

Twitter Sentiment Analysis with Recurrent Neural Networks

Mahtab Tamannaee, Kandasamy Illanko
 {Mahtab, killanko}@ryerson.ca
 Department of Electrical and Computer Engineering
 Ryerson University, Toronto, Canada.

Abstract—One of the most popular topic in the field of deep learning and natural language processing is Sentiment analysis. Sentiment classification in natural language texts is a classification task of finding the polarity of the sentiments before them. One of the best sources of Natural Language texts and the most challenging one is the twitter database. Similar to the other NLP tasks, the deep learning methods have outperformed the other well-known NLP tasks. In this project, we address the document-level sentiment classification task and explore the application of Recurrent Neural Networks on this task from both deep learning and natural language processing point of view. Finally we provide a model by a special type of RNN, Long Short Term Memory network RNN architecture to perform predictions and analysis. This conducted evaluations show that our model is efficient and making it practical classification. Our model reached 77 percent accuracy on 1,600,000 tweets data-set.

Index Terms—Sentiment Analysis, Recurrent Neural Networks, Artificial Intelligence

I. INTRODUCTION

This ability to identify the positive or negative sentiment behind a text is interesting when it comes to social media like twitter. A predictive model could predict sentiment polarities of live tweets which would identifies the user's attitude towards an event. Performing classification with social media data can be challenging as most of the natural language texts in social media include hyperlinks, mentions, or other non-text contents. The contents of each will help the user to include more concept into a tweet or review. However, such content are not structured enough for the machine to learn. The employment of pre-processing ease up the procedure of converting these contents into structured data [1]. Similar to any Machine Learning model, its important to choose and use proper features to train models. Some studies have proved that working on the feature selection stage can potentially improve the classification accuracy. Word embedding and language models are typically applied for feature representation in deep Learning approaches. [1]

A Neural Network Model inputs the words into the input nodes (like word embedding) and the output values of nodes predicts the classification results. Basically, there are two original types of NN architectures that try to capture semantic and syntactic information as follows:

- Convolutional Neural Networks (CNN) is a Feed-Forward Neural Network which has the advantage of being able to handle high computational cost of NN. However, it needs

more training time in comparison to backward Neural Network.

- Recurrent Neural Networks (RNN) is a Backward Neural Network. In this network the output of one state is dependant of its previous state. This model requires large space to train properly. This dependency makes the RNN more superior in learning sequential information. Another advantage of RNN is that reduces the number of parameters as it keeps the same parameters at each step.

As we concluded, unlike feed forward neural networks, RNN can use its internal “memory” to learn a sequence of inputs, which makes it popular for processing sequential data. [9]. Recurrent models match people's intuition of reading word by word and are capable to model the intrinsic relations between sentences. Without any explicit boundary, RNNs could extract the sentence representation and analyze the semantic meaning of it at the same time keeping the word sequence. [3] The advantages of RNN over CNN encouraged us to choose this it for modeling our sentiment classifier. In the next section, we provide a literature review about most important paper works in this domain. The problem statement and our proposed models will be discussed in section IV. In section ?? we will demonstrate the experiments performed and will analyse the evaluation results.

II. RELATED WORK

RNN models have been widely used in Sentiment Classification. There have been several attempts to solve classification problem using machine learning. In this section, we briefly review and discuss the major researches conducted that are relevant to this work. The LSTM models have been shown the best performance between other CNN based methods [1]. Some of the best designs and state of arts are given here and they are all based on LSTM models.

Authors in [3], proposed a model named as Cached LSTM which improves the LSTM ability to carry information for long distances. they presented Cached Long Short-Term Memory neural networks (CLSTM) to capture the overall semantic information in long texts. CLSTM provides a cache mechanism, which divides memory into several groups with different forgetting rates and thus enables the network to keep the document's sentiment information better within a recurrent unit.

sentiment	id	date	query	user_id	text
0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zi ~ Awww, I...
1	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all...

Fig. 1. Twitter Sentiment140 data-set schema.

The authors in [4], proposed a model named as NCSL which includes two target dependent long short-term memory (LSTM) models. Neural Context-Sensitive Lexicon utilizes sentiment lexicons to treat the sentiment score of a sentence as a weighted sum of prior sentiment scores of negation words and sentiment words.

In this article [5], the authors proposed an attention-based model named as TD-LSTM for aspect-level sentiment classification task. They approach this task by an idea of learning aspect embeddings and let aspects participate in computing attention weights. Target-dependent LSTM treats an aspect as a target by using two LSTM surrounding the aspect term. Such model allows different aspects to be given as it concentrate on different parts of a sentence for different aspects.

Ruder S et.al in [6] proposed a hierarchical model named as TC-LSTM. This model performs aspect-based sentiment analysis by modeling the interdependencies of sentences in a review with a hierarchical bidirectional LSTM. Target-connection LSTM facilitates TD-LSTM to learn the relationship between aspect and each context word. They demonstrate this hypothesis for the task of aspect-based sentiment analysis by modeling the interdependencies of sentences in a review with a hierarchical bidirectional LSTM.

Yi Tay et. al in [7], proposed a model named as ATAE-LSTM. Attention-based LSTM with Aspect Embedding consists of aspect attention vector with LSTM hidden vector for the sentiment polarity. By looking at word-aspect associative relationships, they incorporate aspect information into the neural model. In order to model the similarity between aspect and words, they employed a circular convolution and circular correlation.

Yang Z et.al in [8], proposed a model named as Aspect-fusion LSTM. This AF-LSTM model is hierarchical attention networks (HAN) for classifying document. They used an additional attention mechanism and learns associative relationships between aspect and context words by word-aspect fusion attention. The document vector gets constructed in a hierarchical order first by aggregating important words into sentence vectors, and then aggregating important sentences vectors to document vectors.

III. DATASET

We choose sentiment140 (Fig 1), as the base dataset for model development and intuition. Most of the baselines methods are used a social media datasets, we also used twitter dataset for comparison purposes. This dataset consists of 1,600,000 tweets.

Different scores, from the range of zero to four, have been assigned to these tweets. Based on such score assignment,

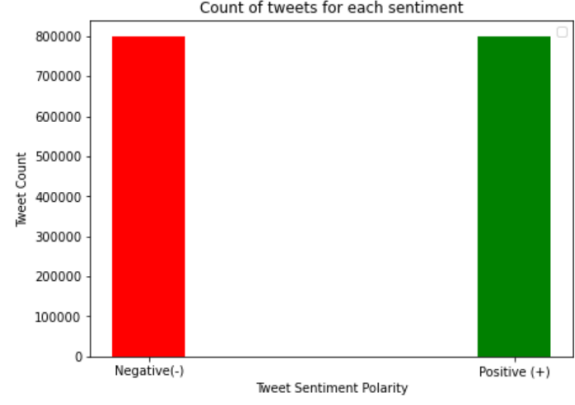


Fig. 2. Twitter Sentiment140 positive and negative polarities.

score = 0 indicates the most negative polarity score, and the score = 4 indicates the most positive polarity score. As you can see in Fig 2, by mapping the Negative polarity to tweets that their score is zero and positive polarity to the ones with score of 4, we can construct our data set with 800,000 positive tweets and 800,000 negative tweets.

We further divided the data set into a train set of size 1280000 and a test set of size 320000.

IV. SYSTEM MODEL AND THE PROBLEM STATEMENT

Sentiment Analysis has been studied at three levels, document level, sentence level, and aspect level [8]. In this project we perform Document level sentiment classification which classifies the sentiment behind a document(ex. tweet, query, or text) to a polarity, negative or positive. This type of sentiment classification can be considered under document classification category in NLP. This opinion classification is a conclusion on overall sentiment of the document. Such sentiment analysis can also understood as a regression task if the model needs to predict a rating score of a review or tweet.

A. Dense Document Vector Representation : Word2Vec

To perform sentiment analysis on document level, we need to express each document with a dense representation which can convey contextual information of each document term with respect to the text its extracted from. For such purpose, language modeling have shown promising word contextual modeling which provide word embedding representations. Employing the word embedding representations constructs a dense document vector which can be used as perfect feature of a document.

For such purpose, a local word embedding model (Word2Vec) was trained and used to construct the embedding layer representing the feature of each text or tweet. The Word2Vec word embedding model has been built using the well known GENSIM Python library.

B. Long Short-Term Memory RNN : LSTM

We propose a architecture for twitter sentiment classification at document-level by utilizing a special type of recurrent

neural networks (RNN), Long Short Term Memory network. The main idea is to use such networks to learn sequential pattern and context of tweets and predict a numerical value which further can be mapped into polarity classification.. This network can handle the processing of long-term dependencies. This advantage comes from the repeating module of LSTM structure which employs four layers of neural network. The LSTM model provides two cell states and decides the output based on cell states. This model uses sigmoid layer. [8].

In Long short-term memory network (LSTM), Considering σ as is the logistic sigmoid function, and \odot is the element-wise multiplication operation, the update of each LSTM Model component can be formalized as:

$$\begin{aligned} \mathbf{i}^{(t)} &= \sigma(\mathbf{W}_i \mathbf{x}^{(t)} + \mathbf{U}_i \mathbf{h}^{(t-1)}) \\ \mathbf{f}^{(t)} &= \sigma(\mathbf{W}_f \mathbf{x}^{(t)} + \mathbf{U}_f \mathbf{h}^{(t-1)}) \\ \mathbf{o}^{(t)} &= \sigma(\mathbf{W}_o \mathbf{x}^{(t)} + \mathbf{U}_o \mathbf{h}^{(t-1)}) \\ \tilde{\mathbf{c}}^{(t)} &= \tanh(\mathbf{W}_c \mathbf{x}^{(t)} + \mathbf{U}_c \mathbf{h}^{(t-1)}) \\ \mathbf{c}^{(t)} &= \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)} \\ \mathbf{h}^{(t)} &= \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \end{aligned}$$

Considering the gates, $\mathbf{i}^{(t)}$, $\mathbf{f}^{(t)}$, $\mathbf{o}^{(t)}$ and $\mathbf{c}^{(t)}$ as the input gate, forget gate, output gate, and memory cell activation vector at time-step t respectively. letting H as dimensionality of hidden layer and d as dimensionality of input, $\mathbf{h}^{(t)}$ corresponds to hidden vector. $\mathbf{h}^{(t)} \in \mathbb{R}^H$. $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o \in \mathbb{R}^{H \times d}$ and $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_o \in \mathbb{R}^{H \times H}$ are train-able parameters.

V. EXPERIMENTS

the RNN should have the following architecture which includes EmbeddingLayer, LSTM, and Dense layer. Since we are going to build a sequence model. We should feed in a sequence of numbers to it. These sequence of numbers are the sequence of word embedding of the terms in each tweet.

This NN classifier outputs a scalar value prediction. This model predicts an score between 0 and 1. Specifying a THRESHOLD will provide classification of the sentiment being positive if the predicted value is above THRESHOLD. In this project, the THRESHOLD has been set to "0.5", therefore, a tweet gets classified as POSITIVE if its predicted value is above "0.5".

Two callbacks gets called at the end of each epoch: First, ReduceLROnPlateau, which updates a Learning Rate at specific epoch. Second, EarlyStopping, which allows a performance measure to be specified and monitored. And steppes the training if it gets triggered.

A. Accuracy and Loss Results and Analysis

accuracyloss In the first step of evaluation, Learning Curve of loss and accuracy of the model on each epoch has been graphed. According to the Fig 4 we can see the accuracy and loss of the model on each epoch. Its crystal clear that the accuracy of the validation set has reached has reach 77 percent. Similarly the value of the loss has declined to 0.4601. Similar trend on both training and validation dataset shows promising facts that the model is not over-fitted or remained untrained.

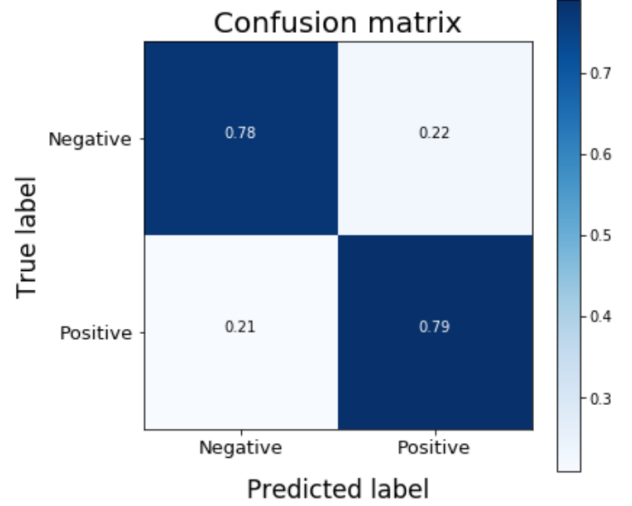


Fig. 3. Confusion Matrix of model classification.

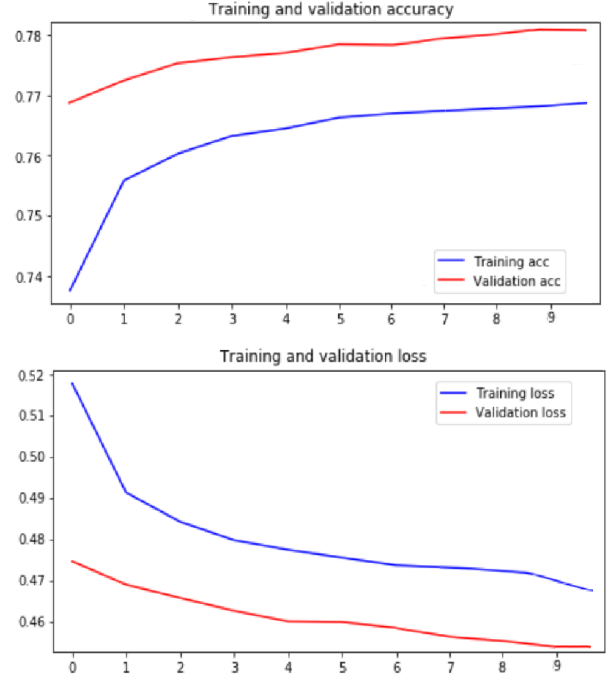


Fig. 4. Twitter Sentiment140 positive and negative polarities.

B. Classification Results and Analysis

The classification accuracy, Precision, Recall, and f1-score has been reported in Table 1. The values of the table is compatible with the Confusion Matrix (Fig Fig 3) which shows the true positive, false positive, true negative, and false negative classifications.

VI. CONCLUSION

In this paper, we applied a LSTM RNN on twitter data to observe the accuracy of this architecture to perform document-level sentiment classification. Building a language model

TABLE I
CLASSIFICATION RESULT SUMMERY

-	precision	recall	f1-score	support
Negative	0.78	0.76	0.77	160539
Positive	0.77	0.77	0.77	159461
Accuracy	-	-	0.77	320000
macro avg	0.77	0.77	0.77	320000
weighted avg	0.77	0.77	0.77	320000

based on local data set data has been shown effective as it provided dense document representation of documents. Training and testing the LSTM model on dense document embedding has reached 77 percent accuracy on 1,600,000 tweeter database.

REFERENCES

- [1] A. Mabrouk, R. P. D. Redondo and M. Kaye, "Deep Learning-Based Sentiment Classification: A Comparative Survey," in *IEEE Access*, vol. 8, pp. 85616-85638, 2020, doi: 10.1109/ACCESS.2020.2992013.
- [2] Z.-H. Deng, K.-H. Luo and H.-L. Yu, "A study of supervised term weighting scheme for sentiment analysis", *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3506-3513, Jun. 2014.
- [3] Xu J, Chen D, Qiu X, Huang X. Cached long short-term memory neural networks for document-level sentiment classification. *arXiv preprint arXiv:1610.04989*. 2016 Oct 17
- [4] Tang D, Qin B, Feng X, Liu T. Effective LSTMs for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*. 2015 Dec 3.
- [5] Wang Y, Huang M, Zhu X, Zhao L. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* 2016 Nov (pp. 606-615).
- [6] Ruder S, Ghaffari P, Breslin JG. A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*. 2016 Sep 9.
- [7] Tay Y, Tuan LA, Hui SC. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *Thirty-second AAAI conference on artificial intelligence* 2018 Apr 26.
- [8] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* 2016 Jun (pp. 1480-1489).
- [9] Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2018 Jul;8(4):e1253.