A Deep Learning Approach to Prioritize Customer Service Using Social Networks

P. R. P. Amora[1], E. M. Teixeira[1], M. I. V. Lima[1], G. M. Amaral[1], J. R. A. Cardozo[2], J. C. Machado[1]

[1] LSBD - Department of Computer Science, Universidade Federal do Ceará, Fortaleza CE, Brazil
{paulo.amora,elvis.teixeira,isabel.lima,gabriel.maia,javam.machado}@lsbd.ufc.br
[2] Digitro Tecnologia SA, Brazil
jose.cardozo@digitro.com.br

Abstract. Social networks provide a rich source of opinionated content, which may give an insight on how customers are reacting to a company's service. An interesting application in this sense is to analyze customer feedback on social media to prioritize service or response times to customers in critical situations or to the most unsatisfied. In this work, we present a deep learning approach using LSTM networks applied to the sentiment analysis of tweets, aiming customer service prioritization through the introduction of a new class to help characterize the level of dissatisfaction with the service provided, as well as a comparative study with more traditional machine learning approaches.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*; *Language parsing and understanding*; I.2.6 [**Artificial Intelligence**]: Learning—*Concept Learning*

Keywords: Sentiment Analysis, Deep Learning, LSTM, Social Networks, Customer Service

1. INTRODUCTION

The widespread use of social networks have made it possible to take advantage of real users' content through posts, comments and reviews. By analyzing this kind of data we are able to extract relevant information, like opinions and emotions. Sentiment analysis (or opinion mining) is a known technique used to classify texts according to their prevalent polarity, in order to infer from texts the sentiment of their author [Liu 2015]. This is helpful when one wishes to identify if a certain opinion is positive or negative. One of the possible applications of the sentiment analysis technique is using its output as a lead on what are the comments that contain more displeasingness with a service provided, when looking to a company's feedback channel, be it a forum or social networks.

A reasonable parameter for prioritizing customer requests is their level of dissatisfaction. Usually, the most dissatisfied customers are the ones that are waiting the longest for a response from the service provider, therefore, if the service provider can infer which customers are more dissatisfied, it makes sense to try to solve the most critical cases first. However, analyzing only the polarity of a sentence may be insufficient for correct customer service prioritization because there may be much more negative messages than positive ones and, in this scenario, the majority of the cases would be prioritized. This fact is not directly related to the quality of the product or service, but reflects the fact that people are probably more likely to post complaints than recommendations or satisfaction notes. This is the main reason for our interest in separating the critical cases from the negative ones, to allow efficient customer service by the provider.

This publication is the result of a project promoted by the Brazilian Informatics Law (Law No. 8248 of 1991 and subsequent updates) and was developed under the Cooperation Agreement 01/2016 between Federal University of Ceara and Digitro Tecnologia SA. This research was also partially supported by LSBD/UFC.

In this work, besides the usual Positive and Negative labels, we introduce a new label, called Critical, in which the most pressing texts are classified. These three classes are used to separate a corpus composed of social media texts, in this case, collected tweets that mention major Brazilian telecommunications operators. The semantic similarity between the Negative and Critical classes presents a challenge in classification work because the separation between them is not clear even for the task of labeling data for supervised learning. Therefore, an LSTM network was leveraged against several classic machine learning approaches in this three-class universe with the goal of classifying the customer messages that deserve immediate attention. The contributions of this work are the following:

—A study of a sentiment analysis problem, using social media in Brazilian Portuguese language and geared towards customer service scheduling;

—A corpus focused on a customer service perspective and a labeled dataset for supervised learning.

—An investigation on how LSTM networks can outperform traditional machine learning approaches in this scenario.

2. THEORETICAL BACKGROUND

2.1 Long Short-Term Memory Units (LSTM)

Recurrent neural networks (RNNs) assume the existence of correlations between different inputs to the network, be them semantic, causal, logical or simply temporal connections. The basic architecture for a RNN is the Simple Recurrent Network (SRN), proposed by Elman et al. [Elman 1990], consisting of a modification of a backpropagation network. In the SRN, the input is partitioned into a stream and the first element is fed into the input layer. The hidden layers take this information and also the contents of a context layer, which represents the internal state of the network. Results from the hidden layers are then used to update the contents of the context layer and more data is fetched from the input stream into the input layer. Like in backprop networks, SRN relies on back-propagation algorithms to update weights and biases, including from context layer to hidden layer.

An RNN, then, possesses directed loops, allowing information about data already processed by the network to be fed backwards and used as input. Maintaining an internal state grants it a dynamical reaction to any series of inputs, where past information plays a part in determining the RNN's reaction to incoming information. However, such traditional RNNs find problems in their gradient-based updates, in particularly the vanishing gradient problem, as evidenced by Hochreiter et al. [Hochreiter et al. 2001]. The Long Short-Term Memory (LSTM) network, proposed by Hochreiter et al.,[Hochreiter and Schmidhuber 1997], avoids such gradient problems. LSTM is a deep learning system, consisting of a cascading number of recurrent network units, each specialized in "remembering" information from past units, be it for a short or long duration.

LSTM units have an inner state vector and gates. A gate is a vector, whose values range from 0 to 1, which controls the flow of information inside the unit. They are computed through logistic functions, and then multiplied by the flowing information. The inner state vector tracks which values from the input stream are to be remembered or forgotten. All vectors have weights and biases associated to them, which are the only parameters of the LSTM and are trained with "backpropagation through time" [Mozer 1995].

Figure 1, adapted from [Olah 2015], illustrates the LSTM unit utilized. The unit t consumes the word t (x_t), along with the output and inner state of unit $t-1$ (h_{t-1} and c_{t-1}), and calculates the values of three gate vectors f_t, i_t and o_t, represented in said order by yellow σs, which are then used with an hyperbolic tangent function ($tanh$) and sum ($+$) and Hadamard product (x) operations to calculate its new inner state vector c_t and its output vector h_t.

Gates i_t, the input gate, and f_t, both regulate the data flow into the unit's inner state. While i_t controls how much of it comes from input, f_t controls how much comes from the last unit's hidden state. The output gate o_t decides this state's influence in the output.

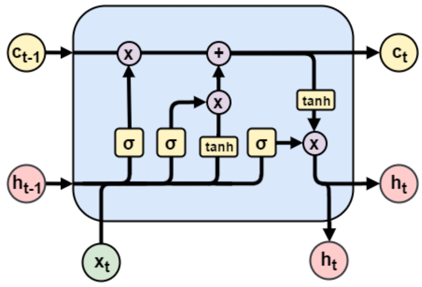

Fig. 1: Flow and structure of a LSTM unit.

2.2 Word Embedding

An embedding layer was employed to transform the textual inputs into numerical features. This layer is also trainable for better representing the words. Assuming that the layer is pre-trained to optimally represent the language's great majority of known and valid words, and providing it with space for new word entries, it then trains itself to better understand new vocabulary. This allows it to evade errors coming from the use of language in the Twitter platform, such as typos, colloquialism, sarcasm and other variants carried with sentimental value over syntactic.

After embedded into numeric vectors, words are given one-at-a-time to LSTM layers consisting of a recurrent unit each, which unfolds through time into a sequence of units. In addition, all LSTM layers apply the dropout technique to avoid overfitting, as described in [Hinton et al. 2012].

The sequence of output vectors of each LSTM layer is given to the next one as input vectors. The last LSTM layer is followed by a series of densely-connected regular neural network layers. Each neuron i in a dense layer processes a matrix X of all the layer's input into an output y_i through a function $y_i = \sigma(W_i X + b_i)$, where σ is a softmax function, and W_i and b_i are trainable parameters. The last dense layer is composed of n neurons, where n is the number of classes, each outputting a value $z_i \in [0, 1]$ for the tweet's belonging to class i. We finally take the highest value to represent the tweet's class.

3. COMPLAINT PRIORITIZATION

When customers manifest publicly their issues with service providers, it can cause harm to a company or product/service image. Besides customer service centrals, customers and users can also manifest their dissatisfaction and problems with the service provided through social networks, which have a much wider range of influence and affect the business directly, even if it's not a standard form of contact. On those grounds, companies today have teams of customer service for social media, with the mission of answering those complaints in able time, to preserve the company image.

This scenario becomes unfeasible when the number of complaints and comments directed at a company is so large that the people assigned to respond them are not enough. Therefore, some kind of screening is necessary in order to allow the customer service teams to perform well. To do this screening and further prioritization, we propose the use of sentiment analysis. However, the prioritization is not straightforward because:

—The number of complaints far outweighs the number of compliments and neutral inputs.
—Given that we have several complaints, which ones seem more urgent?

The first point refers to a known problem in classification tasks, which are unbalanced classes.

To address these points, instead of the traditional Positive/Negative classes, we propose a new label, called Critical, in which the most pressing complaints are classified. Making a binary classification problem a multi-class one has its drawbacks when regarding accuracy. Also, the Critical label is semantically very close to the negative class. The clear advantage is that we now have a class to discriminate which complaints should be prioritized.

To tackle this classification problem, we decided to use LSTM networks along with more traditional machine learning approaches, such as Support-Vector Machines, Naive-Bayes, Decision Trees, Random Forests, K-Nearest Neighbors and an Ensemble Voting classifier composed by all these approaches besides LSTM.

The first step is gathering the data. Given the context of our application, we decided to create the dataset, making use of the Twitter API to collect data from telecommunication companies. The kind of data that we are most interested in are mentions from other users to the company profile, because that's where the complaints come from. We decided to monitor 2 Brazilian companies from 12/16/2016 to 01/11/2017 to obtain this data. From that, 15,991 tweets were collected and 3,000 were randomly selected to be our corpus.

These 3,000 tweets were then manually labeled by one person, according to our three-label class system (Positive, Negative or Critical). As mentioned before, the Critical class is used on the most pressing tweets, which we define as the tweets that might damage more the company's image. The criteria to consider a tweet critical involved mentions of out-of-service times, rage, cursing, among others, and combinations of those. From that, we know who are the most helpless clients and from this characterization, may direct customer support to them.

Table I: Labeling system

Label	Characteristics	Example (translated)
Positive	appreciation; neutral	"Thank you for fixing my Internet, just don't do it again"
Negative	complaint; impatience	"Hey, are you really not going to answer my DM??"
Critical	threat; curse; anger	"I'm so angry, I just want to cancel my account, don't make me sue you"

Afterwards, we need to prepare the data for training. For this, we used known language pre-processing techniques, such as tokenization, stop words removal, case lowering and filtering of irrelevant information (URLs, Twitter mentions, retweet indicators, etc.). Stemming was also considered as part of the pre-processing, but its inclusion did not present a relevant impact on the overall performance. 861 tweets were labeled as Positive, 1,413 as Negative and 685 as Critical. In table I, the main characteristics we used to guide the labeling process are listed, along with an example tweet for each label. Of course the labels given to the messages, specially the critical one, are completely dependent on the application domain, and thus a different set would have been considered critical if the domain was different from telecommunications.

We then vectorize the data, since machine learning algorithms cannot deal with natural language. For LSTM, we used the word embedding layer, which is a special input layer, that is fed each tokenized tweet. It converts each word token into its embedded form, reading the tweet as a stream of vectors. It offers a faithful representation of the words' semantics through its location inside the vector space, boosting the accuracy of sentimental analysis tasks [Socher et al. 2013] by letting it calculate directly on a word's meaning, while the LSTM network itself is able to reason with the text structure.

For the traditional machine learning approaches, a process where a relevance score is assigned to each word in the corpus using TF-IDF (Term Frequency-Inverse Document Frequency) was applied. For each document – a tweet in our case – in the corpus, a vector of TF-IDF scores corresponding to the words contained in it is computed and used as the set of features for training and testing the classifiers.

The next step after vectorization is to train the machine learning models and the LSTM network.

Finally, the output provided by the trained classifiers on real data is used as a decision support mechanism on what clients are the most impacted and in need of immediate assistance.

4. EXPERIMENTAL EVALUATION

We now present how the LSTM perform against the other, more traditional machine learning approaches.

4.1 Setup

The experiments were conducted on an Intel Core I5 3.1GHz running Ubuntu 16.04 LTS, with 8GB of RAM. To implement the traditional machine learning approaches, we used Python 3.4 with SciKit Learn 0.18.1. For LSTM we used Keras[Chollet 2015].

4.2 Experiments

For validation purposes, the dataset was shuffled and divided as following: $\frac{2}{3}$ for training and $\frac{1}{3}$ for testing.

After the pre-processing steps mentioned, we applied the dataset to the models. One thing observed is that some models performed poorly on default parameters. For that reason, some parameters were adjusted. For example, in K-Nearest Neighbors, the parameter K was adjusted to select the optimal value. On SVM, the different kernel types (Linear, RBF, Polynomial and Sigmoid) were tested and the *gamma* factor was varied. Decision Tree had the node split criterion changed between Information Gain and Gini Impurity. The amount of trees was changed in Random Forest.

The LSTM network's optimal architecture, concerning quantities of layers, units and nodes, was decided mainly through trial and comparison. Dropout rates were obtained similarly. The best performing network consisted of three LSTM layers, each with 200 units and dropout rate of 0.3, and one dense layer with three neurons.

Figure 2 depicts the architecture employed. Initially, the tokenized tweet is fed to the word embedding layer (WE). Each word w_i is thus embedded in an input vector x_i. The inputs are sequentially processed by the LSTM layers, whose output passes through a dense activation layer. The three final outputs are the tweet's pertinence levels to each class.

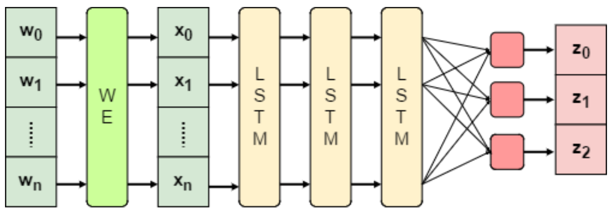

Fig. 2: Architecture of the LSTM network employed.

The embedding layer of the network was initialized by a generator model, trained with the aid of the set of words encountered in the Portuguese Wikipedia. The adopted procedure was the skip-gram model proposed in [Bojanowski et al. 2016], which takes word morphology into account. The skip-gram is an instance of the Word2vec models, a group of embedding generating models which focuses on spatially approximating words with common contexts [Mikolov et al. 2013].

The optimally initialized embedding layer was then given freedom to learn the embedding of new words encountered specifically inside our model's domain, while also changing the vectorization of known words if needed.

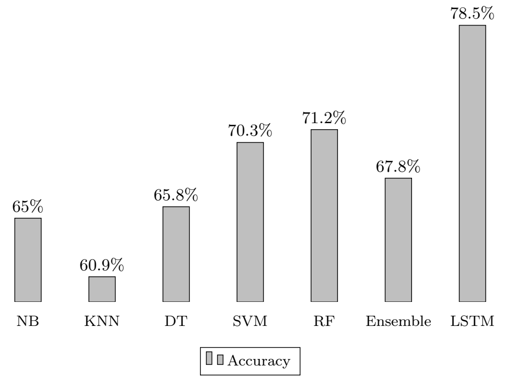

Fig. 3: Overall accuracy

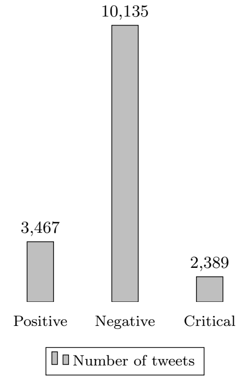

Fig. 4: Class dispersion

4.3 Results

In order to properly evaluate the obtained results, we relied on the following metrics: overall accuracy; precision, recall and f-score of the Critical class. For the specific problem presented in this article, the results of the Critical class are more important than correctly classifying all classes, because we are interested in prioritizing service exclusively to the customers in more pressing situation.

Table II: Results: critical class

	Precision	**Recall**	**F-score**
Naive-Bayes	0.853658	0.432098	0.57377
KNN	0.846153	0.507204	0.634234
Decision Tree	0.76923	0.576131	0.658823
SVM	0.84375	0.555555	0.669975
Random Forest	**0.896103**	0.567901	0.695214
Ensemble	0.68018	0.621399	0.649462
LSTM	0.838488	**0.847222**	**0.842832**

The results regarding the classification of the Critical class are shown in table II. Because the f-score is the harmonic mean of precision and recall, methods with good precision but low recall, like Naive-Bayes, end up having a lower f-score compared to methods which have these two metrics more balanced out. SVM, a method that tends to produce the best results in related works, was slightly outperformed by Random Forest, and by LSTM. As expected, it far outperforms the traditional machine learning approaches, with an F-Score of approximately 0.842.

The overall accuracy of each classification method can be seen on Figure 3. KNN, despite all efforts in adjusting the number of neighbors to an optimal number, still performed poorly compared to the other methods. SVM and Random Forest once again presented good results, but with LSTM outperforming both in accuracy by 8.2% and 7.3% respectively.

Also, for further analysis, a new model with all labeled tweets was trained and the tweets from the original dataset were tested against it. In Figure 4, a chart illustrates the distribution of the classified tweets over the three classes. As expected, the number of negative tweets is considerably higher than the positive ones, which validates our initial claim that customers are more inclined to write complaints rather than compliments on social networks. More importantly, the number of critical tweets is about 25% of the negative ones, which is much more realistic to manage from the customer service point of view. This means that the creation of the Critical class has made it possible to break down the large and unfeasible to handle Negative class, thus allowing service to be directed to a smaller and more relevant set of clients.

5. RELATED WORKS

Sentiment analysis, as defined by [Liu 2015] and [Pang and Lee 2007], is the science that analyzes what are people's opinions and impressions towards a target product or service based on written texts, such as reviews or comments. In this context, social networks are an important source of such content, given that, nowadays, increasingly more businesses have an online presence and interact in real time with their customers.

In sentiment analysis, most works can be classified according to the method they are based on, which is usually either lexicon-based or using machine learning. [Taboada et al. 2011], [Balage Filho et al. 2013] and [Souza and Vieira 2012] are good examples of works that use a thesaurus (lexicon) to identify the polarity of texts written in Portuguese. The main challenge of the lexicon approach is that you have to design an algorithm considering the quality of the lexicon and the application domain. Machine learning approaches, such as ours, have the advantage of not having to concern themselves on how to interpret the words.

The applications of deep learning in the field of sentimental analysis, while new, are increasingly pertinent and promising. As shown by [Rojas-Barahona 2016], different deep networks may hold solid ground on polarity tasks when analyzing large texts, such as reviews. Timmaraju et al [Timmaraju and Khanna 2015] evidences this potential by showing a model based on recurrent networks able to reach test accuracy levels close to those of featured-engineered models without the need for any handcrafted feature. Furthermore, Liu et al. [Liu et al. 2016] shows possible enhancements to these models, as it provides different architectures of deep recurrent networks wielding improved results through multi-task learning. Rojas et al.[Rojas-Barahona 2016] also notes its viable application to twitter data, should the network and the data be finely tuned and balanced. Indeed, the short length, complex language, abbreviated vocabulary and often implicit inflections, such as sarcasm, on text data are constraints that beg for cautious pre-processing of data, as remarked by [Yuan and Zhou 2015].

Works such as [Lee and Dernoncourt 2016] avoid the text length problem by laying tweets in time-sequences, grouped by user, and classifying tweets while holding information of the ones behind them. This approach manages state-of-the-art results.

Adversities emerging from the language, specifically Portuguese, due to its complexity, have been mainly noted and treated by [Alves et al. 2016]. Even though they utilize Naive Bayes and SVM classifiers, their work demonstrates that sentimental analysis applied to tweets under such adversities can wield favorable results if pre-processing is carefully taken care of. They manage it through the manual correction of grammar mistakes, unnecessary characters, abbreviations, slangs and non-words. Our approach shows effectiveness even when data is used as-is. Even the most basic pre-processing had no manual interference.

In [Dosciatti et al. 2013], Support Vector Machines are used to identify six different emotions, listed neutral, displeasure, joy, anger, fear, surprise and sadness, in texts from online newspapers. The results were compared with those of two other experiments, using K-Nearest Neighbors and Naive-Bayes.

The approach taken by [Lee and Dernoncourt 2016] falls short when the tweet sequence consists of quick replies with high variation of tone and subject, as it happens in costumer support exchanges, leaving us with single-tweet analysis. And, unlike [Alves et al. 2016], we focus on making the least amount of text correction possible while maintaining test accuracy levels. We chose instead to train our model to adapt to this new linguistic environment by letting the network's word embedding layer more flexible. We followed the conclusion of [Tang et al. 2014], which states that a word embedding concerned solely on syntactic context performs worse than one unifying syntax and sentiment information. We therefore let the language variations be perceived as sentiment information by the network and observe that it satisfactorily reduces hindrances as slangs, typos and sarcasm.

6. CONCLUSION

We have presented a technique to prioritize customer service through social networks using sentiment analysis. By splitting customers into three classes, according to their level of dissatisfaction, we were able to identify the customers with more urgent needs, thus saving time and making overall service more efficient, in a very effective way. Our experiments show that LSTM networks can largely outperform traditional machine learning approaches in this scenario, and can be a very good fit to solve this problem. We also observed that even with small sized inputs, LSTM was a valid approach.

For future work, customer feedbacks from other social networks that allow more text, like TripAdvisor or Yelp, can be analyzed to validate or enhance the LSTM network architecture.

REFERENCES

Alves, A. L. F., de Souza Baptista, C., Firmino, A. A., de Oliveira, M. G., and de Paiva, A. C. A spatial and temporal sentiment analysis approach applied to twitter microtexts. *Journal of Information and Data Management* 6 (2): 118, 2016.

Balage Filho, P. P., Pardo, T. A., and Aluísio, S. M. An evaluation of the Brazilian Portuguese LIWC dictionary for sentiment analysis. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology (STIL)*. pp. 215–219, 2013.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

Chollet, F. Keras, 2015. Accessed: 2017-05-22.

Dosciatti, M. M., Ferreira, L. P. C., and Paraiso, E. C. Identificando emoções em textos em português do brasil usando máquina de vetores de suporte em solução multiclasse. *ENIAC-Encontro Nacional de Inteligência Artificial e Computacional. Fortaleza, Brasil*, 2013.

Elman, J. L. Finding structure in time. *Cognitive science* 14 (2): 179–211, 1990.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation* 9 (8): 1735–1780, 1997.

Lee, J. Y. and Dernoncourt, F. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827*, 2016.

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

Liu, P., Qiu, X., and Huang, X. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Mozer, M. C. A focused backpropagation algorithm for temporal. *Backpropagation: Theory, architectures, and applications*, 1995.

Olah, C. Understanding lstm networks, 2015. [Online; accessed 2017-04-26].

Pang, B. and Lee, L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2 (1-2): 1–135, 2007.

Rojas-Barahona, L. M. Deep learning for sentiment analysis. *Language and Linguistics Compass* 10 (12): 701–719, 2016.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C., et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631. Citeseer, pp. 1642, 2013.

Souza, M. and Vieira, R. Sentiment analysis on twitter data for portuguese language. In *International Conference on Computational Processing of the Portuguese Language*. Springer, pp. 241–247, 2012.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. D., and Stede, M. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37 (2): 267–307, 2011.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*. pp. 1555–1565, 2014.

Timmaraju, A. and Khanna, V. Sentiment analysis on movie reviews using recursive and recurrent neural network architectures. *Semantic Scholar*, 2015.

Yuan, Y. and Zhou, Y. Twitter sentiment analysis with recursive neural networks, 2015.