Learning Short Text Representation using Non-Negative Matrix Factorization and Word Semantic Correlations

Luepol Pipanmaekaporn[1] Suwatchai Kamonsantiroj[2] and Earn Suriyachay[3]
Department of Computer and Information Science
King Mongkut University of Technology North Bangkok, Bangkok, Thailand, 10800
E-mail: {luepol.p[1], suwatchai.k[2], earn.s[3]}@sci.kmutnb.ac.th

***Abstract*— With the emergence of online medias, short texts have been increasingly available. Because the short texts have a limited contextual information, they are sparse, noisy and ambiguous. Conventional models to text representation are hence not applicable. In this paper, we propose a representation learning model for short texts to tackle this challenge. Our proposed model is based on Non-Negative Matrix Factorization (NNMF). Due to the problem lacking word co-occurrences information, it effectively learns the representation model by incorporating the semantic relationship between words generated by unsupervised learning methods. Finally, the NNMF-based model is effectively solved by using a gradient descent-based algorithm. Experimental results on two benchmark data sets, including sentiment140 and 20newsgroups, demonstrate that our proposed approach achieved the improvement of classification accuracy to state-of-the-art topic discovery techniques.**

Keywords*— *Short text; Non-Negative Matrix Factorization; Topic Modeling

I. INTRODUCTION

The prevalent successes of social-network and e-commerce systems daily generate tremendous amounts of short texts, such as tweets, comments, product reviews and headlines. Discovering knowledge from the short texts is valuable and has gained a lot of attention. However, short texts typically contain a few words and they are noisy and ambiguous [1][3]. For example, the longest messages in twitter are limited to 140 characters and Microsoft's Window Messenger allowing the longest message of 400 characters. Too limited words in short text always pose a great challenge to effectively represent their content and discover knowledge from them, such as text classification [4][6][12], sentiment analysis [2] and text clustering [3][16].

Traditionally, topic modelling has been widely used to discover hidden topics in a collection of short documents with topic models. Generally, there are two groups of topic models. The first scheme relies on generative probabilistic models, such as Latent Dirichlet Allocation (LDA) [5]. The LDA basically learns topics from word co-occurring together in documents and then generates a topic per document model and words per topic model, named as Dirichlet distributions. The second scheme focuses on matrix factorization models, such as Latent Semantic Indexing (LSI) [6] and Non-Negative Matrix Factorization (NNMF) [8]. Among such methods, the NNMF-based models have shown encouraging performance for topic discovery in short texts [9].

Although the conventional topic models have achieved for text modelling, in many cases they are not well applicable for short text collections. This is because a short text only contains a few words, resulting in the difficulty to capture the word co-occurrence information. For many years, many efforts have been done to overcome this challenge. Among such efforts, a popular scheme is to aggregate short texts into the longer ones, known as pseudo-documents to improve the word co-occurrence information [10]. However, unrelated short texts aggregated into the pseudo-documents may cause misleading results. Some studies rely on deriving external knowledge to overcome the problem of word ambiguity. For example, the study in [3] has proposed to apply transfer learning for clustering short text messages by incorporating auxiliary long texts and topic model. In [4], the authors learn universal topics from Wikipedia's articles to infer document's topics for building a short text classifier.

Another strategy focuses on discovering the semantic relationships between words for short text modelling. Several efforts have been also made to discover topics for short texts from resources, such as frequent itemsets extracted from Wikipedia's articles [13] and the word embeddings based on GoogleNews [14]. However, many differences between external documents and the short text may introduce the noise to the discovered topics. Instead, some studies target on discovering the internal semantic relationship between words. For example, using word-pairs co-occurring in a short text have demonstrated to be effective for discovering topics in short texts.

Motivated by these works, we propose a novel NNMF-based model for short text modelling. The key idea is to learn topics of short texts by incorporating semantic relationship between words captured by unsupervised learning algorithms. The word-word matrix is first computed by using word embedding as an input for learning the latent feature matrix of words in NNMF. In this work, we use a skip-gram model with the negative sampling [15] for the word embedding. To obtain the latent feature matrix, autoencoder [18], an unsupervised neural network, is trained by optimizing our objective function. By

978-1-7281-2627-2/19/$31.00 ©2019 IEEE
DOI 10.1109/IIAI-AAI.2019.00130

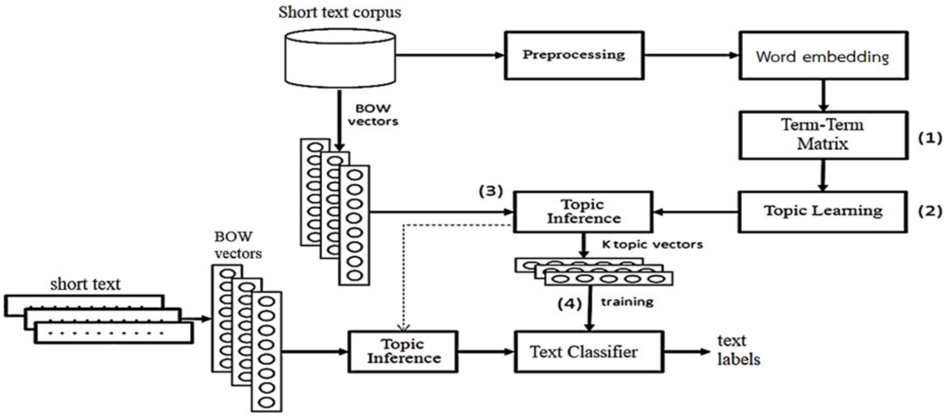

Fig. 1. Our proposed method

incorporating the learned feature matrix, a gradient descent-based algorithm is used to effectively solve the NNMF-based model. We conduct experiments on two common tasks, including sentiment analysis and news title classification, using two real-world short text datasets. The experimental results demonstrated that our proposed model is effective for short text modelling. The main contributions of our work are summarized as follows:

- We propose a novel NNMF-based model that incorporates the semantic relationships between words captured by unsupervised neural networks.
- Most existing topic models directly learn latent factor matrix of words from the word-document matrix that is always sparse and ambiguous in short texts. Our proposed method solves this issue by using the word-word matrix generated by word embedding. Compared to the word-document matrix, the word-word matrix is always less sparsity, but more meaningful information for topic modeling.
- We conduct experiments on real-world short text datasets and compare with the other state-of-the-art methods for short text modelling.

II. Preliminaries

In this section, we will first provide some preliminaries for our proposed method.

A. Non-Negative Matrix Factorization (NNMF)

NNMF [8] is one of the most popular matrix factorization algorithms. NNMF basically decomposes a data matrix into two low-rank non-negative factor matrices. Given X is a word-document matrix of a short-text corpus. In NNMF, the word-document matrix can be approximated by two low-rank matrices $Z \in \mathbb{R}^{M\times K}$ and $U \in \mathbb{R}^{N\times K}$, where $K \ll \min(M,N)$. NNMF is usually formulated as follows:

$$\min_{Z,U\geq 0} \|X - ZU\|_F^2 \tag{1}$$

where Z indicates the latent factor matrix of words and U is the latent factor matrix of documents. In the case of short texts, NNMF may perform the poor performance since the limited length of short text cannot capture document-level word co-occurrences. Our objective is to learn the latent factor matrices for NNMF by capturing the word semantic relatedness.

B. Word embedding with Skip-gram model

Word embedding has been demonstrated to be an effective tool in capturing semantic relationships of the words. One of the most successful word embedding methods is Word2Vec [17]. Word2Vec uses a two-layer neural network to learn dense vectors of words from raw texts. Word2vec includes two training models, namely skip-gram and CBOW continuous Bag-of-Words (CBOW). Skip-gram learns the probability of context words given a word w_t by minimizing the loss function: $E = -\log p(w_{t-j}, \ldots, w_{t+j}|w_t)$ where j is a window size. CBOW learns to predict a word w_t given the context words. In this study, the skip-gram model is used to capture the semantic relationships between words for generating the word-word matrix because experiments have shown its effective use for sparse data compared to CBOW [15]. Figure 2 illustrates the skip-gram architecture.

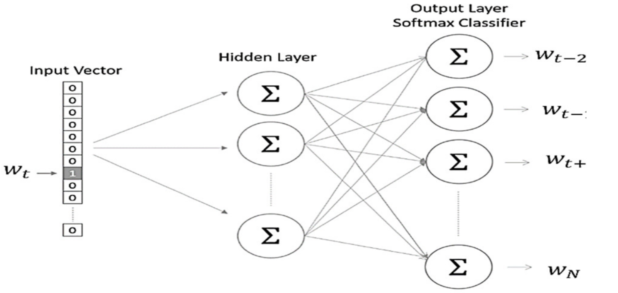

Fig. 2. The structure of skip-gram model

III. Proposed Method

As shown in Figure 1, the proposed method for short text modelling. We summarize it as follows:

(1) Word-Word Matrix: Short texts in a corpus are first preprocessed as the appropriate input for word embedding (i.e., Skip-gram model with negative sampling). By using the skip-gram model, dense vectors of each word in the vocabulary are automatically generated. This allows us to easily construct the word-word matrix S by applying the common similarity measures.

(**2**) **Topic Learning:** Once the word-word matrix was generated. We estimate the word-topic matrix Z using autoencoder. We also formulate an objective function for training the network to obtain the topics.

(3) Topic Inference: In this step, we estimate the document-topic matrix U using a gradient descent-based algorithm.

(**4**) **Text Classifier**: In training process, we build a text classifier based on short texts in the training corpus. In this study, we use Support Vector Machine (SVM) with topic features as input features to classify web short texts associated with pre-defined labels.

A. The Semantic Correlation Matrix

We first process each short text in a collection by removing stop words and stemming. We then segment the preprocessed texts into a collection of sentences as input for training a skip-gram model with negative sampling. The output of the word embedding model is a set of d-dimensional vectors of word (i.e., vocabulary). After representing each word by semantic vector, we compute the correlation between any two words.

$$sim(w_i, w_j) = \frac{C(w_i) \cdot C(w_j)}{|C(w_i)||C(w_j)|} \qquad (2)$$

where $C(w_i)$ indicates the context vector of word w_i derived by using the skip-gram model and $|C(w_i)| = d$ is the length of the word vector w_i. We finally compute word correlation matrix S.

B. The Word-Topic Matrix Estimation

We here formulate the problem of estimating the word-topic matrix Z by using the objective function: $\|S - ZZ^T\|^2$ where S is the word-word matrix. To find the word-feature matrix, autoencoder is trained by using gradient descent algorithm. Generally, the autoencoder learns the output $\hat{X} = \{\hat{x}_1, \hat{x}_2, .., \hat{x}_n\}$ that reconstructs the input $X = \{x_1, x_2, .., x_n\}$ with a hidden $Z = \{z_1, z_2, .., z_k\}$ where $0 < k < n$. The value of hidden layer node z_i is defined as

$$z_i = b_i^{(1)} + \sum_{j=1}^{n} w_{ij}^{(1)} x_j \qquad (3)$$

where $b_i^{(1)}$ is the bias of hidden node i and $w_{ij}^{(1)}$ represents the weight between input node j and hidden node i. The output layer $\hat{X}$ is built by using the activations of the hidden layer as input, bias $b^{(2)}$ and weights $W^{(2)} = \left\{w_{11}^{(2)}, w_{12}^{(2)}, \dots, w_{kn}^{(2)}\right\}$:

$$\hat{x}_i = f\left(\sum_{j=1}^{k} w_{ij}^{(2)} a_j\right) \qquad (4)$$

where $f(z)$ is an activation function and $a_j = f(z_i)$ is the output activation of hidden node j. The autoencoder learns a function that minimizes the difference between the output vectors $\hat{X}$ and input vectors X. Figure 3 shows the structure of the neural network. By training the autoencoder network, we learn topical structure of words from information of word-word matrix. We apply a soft-sign activation function: $f(z) = z/(1 + |z|)$ to the nodes comprising the neural network. Let $S^{(i)} = \langle s_1^i, s_2^i, \dots, s_n^i \rangle$ be a feature vector of word i and $\hat{S}^{(i)} = \langle \hat{s}_1^i, \hat{s}_2^i, \dots, \hat{s}_n^i \rangle$ be output vector for the word predicted by the neural network. The following objective function is used to train the autoencoder network:

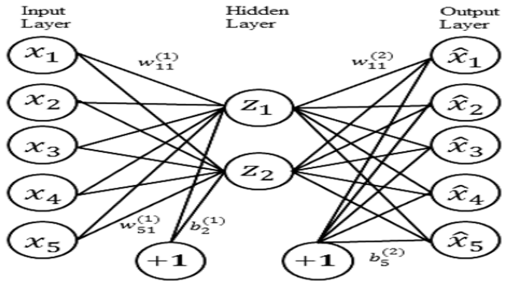

Fig. 3. The autoencoder structure

$$J(W,B) = \underset{W,B}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left\| \hat{S}^{(i)} - S^{(i)} \right\|^2 + \frac{\lambda}{2} \|W\|_2^2 \right\} \qquad (5)$$

where $W\ and\ B$ indicate weights and biases of the training network and λ is a regularization parameter.

The cost function is basically an average sum-of-square error of the differences between input and output vectors of the neural network. We impose a regularization term to the cost function. By imposing the regularization constraint, the magnitude of the weights can be decreased, leading to the generalization improvement. The weight decay parameter λ that controls the relative importance of the regularization term. The parameter λ is automatically determined by cross validation. After trained the network, we extract k dimensional representation for each word vector $S^{(i)} \in S$

$$Z^{(i)} = \langle z_1^{(i)}, z_2^{(i)}, \dots, z_k^{(i)} \rangle \qquad (6)$$

Intuitively, the k vector space can be interpreted as a topical distribution over the word i. A positive value implies the semantic relatedness between this word and the feature. However, negative values can be meaningless in our context. We also apply the following threshold to eliminate these values.

$$z_j^{(i)} = \begin{cases} z_j^{(i)}, & z_j^{(i)} \geq 0 \\ 0, & otherwise \end{cases} \qquad (7)$$

By using (6) and (7), we formulate the non-negative matrix Z that minimizes the distance between S and Z via training the autoencoder network. In this work, we use MATLAB implementation of autoencoder, published by Stanford [20] for the purpose of estimating the word-topic matrix from word semantic correlations.

C. The Document-Topic Matrix Estimation

The learned matrix Z well captures the latent features of words in the text collection. By using this matrix, the representations of text, i.e. the document-topic matrix U, can be estimated by directly projecting the word-document matrix X into the topic space. However, the data sparsity in short text can be problematic for estimating the matrix U. To overcome this problem, we estimate the matrix U that fits the observed data by following the objective function.

$$D(U) = \min_{U} \|X - ZU\|_F^2 \quad s.t. \quad U \geq 0 \tag{8}$$

As seen in Eq. (8), the document-topic matrix U can be estimated by minimizing the Euclidean distance between the word-document matrix X and the product matrix ZU, given the word-feature matrix Z. A gradient descent (GD) algorithm is used to solve the objective function.

$$\widehat{U} = (Z^T Z)^{-1} Z^T X \tag{9}$$

Algorithm 1 describes the overall procedure of our proposed method.

```
Algorithm 1: THE PROPOSED ALGORITHM
    Input:  Word-document matrix $X$;
            The number of latent factors $k$;
    Output: $Z$, $\widehat{U}$;
1. Compute $S$ by using Eq. (2);
2. Compute $Z$ by using Eq.(5) - Eq. (7);
3. Compute $\widehat{U}$ using Eq. (9);
4. return $Z$, $\widehat{U}$;
```

IV. Experiments

In this section, we conducted experiments to evaluate the effectiveness of our proposed method. We first describe the experimental real-world data on two common tasks in web mining: 1) Sentiment analysis and (2) News title classification. We also compared the proposed method with state-of-the-art methods for text representation, including Latent Dirichlet Allocation (LDA), Non-Negative Matrix Factorization (NNMF) and Biterm Topic Model (BTM).

A. Datasets

We conducted this experiment on the two real-world datasets: 1) *Sentiment140* (http://www.sentiment140.com), a tweet sentiment dataset and 2) *20 newsgroup* dataset (http://www.ai.mit.edu/~jrennie/20Newsgroups). The twitter dataset contains 1,600,000 English tweets collected from Twitter during April, 6, 2009 to June 25, 2009. Each tweet was hand-classified with either positive or negative sentiment. All special characters and emoticons are removed. We select only 4,000 tweets in each sentiment category for the evaluation of effectiveness.

For news title classification, we choose the well-known 20 newsgroups dataset, which contains over 20,000 English posts from 20 different newsgroups. For the short text scenario, we only consider the subject filed of each document as training and test data and discard other information, such as main body.

TABLE I. Statistics of two datasets

Dataset	Sentiment140	20newsgroups
#documents	8,000	20,000
#words	2,514	15,980
doc-length	10.69	7.28
#class	2	20

Both the datasets are preprocessed by using standard text processing tasks, including stop word removal and word stemming, to reduce noises in text.

B. Evaluation Metrics

In our experiments, we evaluate the effectiveness of document classification accuracy by using the classical precision (P) / recall (R) as follows:

$$P = \frac{TP}{TP + FP} \quad and \; R = \frac{TP}{TP + FN} \tag{10}$$

We also use the following F-score:

$$F_1 = \frac{2 \times (P \times R)}{(P + R)} \tag{11}$$

Finally, the quality of the classification is measured by averaged precision, recall and F-score.

TABLE II Command Parameter settings used for Word Embeding

Parameter	Description	Value
-train	Name of input file	"word.txt"
-output	Name of output file	"vector.out"
-cbow	Training model 0:Skip-gram model and 1: CBOW model	0
-size	Dimension of vectors	300
-window	Size of window	5
-negative	Training method 0: hierarchical softmax 1: Negative sampling	1
-sample	Threshold of sampling	5
-threads	Number of running threads	12
-binary	Mode of storage 0: common format and1: binary format	0
-mode	output model 0: binary, 1:text	1

C. Comparison Methods

We compare the performance of our proposed model with the following state-of-the-art methods.

TABLE III. PERFORMANCE COMPARISON OF VARIOUS METHODS ON DOCUMENT CLASSICIATION

	News Title			Sentiment140		
	Precision	Recall	F-score	Precision	Recall	F-Score
LDA	0.592	0.573	0.565	0.724	0.730	0.727
NNMF	0.530	0.547	0.536	0.624	0.681	0.651
BTM	0.639	0.604	0.603	0.732	0.718	0.723
Our proposed method	**0.660**	**0.636**	**0.640**	**0.789**	**0.780**	**0.784**

- **Latent Dirichlet Allocation (LDA)**: LDA [5] is a well-known baseline method for topic modelling, which performs well on the normal-length documents. In this work, we use the Gibb sampling-based LDA implementation in Matlab Topic Modeling toolbox 1.4.
- **Non-Negative Matrix Factorization (NNMF)**: NNMF [8] is an unsupervised method that can perform dimension reduction and clustering simultaneously. In our experiments, the NNMF is implemented in Python with gradient descent algorithm.
- **Biterm Topic Model (BTM)**: BTM [11] learns the topics by directly modelling the generation of word co-occurrence patterns in the short text corpus. In BTM, a biterm is an unordered word pair co-occurred in a short text.

In our experiments, we will conduct short text classification on all the datasets. A five-fold cross validation is used to evaluate the performance of the classification, where each corpus is randomly split into training and testing with 70% and 30% ratio respectively. Then, the documents are classified by LIBLINEAR [21], an implementation of Support Vector Machine (SVM). For LDA and BTM, we set parameters $\alpha = 0.1$ and $\beta = 0.01$ since the weak prior can give a better performance for short texts. In LDA and BTM, Gibbs sampling is run for 2000 iterations. For all the comparison methods, we default number of latent factors (i.e. topics) is set to $K = 60$.

For our proposed method, we use the validation set to find the best value of the regularization parameter λ, varying $\{0.01, 0.02, \ldots, 0.90\}$. We finally reached the best performance on news title dataset ($\lambda = 0.07$) and sentiment140 dataset ($\lambda = 0.04$). Table II illustrates common parameter settings used for word embedding in our experiments.

D. Results

Table III illustrates the classification results of all the methods on the two datasets. As seen in Table III, the best results are achieved by our proposed model on both datasets. This demonstrates that our models are effective in the document classification for short texts.

Compared with the conventional topic models, such as LDA and NNMF, our proposed model has a significant improvement in terms of different classification measures, especially a standard NNMF method. This result demonstrates the effective use of word semantic correlations to improve NNMF for addressing the data sparsity issue in short text. In addition, these results confirm that word embedding can effectively capture the semantic relationships between the words. In Table III, we also observe that the BTM model performs better than the other baseline methods (i.e., LDA and NNMF). The major difference between BTM and LDA is that BTM makes use of the term correlations generated from the short text corpus.

We also compare the proposed model with BTM. As seen in Table III, the proposed model perform better than BTM, which attempts to capture the word correlations in short text. This comparison demonstrates that the word correlations obtained by skip-gram model of each corpus play an important role in capturing high quality semantics, given the performance of BTM is not as good as that of the proposed model. In addition, the results based on the tweet dataset are higher than that of the news title dataset because the number of tweet categories is different.

The proposed method based on NNMF has on an average more than 15% improvements over the standard NNMF with respect to precision, recall and F-score. In summary, the classification results have shown that the proposed method is a superior model for short text modelling.

V. CONCLUSIONS

In this paper, we introduce a NNMF-based model to discover topics for short texts. The proposed model basically focuses on alleviating the sparseness of short text data by using the semantic correlations between words learned from a corpus using word embedding algorithm. We show that the word semantics offers less sparsity and more meaningful for topic learning than word co-occurrences widely used in conventional topic models.

To learn topics from short documents, we first apply word embedding in terms of the skip-gram with negative sampling to a collection of labeled short texts and then compute the word-by-word matrix using cosine similarity measure. We also learn topics from the word correlation matrix using autoencoder, a deep neural network. The document-topic matrix is finally estimated by minimizing an objective function. We conducted experiments on the two classification tasks: tweet sentiment classification and news title classification. Experimental results on two benchmark datasets demonstrated that our method significantly improves the classification accuracy compared to state-of-the-art methods for text modelling.

As a future work, we would explore other tasks in short text such as topic discovery and text clustering [1] [4]. We believe that our proposed method is promising for the tasks of short texts.

References

[1] Tommasel, Antonela, and Daniela Godoy. "Short-text feature construction and selection in social media data: a survey." *Artificial Intelligence Review* 49.3 (2018): 301-338.

[2] Zhang, S., Tang, Y., Lv, X., & Dong, Z. “Movie Short-Text Reviews Sentiment Analysis Based on Multi-Feature Fusion”. In *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, 61- 66, 2018.

[3] Jin, Ou, Nathan N. Liu, Kai Zhao, Yong Yu, and Qiang Yang. "Transferring topical knowledge from auxiliary long texts for short text clustering." In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 775-784, 2011.

[4] Phan, Xuan-Hieu, Le-Minh Nguyen, and Susumu Horiguchi. "Learning to classify short and sparse text & web with hidden topics from large-scale data collections." In *Proceedings of the 17th international conference on World Wide Web*. 2008.

[5] Blei D. M., Ng A. Y. and Jordan M. I. “Latent dirichlet allocation”. *Journal of Machine Learning* Res.3 (2003): 993-1022.

[6] Zelikovitz, Sarah, and H. Hirsh. "Transductive LSI for Short Text Classification Problems." In *FLAIRS conference*, 556-561. 2004.

[7] Jianhua Yin and Jianyong Wang. “A Dirichlet Multinomial Mixture Model-based Approach for Short Text Clustering”. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 233–242, 2014.

[8] Daniel D Lee and H Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. *Nature* 401, 6755 (1999), 788–791.

[9] Chen, Y., Zhang, H., Liu, R., Ye, Z., & Lin, J. Experimental explorations on short text topic mining between LDA and NMF based Schemes. *Knowledge-Based Systems*, *163* (2019), 1-13.

[10] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. “Short and Sparse Text Topic Modeling via Self-aggregation”. In *Proceedings of the 24th International Conference on Artifcial Intelligence (IJCAI’15).* AAAI Press, 2270–2276, 2015.

[11] Yan, Xiaohui, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. "A biterm topic model for short texts." In *Proceedings of the 22nd international conference on World Wide Web*, 1445-1456, 2013.

[12] Yang, Lili, Chunping Li, Qiang Ding, and Li Li. "Combining lexical and semantic features for short text classification." Procedia Computer Science 22 (2013): 78-86.

[13] Man, Yuan. "Feature extension for short text categorization using frequent term sets." *Procedia Computer Science* 31 (2014): 663-670.

[14] Li, C., Wang, H., Zhang, Z., Sun, A., and Ma, Z. “Topic modeling for short texts with auxiliary word embeddings”. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval,* 165-174, 2016.

[15] Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie and Yorick Wilks. "A closer look at skip-gram modelling." In *Proceedings of the 5th international Conference on Language Resources and Evaluation*, 1-4, 2016.

[16] Yan, Xiaohui, Jiafeng Guo, Shenghua Liu, Xue-qi Cheng, and Yanfeng Wang. "Clustering short text using ncut-weighted non-negative matrix factorization." In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2259-2262, 2012.

[17] Mikolov, T., Yih, W. T., & Zweig, G. “Linguistic regularities in continuous space word representations”. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746-751, 2013.

[18] Coates, Adam, Andrew Y. Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning." In *Proceedings of the International conference on artificial intelligence and statistics*, 215-223, 2011.

[19] Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. "A closer look at skip-gram modelling." In *Proceedings of the 5th international Conference on Language Resources and Evaluation* (LREC-2006), 1-4, 2006.

[20] Sparse Autoencoder Matlab code avaialble at http://deeplearning.standford.edu/wiki/index.php/UFLDL_Tutorial.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification Journal of Machine Learning Research 9 (2008), 1871-1874.