A Comparative Review Of Sentimental Analysis Using Different Algorithms

1st Shouxi Li
Electrical and Computer Engineering
Lakehead University
0883145

2nd Jiyanbo Cao
Computer Science
Lakehead University
1094314

3rd Vaibhav Bhat
Computer Science
Lakehead University
1107179

4th Vidhi Pitroda
Computer Science
Lakehead University
1099820

***Abstract*—Sentiment analysis is one of the applications of natural language processing, which aims at using technologies such as artificial neural networks to analyze emotions. By converting lexical resources into numerical representations, the networks are able to detect the mathematical meaning behind the numbers, after which machines will be able to make a decision based on the statistical distribution. In this paper, some of the popular algorithms such as convolutional neural networks and recurrent neural networks are reviewed and a brief comparison is conducted. However, it is unclear that if there exists a method that could perform better than others under all circumstances, since the presentation of human languages are significantly diverse due to its creative properties, which is also the reason why the industrial applications of such a technique has not widely spread yet, because of the probability that it could be misleading and confusing. Therefore, we try to measure certain accuracy metrics for all the algorithms and try to compare them.**

***Index Terms*—natural language processing, sentiment analysis, convolutional neural networks, recurrent neural networks**

I. Introduction

In recent years, Internet has proved to become a hub for people to freely express their feelings and opinions. It is quite often that we see comments or reviews for a product describing its advantages and faults for example websites such as Amazon and Flipkart or some post on Facebook which shows the users emotion. Due to this open access to freely express the opinions, there has been an exponential growth in the unstructured data over the internet. Previously, for obtaining this data, companies used to provide surveys, or hire a third-party organisation to get this information for them. These methods were not only slower but are costly as well. Nowadays all this data are easily available over the internet. This data can be converted into valuable information using sentiment analysis. Sentiment Analysis can be referred to as a computational treatment of opinion, sentiment and subjectivity and can be categorised as a shared field for both natural language processing (NLP) and machine learning (ML) which are both contained within the broadly defined field of artificial intelligence (AI) [1]. Sentiment Analysis is mainly used in context with text-based data. The analysis indicates the users expression of positive, negative or neutral sentiments towards products or events [2]. Product owners can get a hold of customer satisfaction whereas users can evaluate the products using sentiment analysis. As we know, most of the data available cannot be directly applied into algorithms. Hence various preprocessing techniques are used such as word2vec, tf-idf, etc. There are also various algorithms used in sentiment analysis which have their own advantages based on the type of data available.

In this paper, different machine learning algorithms which are frequently used for sentiment analysis along with their preprocessing techniques are explained and compared. The remainder of the paper consists of four sections. The next section (Section II) gives a brief overview of the algorithms and the papers referred for explaining these algorithms. We then move on to explaining each algorithm in detail in the next section (Section III). Section IV comprehensively compares these algorithms and the experimental results are discussed and analyzed. In the last section (Section V) we conclude our results evident from our analysis and provide certain fields of application.

II. Literature Review

Sentimental Analysis deals with identifying the underlying attitude that a particular information/data holds towards an entity. Collectively analysing these information over large chunks of data can represent certain strong opinions and hence can have a plethora of applications. In the paper, we try to provide a basic idea of how sentiment analysis has evolved over the years, across different data sources with data being available in large numbers these days and point out certain algorithms and their applications in the real world. This section tries to briefly introduce the gist of the paper making the reader familiar with certain methodologies and research papers.

The paper "A Literature Review on Sentiment Analysis and its Foundational Technologies [1]" explains the concepts of Sentiment Analysis right from its roots back in mid-tolerate 1990s. It discusses certain techniques used in papers [3] such as extracting syntactic structures [4], using scalable vector machines and k-means clustering to identify patterns. It also discusses certain researches which include the bag of words method as well as lexical rule-based method of recognising emotions from text. The paper mentions that in recent times, the whole process has three main aspects, processing, classification and validation. The three major techniques in document processing includes tokenisation (categorizing a block of text in in a sentence) , stopword filtering (Filtering out any word which will be of no use in classification process) and stemming

(Getting the root word of all the words by cleaning out the prefixes and suffixes). In the next aspect which is classification it mentions certain classification algorithms such as support vector machines, Naïve Bayes (The classic Multinomial Naïve Bayes and highly-efficient Naïve Bayes algorithm TWCNB) and k-NN. It also compares the multinomial naïve bayes, TWCNB and SVM classifier on a specific dataset and find out that TWCNB and SVM have a better standing compared to the classic MNB. Although the paper does mention certain contemporary technologies, it does miss out on the latest classification methods used in AI including the usage of neural networks which tend to have higher accuracies.

Diving deep into certain algorithms using neural networks, the paper "Sentiment Analysis of Comment Texts Based on BiLSTM [2]" uses bidirectional long short-term memory for classifying the comments text. It is important that any processing method should consider both the semantic as well as sentiment information of the word. The paper does this by introducing a new word representation method which is a combination of traditional TF-IDF and sentiment information to generate weighted word vectors. These weighted word vectors are then feeded as an input to the Bi-LSTM network. The algorithm is compared across the original neural network methods such as RNN, CNN, LSTM as well as the classic Naïve Bayes and is proved to be highly efficient in terms of precision, recall and F1 score. Furthermore, the paper also tries to mention an area of application to use Bi-LSTM for comment texts.

Whenever neural networks are mentioned, it is indeed necessary to mention deep convolutional neural networks as these have proven to be highly effective over the years. The main difference between any normal convolutional neural network and a deep neural network is that DNN's tend to have more layers as compared to those shallow CNN's. The paper "Twitter Sentiment Analysis with Deep Convolutional Neural Networks [5]" uses DCNN for sentiment analysis on one of the most abundantly used Twitter data. The paper introduces a new model for initializing the parameter weights of the Convolutional neural network. It also compares its results with some of the other methods and finds out that the results are highly is their favour. Additionally, the paper "Aspect-based sentiment analysis using deep networks and stochastic optimization [6]" is another method for unstructured text-based sentiment analysis done using a combination of creating ontologies, word2vec feature extraction method and convolutional neural networks. This method is compared with other techniques as well and it outperforms the state-of-the-art techniques. As mentioned before, due to the abundance availability of data these days, we can perform sentiment analysis not only on the text-based data but other facial and vocal expressions as well. The paper "A survey of multimodal sentiment analysis [7]" tends to cover these techniques. However, as these techniques wont be a part of natural language processing we would not be digging deep into these methods as it puts this research out of our scope.

III. Algorithms

A. Convolutional Neural Networks

One method for sentiment analyzing is to use deep convolutional neural networks (DCNN). By using DCNN, some deep features will be extracted through the network, and then the fully connected layers behind the DCNN will determine whether the corpus or sentence being detected is positive, neutral, or negative. In 2015, Aliaksei S. and Alessandro M. [5] published a paper that introduced a new model to initial the parameter weights of the convolutional neural networks, which claimed to be crucial to train an accurate model without injecting additional features and could obtain a higher accuracy among most of the candidates of the Twitter Sentiment Analysis Campaign. The model being used is built from scratch, which means the model will start the learning curve from the "zero" point. In addition to that, since the computational resources for the authors are quite limited and a DCNN usually relies on highly efficient GPUs, so unlike other DCNNs such as the ResNet, Inception-ResNet, and DenseNet, etc. the network structure here is not as deep as those above. In fact, the convolutional neural network built in the paper only contains one convolution layer, and the main framework of the network is shown in the figure. Basically, the network contains

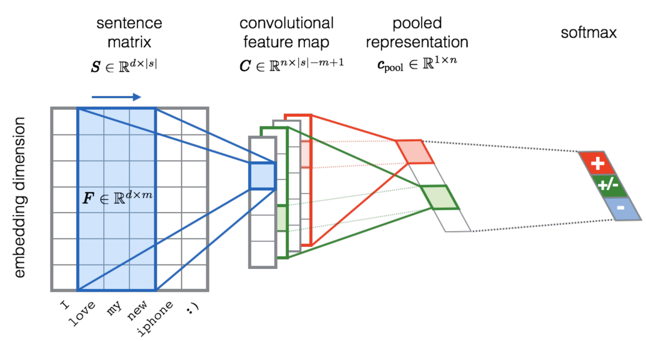

Fig. 1. The Basic Structure of a Convolutional Neural Network [5].

four parts. The first part is to convert a sentence into a matrix, the raw information is from Twitter as a sequence of words: $[w_i, \ldots, w_{|s|}]$, where each word is obtained from a vocabulary V. In this case, words become distributional vectors $\mathbf{w} \in \mathbb{R}^{1\times d}$, and those combined vectors become a matrix $\mathbf{W} \in \mathbb{R}^{d\times|V|}$. Now, for each tweet obtained from Twitter, a sentencing matrix $\mathbf{S} \in \mathbb{R}^{d\times|s|}$ will be built, where the i-th column suggests the position of a word w in a sentence. After this process, the original linguistic inputs become mathematical representations and could be used in further computation and calculation. The second part is to extract the semantic feature out of those mathematical numbers. To accomplish this goal, the convolution operation will be performed in this section. Unlike the "ordinary" convolution operation in the area of signal processing, the convolution here actually does not involve any convolution operation, which could be quite misleading to the readers. In fact, the convolution operation in this section is to map the sentence matrix into a number, which indicates the importance or the weight of a word in a sentence. To perform such operation, a "window" will be required, normally, the "window" is called a filter $\mathbf{F}$, and it could be denoted as $\mathbf{F} \in \mathbb{R}^{d \times m}$, where m indicates the width in a vector, and

$$\mathbf{c}_i = (\mathbf{S} * \mathbf{F})_i = \sum_{i,j} (\mathbf{S}_{[:,i-m+1:i]} \otimes \mathbf{F})_{kj} \tag{1}$$

the operator $\otimes$ indicates the element-wise multiplication and $\mathbf{S}_{[:,i-m+1:i]}$ specifies the matrix slice along columns with the size *m* [5]. Apart from that, an activation function is usually required in this part. The use of an activation function is believed to modify the data, for example, one of the commonly used sigmoid function will rearrange the original output between 0 and 1 to indicate how important this data is, which would also add a nonlinear component into the data. Another commonly used function is the rectified linear unit, which can be shorted as ReLU. The function is as simple as:

$$max(0, x) \tag{2}$$

where *x* is the raw output of the convolution operation [5]. The idea of this function is to keep all the positive value as useful information while discarding all the negative value as useless information. At the end of this procedure, the original matrix will be summed up into a single number. The following part of the network is a pooling layer that can aggregate the information and reduce the representation. The popular choice here is to use max pooling, which means to extract the maximum value among a specific size of data. By doing this, the most "important" value will be captured while those not so "important" data will be filtered, in case the overfitting problem would occur, which is usually due to the remaining of a large amount of "useless" data. At the final stage, the neural network will come up to a conclusion that the sentence goes through the network belongs to which kind of sentiment. By connecting a softmax layer to the network, the final decision will be made by computing the distribution probability over different labels. The function could be denoted as:

$$\begin{aligned} P(y = j|\mathbf{x}, \mathbf{s}, \mathbf{b}) &= softmax(\mathbf{x}^T \cdot \mathbf{w} + \mathbf{b}) f(x) \\ &= \frac{e^{\mathbf{x}^T \cdot \mathbf{w}_j + b_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T \cdot \mathbf{w}_j + b_j}} \end{aligned} \tag{3}$$

where $\mathbf{w}_k$ and b_k are the weight vector and the bias of the *k*-th class [5].

B. BiDirectional Long Short-Term Memory

Usually, the traditional neural network models are not used to dealing with sequence data, because it is ineffective, and it is not designed to describe the correlations between the previous and following sequence. So like text, video, these types of data need other model to deal with. Recurrent Neural network (RNN) is designed to learn sequence input, it connects the input between hidden layers and can learn sequence feature. Every time RNN take an input and give an output then the output together with the next input are the next input for the next step. But for RNN there is a downside, in theory RNNs are capable of handling "long-term dependencies", such as a long sentence. If the information between the very beginning words and the last few words is correlated, even there is a gap between, RNN would learning that. Unfortunately, as the gap grows, RNNs become unable to learn the connection of the information. When RNN is hard to accomplish this mission, we can always make an improvement. Long Short Term Memory networks (LSTMs) is to solve the problem that RNN has. LSTM is an optimized version of RNN. In standard RNNs, they have a chain of repeating modules of neural network, the repeating module will have a simple structure, such as a single function layer. LSTMs base on this repeating module, LSTM increase the complexity of the structure, there are four neural network layers instead of a single neural network layer. A basic LSTM has four layers of the repeating structure which is a cell, an input gate, an output gate and a forget gate. And the cell state is where LSTMs keep the information, the cell state runs straight down the entire chain, and the information changed only caused by some minor linear interactions. It is very easy for information to persist with no change. The LSTM can remove or add information to the cell state, carefully regulated by gates. Gates are a way to optionally let information through. They are composed out of a sigmoid neural network layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means cannot go through, while a value of one mean everything can through. Forget gate layer decide what information should be throw away of the previous one. Input gate layer decide which value should be upgrade, to replace the old one that been throw away. Output gate layer decide which part of information to be the output. Traditional LSTM networks, information can only be evaluated in forward, because the state of time *t* only depends on the information before time *t*. In "Sentiment Analysis of Comment Texts Based on BiLSTM, [2]" in order to make the context information persist, they have presented a Bidirectional LSTM based sentiment analysis for comment texts. Bidirectional LSTM has two separate LSTMs which the input is presented forwards and backwards, and two separate LSTMs share the same output layer [8]. The most important thing for sentiment polarity is the words with sentiment information. Using BiLSTM to capture the context information effectively, and the comment vectors are better represented. For the task of sentiment analysis of comment texts, BiLSTM is used to extract text feature. But first we transfer comment texts to word vectors, and in this paper they are using traditional TF-IDF algorithm to calculate the weight of the words for a better word vector representation. In a single document, the frequency of the word or in many documents, the distribution of the word TF-IDF is the most commonly used. The formulas of TF-IDF are as follows:

$$w(t_i, d) = \frac{tf(t_i, d) \times idf(t_i)}{\sqrt{\sum_{t_i \in d} [tf(t_i, d) \times idf(t_i)]^2}} \tag{4}$$

$$idf(t_i) = \log(N/n_{ti}) + 1 \tag{5}$$

$w(t_i, d)$ denotes the weight of the word t_i in document d, $tf(t_i, d)$ denotes the frequency of the word t_i in document d, N denotes the total number of documents and $n_t i$ denotes the number of documents in which the word t_i appears. And the weight calculation method for word vectors is as follow:

$$w_i = tf - idf_i \cdot e \tag{6}$$

$$e = \begin{cases} \alpha, t_i \text{ is a sentiment word} \\ 1, t_i \text{ is a non-sentiment word} \end{cases} \tag{7}$$

W_i is the weight of the word, $tf - idf_i$ is the value of feature word, and *e* is the weight according to whether the word contain sentiment information, and α>1. After formalize the word vectors, BiLSTM will take word vectors as inputs, which are stored in memory calculated by gates, and it will decide what information should be kept what information should be thrown away. Then the outputs of BiLSTM are used as the final representation of the original content. The outputs are input into a feedforward neural networks classifier. Finally, the sentiment tendency of content is obtained. The structure are shown below: By using BiLSTM model, the context

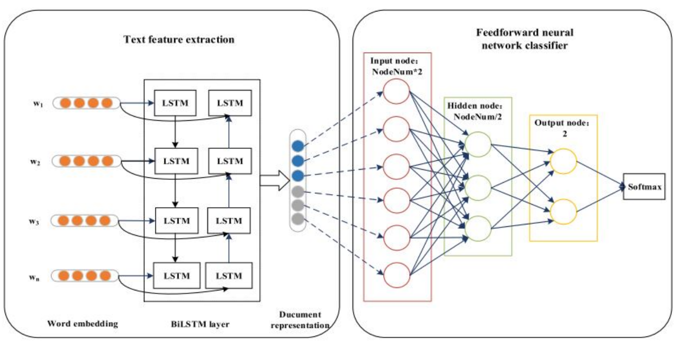

Fig. 2. Structure of BiLSTM based Sentiment Analysis [2].

information are better considered and the text representation of comment are batter obtained, and compare with other traditional method this model has better accuracy.

C. Aspect Based Sentiment Analysis

Sentiment analysis help the customer to decide which product is best suited. Sentiment analysis can be a classification process consist of 3 levels such as document level, sentence level and aspect level [7]. In aspect level features or entities are considered for finding the sentiment. The aim is to find opinion expressed about specific entities (e.g. laptops) and their aspects (e.g. Price) [9]. It comprises of three steps identification, classification and aggregation. In recent years aspect based sentiment analysis has undergone high development because SemEval 2015 [9].
There are total 4 tasks in aspect based sentiment analysis
1. Aspect term extraction
In the given sentence the task is to identify the aspect terms present in the sentence.
Example: I like the food but not the service - [service, food]
2. Aspect term polarity
With the given aspect terms the task is to identify the polarity of each aspect terms whether it is positive, negative or neutral
Example: I like the food but not the service - [service: negative, food: positive]
3. Aspect category detection
The aim is to find the aspect categories in a given sentence. They do not occur as a term in a given sentence.
Example: The laptop is good but it is expensive - [device, price]
4. Aspect category polarity
With the given aspect category the aim is to find the polarity of those terms
Example: The laptop is good but it is expensive - [device: good, price: negative]

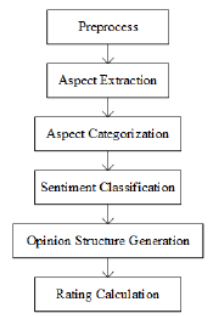

Fig. 3. Architecture of Aspect based sentiment Analysis

1. Initially the dataset is preprocessed which involves the steps called tokenization, stop word removal, stemming [10].

2. After the preprocessing step the result is used to make a distributed semantic model. Some of the semantic model such as: Continuous Bag-of-words (CBOW) which is a neural network language model predict current word using the small context window around the word another model is Global Vectors in which the ratio of probabilities of word to word co occurrence is used to encode the meaning of words [11].

3. In aspect extraction Conditional Random Field algorithm is used to get the label of each token.Some of the methods such as bag-of-word method is used to find the important features from the review sentences. The features can be unigram, bigram or N-grams. Some of the important features used in aspect extraction are:
Bag of n-grams : The occurrence of a n-gram in the context window [11]
Head Words : Word that determines the syntactic category of thatword from the dependency tree [11]

4. In Aspect categorization the category is defined each category has it binary classifier in this way the total classifier is equal to total category. Every sentence is labeled with boolean value as per the category.It will return true if the

sentence has given category otherwise false. Features used are Bag of N-grams and Bag of cluster [10].

5. In sentiment classification follows same procedure as of aspect categorization.The sentiment result can be either positive or negative. Every category has its own classifier to find the sentiment in the given sentence.Thus to find the sentiment of a given category, the defined classifier for that particular category is used [10]. The features used in the sentiment classification are:
Bag of head words: It calculates the syntatic category of the word.
Bag of Clusters: finding the word's group in the window.
Bag of K-skip bigram: It is a bag of n grams skipped over gaps [11].

6. In opinion generation CBOW model is used to find the similarity for every category. The maximum similarity score will be integrated with the extracted aspect.

Another method proposed is to construct a Recurrent neural network to learn a high level feature representation of every word in the sentence [12]. This will make the representation learning for aspect and opinion terms interactive through the underlaying dependency structure among them. The output obtained from RNN is then given to conditional random field(CRF) [13] to learn the relation between high level features and labels. Thus the CRF capture the context around each word for explicit aspect. In this way the aspect and opinion terms are transferred in both ways from parameter learning in CRF to representation learning in RNN. The CRF are statistical modeling method used for structured prediction [13].

IV. Comparison

Sentiment Analysis based on BiLSTM are compared with several traditional sentiment analysis methods, such as RNN, CNN, LSTM, Naive Bayesian. The inputs are the weighted word vectors. The hyperparameters of RNN and LSTM are the same. The CNN method uses a signal channel and the convolution filter size is set to 5. The Naive Bayesian method uses MultinomialNB with the alpha setting of 2 [2].

Method	precision	recall	F_1score
BiLSTM	91.54	92.82	92.18
RNN	87.18	86.60	86.89
CNN	85.10	84.12	84.61
LSTM	88.46	88.03	88.24
Naïve Bayesian	86.02	84.13	85.06

Fig. 4. Comparison result between BiLSTM and other traditional methods [2].

As we can see from the result, BiLSTM based sentiment analysis have all the high accuracy, other traditional methods have lower percentage on precision, recall, and F1 score. RNN has its gradient disappearance problem, CNN cannot modeled the semantic information passed by sequence. LSTM solve the gradient disappearance problem but the information is only transmitted form one direction,front to back. Naive Bayesian has certain error because it determines the probability of posterity though prior knowledge and data. BiLSTM store the sentiment information from both directions, with two LSTM unit, the semantic information of the text is captured more effectively, so the performance is better.

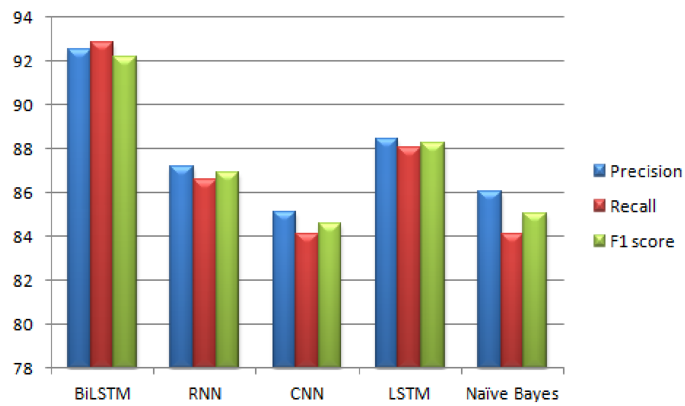

Fig. 5. Comparison result between BiLSTM and other traditional methods [2].

V. Conclusion

In this paper, we explain some of the most commonly used machine learning algorithms for sentiment analysis which includes CNN, RNN, LSTM, Bi-LSTM and Aspect Based sentiment analysis. These algorithms were also compared across their precision, recall and F1 Score. After comparison, it was evident that Bi-LSTM outperformed the algorithms with respect to precision, recall and F1 Score because of its advantage of storing sentiment information from both directions. Furthermore, this algorithm can be implemented in a practical application such as sentiment analysis on movie reviews, twitter data or drug reviews and is expected to perform better compared to the currently used algorithms. As for the convolutional neural network model, unlike the BiLSTM, it does not contain the information from the past "memory" of the data, which could be the reason that this model dos not perform as good as the BiLSTM model.

REFERENCES

[1] Karmaniolos, S., & Skinner, G. (2019, February). A Literature Review on Sentiment Analysis and its Foundational Technologies. In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (pp. 91-95). IEEE.

[2] G. Xu, Y. Meng, X. Qiu, Z. Yu and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," IEEE Access, IEEE, 2019.

[3] Burges, C. J., Smola, A. J., & Scholkopf, B. (1999). Advances in kernel methods. Support Vector Learning

[4] Li, D., & Qian, J. (2016, October). Text sentiment analysis based on long short-term memory. In 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI) (pp. 471-475). IEEE.

[5] Severyn, A., & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 959-962).

[6] Shivaprasad TK, Shetty J (March 2017) Sentiment analysis of product reviews: a review. In: 2017 International conference on inventive communication and computational technologies (ICICCT). IEEE, New York, pp 298–301

[7] Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S. F., & Pantic, M. (2017). A survey of multimodal sentiment analysis. Image and Vision Computing, 65, 3-14.

[8] Graves, Alex, and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." Neural networks 18.5-6 (2005): 602-610.

[9] Pontiki, Maria, et al. "Semeval-2015 task 12: Aspect based sentiment analysis." Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). 2015.

[10] Devina Ekawati ; Masayu Leylia Khodra "Aspect-based sentiment analysis for Indonesian restaurant reviews" 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA) 2017

[11] "UWB at SemEval-2016 Task 5: Aspect Based Sentiment Analysis" Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016),Association for Computational Linguistics,2016

[12] Wang, Wenya, et al. "Recursive neural conditional random fields for aspect-based sentiment analysis." arXiv preprint arXiv:1603.06679 (2016).

[13] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random elds: "Probabilistic models for segmenting and labeling sequence data". In ICML, pages 282–289.