# Web-derived Emotional Word Detection in Social Media Using Latent Semantic Information

Chiyu Cai*†, Linjing Li*, *Member, IEEE*, and Daniel Zeng‡*†, *Fellow, IEEE*
* The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 10090, China
† School of Computer and Control Engineering, University of Chinese Academy of Sciences, China
‡ Department of Management Information Systems, University of Arizona, Tucson, AZ 85721, USA
Email: {caichiyu2014, linjing.li, dajun.zeng}@ia.ac.cn

***Abstract*—Public sentiment permeated through social media is usually regarded as an important measure for public opinion monitoring, policy making, and so forth. However, the deluge of user-generated content in web, especially in social platform, causes great challenge to public sentiment analysis tasks. Therefore, Web-derived Emotional Word Detection (WEWD) is proposed as a fundamental tool aims to alleviate this problem. Most previous works on WEWD focus on rules, syntax, and sentence structures, a few utilize semantic information which has the potential to further increase the accuracy and efficiency of WEWD. In this paper, we propose a Global-Local Latent Semantic (GLLS) framework for WEWD to make a full use of latent semantic information with the help of multiple sense word embedding technology. We devise two computational WEWD models, called Ensemble GLLS (EGLLS) and Deep GLLS (DGLLS). EGLLS exploits an ensemble learning way to fuse the global and local latent semantics while DGLLS takes advantage of deep neural network. We also design an old-new corpus enrich technique to help increase the effectiveness of the overall training and detecting process. To the best of our knowledge, this is the first work which applies multiple sense word embedding and deep neural network in WEWD related tasks. Experiments on real datasets demonstrate the effectiveness of the proposed idea and methods.**

## I. Introduction

As the advent of the era of Web 2.0, netizens have become active content providers. Social media further accelerate this great shift, and equip their users with tools and digital resources to help them generate creative and vivid content. To understand user-generated content is an important part for public sentiment analysis and public opinion monitoring. Among the varies kind of user-generated content, newly created polarity words are the most common one. This kind of words is derived solely from web, and cannot be found in existing dictionary, they are called web-derived emotional word [1]. These web-derived emotional words usually are not only meaningful but also very interesting which include new idioms, abbreviations, hieroglyphs, homophones, mixed languages and so on. For example, the wide spread "orz" derived from Japan means frustrated or worship. It's a hieroglyphic which represents a kneeling, bowing, or comically fallen over person. Taking the word "MambaDay" as another example, it is created to celebrate Kobe Bryant, a famous NBA player. It denotes the day of Kobe's final game and has been widely used as hashtag in Twitter.

With a large volume of web-derived emotional words being used in social platforms and their high diffusion speed, many Natural Language Processing (NLP) tasks like sentiment analysis face huge challenge. Many web-derived words are not listed in existing vocabularies or sentiment lexicons and thus cannot be used directly in analysis, but they actually have a significant impact on the results. The neglect of web-derived emotional words may bring harmful effects. How to effectively and efficiently detect web-derived emotional words or automatically expand sentiment lexicon has been regarded as a highly demanded, valuable, and challenge research problem.

Many models and algorithms have already been proposed to address problems encountered in the field of WEWD. One kind of algorithms use rules and patterns that observed from linguistics [2]. However, the generalization capacity and scalability of these algorithms are poor. Some algorithms utilize well developed NLP tools to integrate syntactic information [3]. A large part of researchers regard WEWD as a classification or clustering problem and resolve it with various machine learning algorithms [4], [5], [6]. Machine learning based algorithms possess good generalization capacity. However, a notable deficiency of those algorithms is that they do not take semantic information into account. In addition, polysemy problem is always ignored in some algorithms which causes some emotional words be neglected as well. For a specific word, semantic information is one of the most important properties which determines how and where to apply it. Making a good use of semantic characteristics and structures can help extract web-derived emotional words a lot. In this paper, we strive to shed some light on this challenge problem.

In this paper, we propose a Global-Local Latent Semantic (GLLS) framework for WEWD. The unique characteristics of the Global Latent semantic Space (GLS, which captures contextual and structure information) and the Local Latent semantic Space (LLS, which captures detailed and comprehensive information) are also explored in the context of WEWD. We then develop two novel models, called Ensemble GLLS (EGLLS) and Deep GLLS (DGLLS) to address the challenge of fusing global and local semantic information through ensemble learning and deep neural network respectively. We also design an old-new corpus enrich technique to help increase the effectiveness of the overall training and detecting process. We validate our finding and models by using a real dataset which consists of 8 million tweets. To the best of our knowledge, this is the first work which applies multiple sense word embedding and deep neural network in WEWD related tasks.

## II. Related Work

### A. Web-derived Emotional Word Detection

Existed algorithms for WEWD can be divided into three categories. Rules and patterns based algorithms are the most primitive way which heavily depend on language, grammar and domain. Wang et al. took advantage of the foregoing degree adverbs rule and other patterns through bootstrapping [7]. The second kind of algorithms always utilizes grammar characteristics, syntax structures, and relations between different types of words. Qiu et al. exploited the relations between emotional words and topics or product features and proposed a novel propagation approach based on dependency trees [8]. The dependency expansion method utilized the dependency between emotional words and degree adverbs both within sentence and among sentences [3]. A considerable part of researches focus on the third kind of algorithms, they regard WEWD as a classification or clustering problem. Hamouda et al. proposed a machine learning based sentiment lexicon extraction algorithm [4]. Fu et al. further considered the imbalance of emotional words and improved the method of Hamouda et al. Velikovich et al. proposed an algorithm based on the graph built from co-occurrence statistics from the entire corpus [1].

### B. Distributed Representations of Words

Distributed representations of words also known as word embedding. Comparing with traditional one-hot representation, word embedding represents words by dense real-valued vectors and reserves more latent semantic information. This property has been experimentally validated by a variety of tests on neural probabilistic language model [9]. Besides the classical Neural Network Language Model (NNLM), log-linear models were presented in 2013, including Continuous Bag-Of-Words (CBOW) model and Skip-gram model [11].

Taking into consideration that many words have more than one sense (or meaning), the concept of multiple sense word embedding was presented [12]. In this concept, one word is corresponded to one global word embedding and multiple local word embeddings, where each local embedding represents one sense of the original word. Neelakantan et al. extended the Skip-gram model and proposed the Multiple Sense Skip-Gram (MSSG) model to address the polysemy problem [13]. In MSSG model, there are three significant hyperparameters which affect the embedding training process and effectiveness a lot, including context window size, dimension of word embedding space, and maximal number of meanings per word.

## III. Problem Statement

We define the problem of web-derived emotional word detection as: *Giving a corpus $\mathcal{C} = \{t_1, t_2, \cdots, t_{|\mathcal{C}|}\}$ and vocabulary resource $\mathcal{V} = \{w_1, w_2, \cdots, w_{|\mathcal{V}|}\}$, where $|\cdot|$ denotes the cardinal. Let $\mathcal{P} = \{negative, neutral, positive\}$ be the sentiment polarity set, and $\mathcal{L} = \{(l_1, p_1), (l_2, p_2), \cdots, (l_{|\mathcal{L}|}, p_{|\mathcal{L}|})\}$ be sentiment lexicon resource, where $l_i \in \mathcal{V}$ is called a emotional (or sentiment) word, its sentiment polarity $p_i \in \mathcal{P}$. We require that a sentiment word has only one unique polarity, i.e., $i = j$, if $l_i = l_j$ holds. Web-derived emotional words are defined as words appear in corpus $\mathcal{C}$ frequently, but cannot be found in the given vocabulary $\mathcal{V}$ and $\mathcal{L}$. WEWD aims to find web-derived words $U = \{u_1, u_2, \cdots, u_{|U|}\}$ from corpus $\mathcal{C}$, and judge the emotional tendency $p_i$ of each web-derived word $u_i$ based on its context and existing sentiment resource.*

## IV. Global-Local Latent Semantic Framework

Given a corpus $\mathcal{C}$, we first extract web-derived word candidates. Then we construct latent semantic spaces to capture both semantic information and syntax information. Finally, we develop computational models and detect web-derived emotional words based on latent semantic information. This is a general and flexible framework. Each constituent of this framework can employ the most suitable method in order to match specific requirements, such as language, size of corpus, and etc. For example, if the corpus $\mathcal{C}$ is in Thai, then the method proposed by Sornlertlamvanich et al. can be employed to recognize web-derived Thai words [14]. Figure 1 summarizes the overall procedure of operations in the framework.

In general, there exist a variety number of ways to construct the latent semantic space. In this paper, we adopt MSSG which can perform word sense discrimination and embedding learning by non-parametrically estimating the number of senses per word type. Two kinds of latent semantic spaces (word embedding spaces) will be provided after training, called global latent semantic space (GLS) and local latent semantic space (LLS), respectively. In addition, we adopt linear chain CRF to extract web-derived words candidate set which has the advantage of integrating multiple features in a uniform framework without concerning their interactions. Main features included $n$-gram, character position, mutual information, left entropy, right entropy, part-of-speech, and etc.

### A. Challenge of Fusing GLS and LLS

No matter GLS or LLS, they both can be used to detect web-derived emotional words independently. However, either GLS or LLS based WEWD only utilizes the semantic information captured by one space, how to make full use of all latent semantic information reserved in both GLS and LLS is still an open question in the field of WEWD. GLS describes context information and each sense embedding in LLS describes one specific meaning of a word. These two kinds of spaces are obtained by using different criteria in the training stage, and thus cannot be merged together directly.

**The Role of GLS.** The contextual information in GLS is adopted to help detect web-derived emotional words. Noted that emotional words may be used in identical or similar context, because they may formally be used to modify similar objects or may follow similar syntactic rules. Take "ugly" and "optimistic" as an illustration, though these two words express

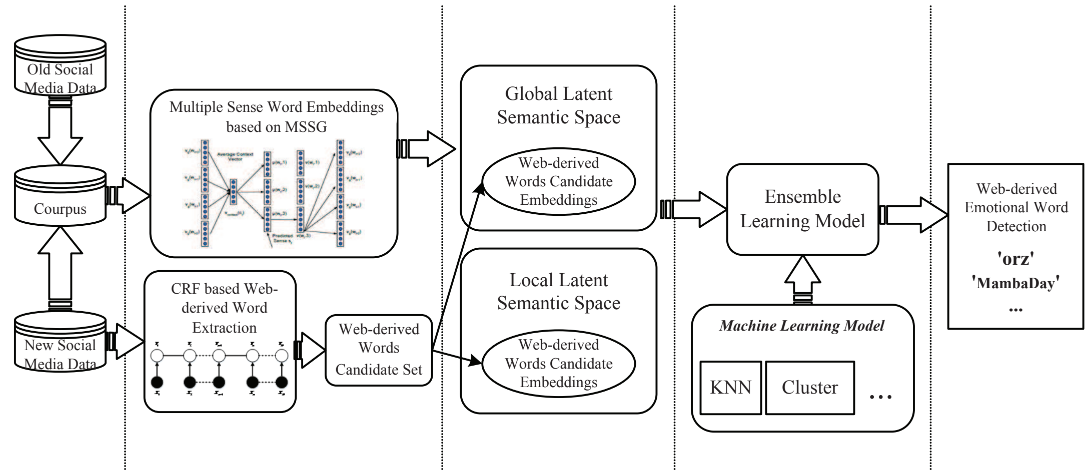


Fig. 1: Ensemble Global-Local Latent Semantic Model for WEWD

different senses, they can be used in an identical context, such as "She is an ugly girl" and "She is an optimistic girl". Hence, comparing with non-emotional words, emotional words may be used in similar syntactic positions or contexts and exhibit a more contextual and syntax structure. This makes emotional words be able to discriminate themselves from other non-emotional words.

**The Role of LLS.** LLS can help solve the problem of polysemy and provide more detailed and comprehensive semantic information. In LLS, a word is associated with multiple sense embeddings, where each sense embedding represents a specific semantic information. Thus, the similarity based on sense embeddings is more reliable than that based on global embeddings.

**Fusing GLS and LLS.** According to the above explanation, we observed that the strengths of GLS and LLS are different and complementary. Based on this, we propose the following hypothesis that *fusing global and local information may achieve a better performance in WDED*. We have also conducted experiments to further validate our hypothesis which are presented in Section VI. In the literature, to our best knowledge, there does exist a mechanism on how to combine these two kinds of embeddings. In that mechanism, a global embedding of the word is regarded as a selector to choose one final sense embedding of it from the set of all its sense embeddings. Through this method, the specific sense corresponding to current context can be selected. This method has been successfully applied in machine translation and named entity extraction [15]. However, in the WEWD related task, consider a word $w$, it is not important which sense is $w$'s actual meaning in a given situation, thus the polarity of $w$ should be judged after considering all possible meanings (senses) of $w$. So the above mechanism of how to fuse global and local (sense) embeddings is not suitable for WEWD.

## V. Ensemble GLLS and Deep GLLS

Based on the above analysis, we propose two automatic methods for fusing global and local semantic information and detecting web-derived emotional words. The first, an Ensemble GLLS (EGLLS) model, uses an ensemble learning way. The second approach relies on deep learning called Deep GLLS (DGLLS). To our best knowledge, this is the first time that GLS and LLS are combined through these manners in the literature.

### A. Ensemble Global-Local Latent Semantic Model

In this subsection, we adopt Nearest Neighbor as the simple method in each semantic space to build the Ensemble Global-Local Latent Semantic model, which is shown in Figure 1. Consider a known emotional word $e \in \mathcal{L}$ and a candidate web-derived emotional word $u$, the global embeddings and the $i$-th sense embeddings of them are denoted as $v_g(e)$, $v_l(e,i)$ and $v_g(u)$, $v_l(u,i)$, respectively. The maximal similarities of $u$ with respect to dictionary $\mathcal{L}$ when computed in GLS is

$$max\_sim_g(u) = \max_{e \in \mathcal{L}} sim(v_g(e), v_g(u)), \tag{1}$$

where $sim(\cdot)$ denotes cosine similarity. While the maximal similarity between $e$ and $u$ in the LLS can be computed as follows:

$$\begin{aligned} i^*, j^* &= \underset{1 \le i,j \le K}{\operatorname{argmax}}\, sim(v_l(e,i), v_l(u,j)), \\ max\_sim_s(e,u) &= sim(v_l(e,i^*), v_l(u,j^*)). \end{aligned} \tag{2}$$

Then, the maximal similarities of $u$ with respect to dictionary $\mathcal{L}$ when computed in LLS is

$$max\_sim_l(u) = \max_{e \in \mathcal{L}} max\_sim_s(e,u). \tag{3}$$

With a trained MSSG model, we now can construct three resulting sets of web-derived emotional words, the first, $E_g$, from the adoption of GLS, the second, $E_l$, from adoption of LLS, and the third, $E_c$, from the combination of GLS and LLS. To make the best use of these three sets, we then conduct an ensemble learning on them to get the final result. This is a highly flexible framework as the machine learning algorithms employed in constructing $E_g$, $E_l$, and $E_c$ can be adapted to the specific applications under consideration.

The construction rule for $E_g$ is,

$$I_g^\delta(u) = \begin{cases} 0, max\_sim_g(u) < \delta \\ 1, max\_sim_g(u) \geq \delta \end{cases}, \quad (4)$$

where $I_g^\delta(\cdot)$ is the characteristic function of $E_g$ with confidence threshold $\delta$, $I_g^\delta(u) = 1$ denotes that the confidence of the web-derived word $u$ is high in GLS and can be regarded as an emotional word candidate.

The construction rule for $E_l$ is,

$$I_l^\varepsilon(u) = \begin{cases} 0, max\_sim_l(u) < \varepsilon \\ 1, max\_sim_l(u) \geq \varepsilon \end{cases}, \quad (5)$$

where $I_l^\varepsilon(\cdot)$ is the characteristic function of $E_l$ with confidence threshold $\varepsilon$, $I_l^\varepsilon(u) = 1$ denotes that the confidence of the web-derived word $u$ is high in LLS and can be regarded as an emotional word candidate.

For $E_c$, GLS based method will be adopted to extract web-derived words as the candidates, then LLS based method will be adopted to judge the emotional tendency of all candidates. We build the model by methods employed for GLS and LLS, the construction rule for $E_c$ is $I_c^\gamma = I_l^\varepsilon \circ I_g^\gamma$, which is the compound of $I_l^\varepsilon$ and $I_g^\gamma$, where $\gamma$ is used to balance the interaction of the two spaces. In fact, it is a candidate threshold which is used to select candidate words in GLS. Finally, an ensemble way is used to merge the above three sets. For parameters $\gamma$, $\delta$ and $\varepsilon$, we select them using $k$-fold cross-validation and take generalization ability into account at the same time.

### *B. Deep Global-Local Latent Semantic Model*

In this subsection, we develop a deep neural network model to fuse global and local semantic information. The architecture of our Deep Global-Local Latent Semantic model for WEWD is shown in Figure 2, the code was implemented using theano[1] library. Our network is composed of a convolutional layer followed by a non-linearity, max pooling, a hidden layer and a softmax classification layer. Where, the ellipsis represents the web-derived word extraction and the multiple sense word embedding training which are the same as those in our EGLLS model. Based on the local embeddings, a sense matrix is built and we then feed it into a Convolutional Neural Network (CNN). CNN provides a higher level semantic feature embedding for all local information, we then merge this high-level feature embedding with the global embedding in the join layer to complete the fusion of global and local information. In the following, we give a brief explanation of the main components of our network architecture.

**Input layer**. The input to the CNN are words where each treated as a sequence of senses: $[s_1, \cdots, s_K]$, where $K$ denotes the number of senses. More formally, let $E \in \mathbb{R}^{d \times K|V|}$ be a trained local word embedding matrix, where each column is a $d$ dimensional vector represents a sense of a specific word from the vocabulary $V$. By selecting the columns of

[1]http://www.deeplearning.net/software/theano/

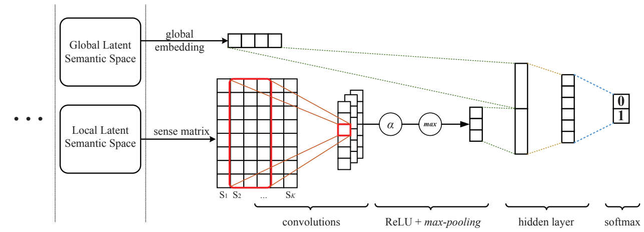


Fig. 2: Deep Global-Local Latent Semantic Model for WEWD

$E$ corresponding to a word, we can construct the following sense matrix:

$$\mathbf{S} = \begin{bmatrix} | & | & | \\ e_1 & \cdots & e_K \\ | & | & | \end{bmatrix}. \quad (6)$$

Each sense matrix $\mathbf{S}$ corresponds to a word, and each column of $\mathbf{S}$ represents one sense of this word.

**Convolutional network**. The convolutional layer is composed of a set of filters $\mathbf{F} \in \mathbb{R}^{d \times m}$, where $m$ is the width of the filter. The convolution operation maps the input matrix $\mathbf{S} \in \mathbb{R}^{d \times K}$ into a vector $\mathbf{c} \in \mathbb{R}^{1 \times (K+m-1)}$ by applying a convolutional filter $\mathbf{F}$ as follows:

$$\mathbf{c}_i = (\mathbf{S} * \mathbf{F})_i = \sum_{k,j} (\mathbf{S}_{[:,i-m+1:i]} \odot \mathbf{F})_{kj}, \quad (7)$$

where $\odot$ is the element-wise multiplication and $\mathbf{S}_{[:,i-m+1:i]}$ is a matrix slice of size $m$ along the columns. Intuitively, the convolution filter, which slides along the column dimension of the input matrix, performs an element-wise product between a column slice of matrix $\mathbf{S}$ and a filter $\mathbf{F}$ and then sums the resulting vector to produce one element of the so-called feature map $\mathbf{c} \in \mathbb{R}^{1 \times (K+m-1)}$. A set of predefined filters form a filter bank $\mathcal{F} \in \mathbb{R}^{n \times d \times m}$, these filters are convolved sequentially with $\mathbf{S}$ to produce a feature map matrix $\mathbf{C} \in \mathbb{R}^{n \times (K+m-1)}$. We chose ReLU activation function for non-linearity in our model, which speeds up the training. The output of the convolutional layer is then passed to the pooling layer which aims to aggregate the information and reduce the dimensions of representation. We transform the resulting feature map into a scalar using *max-pooling*, i.e., we extract the largest value in the map. The final pooled representation is denotes as $\mathbf{c}_{local} \in \mathbb{R}^{1 \times n}$.

**Join layer**. In this layer, we just concatenate the global embedding $v_g$ and local feature $\mathbf{c}_{local}$ into a single vector $\mathbf{c}_{join}$:

$$\mathbf{c}_{join} = [v_g \oplus \mathbf{c}_{local}], \quad (8)$$

where $\oplus$ denotes concatenation.

**Hidden layer**. The joined vector is then passed through a fully connected hidden layer, which aims to capture the relationships between the global and the local representations. Finally, the output of the hidden layer is fed to the softmax layer, which computes the probability distribution over the labels.

**Training**. We train the above model to minimize the categorical cross-entropy through Adadelta, an adaptive learning

TABLE I: Summary of datasets

| Dateset | #Tweets | Year |
|---|---|---|
| **Old dataset** | 6 million | 2013 |
| **New dataset** | 2 million | 2014 |

rate method [16]. The number of mini-batchs is set as 64 for optimization. The gradients are computed by back propogation and the parameters of CNN are trained through stochastic gradient descent algorithm.

## VI. Experiment

### A. Datasets and Evaluation

We collected data from SinaWeibo[2] (a Chinese social media similar to Twitter), in order to build high quality datasets for WEWD related tasks, we adopted the following old-new corpus enrich method. First, we collected about 2 million tweets posted in 2014 and called it new corpus. Then, we crawled about 6 million tweets posted in 2013 and built the old corpus. In the experiments, we only extracted web-derived words from the new corpus, but trained multiple sense word embedding model with all data, as shown in Figure 1. We assumed that the old corpus does not contain timely web-derived words since the evolution of social media is with high speed. This method can reduce the negative influence of frequent irregular string in web-derived words extraction phase and guarantee the converging of training word embedding model. The details of the datasets are summarized in Table I. In the following analysis, the standard evaluation dataset is provided by COAE2014[3] and the original sentiment lexicon is an extended version based on DUTIR-EmotionWords[4]. The widely adopted **precision**, **recall rate** and **F1 score** are adopted as criteria to evaluate different methods.

### B. Results and Discussion

To validate our hypothesis that fusing global and local information may achieve a better performance, we conducted a series of tests using only one latent semantic space, i.e., GLS or LLS. We employed two machine learning algorithms: nearest neighbor algorithm and SVM, the corresponding models are named as N-GLS, SVM-GLS and N-LLS, SVM-LLS. In these tests, the parameter $\theta(S, K, d)$ of MSSG model is randomly set as $(5, 3, 100)$.

Table II summarizes the results for the above models and our EGLLS model, it illustrates that EGLLS model obtains a better result than models which only use one latent semantic space. This empirical results partly demonstrate the positive effect of fusing global and local information, and validate our hypothesis that the proposed GLLS framework can make a more effective use of the semantic information captured by both GLS and LLS to achieve a better performance.

[2]www.weibo.com

[3]The 6th Chinese Opinion Analysis Evaluation

[4]http://ir.dlut.edu.cn/EmotionOntologyDownload

TABLE II: Performance of GLS, LLS, EGLLS

| Methods | $P(\%)$ | $R(\%)$ | $F1$ |
|---|---|---|---|
| **N-GLS** | 37.25 | 56.82 | 45.00 |
| **N-LLS** | 37.89 | 56.12 | 45.24 |
| **SVM-GLS** | 38.63 | 57.81 | 46.31 |
| **SVM-LLS** | 40.38 | 56.99 | 47.27 |
| **EGLLS** | **49.27** | 52.37 | **50.77** |

TABLE III: Performance of EGLLS

| $\theta(S, K, d)$ | $P(\%)$ | $R(\%)$ | $F1$ |
|---|---|---|---|
| (5, 2, 50) | 41.20 | 50.68 | 45.45 |
| (5, 2, 100) | 41.40 | 51.21 | 45.79 |
| (5, 2, 150) | 42.02 | 50.79 | 45.99 |
| (5, 2, 200) | 41.48 | 51.59 | 45.99 |
| (5, 3, 50) | 48.91 | 52.44 | 50.61 |
| (5, 3, 100) | 49.27 | 52.37 | 50.77 |
| (5, 3, 150) | 49.59 | 51.97 | 50.75 |
| (5, 3, 200) | 49.19 | 52.63 | 50.85 |
| (5, 4, 50) | 65.41 | 53.32 | 58.75 |
| (5, 4, 100) | 65.73 | 53.97 | 59.27 |
| (5, 4, 150) | 65.88 | 54.32 | 59.54 |
| (5, 4, 200) | 66.20 | 54.84 | 59.99 |
| (5, 5, 200) | 67.31 | 55.04 | 60.56 |
| (5, 6, 200) | 67.73 | 55.17 | **60.81** |

Next, considering the influence of parameter $\theta(S, K, d)$ in the MSSG model. Table III depicts the performance of EGLLS with different parameter configurations. We can observe that parameter $K$ is a main parameter which affects the final performance most, as the change of parameter $K$ could cause different final results while the performances are near the same with respected to different values of the parameter $d$. To further investigate the role of $K$, we fixed the value of $(S, d)$ as $(5, 200)$ and tuned the value of $K$. Table III shows that there exists a positive correlation between the performance and the value of $K$ when it is small. However, when the value of $K$ exceeds a threshold (depends on $S$ and $d$), the effect of further increasing $K$ becomes insignificant.

For the DGLLS, we set the parameter of word embedding model as $\theta(S, K, d) = (5, 6, 200)$. The CNN network is investigated by varying the number of filters and the filter widths, set to 128, 256 and 2, 3, respectively. For the hidden layer, the number of hidden units is set as 256. Table IV illustrates that incrementing the filter width of the CNN layer boosts the performance by a small margin. We also trained our model with bigger filter size and larger width, but achieved no improvement. In contrast, enormous training time is a passive obstacle. Compared to Table III, DGLLS which selects the appropriate hyperparameters performs better than EGLLS.

We also conducted experiments to compare the performance of our methods with some baseline methods. These baseline methods are derived from related work [3], [4], [5], [7]

TABLE IV: Performance of DGLLS

| CNN Filter Size | CNN Filter Width | $P(\%)$ | $R(\%)$ | $F1$ |
|---|---|---|---|---|
| 128 | 2 | 68.87 | 55.43 | 61.42 |
| 128 | 3 | 69.30 | 55.68 | 61.75 |
| 256 | 2 | 71.62 | 56.15 | 62.95 |
| 256 | 3 | 72.28 | 56.95 | **63.71** |

TABLE V: Performance comparisons

| Methods | $P(\%)$ | $R(\%)$ | $F1$ |
|---|---|---|---|
| Dependency [3] | 58.32 | 55.07 | 56.65 |
| MLBSL [4] | 65.90 | 42.23 | 51.47 |
| OC-SVM [5] | 45.15 | 43.84 | 44.49 |
| Bootstrapping [7] | 59.65 | 53.21 | 56.25 |
| W2V+SVM | 45.01 | 59.54 | 51.27 |
| EGLLS | 67.73 | 55.17 | **60.81** |
| DGLLS | 72.28 | 56.95 | **63.71** |

which have already been introduced in previous sections. Table V describes the experimental results of the GLLS based WEWD and the baseline methods as well. Our GLLS based approach achieves the highest F1 score $63.71\%$ and precision $72.28\%$. Compared with the existed methods, our DGLLS can provide at least a $7.06\%$ improvement in F1 score. This result demonstrates that the proposed EGLLS and DGLLS are effective methods in web-derived emotional word detection.

Finally, we investigated the role of word embedding by substituting the multiple sense word embedding model as Skip-gram model. We employed the famous Word2Vec [11] tool in this experiment and adopted SVM as the computational model, all other parameters are kept the same as previous. The experiment result, denotes as W2V+SVM, is shown in Table V (the third line from the bottom). The F1 score of W2V+SVM is $51.27\%$, which is lower than the value of Dependency [3] and Bootstrapping [7]. This comparison indicates that improvement of the proposed GLLS framework is not directly depend on word embeddings, the fusing of global and local semantic information contributes the most.

## VII. Conclusion

In this paper, we introduced multiple sense word embedding in the field of WEWD and proposed a Global-Local Latent Semantic solution framework. To address the challenge of fusing global and local information, we first analyzed the roles of GLS and LLS, then we developed two models, called Ensemble GLLS and Deep GLLS. We conducted extensive experiments and results of them show that (i) fusing global and local semantic information can provide a positive effect on WEWD; (ii) GLLS framework can use semantic information captured by both GLS and LLS more effectively; (iii) EGLLS and DGLLS can significantly improve the detection performance with respect to most of the existing methods.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 71202169, 71272236, U1435221, 71602184, 71621002, 71672067, 61671450, the Key Research Program of the Chinese Academy of Sciences under Grant No. ZDRW-XH-2017-3, as well as the Early Career Development Award of SKLMCCS.

## References

[1] L. Velikovich, S. Blair-Goldensohn, K. Hannan, and R. McDonald, "The viability of web-derived polarity lexicons," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 777–785.

[2] M. Thelen and E. Riloff, "A bootstrapping method for learning semantic lexicons using extraction pattern contexts," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 214–221.

[3] J. Liang, J. Tan, X. Zhou, P. Liu, L. Guo, and S. Bai, "Dependency expansion model for sentiment lexicon extraction," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03*. IEEE Computer Society, 2013, pp. 62–65.

[4] A. Hamouda, M. Marei, and M. Rohaim, "Building machine learning based senti-word lexicon for sentiment analysis," *Journal of Advances in Information Technology*, vol. 2, no. 4, pp. 199–203, 2011.

[5] F. Lina, X. He, and J. Donghong, "Method of new sentiment word discovery based on oc-svm," *Application Research of Computers*, vol. 7, p. 007, 2015.

[6] L. Sang, X. Xu, J. Zhang, and D. Huang, "New microblog sentiment lexicon judgment based on generalized jaccard coefficient," *Journal of Shandong University*, 2015.

[7] C. Wang and F. Wang, "A bootstrapping method for extracting sentiment words using degree adverb patterns," in *Computer Science & Service System (CSSS), 2012 International Conference on*. IEEE, 2012, pp. 2173–2176.

[8] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Computational Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[12] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 873–882.

[13] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," *arXiv preprint arXiv:1504.06654*, 2015.

[14] V. Sornlertlamvanich, T. Potipiti, and T. Charoenporn, "Automatic corpus-based thai word extraction with the c4. 5 learning algorithm," in *Proceedings of the 18th Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics, 2000, pp. 802–807.

[15] Y. Wu, J. Xu, M. Jiang, Y. Zhang, and H. Xu, "A study of neural word embeddings for named entity recognition in clinical text," in *AMIA Annual Symposium Proceedings*, vol. 2015. American Medical Informatics Association, 2015, p. 1326.

[16] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.