Análise de Desempenho de Redes Neurais Artificiais em Plataformas CPU e GPU Aplicadas no Reconhecimento de Sentimentos em Textos *

Cristiano A. Künas[1], **Leandro P. Heck**[1], **Edson L. Padoin**[1]

[1]Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUÍ)
Santa Rosa – RS – Brasil

{cristiano.kunas, leandro.h}@sou.unijui.edu.br, padoin@unijui.edu.br

Abstract. *Nowadays, a lot of information with all kinds of content circulates on the Internet. Analyzing the feelings exposed can help in understanding what people are talking about a particular company, brand, event or even other people, working as a way to get feedback. In this work, we present an Artificial Neural Network model for sentiment analysis in English language sentences. The Long Short-Term Memory Recursive Artificial Neural Network was implemented for the training of the sentiment analysis model. With the application of RNA developed over a public database with 50,000 movie records using GPU it was possible to reduce the training time of RNAs by up to 91.8% and increase the accuracy to 87.7%.*

Resumo. *Atualmente, circulam pela internet imensas quantidades de informações com todo tipo de conteúdo. Analisar os sentimentos expostos pode auxiliar no entendimento do que as pessoas estão falando sobre uma determinada empresa, marca, evento ou até mesmo sobre outras pessoas, funcionando como uma forma de se obter feedback. Neste trabalho, apresentamos um modelo de Rede Neural Artificial para análise de sentimentos em sentenças para o idioma Inglês. Foi implementado a Rede Neural Artificial Recursiva Long Short-Term Memory para o treinamento do modelo de análise de sentimentos. Com a aplicação da RNA desenvolvida sobre uma base de dados pública com 50.000 registros de filmes utilizando GPU foi possível reduzir o tempo de treinamento de RNAs em até 91,8% e aumentar a acurácia para 87,7%.*

1. Introdução

Inúmeras informações circulam pela internet através de sites, blogs e redes sociais. O conteúdo presente vai desde análise e comentários sobre filmes até conversas e experiências dos seus usuários. Nesse sentido, usando os recursos tecnológicos podemos armazenar, recuperar e analisar uma enorme quantidade destes dados eficientemente. Uma questão é que esses dados não encontram-se estruturados de forma compreensível para um único sistema computacional.

As opiniões são tão importantes para quase todas as atividades humanas que, sempre que precisamos tomar uma decisão, queremos ouvir as opiniões de outros [Liu et al. 2010]. São os principais influenciadores de nossos comportamentos. Indivíduos e organizações estão cada vez mais usando o conteúdo dessas mídias para a tomada de decisões. Porém, os sistemas computacionais ainda encontram grande dificuldade em compreender como os clientes se

*Trabalho desenvolvido com recursos do edital MCTIC/CNPq - Universal 28/2018 sob número 436339/2018-8 e do edital da VRPGPE bolsa PIBIC/UNIJUI.

sentem em relação aos produtos e serviços de uma certa companhia. Em algumas situações, apenas alguns pontos do texto são relevantes: sobre quem se fala, e se o que se fala é bom ou ruim [dos Santos 2017]. A Inteligência Artificial (IA) pode ser aplicada para facilitar a compreensão desses dados, de tal maneira que possamos usar uma sentença em linguagem natural como entrada e extrair um conjunto de dados na saída. Desta forma, a utilização de uma Rede Neural Artificial (RNA) pode auxiliar na extração automática do sentimento ou sensação de sentenças.

Dentre as diferentes subáreas de estudo que o campo de Análise de Sentimento apresenta, a tarefa mais presente na literatura e que abordaremos neste trabalho é a análise de polaridade de um documento, que objetiva classificar textos em uma escala entre positivo e negativo, uma vez que essa área de pesquisa é de grande relevância para consumidores e organizações. Assim sendo, o presente trabalho almeja: i) criar uma rede neural recorrente do tipo *Long Short-Term Memory*; ii) realizar o seu treinamento utilizando uma base de dados pública; e iii) aplicá-la no reconhecimento de sentimentos expressados em textos e/ou postagens.

O restante do trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a metodologia utilizada na implementação e o ambiente de execução utilizado na realização dos testes. Resultados são discutidos na Seção 4, seguidos das conclusões e perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Diversos trabalhos apresentam estudos sobre a Análise de Sentimentos. Este conceito foi introduzido pela primeira vez em 2003 onde técnicas foram usadas na mineração de opinião na história. Algoritmos baseados em aprendizado de máquinas foram usadas em Walker *et al.* (2012) para classificar o posicionamento de uma pessoa em relação a um objeto, ideia ou postura. Neste trabalho, foi classificado o posicionamento que as pessoas têm e aplicaram isso a debates políticos. Eles utilizaram 104 debates de duplo posicionamento oriundos do *convinceme.net* para 14 temas diferentes e tentaram identificar a opinião dos envolvidos. O objetivo principal foi determinar a potencial contribuição de características de diálogo na classificação de debates. O trabalho usou SVM, *Naive Bayes* (NB) e um classificador baseado em regras para fins de classificação e conseguiram precisar a escolha do lado do debate.

Redes neurais profundas foram utilizadas em Hu *et al.* (2015) para a análise de sentimento em *reviews* de produtos eletrônicos, filmes e hotéis. Os autores criaram uma estrutura de classificação que usa 3 métodos diferentes para extração de atributos: baseados em frequência, contexto e etiquetagem de partes de falas. O trabalho comparou o desempenho da estrutura proposta pelos autores com métodos de aprendizado de máquinas (NB e SVM). A análise verificou que a acurácia do modelo de redes neurais profundas foi superior a NB e SVM para um número de amostras grande no conjunto de treinamento (acima de 200000). Todavia, a precisão do modelo proposto diminuiu para casos em que a dimensionalidade das entradas é grande e o número de amostras para treinamento é reduzido.

Análise de sentimento a nível de sentença foi estudado em Li *et al.* (2014). Os autores criaram um banco de dados de sentenças no idioma chinês denominado de *Recursive Neural Deep Model* (RNDM). Ao contrário de outros bancos de dados que possuem um sentimento global para cada sentença, este possui classificações de sentimento para cada oração dentro de cada sentença e para cada palavra dentro de cada oração. O trabalho demonstra o desempenho deste sistema comparado a outros com métodos de aprendizado de máquinas (NB, SVM, e

Regressão Logística).

Estes trabalhos contemplam importantes contribuições no uso de Redes Neurais Artificiais para classificação de sentimentos. Seguindo esta premissa, nossa proposta é avaliar a utilização de uma RNA Recorrente *Long Short-Term Memory*, objetivando aumento de desempenho no treinamento da mesma através da utilização da arquitetura GPU.

3. Metodologia

A implementação da RNA deste trabalho é feita usando a linguagem de programação *Python* [Rossum 1995]. Os principais módulos utilizados são: *Keras* [Moolayil 2019], *Tensorflow* [Abadi et al. 2016] e *Scikit-Learn* [Pedregosa et al. 2011]. O *Keras* é usado para modelar a RNA. O *Tensorflow* é a biblioteca de *back-end* para computação numérica padrão do *Keras*. Para dividir os dados usados neste trabalho, em treino (80%) e teste (20%), usa-se a função *train_test_split()* da biblioteca *Scikit-Learn*.

A base de dados usada nos experimentos foi elaborada no trabalho de Maas *et al.* (2011). Esta base possui uma coleção de 50.000 avaliações do IMDb escritas no idioma inglês. As críticas estão bem balanceadas em 2 classes: positiva e negativa. O pré-processamento destes dados é feito em etapas com objetivo de remover ruído presente nas sentenças. São elas: i) caracteres especiais e sinais de pontuação são removidos; ii) todas as palavras são escritas em letras minúsculas; iii) as *stopwords* são removidas; iv) as sequências são limitadas a um tamanho fixo (300 palavras);

A criação do modelo de RNA é definido como uma sequência de camadas. A classe *Model* da API funcional da biblioteca *Keras* é utilizada no desenvolvimento do modelo de dados. A compilação do modelo configura o processo de aprendizado. Ele define o otimizador (*adam*), a função de perda (*binary_crossentropy*) e as métricas (*accuracy*). Para treinar o modelo, os dados são divididos em lotes (*batch_size*), iterando repetidamente em todo o conjunto de dados para um determinado número de épocas. O modelo é avaliado por meio da função *evaluate()*. Essa função gerará uma predição para cada entrada e saída e coletará pontuação, realiza uma média sobre o valor de *perda* e *acurácia*.

O ambiente de execução é composto de um equipamento com um processador *Intel Core* i7-9750 com 6 *cores* (12 *threads*) de 2.60 GHz de frequência. Este equipamento possui 16GB de Memória RAM DDR4, GPU NVIDIA GeForce GTX RTx 2060 com 6GB de GDDR6 e 1920 CUDA *cores*, utilizou-se do sistema operacional *Linux Ubuntu* 18.04.3 LTS com versão de *kernel* 5.0.0-37. A versão do NVIDIA CUDA *Compiler* utilizada foi a 10.0.130.

4. Resultados

Na Figura 1 é apresentada a *acurácia* (eixo *y*) alcançada a partir do *dataset* de treinamento (40000 registros), com diferentes tamanhos de *batch*, nas execuções tanto em CPU quanto em GPU. Percebe-se que conforme o número de instâncias para cada atualização de gradiente aumenta, a precisão do modelo é reduzida, ou seja, menos os pesos da RNA são atualizados e consequentemente a taxa de perda aumenta.

Na Figura 2 é apresentada a *acurácia* (eixo *y*) obtida a partir do *dataset* de validação com 10000 registros variando-se o *batch_size* e executando em CPU quanto em GPU. A *acurácia* de validação informa o percentual que nosso modelo acertou na predição em relação ao rótulo. Estes testes, comprovam que a *acurácia* da implementação se mantém estável, variando entre $87 \sim 89\%$.

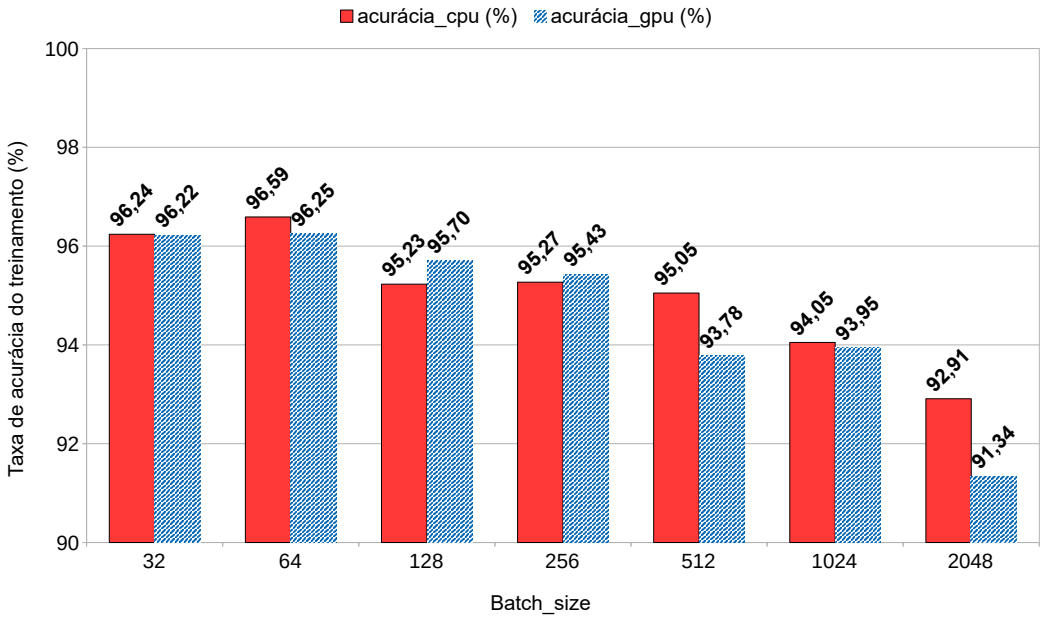

Figura 1. Métricas dos dados de treinamento das execuções em CPU e GPU.

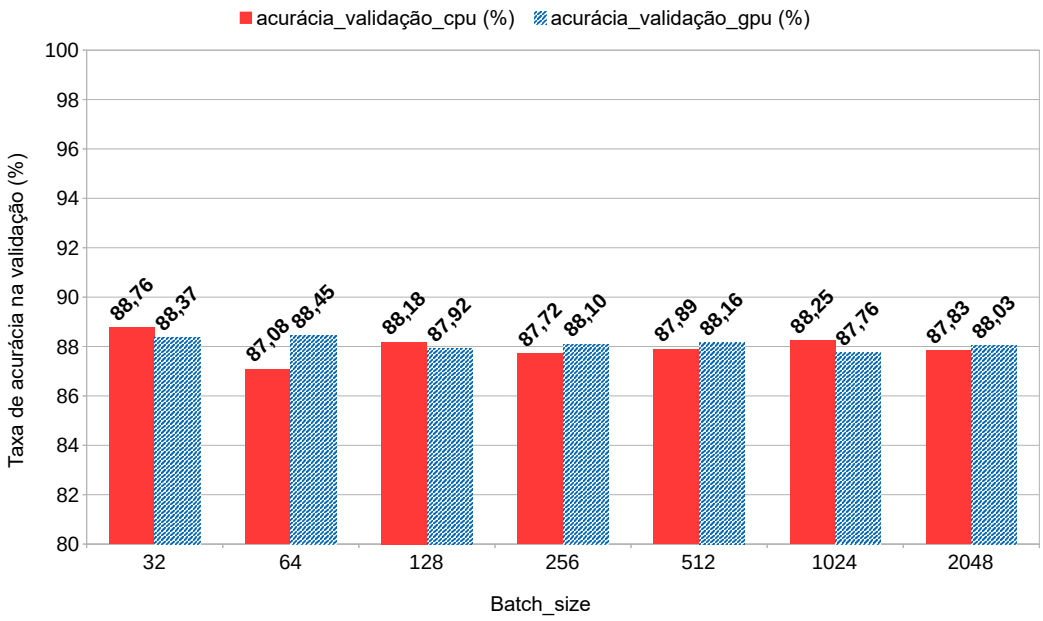

Figura 2. Métricas dos dados de validação das execuções em CPU e GPU.

A Figura 3 apresenta os tempos de treinamento conforme o *batch_size* definido. Percebe-se que conforme o tamanho do *batch* aumenta, o tempo de execução é reduzido consideravelmente em ambas arquiteturas. As execuções em CPU apresentaram um ganho de $3,17$ vezes. Isso reduz o tempo de execução de $1066,02$ segundos para $336,64$ segundos. Nas execuções em GPU, o ganho chegou a $12,15$ vezes, saltando de $2098,66$ segundos para $172,73$ segundos, isto representa um ganho de $91,8\%$. O ganho é bem expressivo se comparado a CPU. Entretanto, nas execuções com *batch_size* menor, é possível perceber que o tempo de execução em GPU praticamente dobra em relação ao tempo da CPU. Isso ocorre devido a GPU levar mais tempo

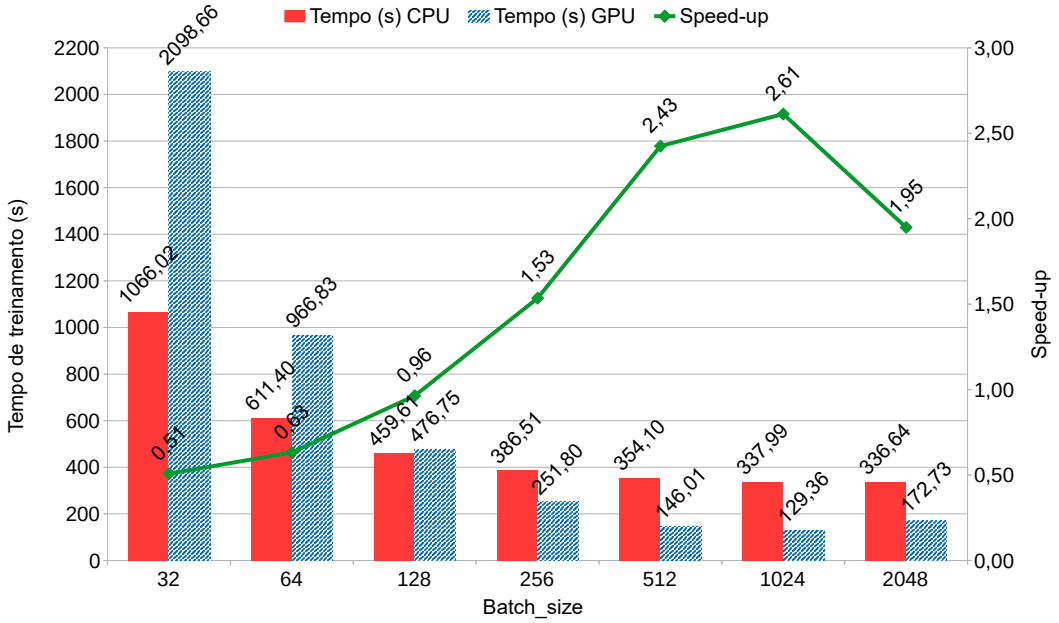

Figura 3. Tempo de treinamento e *Speed-up* da RNA nas arquiteturas CPU e GPU.

para a transferência dos dados, do que para o treinamento propriamente dito. Quando o número de instâncias aumenta, é possível utilizar mais da capacidade que a GPU oferece.

O *speed-up* da aplicação também é demonstrado na Figura 3, apresentando seu melhor caso no treinamento utilizando *batch_size = 1024*. Isto representa um ganho de $2,61$ vezes sobre a CPU, reduzindo o tempo de execução de $337,99$ segundos da CPU para $129,36$ segundos da GPU. Isso se deve ao fato da GPU dedicar mais recursos para o treinamento ao invés de priorizar a transferência dos dados. Entretanto, quando o limite de recursos computacionais é atingido, também provoca perda de desempenho, como pode ser visto ao aumentar o *batch_size* para 2048. Isso acontece devido a ter mais processos executando e dividindo os recursos computacionais.

Para validar a proposta, foi selecionado o modelo treinado que apresentou a menor taxa de *perda* e a maior taxa de *acurácia*, ambos valores obtidos do *dataset* de validação. Isso resultou no modelo treinado em GPU com *batch_size = 2048* (Figura 4). O modelo é carregado pelo algoritmo da RNA, e então as entradas são submetidas para teste. As novas entradas são *reviews* aleatórias obtidas da página oficial do IMDb[1] sobre o filme *Wonder Woman (2017)*.

Os resultados obtidos estão demonstrados na Tabela 1. A primeira coluna indica a *review* selecionada. A segunda coluna é a quantidade de estrelas indicadas pelo autor da avaliação, considerando como sendo uma opinião negativa as avaliações entre 1 e 5 e opinião positiva as avaliações entre 6 e 10. A última coluna indica a predição feita pelo nosso modelo. Essa predição é uma distribuição de probabilidades, em que nosso algoritmo calcula qual a probabilidade de o conteúdo ser uma opinião positiva ou negativa, e então sinaliza para aquela com maior taxa. Ao analisar a tabela, percebe-se que o modelo de RNA testado foi assertivo em todos os casos submetidos. Isto representa uma assertividade média de $94,71\%$. Após a análise e rotulação manual da opinião expressada em cada *review*, é possível comprovar que o modelo de RNA classificou a opinião corretamente.

[1] https://www.imdb.com/

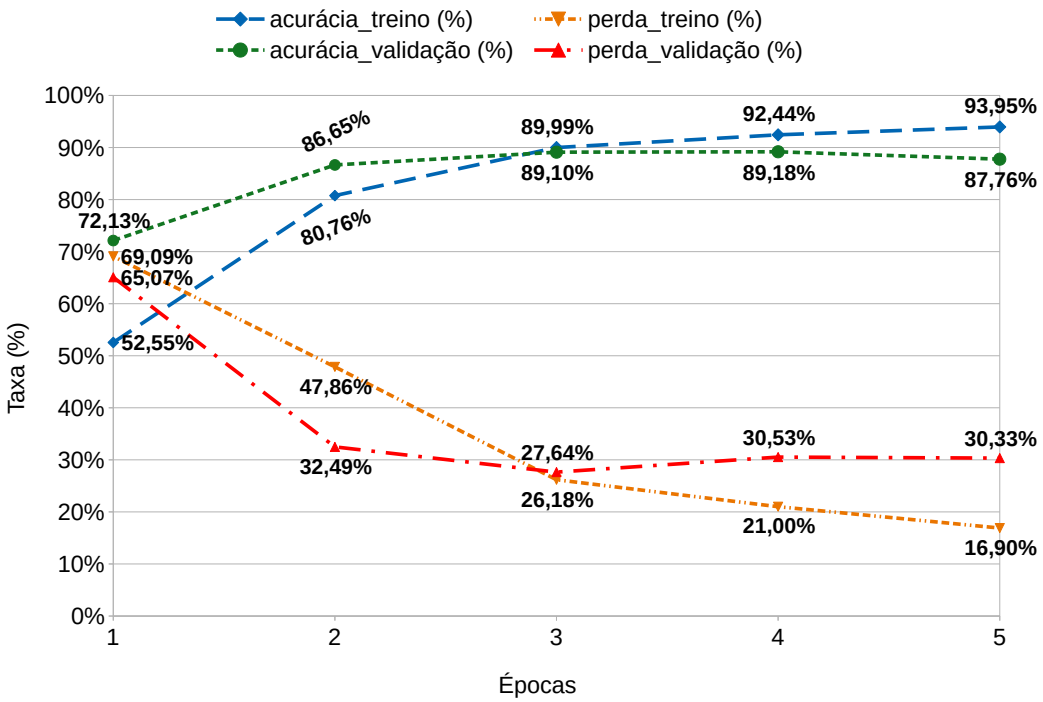

Figura 4. Métricas do treinamento do modelo selecionado.

Review	Avaliação	Predição da RNA
1	1/10	99.22% N
2	5/10	95.80% N
3	7/10	82.83% P
4	9/10	99.74% P
5	1/10	99.54% N
6	10/10	99.71% P
7	9/10	99.93% P
8	5/10	99.01% N
9	8/10	78.01% P
10	3/10	93.40% N

Tabela 1. Resultados da validação da proposta. Predição: negativa (N), positiva (P)

5. Conclusões e trabalhos futuros

Este trabalho abordou o uso de redes neurais artificiais recorrentes *Long Short-Term Memory* na classificação de sentimentos em sentenças. Utilizou-se diferentes configurações para analisar o desempenho da aplicação executada em arquiteturas CPU e GPU. No que tange a taxa de *acurácia*, para arquitetura CPU, a melhor taxa é registrada para *batch* de tamanho 32. Já para arquitetura GPU, a melhor taxa é obtida para *batch* de tamanho 64. A proposta apresentou bom desempenho no que se refere à precisão da RNA. O tempo de execução é reduzido consideravelmente em ambas as arquiteturas ao aumentar o tamanho do *batch*. Dentre os ganhos mais significativos, o tempo de execução em CPU apresentou ganho de $3,17$ vezes. O tempo de execução em GPU obteve redução ganho de $12,15$ vezes. Comparando as arquiteturas, o tempo de execução em GPU apresentou redução de até 61%, isso representa um *speed-up* de $2,61$ vezes sobre o tempo da CPU. Estes resultados indicam que a utilização de GPU reduz o tempo de treinamento de RNAs, apresentando bons rendimentos para tamanhos de *batch* maiores.

Como trabalhos futuros, uma primeira iniciativa poderia ser a utilização de *word embeddings* pré-treinados. Além disso, é possível estudar a influência de outros hiperparâmetros, como taxa de aprendizagem, taxa de *Dropout* e Função de Ativação. Também é possível avaliar base de dados maiores e em outros idiomas. Uma outra análise que pode ser feita é adicionar outras classes de classificação além de positivo e negativo, como por exemplo, muito positivo, muito negativo e neutro. Como também avaliar outras fontes de dados como *twitter*. Outra abordagem a ser estudada é uma melhor separação das bases de treino, teste e validação.

Referências

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

dos Santos, I. P. P. (2017). Análise de sentimento usando redes neurais de convolução. Dissertação (Mestrado em Engenharia Eletrônica), Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brazil.

Hu, Z., Hu, J., Ding, W., and Zheng, X. (2015). Review sentiment analysis based on deep learning. In *2015 IEEE 12th International Conference on e-Business Engineering*, pages 87–94. IEEE.

Li, C., Xu, B., Wu, G., He, S., Tian, G., and Hao, H. (2014). Recursive deep learning for sentiment analysis over social data. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 2, pages 180–185. IEEE.

Liu, B. et al. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2(2010):627–666.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Moolayil, J. (2019). An introduction to deep learning and keras. In *Learn Keras for Deep Neural Networks*, pages 1–16. Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Rossum, G. (1995). Python reference manual. Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, NLD.

Walker, M. A., Anand, P., Abbott, R., Tree, J. E. F., Martell, C., and King, J. (2012). That is your evidence?: Classifying stance in online political debate. *Decision Support Systems*, 53(4):719–729.