Neural Network-based Vector Representation of Documents for Reader-Emotion Categorization

Yu-Lun Hsieh*†, Shih-Hung Liu*, Yung-Chun Chang*‡, Wen-Lian Hsu*
*Institute of Information Science, Academia Sinica, Taipei, Taiwan
†Department of Computer Science, National Chengchi University, Taipei, Taiwan
‡Department of Information Management, National Taiwan University, Taipei, Taiwan
Email: {morphe,journey,changyc,hsu}@iis.sinica.edu.tw

***Abstract*—In this paper, we propose a novel approach for reader-emotion categorization using word embedding learned from neural networks and an SVM classifier. The primary objective of such word embedding methods involves learning continuous distributed vector representations of words through neural networks. It can capture semantic context and syntactic cues, and subsequently be used to infer similarity measures among words, sentences, and even documents. Various methods of combining the word embeddings are tested for their performances on reader-emotion categorization of a Chinese news corpus. Results demonstrate that the proposed method, when compared to several other approaches, can achieve comparable or even better performances.**

***Keywords*-neural network; word embedding; document representation; reader emotion;**

I. Introduction

With the rapid growth of the Internet, sharing information has become exceedingly easy. It is very common nowadays for everyone to instantly share on social media websites one's experiences and emotions regarding virtually anything [1], [2]. By means of modern computational technologies, we can quickly collect and classify data about human emotions for further research. Studies on emotion classification aims to predict the emotion categories (e.g., *happy* or *angry*) of the given text [3], [4]. Moreover, business entities have realized the potential of the crowd on opinions toward their products and services. Therefore, emotion classification has been attracting more and more attention, e.g., [5], [6].

The emotion contained in texts can be roughly divided into two aspects, i.e, writer's and reader's emotions. In short, writer's emotion is the emotion expressed by the author of an article, while reader's emotion is the reader's response after reading it. The writer may directly express her feelings through some emotional words or *emoticons*. However, the reader's emotion can be invoked by not only the received content but also her personal experiences or knowledge [7]. Thus, the stimulus of reader-emotion can be more complicated, which makes recognizing reader-emotion a more challenging task than writer-emotion [8], [9]. For instance, the news title "*Dozens killed after military plane crashes in Indonesian city*" is likely to trigger some *angry* or *worried* emotions in its readers, despite the fact that it is a description of an event which contains no emotional words.

In the recent years, we witness an increasing interest in vector space representations for words and documents through neural network or deep learning models, e.g., [10]–[14]. Thus, we propose a novel approach that uses word embedding learned from neural networks and an SVM classifier to categorize reader-emotions in news articles. We hope to utilize the power of deep learning to capture hidden connections between the words on the surface and the potential invocation of human emotions. Experiments demonstrate that this method can achieve a comparable or even better performance than other well-known text or emotion categorization methods.

II. Previous Work

Past work about emotion detection were mainly focused on the writer's emotion. Several researches considered emoticons from weblogs as categories for text classification. In their studies, emoticons were taken as moods or emotion tags, and textual keywords were taken as features. For example, [15] used emoticons in newsgroup articles to extract instances relevant for training polarity classifiers. Other methods use emoticons as tags to train support vector machines at the document or sentence level [16], [17].

On the other hand, several work were mainly concerned with the textual information only. [18] classified movie reviews into positive and negative emotions. [19] proposed a sentence level emotion recognition method using dialogs as their corpus, in which "Happy", "Unhappy", or "Neutral" was assigned to each sentence as its emotion category.

However, writers and readers do not always share the same emotions regarding the same text. [20] tried to automatically annotated reader emotions on a writer emotion corpus with a reader emotion classifier, and studied the interactions between writers and readers with the writer-reader emotion corpus. Unfortunately, the readers' emotion has not attracted much attention. Classifying emotion from the reader's perspective is a challenging task, and research on this topic is relatively sparse as compared to those considering the writers' point of view. Due to the recent

978-1-4673-6656-4/15 $31.00 © 2015 IEEE
DOI 10.1109/IRI.2015.90

increase in the popularity of Web 2.0, news websites, such as Yahoo! Kimo News, incorporated the technology that allows readers to express their emotions toward news articles through voting. Thus, [7] attempts to classify Yahoo! News articles into 8 emotion classes from the readers' perspectives.

III. Method

We propose a novel usage of word and document embedding for emotion classification. One of the well-known studies on developing word embedding methods was presented in [10]. It estimated a statistical language model, formalized as a feed-forward neural network, for predicting future words in the context while inducing word embeddings (or representations) as a by-product. Such an attempt has already motivated many follow-up extensions to develop similar methods for learning latent semantic and syntactic regularities in various NLP applications. Representative methods include the continuous bag-of-word (CBOW) model and the skip-gram (SG) model [21]. They have been proven to be successful in many tasks including and beyond NLP. However, there is little work done to utilize these methods for use in Chinese text categorization tasks. We will briefly introduce these models in the following sections.

A. Continuous Bag-of-word (CBOW) Model

The concept of *CBOW* is motivated by the distributional hypothesis [22], which states that words with similar meanings often occur in similar contexts and thus suggests to look for word representations that capture their context distributions. Rather than seeking to learn a statistical language model, the *CBOW* model tries to obtain a dense vector representation (embedding) of each word directly [21]. The structure of *CBOW* is similar to a feed-forward neural network, with the exclusion of the non-linear hidden layer. In this way, the model can still retain good performances and be trained on much more data efficiently while getting around the heavy computational burden incurred by the non-linear hidden layer. Formally, given a sequence of words $w_1, w_2, \cdots, w_T$, the objective function of *CBOW* is to maximize the log-probability expressed in (1):

$$\sum_{t=1}^{T} log\hat{P}(w_t|w_{t-c}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+c}), \quad (1)$$

where c is the window size of the training context for the central word w_t, and T denotes the length of the training corpus. The conditional probability P in Eq. (1) is defined by:

$$\hat{P}(w_t|w_{t-c}^{t+c}) = \frac{e^{\mathbf{v}_{\bar{w}_t} \cdot \mathbf{v}_{w_t}}}{\sum_{i=1}^{V} e^{\mathbf{v}_{\bar{w}_t} \cdot \mathbf{v}_{w_i}}}, \quad (2)$$

where $\mathbf{v}_{w_i}$ denotes the vector representation of the word w at position t; V indicates the size of the vocabulary; and $\mathbf{v}_{\bar{w}_t}$ denotes the (weighted) average of the vector representations of the context words of w_t [21], [23], which can be further expressed in the form:

$$\mathbf{v}_{\bar{w}_t} = \sum_{j=-c, j\neq 0}^{c} \alpha_j \cdot \mathbf{v}_{w_{t+j}}, \quad (3)$$

where α_j is a weighting factor associated with the distance between the central word w_t and the context word w_{t+j}.

B. Skip-gram (SG) Model

In contrast to the *CBOW* model, the *SG* model employs an inverse training objective for learning word representations with a simplified feed-forward neural network [14], [21], [24]. Formally, given a sequence of words, $w_1, w_2, \cdots, w_T$, the objective function of *SG* is to maximize the following log-probability:

$$\sum_{t=1}^{T} \sum_{j=-c, j\neq 0}^{c} log\hat{P}(w_{t+j}|w_t), \quad (4)$$

where c is the window size of the training context for the central word w_t; and the conditional probability can be calculated by:

$$\hat{P}(w_{t+j}|w_t) = \frac{e^{\mathbf{v}_{w_{t+j}} \cdot \mathbf{v}_{w_t}}}{\sum_{i=1}^{V} e^{\mathbf{v}_{w_i} \cdot \mathbf{v}_{w_t}}}, \quad (5)$$

where $\mathbf{v}_{w_{t+j}}$ and $\mathbf{v}_{w_t}$ denote the word representations of words at position $t+j$ and t, respectively.

In addition, improvements to the training procedure have been proposed to increase speed and effectiveness. They include the hierarchical soft-max algorithm (*HS*) and the negative sampling algorithm (*NS*) [11], [24], [25].

Both methods adopt a sequential training process for learning the parameters (word representations), so the trained model may be drastically affected by the order of the training samples. Therefore, randomization and multiple iterations of the corpus are often utilized when training these models.

C. Vector Representation for Documents

Recently, neural-network-based approaches for learning vector representations for documents have been proposed [14]. Originally, the word vectors are used to predict the next word in the sentence. This idea can be extended for document vectors in a similar manner. If every document is mapped to a unique vector, which can be thought of as a special word, we can use it to predict other words in the same document. During training, the word vectors will be learned first. Then, a sliding window over the whole document is used to sample every word. Eventually, we can obtain a vector for each document that contains information of embeddings in the whole document. More specifically, document vectors (and word vectors) are learned with a stochastic gradient descent obtained via back-propagation.

For each iteration, the vector for a document is fed through the neural network, then we can compute the error gradient from the network and use the gradient to update the document vector.

Moreover, document vectors have some advantages over traditional bag-of-words models. First, since they are based on word vectors, the semantics of the words can also be incorporated. Second, they can include information from a much broader context, i.e., the whole document. Such feature usually requires a very large n in n-gram models, hence a heavy toll on the memory. Lastly, since the document vectors are learned from sequentially feeding the word vectors into the network, the ordering of the words can also be considered.

IV. Experiments

A. Dataset and Setting

We collected a corpus of Chinese news articles from an online news site[1], in which each article is voted from readers with emotion tags in eight categories: *angry*, *worried*, *boring*, *happy*, *odd*, *depressing*, *warm*, and *informative*. We consider the voted emotions as the reader's emotion toward the news. Following previous studies that used a similar source, we exclude "*informative*" as it is not considered as an emotion category [7], [8]. In this evaluation, we only consider coarse-grained emotion categories (i.e., *positive* and *negative*). Thus, fine-grained emotions like *happy*, *warm*, and *odd* are merged into '*positive*', while *angry*, *boring*, *depressing*, and *worried* are merged into '*negative*'. To ensure the quality of our evaluation, only articles with a clear statistical distinction between the highest vote of emotion and others determined by *t*-test with a 95% confidence level are retained. Finally, 27,000 articles are kept, and divided into the training set and the test set, each containing 10,000 and 17,000 articles, respectively. Each set contains roughly the same amount of *positive* and *negative* articles. For evaluation metrics, we adopt the convention of using accuracy measures as in [7].

At the outset, word vectors are trained using *CBOW* and *SG* models with negative sampling (*NS*) and hierarchical soft-max (*HS*). These word vectors are subsequently used to train document vectors. Afterwards, they are used as representations for the documents and sent to support vector machines (SVM) [26] in order to classify the reader-emotion of the document, denoted as *DV-SVM*. We first experiment with various settings for the dimensionality of the vector, and the best settings are compared with other methods described below.

Several other text representation models are also implemented. First, a baseline system that uses Naïve Bayes for classification, is denoted as *NB* [27]. In addition, we evaluate a probabilistic graphical model called LDA for representation of a document [28] (denoted as *LDA*)[2]. Next, since it has been proved that keywords are very effective in text classification tasks [30], an emotion keyword-based model is also compared (denoted as *KW*). Both *LDA* and *KW* are trained using SVM classifiers. Lastly, *CF* denotes a state-of-the-art reader-emotion recognition method that combines various features including bi-grams, words, meta-data, and emotion category words [7].

Table I
Accuracies (%) of DV-SVM using different models and vector dimensionality.

Dimensionality	***Model***	
	CBOW	SG
10	76.69	75.98
50	83.94	80.48
100	85.97	81.81
150	86.67	82.63
300	**87.37**	**85.47**
400	84.62	83.38

B. Results & Discussion

Table I is a comparison of accuracies by using different dimensionality of the vector to train *DV-SVM*. By using a document vector with just 10 dimensions, *DV-SVM* can achieve a substantial accuracy of over 75% for both models. For the most part, the increase in performance is positively related to dimensionality. We also observed that the difference between the two models, *CBOW* and *SG*, is not very obvious.

Interestingly, it shows that increasing the dimensionality of the vector does not promise to improve the performance. We believe that the drop in accuracy when using higher dimension vectors may be attributed to the over-fitting effect. In the end, we can achieve the best performance for both *CBOW* and *SG* models using a vector dimensionality of 300. In particular, the *CBOW* model reaches a slightly better accuracy of 87.37% than *SG*'s 85.47%. However, it has been reported that the *SG* model is more efficient in other tasks [21]. We conclude that the optimal model and dimensionality for vectors may be closely related to the amount of training samples and the classification complexity.

Table II shows a comprehensive evaluation of *DV-SVM* and other methods. The baseline method *NB* can only obtain a low accuracy of 52.78%. It indicates that using only surface word weightings and ignore inter-word relations can not lead to a satisfactory result. In contrast, the *LDA* model greatly outperforms *NB* with an overall accuracy of 74.16%.

[1] https://tw.news.yahoo.com

[2] The dictionary required by all comparing methods is constructed by removing stop words in [29], and retaining tokens that make up 90% of the accumulated frequency. For unseen events, we use the Laplace smoothing in *NB*. We use a toolkit at http://nlp.stanford.edu/software/tmt/tmt-0.4/ to implement *LDA*.

Table II
COMPARISON OF THE ACCURACIES OF READER-EMOTION CLASSIFICATION SYSTEMS.

Methods	*Accuracy(%)*
NB	52.78
LDA	74.16
KW	80.81
CF	85.70
DV-SVM$_{\text{CBOW300}}$	**87.37**

The ability of including both local and long-distance word relations may be the reason for its success.

Notably, the *KW* model shows substantial effectiveness in categorizing the emotions. It indicates that reader-emotion can largely be recognized by using only the weighting of keywords. As mentioned in Section I, we presume that this is due to the strong relation between specific persons or events and emotions. It is conceivable that articles containing tragic events or notorious people can cause readers to have negative feelings, while joyous events lead to positive emotions.

Meanwhile, the integration of complicated lexical feature sets (e.g., character bi-grams, word dictionary, and emotion keywords) in *CF* allows it to reach a satisfactory overall accuracy around 86%. It suggests that, in order to capture more profound emotions hidden in the text, one has to consider not only surface words, but also the relations and semantics within it. By representing word-word relations through bi-grams and dictionaries, this system can obtain a better result.

On the other hand, document vectors learned in *DV-SVM* can surpass other methods and achieve the best outcome. It demonstrates that document embedding learned from neural networks can successfully encode the complex relations between words in an article into a dense vector. These embeddings can supply substantial discriminating power to a vector-based classifier like *SVM*. Moreover, such an approach has the advantage of requiring very little supervision and feature engineering. It automatically learns the importance or weights of various words inherently.

V. CONCLUSION

In this paper, we present a novel approach for reader-emotion classification using document embedding as features for the SVM classifier. We first investigate the effect of dimensionality on representing a document with a vector, and found that higher dimension does not always guarantee better performance. The choice of the modeling scheme and parameters could be an empirical one. Then, we demonstrate that using document embedding for reader-emotion classification can yield substantial success. In the future, more work can be done on designing different algorithms to combine word vectors that can lead to a better representation of a document. Also, exploring the possibilities of projecting semantic knowledge-base onto the vector space is another interesting line of research. Lastly, we hope to extend this approach to other NLP applications.

ACKNOWLEDGMENT

We are grateful to the anonymous reviewers for their insightful comments. This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 103-3111-Y-001-027 and 103-2221-E-002-106-MY2.

REFERENCES

[1] P. D. Turney, “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 417–424.

[2] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis,” *Computational Linguistics*, vol. 35, no. 3, pp. 399–433, 2009.

[3] C. Quan and F. Ren, “Construction of a blog emotion corpus for chinese emotional expression analysis,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 3, 2009, pp. 1446–1454.

[4] D. Das and S. Bandyopadhyay, “Word to sentence level emotion tagging for bengali blogs,” in *Proceedings of the ACL-IJCNLP 2009 Conference*, 2009, pp. 149–152.

[5] Y. Chen, S. Y. M. Lee, S. Li, and C.-R. Huang, “Emotion cause detection with linguistic constructions,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 179–187.

[6] M. Purver and S. Battersby, “Experimenting with distant supervision for emotion classification,” in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 482–491.

[7] K. H.-Y. Lin, C. Yang, and H.-H. Chen, “What emotions do news articles trigger in their readers?” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 733–734.

[8] ——, “Emotion classification of online news articles from the reader’s perspective,” in *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1, 2008, pp. 220–226.

[9] Y.-j. Tang and H.-H. Chen, “Mining sentiment words from microblogs for predicting writer-reader emotion transition.” in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, 2012, pp. 1226–1229.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[11] F. Morin and Y. Bengio, “Hierarchical probabilistic neural network language model,” in *Proceedings of the international workshop on artificial intelligence and statistics*, 2005, pp. 246–252.

[12] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[14] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.

[15] J. Read, “Using emoticons to reduce dependency in machine learning techniques for sentiment classification,” in *Proceedings of the ACL Student Research Workshop*. Association for Computational Linguistics, 2005, pp. 43–48.

[16] G. Mishne, “Experiments with mood classification in blog posts,” in *Proceedings of the 1st Workshop on Stylistic Analysis Of Text For Information Access (Style 2005)*, 2005. [Online]. Available: http://staff.science.uva.nl/~gilad/pubs/style2005-blogmoods.pdf

[17] C. Yang and H.-H. Chen, “A study of emotion classification using blog articles,” in *Proceedings of Conference on Computational Linguistics and Speech Processing*, 2006.

[18] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[19] C.-H. Wu, Z.-J. Chuang, and Y.-C. Lin, “Emotion recognition from text using semantic labels and separable mixture models,” vol. 5, no. 2, pp. 165–183, 2006. [Online]. Available: http://doi.acm.org/10.1145/1165255.1165259

[20] C. Yang, K. H.-Y. Lin, and H.-H. Chen, “Writer meets reader: Emotion analysis of social media from both the writer’s and reader’s perspectives,” in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*. IEEE Computer Society, 2009, pp. 287–290.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proceedings of Workshop at ICLR*, 2013.

[22] G. A. Miller and W. G. Charles, “Contextual correlates of semantic similarity,” *Language and cognitive processes*, vol. 6, no. 1, pp. 1–28, 1991.

[23] L. Qiu, Y. Cao, Z. Nie, Y. Yu, and Y. Rui, “Learning word representation considering proximity and ambiguity,” in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[24] 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.

[25] A. Mnih and K. Kavukcuoglu, “Learning word embeddings efficiently with noise-contrastive estimation,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2265–2273.

[26] C.-C. Chang and C.-J. Lin, “LIBSVM : a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[27] A. McCallum, K. Nigam *et al.*, “A comparison of event models for naive bayes text classification,” in *AAAI-98 workshop on learning for text categorization*, vol. 752, 1998, pp. 41–48.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[29] F. Zou, F. L. Wang, X. Deng, S. Han, and L. S. Wang, “Automatic construction of chinese stop word list,” in *Proceedings of the 5th WSEAS International Conference on Applied Computer Science*, 2006, pp. 1010–1015.

[30] Y.-C. Chang, Y.-L. Hsieh, C.-C. Chen, and W.-L. Hsu, “A semantic frame-based intelligent agent for topic detection,” *Soft Computing*, pp. 1–11, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00500-015-1695-4