Sentiment Analysis with CharSCNN

Min Xiao
Analytics 2019
Georgetown university
mx61@georgetown.edu

Abstract

This paper reproduced and modified the original CharSCNN model to perform sentiment analysis on Stanford Twitter Sentiment corpus and FinancialBankPhrase Dataset. The model extracts from the character level to the sentence level representation and applies the convolutional neural networks to detect the sentence polarity. When embedding the characters, we used LSTM instead of the CNN in the original model. The model performs well on both datasets. The accuracy reached 70% on the STS data and 81.32% on the financial news dataset.

1 Introduction

1.1 Motivation of Sentimental Analysis

The fast-growing social media and computational capacity make it possible to extract information from text such as news, blogs, social media posts, reviews, and various documents. There is a huge amount of valuable information in the text data. one of the exampleS is to identify the opinions or analyze the sentiment in news articles, reviews, online posts, and etc. To conduct the analysis from the hidden information, researchers have worked on various methods.

Sentiment analysis, more precisely, is a computational detection of opinions, sentiments, and attitudes towards an entity. It can be used in a wide range of fields including e-commerce, political movements, company strategies, marketing campaigns, algorithm trading, and etc. However, sentiment analysis or opinion mining is very challenging since text information is unstructured and noisy and it usually has complicated language structure.

1.2 literature review

Sentiment analysis is an interrelated area of research which uses various techniques including Natural Language Processing(NLP), Information Retrieve, Structuring, and Unstructured Data Mining. Sentiment analysis is not a single problem and it includes multiple sub-tasks and methods. Below is a summary of various common tasks and methods:

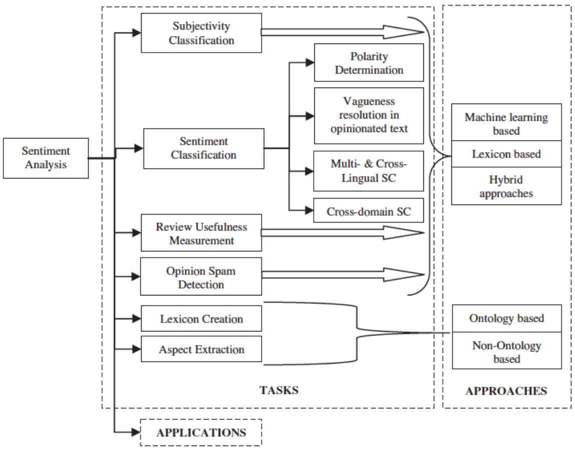


Fig1: Sentiment Analysis Overview

Sentimental classification is the determination of the sentiment of the given text in two or more classes. Polarity detection is to determine the polarity of a sentence, i.e positive, negative or neutral. Polarity detection can be performed using machine learning based methods(Pang and Lee., 2002), (Saleh et al., 2011), lexicon based methods (Tang et al., 2009) and hybrid-based methods (Chen et al., 2011).

In this paper, we will focus on the polarity under the subtask sentimental classification. (GO et al.,

2009) applied machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) on twitter messages. (Moraes et al., 2013) presented an empirical comparison between SVM and ANN regarding document-level sentiment analysis. Deep learning is also widely used in sentiment analysis, (Santos et al., 2014) use convolutional neural networks in sentiment classification and embedded from character level to sentence level representation.

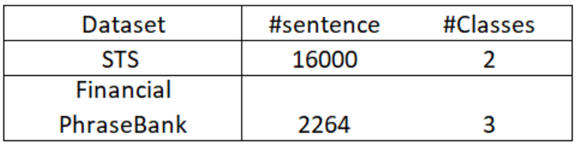
1.3 Paper Structure

In this paper, we will walk through and reproduce the CharSCNN model in (Santos et al., 2014) to compute a score for each sentiment label. The network extract features from character-level up to sentence level so that it can explore the richness in the text information. Some of the model details may differ from the original article.

2 Dataset

In this paper, we used two datasets. One is the common sentimental analysis corpus STS and another one is a finance news dataset.

Stanford Twitter Sentiment corpus(STS) contains 1.6 million tweets and labelled as positive/negative. In our experiment, we sample 160k tweets as the dataset we use.
Finance bank phrase includes 2264 news titles with sentiment labels as positive, neutral and negative



Dataset	#sentence	#Classes
STS	16000	2
Financial PhraseBank	2264	3

table 1: datasets

3 Model Architecture

In the model, we have 3 level of embeddings. For a given sentence, we will combine the word embeddings and character embeddings and then extract the sentence representation using Convolutional neural networks.
After completing the text representation, we will use 2 dense layers to classify the sentence to positive or negative.

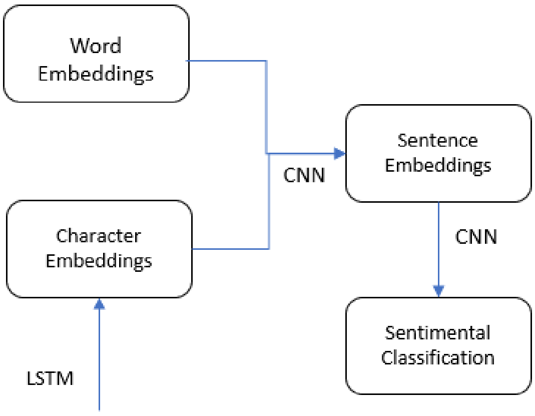
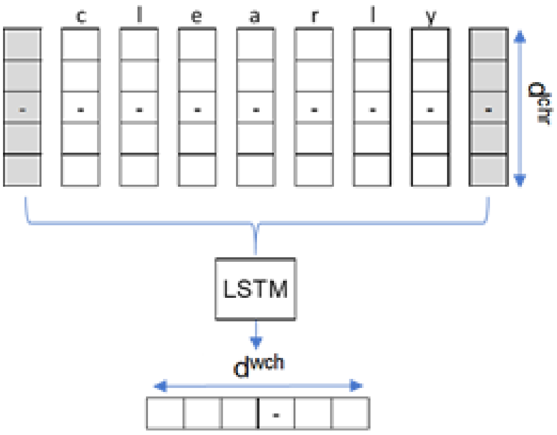


Fig2: Model Structure

- Word Emeddings
  Word Embeddings use an encoded matrix $w^{wrd} \in \mathcal{R}^{d^{wrd} * V^{wrd}}$ to represent single words in the vocabulary. V^{wrd} is the vocabulary size and each column r^{wrd}in the matrix represents a single word. The most straightforward way is to use a one-hot vector. Modern NLP researchers have proposed some effective algorithms. For example, word2Vec (Mikolov et al., 2014) from Google use a shallow neural network to get the vector representation. We can choose to train the embeddings or use the pre-trained.

- Character-level Embeddings
  When extracting the information from text, we need take into consideration of the morphological and shape information. Thus, we need embed the character level information into the model as it can capture more local features.
  Similarly, character embeddings use a numerical matrix W^{chr} to represent characters. Each of the columns $r^{chr} \in W^{chr}$ represent a single character.
  For a given word w which is composed of M characters $\{c_1, c_2, ..., c_M\}$, there are local features regarding the characters instead of the universal character embeddings.The second step is to produce local features around each character. Here we apply a Long Short Term Memory Netork(LSTM) on the character embedding vectors$\{r_1^{chr}, r_2^{chr}, ..., r_M^{chr}\}$ for a given word to produce a local character feature r^{wch}.



figureFigure caption

- Sentence-level Embeddings
  Last step is to conduct the sentence level embeddings. The common challenges in sentence embeddings are that sentence have different lengths and important information can appear at any part of the sentence. To tackle these challenges, we will apply the convolutional layers to extract the sentence level representation. First, we combine word embeddings and character-level embeddings, we have the joint embeddings vectors $\{u_1, u_2, ..., u_N\}$ for a given sentence x with N words. Here $u_n \in \mathcal{R}^{(d^{wrd}+d^{wch})}$ and sentence $x \in \mathcal{R}^{(d^{wrd}+d^{wch})N}$. Let's define Z_n as the concatenation the concatenaiton of a sequence of k embeddings of u_n. Then we apply a convolutional layer to produce the sentence representation, the filter $W^1 \in \mathcal{R}^{cl(d^{wrd}+d^{wch})k^{wrd}}$.

$$r[sent]_j = max_{1<n<N}[W^1 Z_n + b^1]$$

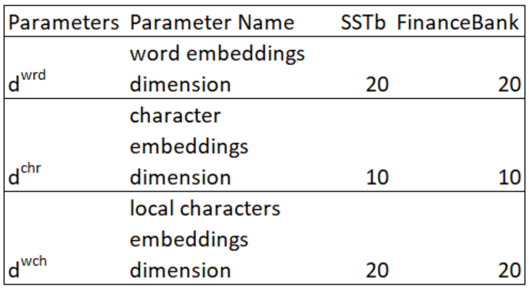
- Dense layers

  After obtaining the sentence representation, we apply another 2 neural network layers to compute a score for each sentiment label:

$$s(x) = W^3 h(W^2 r_x^{sent} + b^2) + b^3$$

4 Experiment and Results

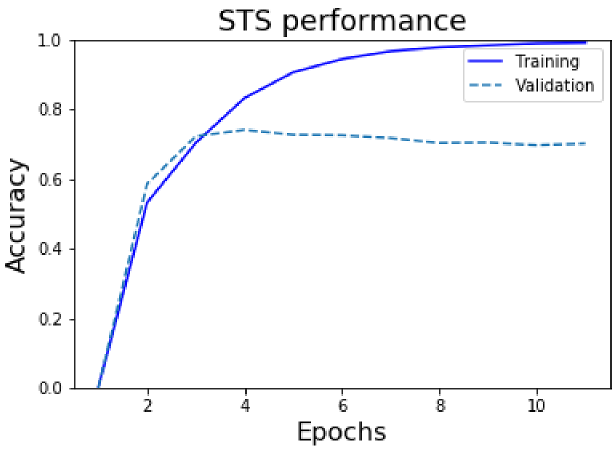
- Model setting-up:
  The model has a set of hyperparameters. Here is the table of all hyperparameters.

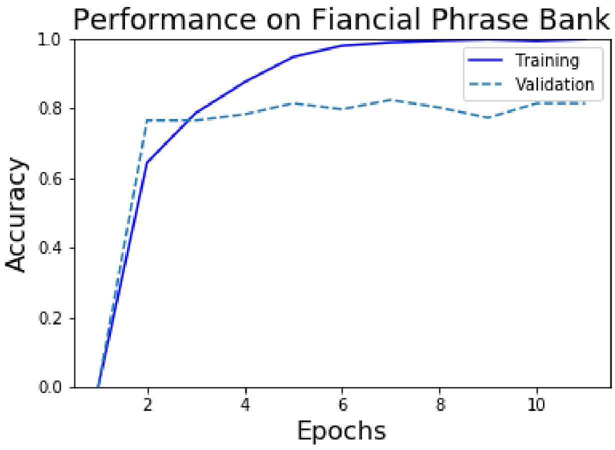


Parameters	Parameter Name	SSTb	FinanceBank
d^wrd	word embeddings dimension	20	20
d^chr	character embeddings dimension	10	10
d^wch	local characters embeddings dimension	20	20

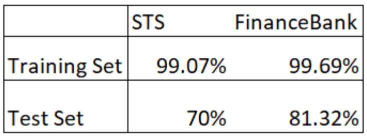
Table1: model hyper-parameters

- Experiment result:
  First, we train the model and apply the model to the test data. Here are the trajectories of the training and test data.





The training and test accuracy of the two datasets is as below. Both the training datasets can reach 98% accuracy. The accuracy test datasets are around 73% and 70% respectively.



	STS	FinanceBank
Training Set	99.07%	99.69%
Test Set	70%	81.32%

- Discussion
  In the original paper, they used the CNN when implementing the character embeddings. For the STS corpus, the result reached 86.4% accuracy as they used the pre-training embeddings and 81.9% if not using pre-trained ones. In our mode, the STS data reach 70% accuracy, which is so powerful but still a fair classification. However, the model works well on the FinanceBank, with 81.32% as the accuracy.

5 Conclusion & Future Work

Conclusion:
In this project, we applied CharSCNN on the STS and FinanceBank Dataset to do the polarity detection. The model extracted from the character level to sentence level information in their representation. We applied RNN as well as the CNN to embed the information and implement the classification. Models work slightly better on the financial news data.

Future Work

- Combine the deep learning models with linguistics principals Currently, the model focuses more on using the deep learning and complicated models to extract the information. For the next steps, when setting up the model, we should take the linguists into consideration as sentiment analysis partly depends on various language rules. More inputs from linguistics can improve the model performance.

- We have a classification on the financial news data with a small dataset. To make use of the model, we can apply the model on model news such as Reuters news to predict the sentiments of the stock market. Then compare the sentiment analysis result with the market performance as we expect the news will reflect and influence the market sentiment.

References

Alec Go, Richa Bhayani, and Lei Huang. 2009. *Twitter Sentiment Classification using distant supervision*. Techincal report, Stanford University,.

Rodrigo Moraes,Joo Francisco Valiati,Wilson P.Gavio Neto 2013. *Document-levels sentiment classification: An empirical comparison between SVM and ANN*. Expert Systems with Applications. 40(2013):621–633.

Cícero Nogueira dos Santos, Maíra A. de C. Gatti 2014. *Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts*. COLING.

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean 2014. *Efficient Estimation of Word Representations in Vector Space*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In ICLR,2002.

M.Rushdi Saleh, M.T.Martn-Valdivia, A.Montejo-Rez and L.A.Urea-Lpez 2011. *Experiments with SVM to classify opinions in different domains*. Expert Systems with Applications. 36(2011):14799–14804.

Huifeng Tang, Songbo Tan and Xueqi Cheng 2009. *A survey on sentiment detection of reviews*. Expert Systems with Applications. 36(2009):10760–10773.

Long-Sheng Chen, Cheng-Hsiang Liu, Hui-Ju Chi 2011. *A neural network based approach for sentiment classification in the blogosphere* . Journal of Informetrics. 5(2011):313–323.