Keyword extraction for social media short text

Dexin Zhao, Nana Du, Zhi Chang, Yukun Li

Tianjin Key Laboratory of Intelligent Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China
qiqiharxin@163.com

Abstract—**With the booming development of social media in recent years, researchers have begun to pay more attention to extracting personal profiles from information. Keyword extraction plays an important role in extracting personal profiles. However, most of the previous studies are only valid for ordinary text, but not ideal for social media short text. In this paper, we propose an improved method for keyword extraction based on Word2vec and Textrank to solve the unique problem of social media short text. Our approach uses the Word2vec to capture the semantic features between words in selected text, and meanwhile naturally fuses the word frequency, semantic relation and directional relation into Textrank to extract keywords. We conduct the experiments on the three datasets. The experimental results show the superior performance of our method in keyword extraction.**

Keywords—short text; word2vec; textrank; keyword extraction

I. INTRODUCTION

In recent years, the widespread use of online social media provides users with an opportunity to share their interests. With the continuous study of human, the concept of user profile gradually has been formed. User profile is a description of the user or product feature attributes from each dimension, and it provides an effective source of information for user analysis. The main work of user profile is keyword extraction.

However, some inherent characteristics of social media make it difficult to extract keywords. Such as Sina Weibo, it is a microblog service platform, on which user can publish short messages visible to the public or a group specified by the user. The number of words are limited and its content is clutter and not standard, besides, users often publish some meaningless content. Hence it is difficult to extract keywords from social media text.

The main task of this paper is to extract the effective keyword information from the published social media information by user. We will show how to use Word2vec and improved Textrank method to improve results of keyword extraction for social media short text.

The rest of this paper is organized as follows. Section 2 reviews related work on user profile and keyword abstraction. Key techniques involved in this paper are described in detail in Section 3. The experimental results are illustrated and discussed in Section 4. Finally, this paper is summarized in Section 5.

II. RELATED WORK

Keyword extraction as a classical task in the field of natural language processing, this research has been going on for a long time. The task of keyword extraction is generally divided into two steps: The first stage is to generate the candidate keyword set, the second stage is to select keywords by supervised learning or unsupervised learning algorithm. Early methods of supervised learning treat the keyword extraction task as a simple binary-class task (Frank et al., 1999; Turney, 1999; Wolf, Noam et al., 2013; Ludovic Jean-Louis, 2014)[1-4], positive class is keyword, negative class is non-keyword. But each candidate keyword is independent and cannot be compared. At present, most of the keyword extraction methods are unsupervised, they are divided into the following four types: Graph-based ranking method, the basic idea is to create a graph and generate keywords according to the importance (Mihalcea and Tarau, 2004; Wan and Xiao, 2008a;Yang, Fan, 2016)[5-7]. Topic-based clustering method (Grineva et al., 2009; Liu et al., 2010; Maryam Habibi et al., 2015)[8-10], this method needs to consider the subject of the given document, and cluster according to different themes. Besides, the method based on simultaneous learning, text summarization and keyword extraction can interact with each other. The last, the method based on language model has good performance in keyword extraction task.

In recent years, machine learning-based methods and deep learning-based methods for keyword extraction have been widely adopted due to their excellent performance (Li, Wang, Zhou & Lee, 2011; Santosh Kumar Bharti, 2017; Li Yuepeng, 2015)[11-13]. How to extract complex features rather than simple features and figuring out which types of features are more valuable are essential in machine learning-based method. Up to this day, a variety of feature extraction methods have been proposed, lexical–syntactic patterns and many other novel models (Xia & Zong, 2010)[14]. Nevertheless, due to the inherent characteristics of social media short text, its application effect is not ideal. Semantic features can reveal the deep and implicit semantic relationships between words, which is beneficial to abstract semantic features. Hence, the keyword extraction based on semantic features should be able to obtain better results.

Word2vec is a tool based on deep learning, they are widely applied in text. Word2vec can learn the vector representations of words in the high-dimensional vector space.

978-1-5386-4806-3/17 $31.00 © 2017 IEEE
DOI 10.1109/WISA.2017.12

The Textrank algorithm is often used in ranking and keyword extraction. However, it only considers the word itself, the semantic relation between words is not considered. So we try to combine the two methods to improve the weights of the graph nodes, in addition, the factors of word frequency and directional relation are also considered. And then, we apply the improved ways to extract keyword for social media short text. The experimental results show that our method can obtain a better performance.

III. KEYWORD EXTRACTION MODEL

As we know, the Word2vec method is often used in text processing. Some researches show that Word2vec has a good effect on solving the semantic relations in text, it can also be used to set the frequency of words, and can filter some words which appear too little. Textrank is often used for keyword extraction, but the relationship between the Textrank nodes in the graph does not consider their semantics, frequency and so on. Therefore, in view of the special features of the social media text, we combine these two algorithms and improve the Textrank algorithm to extract the keywords, we will test in the next chapter. This part mainly introduces the method of combining Word2vec and Textrank to carry out the specific process of extracting keywords for social media short text. We will explain it in two parts in the following.

The general framework is shown in Figure 1:

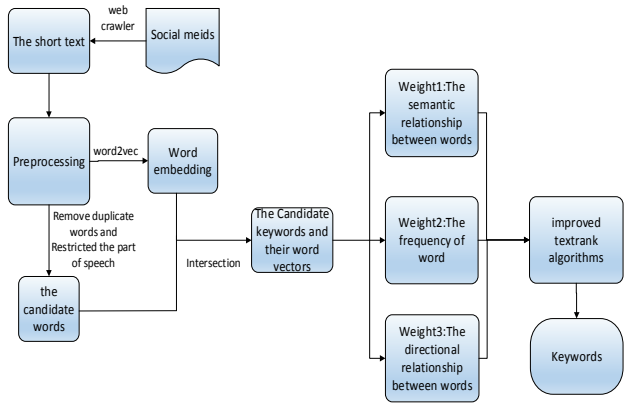

Fig. 1. The general framework of our work

A. Candidate keywords extraction

Step 1 Data preprocessing: Firstly, we need to preprocess the social media text, we adopt Jieba tool to process, this step includes: removal of stop-words, removal of certain classes of characters, such as numbers, special characters, segmenting texts into words. In this step, we do not remove duplicate words because it will affect the training of Word2vec. Besides, we delete words appearing more than 100 times since they cannot be recognized easily and don't contain valid information.

Step 2 Word2vec training: We will deal with the data obtained in the previous step in two parts. In this part, we use of Word2vec to train the preprocessed data, the data after the preprocessing will convert to word embedding. We use the Gensim library to invoke the Word2vec model in python. We will train these samples in the form of a txt file and get the corresponding word embedding, in addition, we set the word frequency to be greater than or equal to 5, so that we can filter out the low frequency words. The following table 1 introduces the main parameter values of the Word2vec training model in this experiment. First, we construct a unique dictionary by training the text, and then get the vector expression of the word by training. This paper sets the word vector to 200 dimensions.

TABLE I. THE KEY PARAMETERS OF TRAINING COMMAND.

Parameters	Explanations	Default Values
Train	Name of input file	Train.txt
Output	Name of output file Vectors.txt	Vectors.txt
cbow	Choice of training model 0: Skip-gram model 1: CBOW model	0
Size	Dimension of vectors	200
Window	Size of training window	5
Negative	Choice of training method 0: Hierarchical Softmax method 1: Negative Sampling method	0
hs	Choice of training method 0: Negative Sampling method 1: Hierarchical Softmax method	1
Sample	Threshold of sampling	le-3
Threads	Number of running threads	1
Binary	Mode of storage 0: Common format 1: Binary format	0

Step 3 Get the candidate keywords and the word vector: We delete duplicated words from initially extracted data and then limit their POS, table 2 shows the reserved parts of speech for the keywords. Now we take the intersection of the newly obtained data and the trained data by Word2vec to obtain the candidate keywords and their word vector.

TABLE II. THE PARTS OF SPEECH.

Language	Part of speech coding
Chinese text	an, i, j, l, n, nr, ns, nt, nz, t, v, vd, vn

B. Improved Textrank algorithms

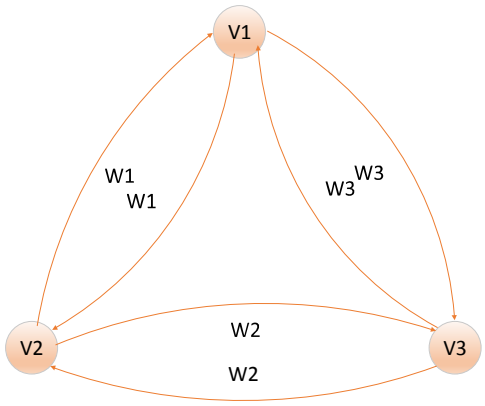

Fig. 2. The process of Textrank

As shown above, V stands for candidate keywords, and W is the weight between two words. In this paper, we consider the weight problem of keyword extraction from three aspects:

1) The semantic relationship between words: if the semantic relevance between the two words is higher, the weight between them will be higher.

For it, we use the Word2Vec tool to train word vector, we use the cosine similarity to calculate the similarity between two words by word vector and form the similarity relation matrix between words, see the following formula (1), V_i, V_j respectively represent two words, $Sim(V_i,V_j)$ refers to the similarity between two words.

$$Sim(V_i,V_j)=\cos\theta=\frac{V_i.V_j}{||V_i||.||V_j||} \tag{1}$$

Equation (2) represents the similarity matrix between words. w_{ij} represents the cosine similarity between the word V_i and the word V_j for the word vector, $w_{ij}=Sim(V_i,V_j)$.

$$M_\alpha(Sim(V_i,V_j))=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{2}$$

2) The frequency of word: If the more the word appears in the text, the higher the value of the word. We give the formula (3).

$$W_\beta(v_i,v_j)=\frac{C(v_j)}{\sum_{v_k\in Out(v_i)}C(v_k)} \tag{3}$$

$C(V_j)$ represents the number of times that the word V_j appears in the text, $\sum_{v_k\in Out(v_i)}C(v_k)$ represents the sum of each word occurrences in out degree of V_i , $W_\beta(V_i,V_j)$ is the ration of the frequency of V_j and the sum of the frequencies of V_i to each word, it represents the influence of word frequency. Form a transfer matrix (4):

$$M_\beta(W_\beta(v_i,v_j))=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{4}$$

3) The directional relationship between words: The word V_i points to another word V_j, so the importance of the word V_i is passed to the word V_j, in addition, the more words point to the word V_j, that is, the more words are associated with the word V_j, the word V_j is more important. About this point, we retain the original Textrank method to deal with this problem, given the formula:

$$W_\gamma(v_i,v_j)=\frac{1}{|Out(v_i)|} \tag{5}$$

$W_\gamma(V_i,V_j)$ means that the effect of V_i on V_j, $|Out(v_i)|$ shows that the number of V_i points to other words. The transfer matrix as follows:

$$M_\gamma(W_\gamma(v_i,v_j))=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{6}$$

For any node V in the figure, its importance consists of the above three aspects: the semantic relationship between words, the frequency of word, the directional relationship between words, we set that α, β, γ respectively represent the proportion of these three different effects, and then, $\alpha+\beta+\gamma=1$. According to the theory of link analysis, the importance of nodes can be calculated by iteration if given the transfer probability between nodes in a given graph. The matrix M represents the transfer matrix between words, as shown in equation (7):

$$M=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \tag{7}$$

Column j in M represents the probability distribution of jumping from node V_j to the other nodes, and the sum of the probabilities of each column is 1. P_{ij} represents the transition probability from node j to node i, we incorporate the three aforementioned weights into it, and then, we get a new transfer matrix (8).

$$M'=\alpha M_\alpha(Sim(w_i,w_j))+\beta M_\beta(W_\gamma(v_i,v_j))+\gamma M_\gamma(W_\gamma(v_i,v_j)) \tag{8}$$

Bring formula (8) into the Textrank formula, we call this W-Textrank, update the formula as follow:

$$WS'(V_i) = d*M'*WS'(V_j) + (1-d)*\frac{e}{n} \quad (9)$$

Here, e is a vector with all components of value 1 and dimension n. We stop the iteration calculation when the two adjacent calculative results is similar, now the iterative calculation results have converged. And then, the current weights of all vocabulary nodes are sorted in descending order, we choose Top N words as the keywords for the document. Table 3 describes the main parameters of the W-Textrank algorithm in this experiment.

TABLE III. THE KEY PARAMETERS OF W-TEXTRANK ALGORITHM

Parameters	Explanations	Default Values
α	The coefficient of semantic and syntactic relationship between words	0.2
β	The word occurrence frequency	0.4
γ	The correlation coefficient of words with other words	0.4
d	Damping coefficient	0.85

The procedure of keyword extraction is described below:

Algorithm 1 keyword extraction

Input: Weibo/Tweet Collection
Output: Keyword
procedure extraction process
1. Vector set ⟵ clean words
2. Cosine similarity array ⟵ Vector set
3. for i in range(iteration) set the number of iterations is 130 , iteration =130
4. P=d.M'.p+(1-d).u , d=0.85 Calculate the improved Textrank value
5. srt =sort(p)
6. Choose top N keyword from srt
7. end for
8. end

IV. EVALUATION AND ANALYSIS

A. Data set

We now describe the training dataset in detail. We make use of Wikipedia, Sina Weibo and Twitter to obtain data. Firstly, more than 100,000 data crawled are downloaded from Sina Weibo, more than 30000 Twitter data are downloaded from data mining competition Tianchi, and then, 20 long documents are downloaded from Wikipedia. Finally, we use the above three datasets to compare. Considering that each Wikipedia document has a different theme, we will extract keywords of different themes, we choose five documents of them to compare; similarly, Twitter data are also based on different themes, we only choose one theme of the Twitter data to experiment; Sina Weibo data is crawled by the user, there is no specific theme, in this paper, we selected 22,000 data for the experiment. Table 4 shows the comparison of the different datasets in this experiment.

TABLE IV. COMPARISON OF DIFFERENT DATA SOURCES

Dataset	Publish Time	Data amount	Description
Sina Weibo	2014	2.32MB 22000texts	These texts are short, messy and have a lot of network terminology.
Twitter	2015	1.61MB 14555texts	It is a foreign social software and its feature like Sina Weibo.
Wikipedia	2009	244KB 5 documents	Long text, formal format,clear semantics

B. Experimental Design

We use the Jieba tool in python for preprocessing in our experiment. The obtained dataset after data preprocessing is dealt separately by two steps: (1) we convert data to word vector by Word2vec tool. (2) we remove duplicated words and limit their part of speech on the data after data preprocessing. Step (1) and step (2) are parallel operations. And then, these data in the above two steps are intersected with each other to obtain the candidate keyword set and these word vector. Finally, we will use the W-Textrank algorithm to calculate the importance ranking of each word, and we choose the TOP N words as the keywords.

In this paper, the TF-IDF and the traditional Textrank algorithm as the base algorithm to compare with our approach. We use accuracy rate, recall rate and F-measure to evaluate the effectiveness of our approach.

$$P = \frac{\text{The number of keywords extracted correctly}}{\text{The number of all keywords extracted}} \quad (10)$$

$$R = \frac{\text{The number of keywords extracted correctly}}{\text{The number of standard keywords}} \quad (11)$$

$$F\text{-}measure = \frac{2*PR}{P+R} \quad (12)$$

C. Experimental Results and Analysis of the Result

Table 5, 6 and 7 list the results obtained. In this part, we select five documents from Wikipedia data set to experiment; we evaluate the result of Twitter data and Sina Weibo data according to the different number of keywords. Table 5, 6 and 7 are the experimental results of Wikipedia, Sina Weibo, Twitter respectively. The experimental results are as follows: Table 5 is an experiment on Wikipedia's long text. The result shows that the three algorithms have very close experimental result for long texts. In the five documents, the W-Textrank algorithm is a little better than the other two methods for the three documents .

TABLE V. THE EXPERIMENTAL RESULT OF WIKIPEDIA

	Doc1			Doc2			Doc3			Doc4			Doc5		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TF-IDF	30%	30%	30%	30%	33.3%	31.6%	40%	26.7%	32%	20%	16%	17.8%	30%	28.6%	29.3%
Textrank	20%	20%	20%	20%	22.2%	21%	45%	30%	36%	40%	35.6%	37.8%	40%	38%	39%
W-Textrank	40%	40%	40%	25%	27.8%	26.3%	50%	33.3%	40%	30%	24%	26.7%	40%	38%	39%

TABLE VI. THE EXPERIMENTAL RESULT OF SINA WEIBO

	N=5			N=10			N=15			N=20		
	P	R	F	P	R	F	P	R	F	P	R	F
TF-IDF	20%	5%	8%	30%	15%	20%	33.3%	25%	28.6%	35%	35%	35%
Textrank	40%	10%	16%	20%	10%	13.3%	33.3%	25%	28.6%	30%	30%	30%
W-Textrank	60%	15%	24%	50%	25%	33.3%	40%	30%	34.3%	45%	45%	45%

TABLE VII. THE EXPERIMENTAL RESULT OF TWITTER

	N=3			N=5			N=7			N=10		
	P	R	F	P	R	F	P	R	F	P	R	F
TF-IDF	66.7%	20%	30.8%	60%	30%	40%	71.4%	50%	58.8%	70%	70%	70%
Textrank	66.7%	20%	30.8%	80%	40%	53.3%	71.4%	50%	58.8%	80%	80%	80%
W-Textrank	66.7%	20%	30.8%	80%	40%	53.3%	85.7%	60%	70.6%	80%	80%	80%

Table 6 shows the results of the experiment on Sina Weibo under different number of keyword, we set the number of extracted keywords N are 5, 10, 15, 20, in addition, the number of artificial extraction keywords is 20. We can see from the table 6, no matter how much the number of extracted keywords, the W-Textrank algorithm is always higher than the other two algorithms. When N equals 5, the effect of three indicators is the best.

Table 7 indicates an experiment on the Twitter data set, because the size of the Twitter data set is smaller than the size of Weibo, the number of extracted keywords is changed to 3, 5, 7, 10. We can see that the accuracy rate, recall rate and F-measure of the three algorithms are same when N equals 3; the accuracy of W-Textrank algorithm is as high as Textrank algorithm when N equals 5 and 10, and TF-IDF algorithm is relatively low; the W-Textrank algorithm is higher than the other algorithms when N equals 7. In addition, we can clear see that the test results for the Twitter dataset are significantly higher than the results of the other datasets, because the Twitter data comes from the data mining competition and it contains a clear theme.

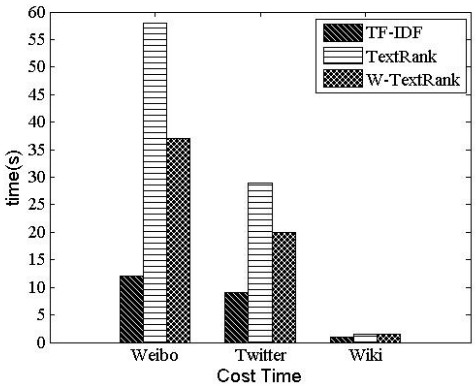

Fig. 3. Time spent in the experimental model

In the experiment of time spent, as shown in Figure 3，in the three different data sets, the time spent is a big gap, so we set the time coordinates in seconds and ignore the decimal point. It can be seen that although the time spent in the W-

Textrank algorithm is more than the TF-IDF algorithm, it is much smaller than the Textrank algorithm, this is because the Textrank algorithm has iterative calculations that are time-consuming, and this paper proposed the W-Textrank algorithm based on the Textrank algorithm, and we have dealt with the choice of candidate keywords to significantly save time.

We can draw the following conclusions from the above results: (1) in the longer and content-rich article, the accuracy rate is not much different from Textrank，TF-IDF and the W-Textrank. (2) the W-Textrank algorithm is superior to the other two algorithms for the social media short text. (3) the time spent in the W-Textrank algorithm is less than the Textrank algorithm.

After proving the validity of our algorithm, we extracted keywords of all information published from one user in Twitter, and visualized it with using the word cloud. The result is shown below:

Fig. 4. Twitter user profile

V. CONCLUSION

Differing from the general keyword extraction, this paper is mainly for the social media short text, our approach is considerd the feature of the social media short text and the relationship between words rather than a simple keyword extraction. In this paper, we use two methods, one is Word2vec and another is Textrank algorithm, we use Word2vec to convert text to word vector and improve the Textrank algorithm from three aspects, finally, we extract keywords with improved method. This paper compared the three data sets. The experimental results show that our approach has a better effect on social media short text, but also time spent is less. On this basis, texts posted in Twitter from a user are used to extract the keywords and it lays the foundation for user profile. In the future work, we will explore other aspects of the user profile.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (No.61202169) and Natural Science Foundation of Tianjin (Grant No.15JCYBJC46500).

REFERENCES

[1] Frank E, Paynter G W, Witten I H, et al. Domain-Specific Keyphrase Extraction[C]// ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November. 1999:283-284.

[2] Turney, P.D. Learning to Extract Keyphrases from Text. National Resea rch Council, Institute for Information Technology, Technical Report ER B-1057, 1999.

[3] Wolf N, Zhu Z, Semret N, et al. Providing product recommendations through keyword extraction from negative reviews: US, US 8515828 B1[P]. 2013.

[4] Jean-Louis L, Zouaq A, Gagnon M, et al. An Assessment of Online Semantic Annotators for the Keyword Extraction Task[C]// Pacific Rim International Conference on Artificial Intelligence. Springer International Publishing, 2014:548-560.

[5] Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain: Association for Computational Linguistics.

[6] Wan, X., & Xiao, J. (2008a). CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of the 22nd International Conference on Computational Linguistics COLING '08 (Vol. 1, pp. 969–976). Stroudsburg, PA: Association for Computational Linguistics.

[7] Onan A, Korukoglu S, Bulut H. Ensemble of keyword extraction methods and classifiers in text classification[J]. Expert Systems with Applications, 2016, 57(C):232-247.

[8] Grineva M, Grinev M, Lizorkin D. Extracting key terms from noisy and multitheme documents[C]// International Conference on World Wide Web, WWW 2009, Madrid, Spain, April. DBLP, 2009:661-670.

[9] Liu, Z., Huang, W., Zheng, Y., & Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9–11 October 2010, MIT Stata Center, A meeting of SIGDAT, a Special Interest Group of the ACL. (pp. 366–376). Stroudsburg, PA: ACL.

[10] Habibi M, Popescu-Belis A. Keyword Extraction and Clustering for Document Recommendation in Conversations[J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2015, 23(4):746-759.

[11] Li, S., Wang, Z., Zhou, G., & Lee, S. Y. M. (2011). Semi-supervised learning for imbalanced sentiment classification. In Proceedings of the 22nd international joint conference on artificial intelligence (pp. 1826–1831)

[12] Bharti S K, Babu K S. Automatic Keyword Extraction for Text Summarization: A Survey[J]. 2017.

[13] Li Y, Jin C, Ji J. A Keyword Extraction Algorithm Based on Word2vec[J]. e-Science Technology & Application, 2015.

[14] Xia, R., & Zong, C. (2010). Exploring the use of word relation features for sentiment classification. In Proceedings of the 23rd international conference on computational linguistics (pp. 1336–1344). Association for Computational Linguistics.