

# Word Embeddings-based Sentence-Level Sentiment Analysis considering Word Importance

**Toshitaka Hayashi, Hamido Fujita**

Iwate Prefectural University, 152-52 Sugo, Takizawa, 020-0693, Japan

E-mail: g236r002@s.iwate-pu.ac.jp, HFujita-799@acm.org

---

*Abstract: Word2vec has been proven to facilitate various Natural Language Processing (NLP) tasks. We suppose that it could separate the vector space of word2vec into positive and negative. Hence, word2vec can be applied to Sentiment Analysis tasks. In our previous research, we proposed the word embeddings (WEMB) based Sentence-level Sentiment Analysis method. Word's vectors from WEMB are utilized to calculate the sentence vector. Training of the classification model is done using sentence vector and the polarity. After training, the model predicts the polarity of the unlabeled sentence. However, the sentence vector was insufficient because the method treats all words with the same weight for calculating a sentence vector. In this paper, we propose a method to solve this problem. We consider word weight according to their importance for calculating sentence vector. The proposed method is compared with the method without word importance, and the accuracy is improved. However, there is still a grim difference with state of the art. We discuss the next improvement and present future work.*

*Keywords: Sentiment Analysis; Polarity Classification; Word Embeddings; Word Importance*

---

## 1 Introduction

A human can detect emotion or sentiment in written language. However, social media and other tools have increased the number of sources and volumes of information; it is too voluminous and complex for humans. Sentiment Analysis [3, 4, 10] [18, 22] (SA: also known as Opinion Mining) is the challenge. SA is one of the most active research areas in Natural Language Processing (NLP) and machine learning [10], with a particular interest in the classification of text into positive, negative, neutral. Such a task is known as the Polarity Classification problem. The goal is classifying the polarity of a given text at the document, sentence, or feature/aspect level.

There are two main types of methods for SA, lexicon-based approach or machine learning based approach. Lexicon-based approach processes the text data by keyword matching using sentiment lexicon. Lexicon has sentiment information of words. For an example of the lexicon, SentiWordNet [1, 2] is proposed by Esuli *et al.*, and SenticNet [7, 8, 9] [11, 12] is proposed by Cambria *et al.* Lexicon alone does not give much accuracy [38], but combining it with a language rule called Semantic Rule [23, 31, 40] produces good results [11, 24, 25]. Semantic Rule is utilized to handle the exception of the language like negation. It has a good impact on polarity classification. However, the method requires the definition of Semantic Rule, which has to be done manually.

Machine learning based approach [28] aims at building predictive models for the sentiment, which use supervised learning. In this approach, a model is created to predict unlabeled vectors by training feature vectors and labels. There is an issue, the feature vector of the text is required to apply supervised learning to SA. The main problem is how to extract a feature vector from the text data.

As one of the conversions from text into the vector, word2vec [35] is introduced, which trains text corpus and outputs Word Embeddings (WEMB) that are the set of word vectors. The idea of word2vec (WEMB) originated from the concept of distributed representation of words. Word2vec has been proven to facilitate various NLP tasks [5, 14, 20]. It can be expected to prove polarity classification too.

In light of such trends, we use a machine learning based approach because it can do automatically. Also, we use word2vec because it has proven various NLP tasks. Our goal is to extract a feature vector of a sentence. WEMB is utilized to obtain word vectors, and the word vectors are combined to make a sentence vector. The main problem is how to combine word vectors to make sentence vector.

In our previous research [34], we propose the method for Sentence-level Sentiment Analysis using a supervised approach with WEMB-based feature extraction. Our process extracts each word vector in a sentence from WEMB. Also, sentence vector is calculated using simple calculation such as the average, the variance, and the geometric mean of the word vectors. However, there is a problem that the method treats important words for polarity and not important words for polarity with the same weight.

In this paper, we consider word importance to solve this problem. In our proposed method, a word that has a bias in the rate of occurrence due to polarity is considered important. Hence, word importance is calculated from training data. Also, sentence vector is calculated by a weighted average using word vector and word importance. It makes a better result than our previous research [34]. The rest of the paper is structured as follows. Section 2 summarizes the previous work regarding machine learning based Sentiment Analysis, word2vec and WEMB, and WEMB based method. In Section 3, our proposed method about how to calculate

word importance and how to calculate sentence vector are written. In Section 4, experiment results for public datasets are given. In our experiment, Twitter datasets and movie review datasets are utilized. In Section 5, the discussion about our method is done. In Section 6, conclusion and future work are written.

## **2 Related Work**

### **2.1 Machine Learning-based Sentiment Analysis**

In this paper, a machine learning based approach is presented. Machine learning based sentiment analysis aims at building predictive models for sentiment by training labeled datasets. This approach builds a feature vector of each text entry from certain aspects or word frequencies. Standard machine learning tools train the feature vectors and the labels to establish the corresponding model is validated against labeled texts [15].

The main problem of machine learning based sentiment analysis is how to extract feature vector from the text. For an example of the feature vector, five features are described in the review of Pang and Lee [4], which are Term presence/frequency, Term based features beyond unigrams [17], Part of Speech, Syntax, and Negation. Also, Socher et al. propose Sentiment Treebank [28], a model in which semantic composition is considered hierarchically. It makes good results for polarity classification.

### **2.2 Word2vec and Word Embeddings**

Mikolov et al. introduced word2vec [35] that can obtain word vectors called word embeddings by training text corpus. The idea of word2vec is originated from the concept of distributed representation of words. Word2vec algorithms such as Skip-gram, CBOW [35, 36] and GloVe [16] have been proven to facilitate various NLP tasks, such as word analogy [35], parsing [20], POS tagging [5], aspect extraction [28], topic extraction [37], translation [14], WordNet [14, 26] and so on. These approaches have shown the ability to improve classification accuracy. However, these have limited challenge to polarity classification. In this paper, the polarity classification is improved using our approach based on word2vec.

## 2.3 Word Embeddings-based Sentiment Analysis

In previous work, we have presented a WEMB-based sentiment analysis [34], which is a supervised method using WEMB. As shown in Figure 1, it is divided into two steps: Training and Classification. In the training part, sentence vector is extracted using WEMB. In the process, the vectors of each word in the sentence are extracted from WEMB. Then, sentence vector is calculated by average, variance, and the geometric mean of the word vectors. After that, the Classifier is trained using sentence vectors and polarities. In the classification part, sentence vector is calculated in the same way as training. Prediction is made using Classifier that is trained in the training part. In the sentence vector calculation, all words are treated with the same weight; this causes the inappropriate assignment to accuracy. Hence, considering word importance; it becomes important to evaluate and enhance the accuracy of the system. In the following section, we are presenting the proposal.

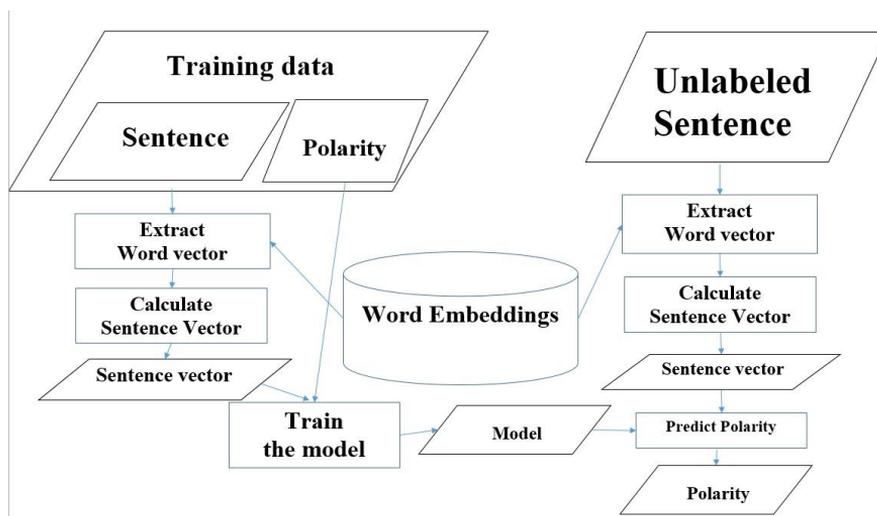


Figure 1  
Word Embeddings-based Sentiment Analysis

## 3 The Proposed Method

The proposed method is described in this section. Section 3.1 shows the general framework of our proposed system which consists of three procedures: (1) Calculate word importance, (2) Extract sentence vector, (3) Training and Classification. Section 3.2, 3.3, 3.4 shows each procedure.

### 3.1 General Framework

Our approach is shown in Figure 2. The proposed method is tackling the problem of sentence vector calculation. In this paper, we could improve the classification accuracy using word importance, compared with previous studies reported in [34]. Word importance could provide better classification accuracy in comparison to previous research.

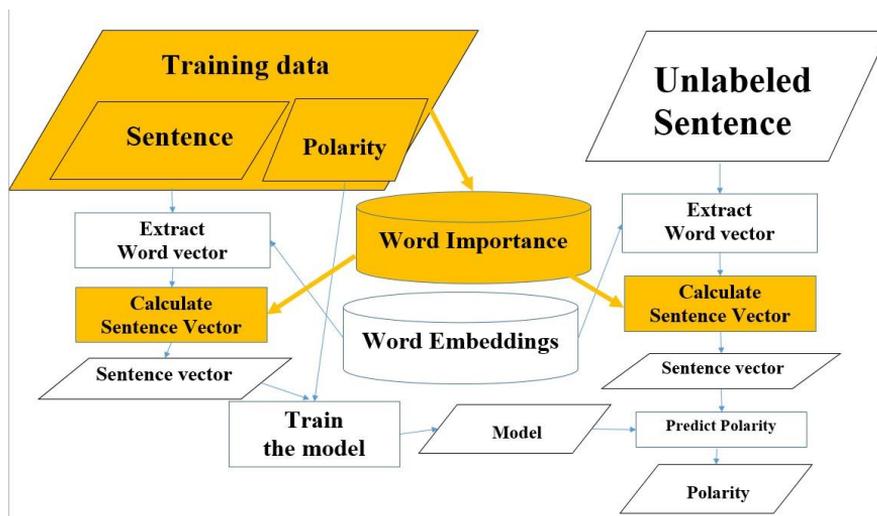


Figure 2

Word Embeddings-based Sentiment Analysis with Word Importance

Our proposal requires three elements, (1) Training data which consists of sentences and polarities, (2) WEMB, (3) classification algorithm. Also, it consists of three steps, (1) Build word importance, (2) Training, (3) Classification. Each step is described as follows:

- (1) The list of word importance is built by training data as written in Section 3.2.
- (2) The input of the training step is the training data which is consist of training sentences and polarities. The output of the training step is a classification model. Training step is carried out as in the following process:
  1. Extract sentence vectors from training sentences as written in Section 3.3.
  2. Train sentence vector and polarity using a classification algorithm as Section 3.4.
- (3) The input of the classification step is an unlabeled sentence. The output of the classification step is polarity prediction of the unlabeled sentence. Classification step is done as the following process:

1. Extract sentence vector from unlabeled sentence as explained in Section 3.3.
2. Predict the polarity from sentence vector using the trained classification model as Section 3.4.

## 3.2 How to Calculate Word Importance

The goal of this process is to calculate word importance of all words in the training data. Hence, the output is list of word importance. Word importance is calculated from training data. Training data consists of sentences and polarities.

In this work; a word of high word importance has a bias of occurrence rate due to polarity. Therefore, we propose the notion of word importance calculation as represented in Algorithm 1.

**Algorithm 1** : Word Importance Calculation

**Input** ( Training Data )

**Output**( List of Word Importance )

1. Decide which polarity is major and which polarity is minor by comparing to the number of appearance in positive sentences and negative sentences.

IF  $|Word_i \text{ in Positive Sentences}| > |Word_i \text{ in Negative Sentences}|$ :

*major Polarity*( $Word_i$ ) = *Positive*

*minor Polarity*( $Word_i$ ) = *Negative*

ELSE:

*major Polarity*( $Word_i$ ) = *Negative*

*minor Polarity*( $Word_i$ ) = *Positive*

2. Calculate word importance as given in formula (1)

$$WI(Word_i) = 1 - \frac{|Word_i \text{ in Sentences of minor Polarity}|}{|Word_i \text{ in Sentences of major Polarity}|} \quad (1)$$

### 3.3 How to Process Sentence to Sentence Vector

In order to apply a classification algorithm, sentence vector should be extracted from sentence. The sentence vector is calculated using word vectors and word importance. We propose how to extract sentence vector from sentence as follows:

1. Let Sentence as “ $Word_1 Word_2 \dots Word_i \dots Word_n$ ”
2. Extract word vectors of all words of a sentence from WEMB. If a word does not exist in WEMB, ignore the word. Each word vector is as shown in formula (2), where  $m$  is the dimension of the word vector.

$$\overrightarrow{Word_i} = (num_{i1}, num_{i2}, \dots, num_{id}, \dots, num_{im}) \quad (2)$$

3. Extract  $I_i$ , which is word importance of  $Word_i$  from word importance list

$$I_i = WI(Word_i)$$

4. Calculate  $I_{total}$  which is the SUM of word importance in the sentence as given in formula (3).

$$I_{total} = \sum_{i=1}^n I_i \quad (3)$$

5. Calculate the weight of a word  $W_i$  as given in formula (4).

$$W_i = \frac{I_i}{I_{total}} \quad (4)$$

6. Calculate Sentence Vector as given in formula (5).

$$\overrightarrow{Sentence} = (value_1, value_2, \dots, value_d, \dots, value_m) \quad (5)$$

$$value_d = \sum_{i=1}^n (W_i num_{id})$$

### 3.4 Training and Classification

The classification model is trained using training sentence vectors and the labels (Polarities). Also, the classification model is utilized to predict polarity for unlabeled sentence vector. In this paper, we apply XGBoost classification algorithm to this procedure. XGBoost [33] is utilized in our training and classification process. It is a package of gradient boosting, which produce a prediction model from an ensemble of weak decision trees [33]. XGBoost works well for high dimension features. We utilize the scikit-learn of Python because it is easy for training. The parameters of XGBoost that is utilized in our experiment

are shown in Table 1. Each parameter is explained in the documentation of XGBoost that is available at:

[http://xgboost.readthedocs.io/en/latest/python/python\\_intro.html#setting-parameters](http://xgboost.readthedocs.io/en/latest/python/python_intro.html#setting-parameters).

Table 1  
Parameters for XGBoost

Parameter	value
test_size	0.2
Objective	“binary: logistics”
eval_metric	“error”
eta	0.1
max_depth	10
number of iteration	500

## 4 Experiment

The proposed approach has been validated against the data listed in Section 4.1.

In our experiment, pre-trained WEMB are utilized. The embeddings are listed in Section 4.2. Our measurement of evaluation is listed in Section 4.3. Experiment results are shown in Section 4.4

### 4.1 The Data

Public datasets for Sentence-level SA are utilized for evaluating our method. In this paper, three Twitter datasets (TSATC, Vader, STS-Gold) and two movie review datasets (PL05, IMDB) are utilized. The balance of each dataset is shown in Table 2.

Table 2  
The balances of the datasets

Dataset	Domain	Positive	Negative	Total
TSATC	Twitter	790178	788436	1578614
Vader	Twitter	2901	1299	4200
STS-Gold	Twitter	632	1402	2304
PL05	Movie	5331	5331	10662
IMDB	Movie	25000	25000	50000

The datasets are shown as follows:

### 4.1.1 Twitter Datasets

Twitter Sentiment Analysis Training Corpus (TSATC) contains approximately a million and a half classified tweets, and each row is marked as 1 for positive sentiment and 0 for negative sentiment. The dataset is based on data from two sources: the University of Michigan Sentiment Analysis competition on Kaggle and the Twitter Sentiment Corpus by Niek Sanders. It is available at:

<http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/>.

Vader contains 2901 positive and 1299 negative tweet-like messages. It is available at:

[https://github.com/cjhutto/vaderSentiment/blob/master/additional\\_resources/hutto\\_ICWSM\\_2014.tar.gz](https://github.com/cjhutto/vaderSentiment/blob/master/additional_resources/hutto_ICWSM_2014.tar.gz).

STS-Gold has been generated for SA evaluation in the Twitter domain. The dataset contains 632 positive and 1402 negative sentences. It is available at [https://github.com/pollockj/world\\_mood/blob/master/sts\\_gold\\_v03/sts\\_gold\\_tweet.csv](https://github.com/pollockj/world_mood/blob/master/sts_gold_v03/sts_gold_tweet.csv).

### 4.1.2 Movie Review Datasets

PL05 consists of 5331 positive and 5331 negative processed sentences of movie reviews. The dataset is introduced by Pang/Lee ACL 2005. It is available at:

<http://www.cs.cornell.edu/people/pabo/movie-review-data/>.

IMDB contains 25000 positive and 25000 reviews from the review site. The target of the review is movies in the online platform. It is available at:

<https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset>.

## 4.2 Pre-trained Word Embeddings

In our experiment, two different pre-trained WEMB (Google News Embeddings and Glove Twitter Embeddings) are utilized, which WEMB is as follows:

Google-news-vectors-negative-300.bin is published by Google. The Embeddings is trained on the part of Google News dataset (about 100 billion words). It is available at: <https://code.google.com/archive/p/word2vec/>.

Glove.twitter.27B is trained on 2 billion tweets (about 27 billion tokens). It is available at: <https://nlp.stanford.edu/projects/glove/>.

Also, the number of vocabulary and dimensions for each pre-trained WEMB are shown in Table 3.

Table 3  
Number of vocabulary and dimensions for pre-trained word Embeddings

Pre-trained Word Embeddings	Number of Vocabulary	Dimensions
GoogleNewsNegative300.bin	3,000,000	300
Glove Twitter.27B	1,200,000	200

### 4.3 Measurements of the Evaluation

In this paper, four measurements of evaluation are utilized, Which, is shown in given formulas (6)-(9). Also, the confusion matrix is shown in Table 3.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

$$precision = \frac{TP}{TP + FP} \quad (7)$$

$$recall = \frac{TP}{TP + FN} \quad (8)$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \quad (9)$$

Table 4  
Confusion Matrix

	Actual Positive	Actual Negative
Predict Positive	TP	FP
Predict Negative	FN	TN

### 4.4 Experiment Result

The experiment for TSATC is done using 4-fold cross-validation. The data have been separated into several folds identified by an ID, as shown in Table 4. Each fold has 400,000 sentences. However, the last fold has only 378,614 sentences. The number of positive/negative sentence in each fold are shown in Table 4.

Table 5  
Data balance for TSATC

ID	Positive	Negative	Total
0000000-0399999	233453	166547	400000
0400000-0799999	212330	187670	400000
0800000-1199999	177064	222936	400000
1200000-1578613	167331	211283	378614
Total	790178	788436	1578614

The classification result for TSATC is shown in Table 5. Two different pre-trained WEMB (Google News Embeddings and Glove Twitter Embeddings) are utilized for feature extraction. The dimension in Table 5 is the number of dimensions of the feature vector. The accuracy is 77.7% using Google News Embeddings. Also, precision, recall, and F-score are shown in Table 5.

Table 6  
4-fold cross-validation result for TSATC (%)

Word Embeddings	Dimension	Accuracy	Precision	Recall	F1
GoogleNews300d	300	<b>77.7</b>	<b>77.6</b>	<b>77.4</b>	<b>77.5</b>
GloveTwitter200d	200	76.9	76.7	77.4	77.0

Further experiments for other datasets are also, carried out. However, the dataset is small for cross-validation. Hence, TSATC model is utilized to do prediction on other data set. The results are shown in Table 6. Overall, Google News Embeddings has produced better results than Glove Twitter Embeddings. It has 88.5% accuracy for STS-Gold dataset and 84.8% F1-score for Vader dataset.

Table 7  
Experiment results for datasets using TSATC model (%)

Dataset	Embeddings	Accuracy	Precision	Recall	F1-score
STS-Gold	Google	<b>88.5</b>	<b>77.5</b>	88.6	<b>82.7</b>
	Glove	86.0	72.1	<b>89.7</b>	79.9
Vader	Google	<b>79.5</b>	<b>86.4</b>	83.4	<b>84.8</b>
	Glove	77.1	83.4	83.4	83.4
PL05	Google	64.4	60.9	80.5	69.3
	Glove	61.8	58.2	83.5	68.6
IMDB	Google	71.5	76.3	62.5	68.7
	Glove	70.7	71.9	68.0	69.9

## 5 Discussion

In this section, we discuss the comparison result with other papers and highlighted issues in our experiment. We could prove better performance using the proposed method in relation to other previous method [34] using the evaluation on the effect of word importance. Also, we compare the proposed method with state of the art reported in [26] to better evaluation finding more improvement of WEMB based SA. The experiment carried out in this paper provide more better outcome in terms of prediction accuracy in relation to [26] [34].

A comparison of result for TSATC is shown in Table 7 in which, we have compared on accuracy and processing speed relative to our methods. We do not include training time as processing time. Our training time for 1.2 million 900-dimensional data is almost 10 hours (on our environment, CPU: Intel Core i5-6600L 3.5 GHz Quad-Core, RAM: 32 GB DDR4-2400, SSD: Samsung 850 EVO-Series 500GB).

Table 8

Comparison of accuracy of 4-fold cross-validation result for TSATC against previous research

Author	WEMB	Dimension	Accuracy (%)	Speed(Sentence/s)
Our previous method[34]	GoogleNews	900	76.3	452
	GloveTwitter	600	76.4	948
		300	75.6	<b>1719</b>
		75	71.2	<b>2479</b>
Our method	GoogleNews	300	<b>77.7</b>	712
	GloveTwitter	200	<b>76.9</b>	1129

However, processing speed is slower than the previous one. Hence, considering word importance requires processing time. Also, the number of dimensions of the feature vector decreased to one third. We think the processing time can be reduced using GPU as will be shown in our future paper.

Also, our method is compared against state of the art. The research of Araque *et al.* [26] is utilized for comparison. Their method is based on similarity-based approach. The similarities are calculated against certain words which becomes a feature vector. The comparison results of STS-Gold, Vader, IMDB, and PL05 are shown in Table 8. In the paper of Araque *et al.* [26], many experimental results are shown. In Table 9 Our method has achieved a better outcome comparing with the work reported in [26].

Table 9  
Comparison of F1-score against [26]

Method		Vader	STS-Gold	IMDB	PL05
[26]	LIU_WPath+Liu_Embeddings+W2V	90.28	82.95	<b>89.06</b>	78.19
	SWN_WPath+SWN_Embeddings+W2V	89.85	82.01	88.80	78.08
	ANEW_WPath+ANEW_Embeddings+W2V	86.91	79.91	88.85	78.03
	AFINN_WPath+AFINN_Embeddings+W2V	<b>90.41</b>	<b>83.39</b>	89.00	<b>78.29</b>
<b>Our Method</b>		84.88	82.71	68.72	69.30

Our method has a fair result for Twitter data (Vader, STS-Gold). However, our method is insufficient for movie review data (IMDB, PL05). There are differences between our method and Araque et al. [26]. In our method, only word embeddings are utilized for feature vector. On the other hand, in their method, word embeddings and lexical resources are combined to get semantic similarity. Also, the usage of WEMB is different. In our method, WEMB is utilized for doing the weighted average calculation for making a sentence vector. In their method, WEMB is utilized for getting similarity, and the similarity becomes a feature vector. The difference of similarity based method and our method is shown in Table 9. In Table 9, each method is using only word embeddings.

Table 10  
Comparison of F1-score against Similarity-based Method only use WEMB

Approach	Vader	STS-Gold	IMDB	PL05
Similarity based Method[26]	<b>88.19</b>	<b>83.75</b>	<b>88.55</b>	<b>76.25</b>
Our Method	84.88	82.71	68.72	69.30

Overall, the similarity-based method has better accuracy than our direct calculation. In Future work, we are modifying our results especially result for movie review based on graph embedding.

Further issues are highlighted in from our experiment as follows.

## 5.1 Unknown Words Challenge

The sentence vector is calculated by word vectors that are gotten from word embeddings. However, some word is not included in word embeddings. Hence the word's vector is not sufficient. In the worst case scenario, words in the sentence could be unknown words. We have investigated the influence of unknown words. We define *KnownRatio(KR)* in as in formula (13).

$$KnownRatio = \frac{|Word \in Sentence \& Word \in WordEmbeddings|}{|Word \in Sentence|} \quad (13)$$

We investigate *KR* for each word embeddings and dataset pairs. The average of *KR* for each pair and Accuracy of our previous method (without weight) and this research (with weight) is as shown in Table 10.

Table 11  
Known Ratio for Dataset and Word Embeddings pairs

Dataset	WEMB	Average of KR	Accuracy (with weight)	Accuracy (without weight) [34]
TSATC	GoogleNews300	<b>0.84</b>	<b>77.7</b>	76.1
	GloveTwitter200	0.76	76.9	76.4
PL05	GoogleNews300	0.72	64.4	67.6
	GloveTwitter200	<b>0.94</b>	61.8	59.6

As a result of considering Table 8, when word importance is considered, if the average of *KR* is high the accuracy is high. On the other hand, when word importance is not considered, if the average of *KR* is high the accuracy is low. This result shows that all word's vectors are not necessarily needed; but instead important word's vectors become important for such computation. We think all words should be known for concluding which ones are important or else. In most cases, the unknown word is caused by mistyping or orthographical variants. In order to solve this problem, the preprocessing algorithm for mistyping or orthographical variants can provide better solution. Also, creating WEMB which has robust vocabulary will be also, other good solution.

## 5.2 Sentence Vector Calculation

In our previous research, all words are utilized with the same weight for making a sentence vector. In this research, the weight of words are changed according to their importance, and it makes better accuracy than the method without word importance [34]. Hence, word importance is considered as effective approach for improving accuracy. Moreover, in this research, the dimension of each word vector is a black box; hence, the dimension of the sentence vector is also a black box. We think Similarity-based method does not have this problem.

## 5.3 Difference between Twitter and Movie Review

In our experiment, the result for Twitter data is sufficient. However, the result for movie review data is insufficient because there is a difference between Twitter and Movie review. In our opinion, movie review data include words titles of the movies. In our proposed word importance calculation, some words of the movie's title are considered as important. Also, if the movie's title is transformed into a vector, it has polarity. Hence, a named entity should be considered to solve this problem.

## Conclusions

In this paper, we proposed word embeddings based Sentence-level Sentiment Analysis method using word weight according to their importance. The word importance is calculated by training data, and it is utilized for sentence vector calculation. In our experiment, we confirm that word importance makes a better effect on accuracy. However, there is still an accuracy difference between our method and state of the art [26]. Our method has a fair result for Twitter data, but it has a problem with movie review data. The weighted average of word vectors is insufficient for sentence vector.

There is much future work for improving accuracy and doing an extension.

We consider future work for improving accuracy as follows:

- A preprocessing algorithm for fixing mistyping or orthographical variants is required to solve an unknown word problem.
- Create WEMB that have much vocabulary to solve an unknown word problem.
- We need many experiments for finding the best calculation for sentence vector.

Our thought is as follows:

- The meaning of the word should be considered using a named entity. Also, using the relational graph structure of the language [21, 27] will be suitable for meanings.
- The similarity-based method outperforms our method. Hence, the usage of WEMB should be reconsidered.

We consider the following extension as future work:

- In this paper, only binary classification for positive and negative is considered. However, neutral polarity and polarity intensity should be considered too.
- This method should be applied to other SA problems such as irony, sarcasm detection [13] or emotion detection [6, 29].
- Our method is a supervised approach. Hence, it requires many training data. However, labeling data is a difficult/hard task. The classification algorithm that works well with small training data is required. We think Semi-supervised learning [19, 30] will solve this problem.

## References

- [1] A Esuli, F Sebastiani, "Determining the semantic orientation of terms through gloss classification," Proceedings of the 14<sup>th</sup> ACM International Conference on Information and Knowledge Management, pp. 617-624, October 2005
- [2] A Esuli, F. Sebastiani, "SentiWordNet – A Publicly Available Lexical Resource for Opinion Mining." In Proceedings of the 5<sup>th</sup> Conference on Language Resources and Evaluation, pp. 417-422, May 2006

- [3] B Liu, "Sentiment Analysis and Opinion Mining." Morgan and Claypool Publishers: Synthesis Lectures on Human Language Technologies, 1<sup>st</sup> edition, USA, May 2012
- [4] B Pang, L Lee, "Opinion mining and sentiment analysis," *Foundation and Trends in Information Retrieval*, Volume 2, pp. 1-135, January 2008
- [5] C Lin, W Ammar, C Dyer, L Levin. "Unsupervised POS Induction with word embeddings." In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1311-1316, May 2015
- [6] E Cambria, A Livingstone, A Hussain, "The Hourglass of Emotions," *Cognitive behavioural systems*, pp. 144-157, January 2012
- [7] E Cambria, C Havasi, A Hussain, "SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis", *Proceedings of FLAIRS*, pp. 202-207, May 2012
- [8] E Cambria, D Olsher, D Rajagopal, "SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis", *Proceedings of AAIL*, pp. 1515-1521, July 2014
- [9] E Cambria, R Speer, C Havasi, A Hussain, "SenticNet: A Publicly Available Semantic Resource for Opinion Mining," *Proceedings of Commonsense Knowledge Symposium*, pp. 14-18, November 2010
- [10] E Cambria, S Poria, A Gelbukh, M Thelwall. "Sentiment analysis is a big suitcase." *IEEE Intelligent Systems* 32, pp. 74-80, November 2017
- [11] E Cambria, S Poria, D Hazarika, K Kwok. "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings". In: *AAAI*, pp. 1795-1802, February 2018
- [12] E Cambria, S Poria, R Bajpai, B Schuller, "SenticNet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives", *Proceedings of COLING*, pp. 2666-2677, December 2016
- [13] E Sulis, DIH Farías, P Rosso, V Patti, G Ruffo, "Figurative messages and affect in Twitter," *Knowledge-Based Systems*, Volume .108, pp. 132-143, September 2016
- [14] J Goikoetxea, A Soroa, E Agirre, "Bilingual embeddings with random walks over multilingual wordnets," *Knowledge-Based Systems*, Volume 150, pp. 218-230, June 2018
- [15] J Lin, W Mao, D D. Zeng, "Personality-based refinement for sentiment classification in microblog," *Knowledge-Based Systems*, Volume 132, pp. 204-214, September 2017

- [16] J Pennington, R Socher, C Manning. "Glove: Global vectors for word representation." In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) October 2014
- [17] JPP Duran, G Sidorov, HG Adorno, I Batyrshin, EM Melendez, GP Duran, LC Hernandez, Algorithm for Extraction of Subtrees of a Sentence Dependency Parse Tre, Acta Polytechnica Hungarica, Volume 14(3), pp. 79-98, March 2017
- [18] K Ravi, V Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," Knowledge-Based Systems, Volume 89, pp. 14-46, November 2015
- [19] L Oneto, F Bisio, E Cambria, D Anguita. "Semi-supervised learning for affective common-sense reasoning." Cognitive Computation, Volume 9, pp. 1842, February 2017
- [20] M Bansal, K Gimpel, K Livescu. "Tailoring continuous word representations for dependency parsing." In: ACL (2), pp. 809-815, June 2014
- [21] M J Cobo. "A Relational Database Model for Science Mapping Analysis," Acta Polytechnica Hungarica, Volume 12, pp. 43-62, June 2015
- [22] M Soleymani, D Garcia, B Jou, B Schuller, SF Chang, M Pantic, "A survey of multimodal sentiment analysis," Image and Vision Computing 65, pp. 3-14, September 2017
- [23] O Appel, F Chiclana, J Carter, H Fujita "Successes and challenges in developing a hybrid approach to sentiment analysis," Applied Intelligence Volume 48, pp. 1176-1188, July 2018
- [24] O Appel, F Chiclana, J Carter, H Fujita, "A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level," Knowledge-Based Systems, Volume 108, pp. 110-124, September 2016
- [25] O Appel, F Chiclana, J Carter, "Main Concepts, State of the Art and Future Research Questions in Sentiment Analysis," Acta Polytechnica Hungarica, Volume 12(3), pp. 87-108, March 2015
- [26] O Araque, G Zhu, Carlos A. Iglesias, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," Knowledge-Based Systems, Volume 165, pp. 346-359, February 2019
- [27] P Goyal, E Ferrara, "Graph embedding techniques, applications, and performance: A survey," Knowledge-Based Systems, Volume 151, pp. 78-94, July 2018
- [28] R Socher, A Perelygin, J Wu, J Chuang, C Manning, A Ng, C Potts. "Recursive deep models for semantic compositionality over a sentiment treebank." In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), pp. 1631-1642, October 2013

- [29] S Poria, A Gelbukh, E Cambria, A Hussain, GB Huang, EmoSenticSpace: “A Novel Framework for Affective Common-Sense Reasoning,” Knowledge-Based Systems, Volume 69, pp. 108-123, October 2014
- [30] S Poria, E Cambria, A Gelbukh, “Aspect extraction for opinion mining with a deep convolutional neural network.” Knowledge-Based Systems 108: pp. 42-49, September 2016
- [31] S Poria, E Cambria, G Winterstein, GB Huang, “Sentic patterns: Dependency-based rules for concept-level sentiment analysis.” Knowledge-Based Systems, Volume 69, pp. 45-63, October 2014
- [32] SL Lo, E Cambria, R Chiong, D Cornforth. “A multilingual semi-supervised approach in deriving Singlish sentic patterns for polarity detection.” Knowledge-Based Systems 105, pp. 236-247, August 2016
- [33] T Chen, C. Guestrin. (2016) “XGBoost: A Scalable Tree Boosting System.” In Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16) ACM, New York, NY, USA, pp. 785-794, August 2016
- [34] T Hayashi, H Fujita, “Sentence-level Sentiment Analysis using feature vectors from word embeddings,” In Proceedings of THE 17<sup>th</sup> International Conference on Intelligent Software Methodologies, Tools and Techniques (SOMET 18) pp. 749-758, September 2018
- [35] T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean, “Distributed representations of words and phrases and their compositionality,” Advances in neural information processing systems, Volume 26, pp. 3111-3119, December 2013
- [36] T Mikolov., K Chen, G Corrado, J Dean, “Efficient estimation of word representations in vector space.” CoRR, arXiv:1301.3781; January 2013
- [37] X Fu, X Sun, H Wu, L Cui, J Z Huang, “Weakly supervised topic sentiment joint model with word embeddings,” Knowledge-Based Systems, Volume 147, pp. 43-54, May 2018
- [38] X Zou, J Yang, J Zhang, H Han, Microblog sentiment analysis with weak dependency connections, Knowledge-Based Systems, Volume 142, pp. 170-180, February 2018
- [39] Y Li, Q Pan, T Yang, SH Wang, JL Tang, E Cambria. Learning word representations for sentiment analysis. Cognitive Computation, Volume 9, pp. 843-851, June 2017
- [40] Y Xie, Z Chen, K Zhang, Y Cheng, D K Honbo, A Agrawal, AN Choudhary. MuSES: a multilingual sentiment elicitation system for Social Media Data. IEEE Intelligent Systems, Volume 29, pp. 34-42, January 2014