Comparison of Machine Learning Algorithm and Deep Learning Algorithm in Sentiment Analysis Task

Yuwei Zhang, Jeremy Zeng, Dean Huang, Zihao Lin

Nov. 15th. 2020

Abstract

The goal of our study is to provide a comprehensive analysis and evaluation of several state-of-the-art machine-learning methods as well as deep learning algorithms for the automated classification of sentiment analysis. The sentiment analysis algorithms we are evaluating and comparing are logistic regression, k-nearest neighbors algorithm, random forest, fasText, Long short-term memory (LSTM), Attention-Based BiLSTM, and CNN. To compare the performance of these methods, we will use the Yelp dataset provided by Kaggle [1] to perform sentiment analysis on the user reviews.

keywords: Sentiment analysis, Machine learning methods, Deep learning methods

Part 1 Introduction

Sentiment analysis is a text analytics field that analyzes and extracts people's opinion towards different topics through detecting the contextual polarization of the text. With the increase in popularity of social media and microblogging, sentiment analysis has become a crucial tool for understanding the aggregated public opinion on different topics. Through grasping the sentiments of the general public, companies and organizations are able to make effective decisions or changes that gravitate towards the public interest.

The rest of this paper is organized as follows: Background section will give an overview of the content and related work, Approach/Methodology section will describe the technical details of the methods used in the study, Results section will include a written summary of the findings, including how we tested and validated our approach, and Conclusions section will summarize the work, describing the problems we solved and approach to solving it.

Part 2 Background

Several related works have been performed earlier to compare various classification algorithms used in sentiment analysis. The authors of [2] have conducted comparative study of Naïve Bayes, Max Entropy, Boosted Tree, and Random Forest for sentiment analysis. The study result showed Random Forest Classification was most suited for sentiment analysis due to its high accuracy and performance, simplicity in understanding, and improvement in results over a period of time. However, it required high training time and processing power due to aggregation of decision trees. In addition, the authors of [3] have conducted comparative study of Naïve Bayes, Support Vector Machine, K-Nearest Neighbors, KStar, and Decision Tree for sentiment classification of movie reviews. The results of the study showed SVM as the most accurate method for correctly calculating the sentiments of movie reviews. The results of both studies showed every kind of classification model having its benefits and drawbacks. Therefore, it is crucial to look at factors like nature of the problem, basis of resources, accuracy requirement, and training time available when selecting the appropriate classification models to perform sentiment analysis.

Similarly, multiple studies on sentiment analysis have been conducted using various deep learning methods. RNN is the most common deep learning algorithm used to deal with sequential data. Hochreiter and Schmidhuber introduced the Long Short-Term Memory (LSTM)

model in 1997 [4], which is very monumental to the NLP field. In the same year, M. Schuster and K. Paliwal created a bidirectional RNN model [5]. These two methods played an important role in sentiment analysis. In 2019, G Xu tried using the BiLSTM method to conduct sentiment analysis on comment texts, and achieved an accuracy of 88.64% [6]. Compared to traditional RNN methods such as LSTM, BilSTM will lose some information when dealing with lengthy texts. Hence, Kyunghyun Cho published another article in 2015 in which he introduced the attention mechanism aiming to solve this problem [7]. In 2016, Y Wang used the attention-based LSTM for aspect-level sentiment classification with the accuracy of 89.9% [8]. In addition to RNN based models, CNN is another type of deep learning algorithm. Although CNN is often applied to computer vision, it can also be used in NLP tasks. In 2019, G Xu used CNN to analyze the sentiment of comment texts and the accuracy was 85.1% [6].

The goal of this study is to perform a similar comparative study on sentiment analysis using various machine learning algorithms. However, different from the past comparative studies, we will incorporate neural network methods into our comparative study. The neural network methods we incorporated are fasText, LSTM, Attention-Based BiLSTM, and CNN.

Part 3 Approach / Methodology

I. Text Pre-Processing

There were altogether 5.2 million user reviews in the original dataset, and we randomly sampled 100,000 reviews that were in English to conduct sentiment analysis. Next, we re-categorized the ratings column into three levels: Level 0 for 1 or 2 stars (negative), Level 1 for 3 stars (neutral), and Level 2 for 4 or 5 stars (positive). Before applying any sentiment analysis method, we executed data pre-processing to reduce computational complexity and generate higher quality of text classification. The goal of data pre-processing is to reconstruct raw data into a cleaner and more efficient format. The typical pre-processing procedures are made up of four steps: part-of-speech tagging (POS), stemming and lemmatization, stop-words removal, and negations handling [9]. We chose to only perform stop words removal and stemming on yelp reviews due to the simplicity of its linguistic structures. To remove stop words, we utilized the stopwords package from NLTK to eliminate all stopwords in English. We also utilized the PorterStemmer package from NLTK to reduce inflected words to their word stem. In addition, we removed all punctuations and numerical digits, and converted all alphabets to lower-case.

II. Word Embedding

Three types of word embeddings were employed in our study: TF-IDF, Word2vec, and Doc2Vec. TF-IDF is applied to evaluate how relevant each word is to a review in the collection of reviews. This is done by multiplying the word frequency in a review to the inverse review frequency of the word across the collection of reviews. We utilized the TfidfVectorizer package from sklearn to convert the reviews to a document-term matrix, with each entry equating to the corresponding TF-IDF values.

Word2Vec was applied to generate a word vector representation with a fixed dimension. We chose the CBOW algorithm with output vector dimension equal to 200 and window size equal to 5. There was no limit on the maximum word frequency, and to take every word into account, the minimum word sequence was set to be 1. For each review, we used the mean of word vectors to represent the feature vector of the review.

Doc2vec was applied to create a numeric representation of a document, regardless of its length. The concept was developed by Mikilov and Le using the word2vec model and adding another vector such as Paragraph ID [10]. There are two types of doc2vec models. One is called Distributed Memory version of Paragraph Vector (PV-DM), and another is called Distributed Bag of Words version of Paragraph Vector (PV-DBOW). The concept of PV-DM model is similar to the CBOW model of word2vec, and the concept of the PV-DBOW model is similar to the Skip-gram model of word2vec. However, the execution time of these algorithms are faster. In contrast to word2vec, doc2vec consumes less memory because the saving of the word

vectors is not needed. PV-DBOW was used as an embedding method for k-nearest neighbors in our study.

III. Machine Learning Method

(a) **KNN**

Since the only assumption of K-Nearest Neighbors is that similar things exist in close proximity, the algorithm is simple and easy to implement. This supervised machine learning algorithm is based on the distance metric such as Euclidean distance or overlap metric between a test sample and the specified training sample. For text classification, KNN will output class labels as the result. The most important parameter in KNN is the value of k, which is the number of nearest neighbors to be used by the classifier. In general, a larger value of k helps to reduce the effect of noise on the classification. However, the value of k cannot be too large because it will result in a reduction of the accuracy of the classification. After calculating the distance and adding it along with the index to an ordered collection, the algorithm sorts the ordered collection of distance and indices from smallest to largest by distance and picks the first K entries from the sorted collection. For classification tasks, KNN will return the mode of the k labels. In contrast, for regression tasks, KNN will return the mean of the K labels [11]. One of the biggest disadvantages of KNN is that accuracy depends on the quality of the data. In addition, KNN requires high memory space to store all training data.

(b) **Random Forest**

Random forest is an ensemble learning method formally proposed in 2001 by Leo Breiman and Adèle Cutler. Just like its name implies, random forest is made up of a large number of decision trees that operate as an ensemble [12]. Each decision tree in the random forest will provide a class prediction and the class with the most votes will represent the model's final prediction. During each split, instead of considering all features, only a random subset of features will be considered, usually equal to the square root of the total features. Random forest is one of the best classification methods because of its capability to classify large amounts of data with accuracy. By picking the most popular outcome out of all individual decision trees, random forest can provide accurate prediction with low variance. In addition, unlike many machine learning methods, random forest does not require much tuning.

IV. Nerual Network

(a) **Fasttext**

FastText is a library created by Facebook's AI Research (FAIR) lab that contains pre-trained word embeddings and text classification, which are based on the same principle as Word2vec, CBOW, and skip-gram algorithm[13]. The input of fastText is cleaned text without embeddings, and its neural network contains a simple hidden layer and hierarchical softmax classification layer. Compared to self-supervised training in word2vec, supervised learning in fastText is faster in training speed and prediction.

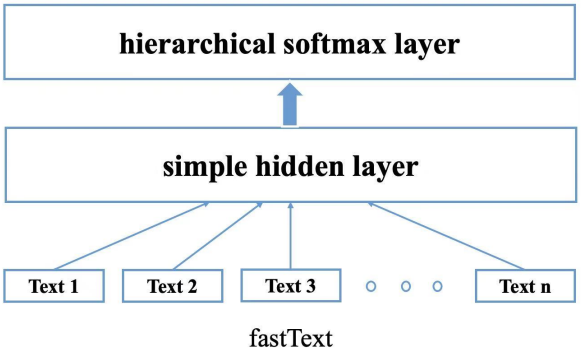

Figure 1: Fast text neural layer diagram

(b) **RNN: BiLSTM**

Recurrent neural network contains a cycle within its network connections, which enables the output of a unit to be directly or indirectly dependent on the previous outputs as an input. This works well in natural language processing that has sequence characteristics. Long short-term memory (LSTM) is an application of RNN. LSTM networks divide the context management problem into two sub-problems: removing no longer needed information from the context, and adding information likely to be needed for later decision making [14]. Traditional RNN always loses the information that is too far away from current words; however, LSTM could solve this problem by adding a forget gate, an input gate, and an output gate which determines the information to be kept and the information to be dropped. Bidirectional LSTM (BiLSTM) is an improved version of LSTM, which is more powerful. BiLSTM has two LSTM layers, one of which is forward and one of which is backward. The final BiLSTM model includes three kinds of layers: embedding layer, BiLSTM layer, and fully connected layer. The embedded dimension is 300. In the LSTM layer, there are two layers inside it, with the dropout probability equal to 0.5. The fully connected layer has an output with three classes (positive, negative, and neutral).

(c) **RNN: Attention-Based BiLSTM**

The attention mechanism is applied to take the entire encoder context into account [14]. In the attention-based BiLSTM classification model, an attention layer is added after the BiLSTM layer. BiLSTM only uses the last time state as the input to do the softmax classification. However, the attention layer will first calculate the weight of each time state, and then uses the weighted sum of each state as the input of the softmax classifier.

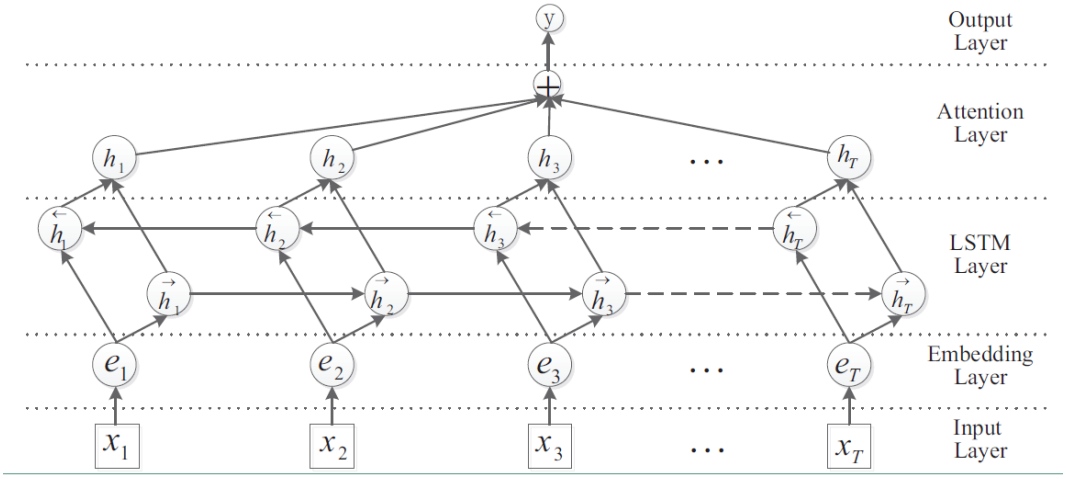

Figure 2: Attention-based Bi-LSTM layer diagram

(d) **CNN**

Convolutional neural networks, also known as CNN, is another deep neural network that could be used for sentiment analysis. CNN contains an input and an output layer plus multiple convolutional and pooling hidden layers. The ReLU layer is used as an activation layer followed by fully connected layers and normalization layers [15]. Although CNN is most commonly used in visualization fields, CNN can also be applied to sentiment analysis. Our CNN model contains four convolution layers.

Part 4 Result

For each classification method, we splitted the dataset into training dataset and testing dataset. Models were trained using the training dataset, and the trained model was used to predict the class labels of the test dataset. The metrics that were used for performance measures were accuracy and recall. F1-score, the harmonic mean of precision and recall, was also applied to address the potential impact of data imbalance [16]. These metrics were not only used to

evaluate, compare the performance of our machine learning models and deep learning models but also used to identify potential problems like underfitting and overfitting.

Table 1: Results

model	Accuracy		Recall		F1	
	train	test	train	test	train	test
Random Forest	100%	80.50%	100%	80.50%	100%	80.50%
Logistic Regression	86.80%	84.90%	86.80%	84.90%	86.8%	84.90%
K-nearest Neighbors	78.10%	72.20%	78.10%	72.20%	78.10%	72.20%
fastText	96.11%	85.90%	96.12%	85.90%	96.12%	85.90%
Bi-LSTM	84.75%	83.61%	84.75%	83.61%	84.75%	83.61%
Attention-Based BiLSTM	85.12%	83.04%	85.12%	83.04%	85.12%	83.04%
CNN	84.84%	83.95%	84.84%	83.95%	84.84%	83.95%

In addition to the above metrics, we evaluated deep learning methods on convergence of loss and accuracy of training set in iteration. The following three plots show the loss and accuracy curves of BiLSTM, attention-based BiLSTM and CNN methods. According to the plots, the accuracy of BiLSTM method remains stable after 10k interactions, however, the loss continues to decrease. The accuracy of attention-based BiLSTM decreases a little and the loss increases after 10k interactions, which might be due to overfitting. Same situation happens after 2000 iterations in CNN method.

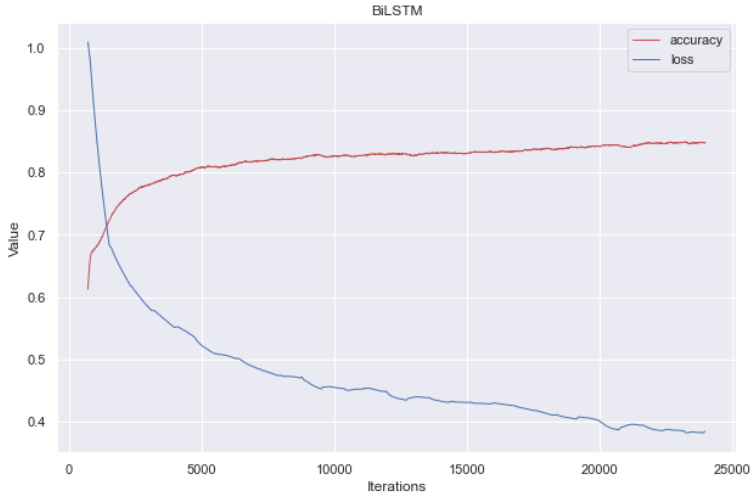

Figure 3: Training result of BiLSTM

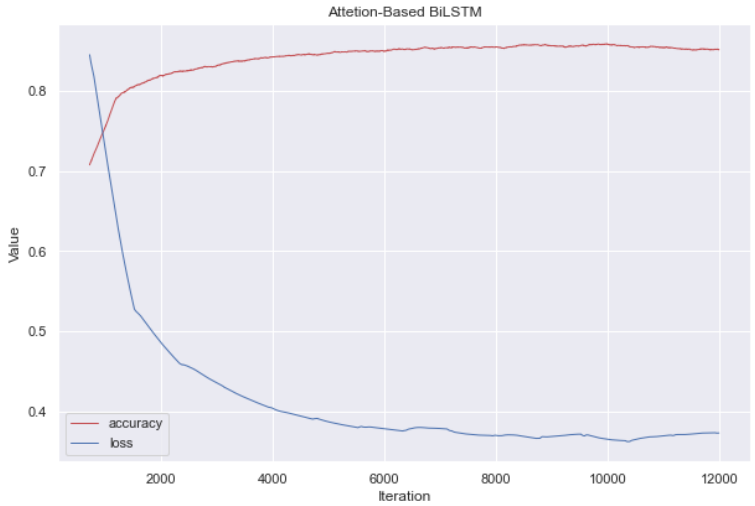

Figure 4: Training result of attention-based BiLSTM

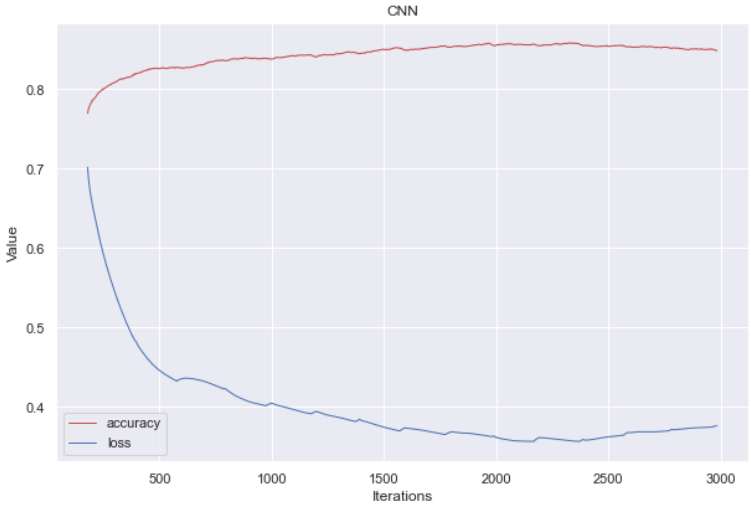

Figure 5: Training result of CNN

According to our results, models with the best performance are fastText and logistic regression, having the best results in all metrics on training and testing data.

Part 5 Conclusion

The table above shows the advantages and disadvantages of various models. For sentiment analysis, logistic regression and fastText are the best methods to implement, since both the risk-benefit trade-off and computational cost for these two methods are small. Random Forest and K-nearest Neighbors are not recommended for analyzing very large dataset due to high memory consumption when executing the models. Even though the model is more difficult to tune, LSTM is a good method for text generation or analyzing texts with sequence characteristics.

Table 2: Comparison of different methods

Model	Training Speed	Predict Speed	Accuracy	Difficulty of Parameter tuning
Random Forest	Medium	Medium	Medium	Easy
Logistic Regression	Fast	Fast	High	Easy
K-nearest Neighbors	N/A	Medium	Low	Easy
fastText	Fast	Fast	High	Easy
BiLSTM (CPU)	Slow	Slow	Medium	Difficult
Attention-Based BiLSTM (CPU)	Slow	Slow	Medium	Difficult
CNN (CPU)	Slow	Slow	Medium	Difficult

Speed: slow>= 10 minutes, 10 minutes >medium>=4 minutes, fast<4 minutes
Accuracy: high: >= 85%, medium: 80%-85%, low: <80%

Part 6 Summary of Team Roles

Each team member worked on understanding the data, building models, providing ideas for the presentation, and preparing slides. All team members contributed equally.

Dean Huang played a role like a coordinator and checker. He makes sure everyone gets involved, holds meetings for discussion. He did the work of preprocessing the data, and corresponding analysis work.

Yuwei Zhang played a role like a programmer. She makes sure everyone's code is "readable", and provides some suggestions when others meet some problems in terms of coding. She also did corresponding analysis work.

Zihao Lin plays a role also like a programmer and writer. He provided suggestions when others meet some problem, and used Latex to convert the final report. He worked on the deep learning model: Bi-LSTM, Attention-Based Bi-LSTM model and CNN model, which is the most difficult part of our project.

Jeremy Zeng played a role like a presenter and writer, he put team presentations and reports together, including the presentation slides and recorded videos. He also did corresponding analysis work.

References

[1] Yelp, Inc. Yelp Dataset. 26 Mar. 2020, www.kaggle.com/yelp-dataset/yelp-dataset.

[2] Gupte, Amit, et al. "[PDF] Comparative Study of Classification Algorithms Used in Sentiment Analysis: Semantic Scholar." [PDF] Comparative Study of Classification Algorithms Used in Sentiment Analysis — Semantic Scholar, 1 Jan. 1970, www.semanticscholar.org/paper/Comparative-Study-of-Classification-Algorithms-used-Gupte-Joshi/466788e0ba1f608981ca5422ddfb5bfedeef75d0.

[3] Elmurngi, Elshrif & Gherbi, Abdelouahed. (2018). Detecting Fake Reviews through Sentiment Analysis Using Machine Learning Techniques.

[4] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[5] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE transactions on Signal Processing 45.11 (1997): 2673-2681.

[6] Xu, Guixian, et al. "Sentiment analysis of comment texts based on BiLSTM." Ieee Access 7 (2019): 51522-51532.

[7] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International conference on machine learning. 2015.

[8] Wang, Yequan, et al. "Attention-based LSTM for aspect-level sentiment classification." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.

[9] Kolchyna, Olga, et al. "Twitter sentiment analysis: Lexicon method, machine learning method and their combination." arXiv preprint arXiv:1507.00955 (2015).

[10] Shperber, G. (2019, November 05). A gentle introduction to Doc2Vec. Retrieved November 18, 2020, from https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e

[11] Harrison, Onel. "Machine learning basics with the k-nearest neighbors algorithm." Towards Data Science. September 10 (2018).

[12] Amrani, Y., Lazaar, M., & Kadiri, K. (2018, March 12). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. Retrieved November 18, 2020, from https://www.sciencedirect.com/science/article/pii/S1877050918301625

[13] Python Module · FastText. fasttext.cc/docs/en/python-module.html.

[14] Jurafsky, Dan. Speech & language processing. Pearson Education India, 2000.

[15] Convolutional neural network. (2020, November 12). Retrieved November 18, 2020, from https://en.wikipedia.org/wiki/Convolutional_neural_network

[16] Sasaki, Yutaka. (2007). The truth of the F-measure. Teach Tutor Mater.