0.32	87.08%	89.77%	84.87%	86.87%
400	0.32	86.96%	89.35%	85.04%	86.78%
500	0.33	86.73%	89.73%	84.38%	86.59%
600	0.32	86.87%	89.88%	84.60%	86.79%

Na Tabela IV observamos que a função de ativação ReLU obteve melhores resultados nas métricas de acurácia, revocação e medida-F. Entretanto a função *sigmoid* obteve melhor precisão.

Tabela IV
SENSIBILIDADE EM RELAÇÃO A FUNÇÃO DE ATIVAÇÃO

act_func	Erro	Acurácia	Precisão	Revocação	Medida-F
relu	0.30	87.98%	87.59%	89.67%	**88.27%**
softmax	0.33	86.88%	89.60%	84.57%	86.62%
tanh	0.28	87.82%	88.53%	87.56%	87.73%
sigmoid	0.31	87.32%	90.70%	84.53%	87.15%

Na Tabela V, como era esperado, os filtros de tamanho 1 e 2 não são capazes de encontrar relações semânticas fortes entre as palavras comparado com os filtros de tamanho maior ou igual a 3. Com relação ao número de filtros, não se constatou variação significativa entre os valores analisados, conforme Tabela VI.

Tabela V
SENSIBILIDADE EM RELAÇÃO AO COMPRIMENTO DO FILTRO

$filter\_len$	Erro	Acurácia	Precisão	Revocação	Medida-F
1	0.34	85.60%	87.83%	84.35%	85.65%
2	0.33	86.00%	88.09%	84.49%	85.87%
3	0.32	86.98%	88.41%	85.92%	86.80%
4	0.33	86.64%	89.03%	85.11%	86.65%
5	0.32	86.86%	87.82%	86.88%	**87.01%**
6	0.32	87.16%	89.32%	85.45%	86.99%
7	0.32	87.09%	89.70%	84.88%	86.85%

Tabela VI
SENSIBILIDADE EM RELAÇÃO AO NÚMERO DE FILTROS

nb_filter	Erro	Acurácia	Precisão	Revocação	Medida-F
10	0.33	86.65%	86.53%	88.15%	86.97%
50	0.32	86.91%	89.72%	84.55%	86.69%
100	0.31	87.30%	89.07%	85.76%	87.06%
150	0.32	86.88%	89.60%	84.98%	86.84%
200	0.32	87.04%	88.03%	87.03%	**87.17%**
250	0.32	87.00%	89.85%	84.47%	86.72%
300	0.32	87.25%	89.77%	84.99%	86.99%
600	0.33	86.62%	88.82%	85.24%	86.62%

O número de instâncias que são avaliadas antes da atualização de peso na rede, definida por $batch\_size$, apresentou resultados curiosos. Observando a Tabela VII, notamos que ao aumentarmos os valores do tamanho de $batch\_size$ piores eram os resultados e maior o erro.

Tabela VII
SENSIBILIDADE EM RELAÇÃO AO TAMANHO DO LOTE

$batch\_size$	Erro	Acurácia	Precisão	Revocação	Medida-F
10	0.33	86.78%	87.14%	87.59%	**87.01%**
20	0.34	86.47%	89.34%	83.99%	86.18%
30	0.33	86.92%	88.54%	85.59%	86.69%
40	0.34	86.39%	88.54%	85.12%	86.40%
50	0.34	86.82%	88.74%	85.66%	86.82%
60	0.35	86.79%	89.56%	84.36%	86.49%
80	0.35	86.39%	88.07%	85.36%	86.30%
100	0.37	86.18%	87.97%	85.25%	86.21%

V. CONCLUSÕES

Neste trabalho, apresentamos um Rede Neural Convolucional para classificação de sentenças com uma medida-F de 88,27%. Além disso, realizamos uma vasta serie de experimentos dos parâmetros usados e quais suas relevâncias no desempenho do modelo. Definimos ainda que o comprimento máximo da sentença seria de 200, que escolheríamos os 20000 *tokens* mais frequentes para treinar nossa WE de 150 dimensões, com 200 filtro de tamanho 5, usando a função de ativação não-linear ReLU e *bacthsize* de tamanho 10. Adicionalmente montamos um *Córpus* com 75098 sentenças etiquetadas em positivo e negativo.

Para trabalhos futuros espera-se: (i) Testar diferentes arquiteturas com múltiplos filtros, camadas escondidas e outras

funções de ativação. (ii) Extrair uma grande quantidade de dados para treinar uma WE mais representativa. (iii) Comparação a WE treinada com uma pre-treinada.

REFERÊNCIAS

[1] B. Liu, *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.

[2] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[3] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1746–1751.

[4] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *CoRR*, vol. abs/1404.2188, 2014.

[5] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *CoRR*, vol. abs/1412.1058, 2014.

[6] P. Wang, J. Xu, B. Xu, C. Liu, H. Zhang, F. Wang, and H. Hao, "Semantic clustering and convolutional neural network for short text categorization," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, 2015, pp. 352–357.

[7] Y. Goldberg, "A primer on neural network models for natural language processing," *CoRR*, vol. abs/1510.00726, 2015.

[8] M. Iyyer, V. Manjunatha, J. L. Boyd-Graber, and H. D. III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, 2015, pp. 1681–1691.

[9] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, 2014, pp. 28–37.

[10] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, 2012, pp. 90–94.

[11] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, 2011, pp. 42–47.

[12] J. Martineau and T. Finin, "Delta TFIDF: an improved feature space for sentiment analysis," in *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*, 2009.

[13] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, 2009, pp. 1275–1284.

[14] M. Taboada, J. Brooke, M. Tofiloski, K. D. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[15] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Languages in Social Media*, ser. LSM '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 30–38.

[16] R. F. de Sousa, "Abordagem top(x) para inferir comentários mais importantes sobre produtos e serviços," Master's thesis, Universidade Federal do Piauí, 2015.

[17] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, 2002, pp. 417–424.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[19] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[20] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, 2014, pp. 1555–1565.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 2013, pp. 3111–3119.

[22] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 919–927.

[23] E. Cambria, B. W. Schuller, B. Liu, H. Wang, and C. Havasi, "Statistical approaches to concept-level sentiment analysis," *IEEE Intelligent Systems*, vol. 28, no. 3, pp. 6–9, 2013.

[24] B. Liu, *Sentiment Analysis and Opinion Mining*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[25] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain.*, 2004, pp. 271–278.

[26] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *CoRR*, vol. abs/1510.03820, 2015.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1532–1543.