CmpE 492 Project

Twitter Sentiment Analysis for Brands and Corporations

1. Introduction and Motivation

Sentiment analysis (i.e. opinion mining) is a technique in natural language processing (NLP) that aims to identify and classify the emotion (positive, negative, neutral) related to a text. It is especially helpful for brands and corporations' marketing teams, making it easier to identify and analyze customer feedback automatically. Brands and corporations use sentiment analysis to track the customer opinions about their brands and understand the feedback.

In light of this information, this project aims to create a high-functioning sentiment analysis tool that successfully identifies the sentiment of a tweet or short customer feedback. Since people often form opinions and post them on Twitter for many people to see, it was important that this system also performs well on tweets, as well as general feedback.

2. State of the Art

As stated above, many brands are interested in sentiment analysis tools. When there is that much demand, it is safe to say that there should be a large supply too. However, the quality of some of the current supply of sentiment analysis tools could be said to be questionable. Two of the state-of-the-art solutions to sentiment analysis are as follows:

- MonkeyLearn: MonkeyLearn is a company that primarily works on supplying machine learning solutions to brands. Most of their solutions do not need any extra coding work, therefore are very practical for most small companies. Their sentiment analysis tool which also allows tagging and topic classification is while practical, not necessarily always well-performing.
- Brandwatch: Brandwatch is a company that works on supplying social media analysis tools to corporations like Unilever, Havas and Nestle. Unlike MonkeyLearn, they don't allow the users to customize what they want to do. They make social media analysis using ML techniques for opinion mining and topic classification, however they don't share their specific techniques.

In light of this, it can be said that current techniques either lack customization or high-performance. In addition to the commercial tools like MonkeyLearn, there are many research papers about sentiment analysis techniques. One of them is by Eryka Probierz *et. al.*[1] and identifies the sentiments of Covid-19 related tweets using Logistic Regression and Convolutional Neural Networks. Another research that is closer to this one on sentiment analysis is made by Salhi *et al*[2], that explores electronic reputations of companies in social media, using sophisticated preprocessing, SVM and Logistic Regression methods. While all of these researches are very helpful and successful, they don't include state of the art methods like LSTMs, which this project makes use of.

3. Methods

In this project, I have made use of a Bidirectional LSTM network structure along with some elaborate preprocessing. Preprocessing includes:

- deconstruction of short forms like isn't, haven't, I've, you're to longer forms like is not, have not, I have, you are etc.
- removing mentions (@'s)
- removing # marks but leaving the actual hashtags
- splitting words and punctuation marks with a single space
- getting dependency relations and part-of-speech information of each word in a sentence/document

After preprocessing, word embeddings for document words, part-of-speech tags and dependency relation words are calculated with training a Word2Vec model. Part-of-speech tags and dependency relation information is retrieved by using StanfordNLP's[3] Stanza library. This concludes the part before feeding all this data to the neural network.

---

[1] Eryka Probierz et al. (2021). *Twitter Text Data from #Covid-19: Analysis of Changes in Time Using Exploratory Sentiment Analysis.* ISAIC 2020. Retrieved from https://www.researchgate.net/publication/349807254_Twitter_Text_Data_from_Covid-19_Analysis_of_Changes_in_Time_Using_Exploratory_Sentiment_Analysis_Twitter_Text_Data_from_Covid-19_Analysis_of_Changes_in_Time_Using_Exploratory_Sentiment_Analysis

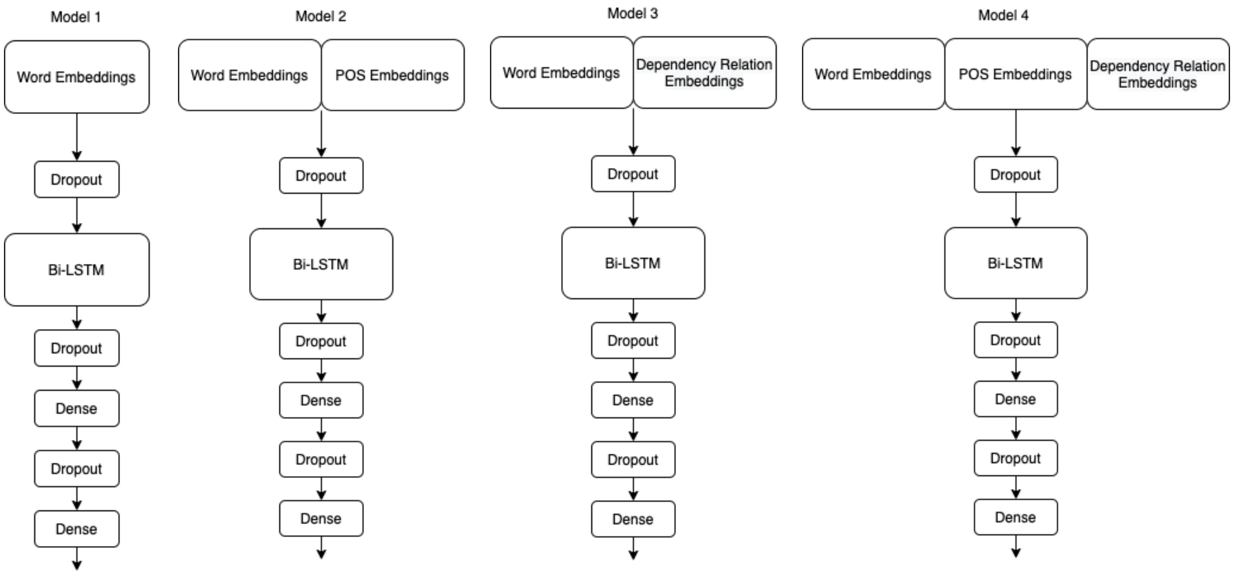
[2] Salhi et al. (2021, March). *Using E-Reputation for Sentiment Analysis: Twitter as a Case Study.* International Journal of Cloud Applications and Computing, *11*(2). Retrieved from https://www.researchgate.net/publication/349895914_Using_E-Reputation_for_Sentiment_Analysis_Twitter_as_a_Case_Study

[3] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Association for Computational Linguistics (ACL) System Demonstrations. 2020. [pdf][bib]

The neural network used is a Bidirectional LSTM architecture with embedding layers concatenated (word embeddings only, word embeddings+part-of-speech embeddings, word embeddings + dependency relation embeddings or word embeddings + part-of-speech embeddings + dependency relation embeddings depending on the model). The idea of concatenating part-of-speech and dependency relation embeddings with word embeddings comes from Zeynep Yirmibeşoğlu and Tunga Güngör's paper about Multiword Expression Identification[4]. In my adaptation of their technique, I trained word embeddings for words, part-of-speech tags and dependency relations, then created 3 different embedding layers for these and concatenated subsets of them to feed to the Bidirectional LSTM model. The different subsets that I tried are as follows:

- Model 1: Document word embeddings
- Model 2: Document word embeddings + Part-of-speech tag embeddings
- Model 3: Document word embeddings + Dependency relation embeddings
- Model 4: Document word embeddings + Dependency relation embeddings + Part-of-speech tag embeddings

Schemas for the models are shown in Figure 1 below.



[4] Yirmibeşoğlu and Güngör. (2020). *ERMI at PARSEME Shared Task 2020: Embedding-Rich Multiword Expression Identification.* Boğaziçi University, Istanbul, Turkey.

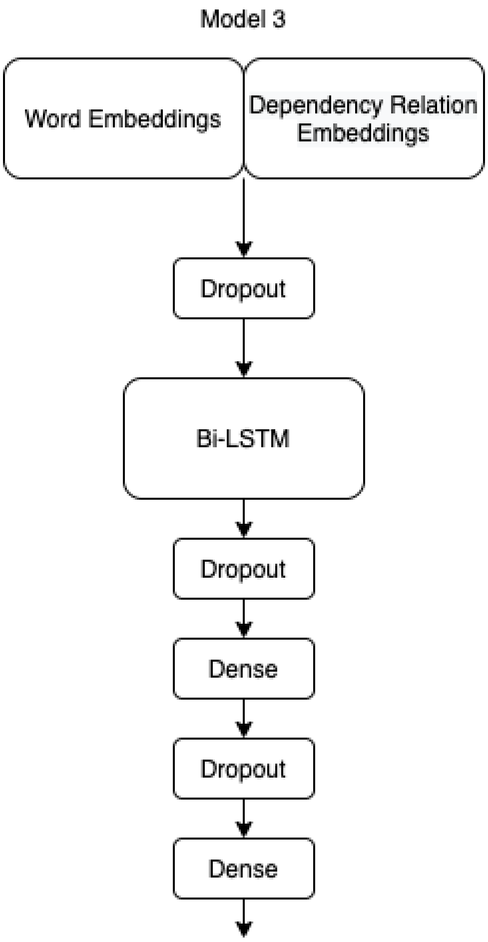


Figure 1: Bi-LSTM models used in this project.

The input to those models are an embedding layer, specifically the embeddings shown in the figure 1, concatenated to a single embedding layer. Word embeddings are trained using Gensim's[5] Word2Vec model and then fed to the network implemented using Keras[6] with TensorFlow[7] backend. Embedding vectors have 100 dimensions. The raw corpora[8] contains 14640 tweets that tags different US airline companies, the sentiments of the tweets (positive, negative, neutral) and other information like negative reason, airline name, retweet count, etc. Only tweet texts and their sentiments are used for this paper's purposes.

4. Results

---

[5] Rehurek, R., & Sojka, P. (2011). *Gensim–python framework for vector space modelling*. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2).

[6] Chollet, F. et al.. (2015). *Keras*. GitHub. Retrieved from https://github.com/fchollet/keras

[7] Abadi, M. et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org.

[8] Figure Eight. (2019, October). Twitter US Airline Sentiment, Version 4. Retrieved from https://www.kaggle.com/crowdflower/twitter-airline-sentiment .

Test results for all 4 models are as follows:

	Accuracy
Model 1 (word embeddings)	81.22%
Model 2 (word+PoS embeddings)	80.33%
Model 3 (word+dependency rel. embeddings)	80.81%
Model 4 (word + PoS + dep. rel. embeddings)	80.19%

From this data, it can be concluded that the most accurate model for the corpus is Model 1, which uses only word embeddings as the embedding layer, closely followed by Model 2. Although there are small differences, it seems like the additional information fed into the model lowers the accuracy score.

Among other models that used the same dataset to implement sentiment analysis models with LSTM, Model 1 is above the average by 2.5% accuracy, therefore it improves most of the models proposed before.

5. Conclusion and Discussion

In this project, I conducted a comparison between 4 models for sentiment analysis using Bi-LSTM model. While at first one could think that the usage of part-of-speech tags and dependency relations may improve the results of Bidirectional LSTM model, on the contrary, it seems that the extra information lowers the accuracy scores of the model. However, the decrease in accuracy is not much when Model 1 and Model 3 are compared. In my opinion, this decrease could be eradicable if the corpus is enlarged or the tweets in the corpus were longer.

Since tweet data is very different from regular language, this may explain why data like dependency relations and part-of-speech information have a negative effect on the accuracy of the model. With a corpus that resembles the natural language more, part-of-speech and dependency relation embeddings would probably be more meaningful.

6. Future Work

Apart from changing the network structure, this work could be further improved by adding tagging or turning it into intent analysis. By turning it into intent analysis, it could attract more attention from large brands and corporations.

This project should also be tried with a more balanced and wide ranging corpus that includes longer customer reviews. By that approach, part-of-speech tags and dependency relations would be more useful in model's predictions.

7. References

MonkeyLearn API Documentation. (n.d.). MonkeyLearn website. Retrieved from https://monkeylearn.com/api/v3/#introduction

Brandwatch Platform Introduction. (n.d.). Brandwatch website.Retrieved from https://www.brandwatch.com/

Yirmibeşoğlu and Güngör. (2020). *ERMI at PARSEME Shared Task 2020: Embedding-Rich Multiword Expression Identification.* Boğaziçi University, Istanbul, Turkey.

Figure Eight. (2019, October). *Twitter US Airline Sentiment*, Version 4. Retrieved from https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Abadi, M. et al. *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org.

Chollet, F. et al.. (2015). *Keras*. GitHub. Retrieved from https://github.com/fchollet/keras

Rehurek, R., & Sojka, P. (2011). *Gensim–python framework for vector space modelling.* NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. Stanza: *A Python Natural Language Processing Toolkit for Many Human Languages.* In Association for Computational Linguistics (ACL) System Demonstrations. 2020. [pdf][bib]

Salhi et al. (2021, March). *Using E-Reputation for Sentiment Analysis: Twitter as a Case Study*. International Journal of Cloud Applications and Computing, 11(2). Retrieved from https://www.researchgate.net/publication/349895914_Using_E-Reputation_for_Sentiment_Analysis_Twitter_as_a_Case_Study

Eryka Probierz et al. (2021). *Twitter Text Data from #Covid-19: Analysis of Changes in Time Using Exploratory Sentiment Analysis.* ISAIC 2020. Retrieved from https://www.researchgate.net/publication/349807254_Twitter_Text_Data_from_Covid-19_Analysis_of_Changes_in_Time_Using_Exploratory_Sentiment_Analysis_Twitter_Text_Data_from_Covid-19_Analysis_of_Changes_in_Time_Using_Exploratory_Sentiment_Analysis

Doğan, Ö. (2020, 25 Dec). *PyTorch Kütüphanesi Nedir? Nasıl Kullanılır?.* Teknoloji.org. Retrieved from https://teknoloji.org/pytorch-kutuphanesi-nedir-nasil-kullanilir/

Yamashita, R. et al. (2018). *Convolutional neural networks: an overview and application in radiology*. Retrieved from https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9

Rysbek, D. (2019). *Sentiment Analysis with Recurrent Neural Networks on Turkish Reviews Domain.* Middle East Technical University, Ankara. Retrieved from https://etd.lib.metu.edu.tr/upload/12623186/index.pdf

Mayers, G. (2020, 7 May). *Sentiment Analysis from Tweets using Recurrent Neural Networks.* Retrieved from https://medium.com/@gabriel.mayers/sentiment-analysis-from-tweets-using-recurrent-neural-networks-ebf6c202b9d5

Saha, S. (2018, 15 Dec). *A Comprehensive Guide to Convolutional Neural Networks* — the ELI5 way. Towards Data Science. Retrieved from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

PyTorch official documentation. (n.d.). PyTorch website. Retrieved from https://pytorch.org/

Keras official API Documentation. (n.d.) Keras website. Retrieved from https://keras.io/guides/functional_api/

TensorFlow official documentation. (n.d.). TensorFlow website. Retrieved from https://www.tensorflow.org/

Grattarola, D., *Twitter sentiment classification,* (2018), GitHub repository, https://github.com/danielegrattarola/twitter-sentiment-cnn

Joshi, K., *CNN for Twitter Sentiment Analysis*, (2017), GitHub repository, https://github.com/thekrutarth/CNN-for-Twitter-Sentiment-Analysis

Fatouros, G., *Twitter Sentiment Analysis with RNN,* (2019), GitHub repository, https://github.com/giorgosfatouros/Twitter-Sentiment-Analysis-with-RNN/

Rachman, A., *Twitter Sentiment Analysis Using Recurrent Neural Network,* (2019), GitHub repository,https://github.com/andikarachman/RNN-Twitter-Sentiment-Analysis