DOI: 10.1002/int.22290

RESEARCH ARTICLE

Short-text feature expansion and classification based on nonnegative matrix factorization

Ling Zhang[1] | **Wenchao Jiang**[1] | **Zhiming Zhao**[2]

[1]School of Computers, Guangdong University of Technology, Guangzhou, China

[2]Multiscale Networked System (MNS) Group, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

Correspondence
Wenchao Jiang, School of Computers, Guangdong University of Technology, 510006 Guangzhou, China.
Email: jiangwenchao@gdut.edu.cn

Funding information
Guangdong Provincial Transport Department, Grant/Award Numbers: Tec-2016-02-30, 2018A030313061; Natural Science Foundation of Guangdong Province; Guangdong Science and Technology Plan, Grant/Award Numbers: 2017B010124001, 201902020016, 2019B010139001

Abstract

In this paper, a non-negative matrix factorization feature expansion (NMFFE) approach was proposed to overcome the feature-sparsity issue when expanding features of short-text. First, we took the internal relationships of short texts and words into account when segmenting words from texts and constructing their relationship matrix. Second, we utilized the Dual regularization non-negative matrix tri-factorization (DNMTF) algorithm to obtain the words clustering indicator matrix, which was used to get the feature space by dimensionality reduction methods. Thirdly, words with close relationship were selected out from the feature space and added into the short-text to solve the sparsity issue. The experimental results showed that the accuracy of short text classification of our NMFFE algorithm increased 25.77%, 10.89%, and 1.79% on three data sets: Web snippets, Twitter sports, and AGnews, respectively compared with the Word2Vec algorithm and Char-CNN algorithm. It indicated that the NMFFE algorithm was better than the BOW algorithm and the Char-CNN algorithm in terms of classification accuracy and algorithm robustness.

KEYWORDS
correlation, feature extension, nonnegative matrix factorization, short text classification

1 | INTRODUCTION

Short texts are convenient in human communication and have prevalent on social networks nowadays. Short text classification is one of the challenges due to its natural sparsity, noise words, syntactical structure, and colloquial terminologies.[1] Those topics attracted lots of research attention in the field of short text expansion and classification research.

Due to the imitation of words and low-frequency of terms in short text, the bag-of-words (BOW) representation has limits in analyzing short texts.[2] One possible solution for handling sparsity is to expand short text by appending new features based on semantic information extracted from Web searching, lexical databases, or provided by machine translations,[3] which are called an external resource-based approaches. Web searching[4] based feature extension technologies need to interact frequently with search engines and result in high communication overhead and low efficiency for data analysis. Knowledge bases or lexical databases, such as Wikipedia and HowNet for concept taxonomies[5–7] or topic models[8,9] are used to enrich short text representations. However, these feature extension method has high dependencies on the integrity of external resources, and often time consuming. Moreover, these predefined topics and categories are domain-specialized or language-specific.

Using rules or statistical information hidden in the context of short texts is another kind of approaches to extend features, which are called the self-contained resource approaches.[10–15] Mining hidden information in short texts plays a key role in feature extension. A self-aggregation-based topic model (SATM)[12] has been reported recently, which assumes short texts are sampled from long pseudo-documents, and then topic modeling is conducted by finding "document-ship" for each short text. Sikdar et al.[10] described a deep learning approach to recognize Amharic named entities from a large data set annotated with six different classes, trained on various language-independent features together with word vectors, which were the semantic information taken by an unsupervised learning algorithm, word2vec. The word vectors were merged with a set of specifically developed language-independent features and together fed to the neural network model to predict the classes of the words. Zhang et al.[11] proposed a character-level convolutional network model for short text classification without any knowledge of the syntactic or semantic structures of a language. Nevertheless, these works ignore the relevance of the words in short texts. In the case of limited words, the association between words can be used as additional information to serve as an important basis for feature expansion and solve the problem of sparse features of the short text.

This paper considers two forms of information: inter-type and intra-type relationships between words and short texts. Based on these two kinds of data relations, the feature space is obtained by dimension reduction of word clustering indicator, which is obtained by non-negative matrix tri-factorization.[16] Then, according to the correlation between words, closely related features in the feature space are selected to expand the text feature vector, and this can effectively solve the problem of feature sparseness.

2 | RELATED WORKS

Feature expansion is essential to classify short texts, and it has been mainly focusing on two kinds of approaches by now, Latent Dirichlet Allocation (LDA) topic model[17–19] and Word Embedding.[18,20–26] Xu[18] used LDA for clustering words or documents into "topics," and based on a "topic-word" probability distribution model, the closely-related words were found and

selected out to expand feature space of words. Xia et al. chose the liveness of each user as a feature and modeled it as the weighted value for the user. They improve the precision of topic detection and tracking, by including the user feature into LDA model to expand the feature of short texts.[17] Yu et al.[19] used the Dirichlet Multinomial Mixture (DMM) model as the main framework and extended short texts with the potential feature vector representation of the words by combining the user-LDA topic model, and achieved a good performance as an external extension of short texts. The complexity of probabilistic graphical model hampers the development of LDA, and the computational cost of LDA results in bigger penalty compare with the improvement of this algorithm.

On the other hand, word embedding presents another kind of words representation, converting per word into a continuous vector space with dimensionality reduction.[27,28] Semantic expansion of words is then obtained by clustering of vectors. Recently, research have widely employed deep learning-based approaches for word embedding model. Google developed a Word2Vec tool based on Bengio neural network for word embedding.[14] In fact, Word2Vec predicted words based on their context by using one of two distinct neural models: CBOW[23,26,28,29] and Skip-Gram.[10,17,20,22,24,25,30]

Wang et al. proposed a framework to expand short texts, based on skip-gram model to learn word embeddings from large-scale unstructured text data. By using additive composition over word embeddings from context with variable window width, the representations of multiscale semantic units in short texts were computed.[25] In literature [24], distributed word embeddings were learned by skip-gram algorithm through a neural network architecture, and then they were combined into a sentence representation to predict the semantic relations between short texts. Liang et al.[30] proposed a global and local word embedding-based topic model (GLTM) for short texts. They trained global word embeddings from large external corpus and employed the continuous skip-gram model with negative sampling (SGNS) to obtain local word embeddings. Utilizing both the global and local word embeddings, their method could distill semantic-related information between words which could be further leveraged by Gibbs sampler in the inference process to strengthen semantic coherence of topics.

Xun et al.[29] used Continuous Bag of Words (CBOW) to provide additional semantics for short text corpus and incorporated it into each short document's model to establish a Gaussian topic in the vector space. In addition, a discrete background mode over word types was also added to complement the continuous Gaussian topics model. In literature [26], by using word embedding features, Sang et al. expanded and enriched the words density in the short texts and semantic similarities of short texts were calculated for effective learning. This method combined external sources of word semantic information with the short text structure information. Pascual et al. presented a Contextual Specificity Similarity (CSS) algorithm[28] for document similarity measure, where documents were represented as arrays of their word vectors, and then Inverse Document Frequency (IDF) of the words were added into to define the closeness degree between documents.

Although Word2Vec has an outstanding performance in synonymous words analysis, it still relies on local context so much, lacking of global statistical information of short texts. Accordingly, in 2014, Pennington et al. presented a new model based on the words ice and steam to illustrate how to generate meaning from word occurrence, and how to result a global word vectors representing that meaning.[13] They defined it as GloVe, whose training was performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showed interesting linear substructures of the word vector space.[25] Comparative study[31] showed that its effectives for the Arabic language processing, and pointed out that the

appropriate starting point for word vector learning might be indeed with ratios of co-occurrence probabilities rather than the probabilities themselves. The shortcoming of GloVe was also mentioned in literature [32], demanding a large-scale corpus and big enough storage resource.

Both approaches mentioned above cannot work without huge corpus data support. Opposite to the large-scale learning algorithms, this paper studies on feature expansion by short text itself. There are three aspects of relations taken into consideration, including word-to-word, word-to-text, and text-to-text, to make use of more relatedness information from short text. We use this method as an alternative to the aforementioned relation features, in cases where only limited amounts of training data are available.

3 | ALGORITHM FRAMEWORK

Given a short text set $T = \{t_1, ..., t_m\}$ and a word set $W = \{w_1, ..., w_n\}$. The goal is to group the texts $\{t_1, ..., t_m\}$ into k clusters, in the meantime also grouping the words $\{w_1, ..., w_n\}$ into k clusters. The relationship matrix $\boldsymbol{R}$ describes the inter-type relationships between texts and words. The correlation matrix $\boldsymbol{A}_t$ and A_w represent the intra-type relationships of texts and words, respectively. The clustering indicator matrix $\boldsymbol{F}$ represents the clustering result of words, whose element $\boldsymbol{F}_{ij}$ represents the possibility that w_i belongs to cluster k_j. Similarly, the clustering indicator matrix $\boldsymbol{G}$ represents the clustering result of short texts. Since the short text category label of training set is known, the matrix $\boldsymbol{G}$ can be obtained. In this way, the feature expansion for short texts is transformed into the clustering of texts and words jointly.

The overall framework of our algorithm is based on nonnegative matrix factorization, including four steps: feature space establishment, feature expansion, feature space updating, and short text classification, as shown in Figure 1.

The feature space of the short text itself describes the possibility of the word belonging to the category. Based on training texts, we construct a relationship matrix to describe the membership of word-to-text, and two correlation matrixes to describe intra-type relation of text-to-text and word-to-word, respectively. Under the manifold regularization, the nonnegative matrix factorization algorithm is used to build the words clustering indicator matrix. After removing some evenly distributed features in the indicator matrix, a dimension-reduced feature space is constructed. The feature of the short text is to extend by the correlation between the features in the feature space and the text features. The updating of feature space is to predict the clustering indicator value of the unknown feature with the clustering indicator average value of the known feature in the same text, and then add the new feature into the feature space. The classifier is to divide the testing samples into different categories by using an SVM algorithm.

4 | FEATURE SPACE CONSTRUCTION BASED ON DNMTF

4.1 | Nonnegative matrix tri-factorization

The feature space is constructed by factorization of the relationship matrix. First, according to the label data of the short text training set, the clustering indicator matrix $\boldsymbol{G}$ can be directly obtained, which is part of the relationship matrix R in the nonnegative matrix tri-factorization.[33] Then, with manifold regularization constraint added, word clustering indicator matrix F is obtained by decomposition.

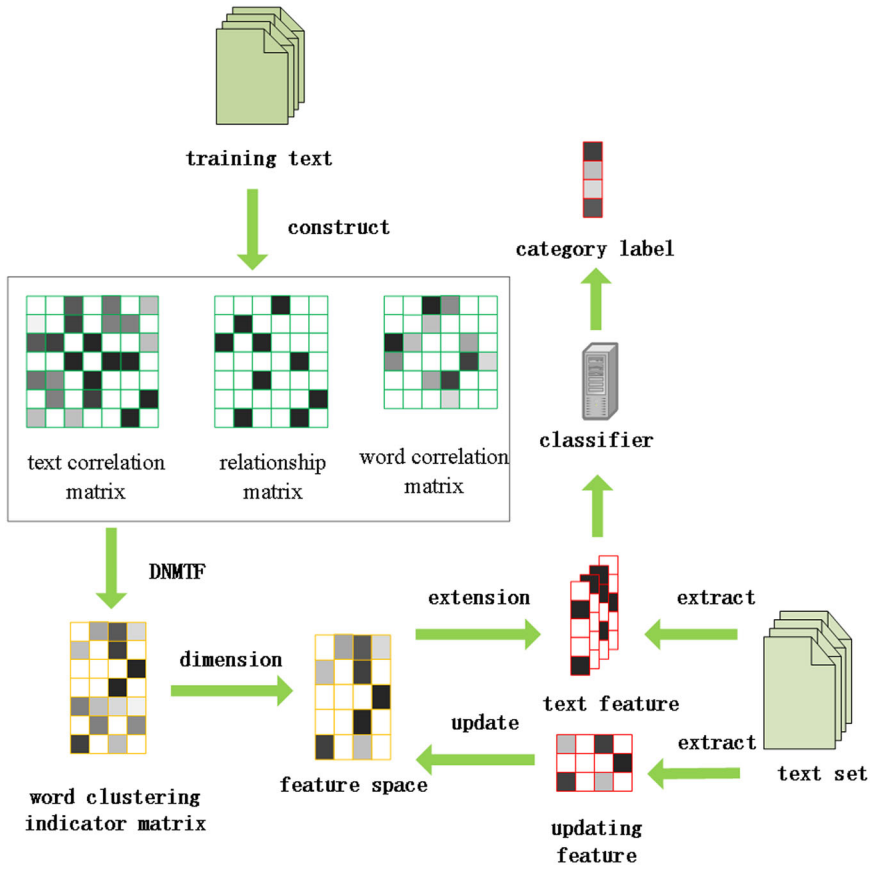

FIGURE 1 Framework of the proposed algorithm [Color figure can be viewed at wileyonlinelibrary.com]

The relation matrix R is decomposed into three matrices, F, S, and G, noted as $R \approx \boldsymbol{FSG}^{\mathrm{T}}$. Matrix F and G are clustering indicator matrix corresponding to two types of entities, respectively, and matrix S is an equilibrium matrix with multidimension, which would guarantee the accuracy of low-dimensional matrix representation.

4.2 | Construction of relationship and correlation matrix

The construction of the relationship matrix R follows the natural relationship between text and word. If the word w_i appears in the text t_j, then $R_{ij} = 1$, otherwise $R_{ij} = 0$.

The construction of the correlation matrix A_t and A_w is based on statistics information between text and words. The calculation of correlation strength between two samples x_i and x_j is shown in the following equation:

$$A_{ij} = \frac{B(x_i, x_j)}{\Sigma_{x_a, x_b \in T(W)} B(x_a, x_b)}, \tag{1}$$

where $B(x_i, x_j)$ is the number of words (text) co-occurrence by sample x_i and x_j in T (word set W).

4.3 | Relationship matrix factorization with manifold regularization

According to the manifold hypothesis,[34] if two samples x_i and x_j are similar in geometric structure, then the practical significance of these two samples is also similar, which is reflected in clustering labels. Therefore, we propose a novel algorithm based on the graph dual regularization non-negative matrix tri-factorization algorithm (DNMTF)[35] to capture the intra-type and inter-type relationship among entities. The relationship matrix factorization based on manifold regularization is shown in the following equation:

$$J_1 = ||R - FSG^T||^2 + \mu tr(F^T L_w F) + \phi tr(G^T L_t G) \text{ s. t. } F, S, G \geq 0, \tag{2}$$

where $\mu, \phi > 0$ are the regularization parameters, used to balance the reconstruction error of DNMTF in the first item and graph regularizations in the second and third terms in Equation (2). $L_w = D_w - A_w$ is the graph Laplacian of the data graph which reflects the label smoothness of the data points, and $L_t = D_t - A_t$ is the graph Laplacian of the feature graph which reflects the label smoothness of the feature D_w and D_t are diagonal matrix, whose entities are column sum of Aw and At, noted as $D_{ii}^w = \sum_j A_{ij}^w$, $D_{ii}^t = \sum_j A_{ij}^t$, respectively.

Since labels of the training set are known already, the clustering indicator matrix G can be directly obtained as part input of J_1. The objective function in Equation (2) can be rewritten into the following equation:

$$\begin{aligned} J_1 &= tr((R - FSG^T)(R - FSG^T)^T) + \mu tr(F^T L_w F) + \phi tr(G^T L_t G) \\ &= tr(RR^T) - 2tr(RGS^T F^T) + tr(FSG^T GS^T F^T) + \mu tr(F^T L_w F) + \phi(G^T L_t G). \end{aligned} \tag{3}$$

Introduce Lawrencian multiplier $\alpha_n \times k$, $\beta_m \times k$ and $\gamma_k \times k$ for constraint $F \geq 0$, $G \geq 0$, and $S \geq 0$, respectively. Accordingly, the Lawrencian function is shown in the following equation:

$$\begin{aligned} L = {} & tr(RR^T) - 2tr(RGS^T F^T) + tr(FSG^T GS^T F^T) + \mu tr(F^T L_w F) + \phi tr(G^T L_t G) + tr(\alpha F^T) \\ & + tr(\beta G^T) + tr(\gamma S^T). \end{aligned} \tag{4}$$

In solving the matrix S, we take the matrix F and G as the given conditions, and then let the partial differential $\frac{\partial L}{\partial \boldsymbol{S}} = 0$, then we derive the following equation:

$$\gamma = 2F^T RG - 2F^T FSG^T G. \tag{5}$$

Using KKT condition[36] $\gamma_{ij} S_{ij} = 0$. Then we can get the following equation:

$$[F^T RG - F^T FSG^T G]_{ij} S_{ij} = 0. \tag{6}$$

According to Equation (6), matrix S follows the following updating, as shown in the following equation.

$$S_{ij} \leftarrow S_{ij} \frac{[F^T RG]_{ij}}{[F^T FSG^T G]_{ij}}. \tag{7}$$

In solving the matrix F, we take the matrix S and G as the given conditions, and then let the partial differential $\frac{\partial L}{\partial \boldsymbol{F}} = 0$. Then we get the following equation:

$$\alpha = 2RGS^T - 2FSG^T GS^T - 2\mu L_w F. \tag{8}$$

Replace $L_w = D_w - A_w$ into Equation (8) and use KKT condition[36] $\alpha_{ij} F_{ij} = 0$. Then we can get the following equation:

$$[RGS^T - FSG^T GS^T - \mu D_w F + \mu A_w F]_{ij} F_{ij} = 0. \tag{9}$$

According to Equation (9), matrix F follows the following updating, as shown in the following equation:

$$F_{ij} \leftarrow F_{ij} \frac{[RGS^T + \mu A_w F]_{ij}}{[FSG^T GS^T + \mu D_w F]_{ij}}. \tag{10}$$

The feature space construction process is described in Algorithm 1.

Algorithm 1. Feature space construction

Input: the number of clusters k, regularization parameters μ, ϕ, and maximum number of iterations I, relationship matrix R, correlation matrices A_t, A_w, clustering indicator matrix G.

Output: feature space H.

Steps in Detail: F

while not convergent and number of *iterations* $< I$

  Update S by Equation (7)

  Update F by Equation (10)

end while

  Get H by dimension reduction of F

return H

5 | FEATURE EXTENSION BASED ON SELF-RESOURCES

5.1 | Feature expansion

Suppose there are p feature words in the feature space $H_{p \times k}$, which is the output of Algorithm 1. Then, from space H, there are q ($p >> q$) features f_i ($i = 1, ..., q$) are chosen out to compose of a subset of the feature space H, denoted as $H^*_{q \times k}$, which contains and only contains those q

features. Then, multiply H^* with feature space H to get matrix $E_{q\times p}$, as shown in the following equation:

$$E = H^* \cdot H^T \tag{11}$$

where the matrix E describes f_i $(i = 1, ..., q)$ correlation with all features in space H.

To select features for expansion conveniently, the matrix E is compressed, and the values of each column are added and the mean is calculated to get the vector e with dimensions p, as shown in the following equation:

$$e\left(j\right) = \frac{\sum_{i=1}^{q} E_{ij}}{q}, \quad j = 1 \cdots p. \tag{12}$$

Vector e describes the relevance between each feature word in the feature space H and feature representation f_i $(i = 1, ..., q)$ in the subspace H^*. In addition to the existing text features, the first K features are selected to expand the short text according to the relevance in e.

5.2 | Feature space update

In the process of extending the features of the short text, there is a possibility: some features extracted from the short text are not included in the feature space H. At this time, the feature space has an insufficient feature expansion. Therefore, before the feature expansion of the short text, the text features should be first detected to see whether an update of space H to cover all new text features is needed. There are two kinds of new features needed to update:

(1) the feature does not exist in the feature space H; and
(2) the feature is not the one that had been deleted after dimension reduction on clustering indicator matrix.

Suppose there are features needed to be updated, and their corresponding clustering indicator matrix is H^{**}. Due to the correlation between input data, H^{**} can be calculated based on H^*, as shown in the following equation:

$$H_i^{**}(j) = \frac{\sum_{g=1}^{q} \boldsymbol{H}_{gj}^*}{q}, \quad j = 1 \cdots k, \quad i = 1 \cdots a. \tag{13}$$

Finally, H^{**} is incorporated into H to obtain an enlarged feature space, based on which feature expansion is carried out. Here, H^* is a subset of the feature space H.

5.3 | Algorithm description

Algorithm 2. Feature expansion

Input: short text set $T = \{t_1, ..., t_g\}$, the number of clusters k, feature space, the number of features to be expanded K

Output: feature vector $v(t)$ of T

Initialize the $v(t_i) = \{0\}$ based on H

for each $t_i = \{f_1, ..., f_{q+a}\}$ of T

Get H^* of t_i

if $a \neq 0$

for each f_b $(b = 1, ..., a)$

Get H_i^{**} by Equation (13)

Update H

end for

end if

Get E by Equation (11)

Get $\boldsymbol{e}$ by Equation (12)

for each $d = \{1, ..., K\}$

Select the features fc with the maximal value in e

if $f_c \notin t_i$

Add f_c to t_i

d++

end if

end for

for each feature $f_d \in t_i$

the f_d position $v(t_i)$ is set to 1

end for

end for

return $v(t_i)$

6 | EXPERIMENTS AND DISCUSSION

6.1 | Experimental data sets

This paper verifies the effectiveness of the proposed method using three data sets. In the experiment, the open source tool libsvm is used as the text classifier. The first data set, Web snippets, obtained from Web search by Phan et al.,[37] is a commonly used short text classification test set. The data set contains eight categories, including 10,060 training sets and 2280 test sets, with an average text length of 17.93. Specific information is listed in Table 1.

The second data set is the Twitter100k, published by Hu et al.[38] The text is written by users in an informal language and is subject to the number limitation of words. Without class label in this data set, only sports-related data are selected out, and used as experimental data for sport-item data classification after they are manually tagged and the final six items, including 3000

TABLE 1 Web snippets data set

Class	Training set	Testing set
Business	1200	300
Computers	1200	300
Culture-Arts-Entertainment	1880	330
Education-Science	2360	300
Engineering	220	150
Health	880	300
Politics-Society	1200	300
Sports	1120	300

training sets and 630 test sets, are left with an average text length of 12.95. The specific information is listed in Table 2.

The third data set is the AGnews data obtained by Zhang[39] and the four classes with the largest amount of are selected to construct the data set, including 120,000 training sets and 7600 test sets, with an average text length of 38.82. The specific information is listed in Table 3.

6.2 | Parameters selection

In Equation (2), the regularization parameters μ and ϕ are selected according to one of the three evaluation indexes, Purity,[40] Normalized Mutual Information (NMI)[41] and Adjusted Rand Index (ARI).[42] Purity calculates the proportion of correctly clustered documents in the total number of documents. NMI measures the degree of similarity between the two clustering results, and ARI measures the degree of coincidence between the clustering results and the real situation. In the process of relationship matrix factorization, the regularization parameter is set to $\mu = \phi$. Based on different value of μ, the DNMTF method with random initialization is carried out for 50 times, and the comparison results are shown in Figure 2.

From Figure 2, we can see that the clustering accuracy arrives the highest when $\mu = 0.6$, with any one of three evaluation indexes. Accordingly, in the following experiments of matrix factorization, we set up the regularization parameter to be $\mu = 0.6$.

TABLE 2 Twitter sports data set

Class	Training Set	Testing Set
Baseball	500	100
basketball	500	100
Football	400	80
Golf	400	50
Rugby	800	200
Swimming	400	100

TABLE 3 AGnews data set

Class	Training Set	Testing Set
World	30,000	1900
Sports	30,000	1900
Business	30,000	1900
Sci/Tech	30,000	1900

The Web snippets data set has 4775 features, Twitter sports data set has 1248 features, and AGnews data set has 6582 features. The selection of feature extension number K directly affects the classification results. Therefore, different parameters K are selected on three data sets for comparative experiments, and the results are shown in Figure 3A-C, respectively. We can see that no matter which data set, even if there is only one feature is added, and the accuracy of classification results increase rapidly to be close to the optimal value 1. The reason for that is the feature with the strongest relevance to the short text is found in the feature space according to Equation (12), which must be the most indicative feature in a certain category. Expansion by this feature will allow other short texts of the same category to enlarge their feature representation, in case they did not have it before. The similarity between the sparse feature vectors of the same category is greatly improved, which has a positive impact on the classification results.

When the number of extended features gradually increases, the accuracy of classification results increases comparatively constant until it reaches the peak point of each data set, then it begins to decline slightly, as shown in Figure 3A–C.

6.3 | Compared algorithms

To verify the effect of NMFFE algorithm, we compare NMFFE with BOW and Char-CNN, namely word bag method and character-level convolutional neural network method without considering semantic information. The results are shown in Table 4 and the corresponding best results are all in bold font. In the study [11], the accuracy of BOW algorithm and Char-CNN algorithm on AGnews data set was 88.81% and 87.18%, respectively. In our experimental environment and data processing operations, our experimental results shown in Table 4 are little different with those presented by study [11].

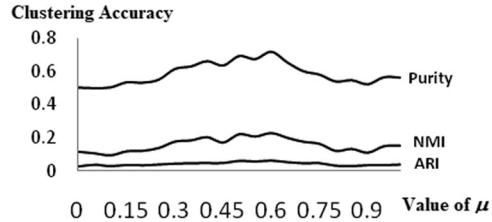

FIGURE 2 Effect of different regularization parameter μ

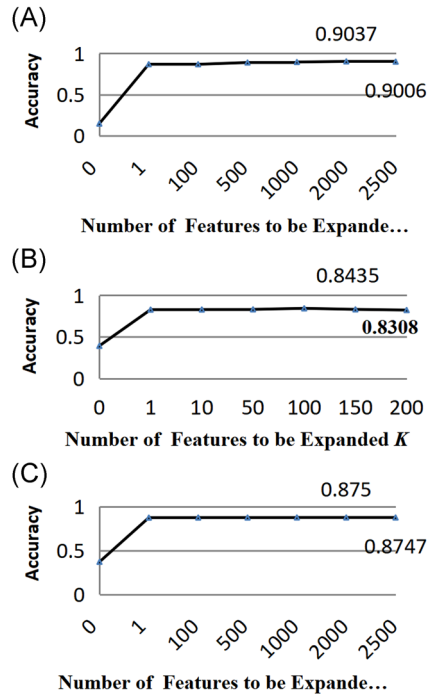

FIGURE 3 Results of parameter *K* on three data sets. (A) Web snippets data set with different *K*. (B) Twitter sports data set with different *K*. (C) AGnews data set with different *K* [Color figure can be viewed at wileyonlinelibrary.com]

From Table 4, we can find that in the respect of data set size, the Char-CNN algorithm performs well in big data sets but perform less in small data sets, where the limited training data cannot cover the overall distribution of data, and lead to the over-fitting of convolutional neural network.

In the respect of data integrity, text length of the AGnews data set is relatively long, and its sufficient corpus makes the three algorithms perform well in text classification. The accuracies of their classification results have small differences. The similarity between test data set and training data set of Web snippets (co-occurrence of keywords) is not as high as the other two data sets, making the BOW algorithm based on word frequency statistics on this data set less effective.

The overall performance of the proposed NMFFE algorithm achieves better classification results than those of the other two algorithms, and the robustness on data sets with different sizes is better than the two latter. BOW algorithm and Char-CNN algorithm are more suitable for large-scale data sets.

TABLE 4 Comparison results of classification accuracy on three data sets

Data sets	BOW	Char-CNN	NMFFE
Web snippets	0.646	0.625	**0.9037**
Twitter100k	0.7346	0.5234	**0.8435**
AGnews	0.8421	0.8571	**0.875**

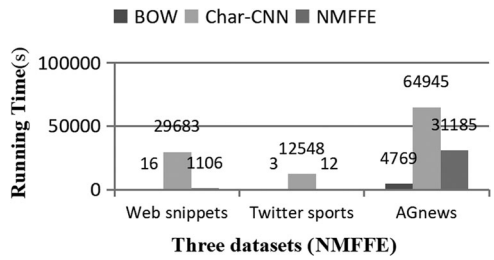

FIGURE 4 Comparation of running time

The running time of the three algorithms is compared on three data sets, and the results are shown in Figure 4. The execution time of BOW algorithm is shorter than the other two algorithms, and it is more obvious on large data sets, mainly because the model of BOW algorithm is relatively simple. NMFFE algorithm takes the longest time in the feature expansion process, because it involves a lot of matrix operations. When the number of feature extensions K increases, the running time also increases. The Char-CNN algorithm model consists of six convolution layers and three full connection layers.

7 | CONCLUSIONS

Different from vector-form based feature expansion method of short texts, we proposed a method using K relevant features as a self-contained subset to extend feature space of short texts. Without relying on the external resources, words clustering indicator matrix was obtained from text data set itself through graph dual regularization non-negative matrix tri-factorization (DNMTF). After dimension reduction, feature space was obtained as the basis for feature expansion, and then the most relevant features extracted within the data set itself were selected to enlarge the feature space of short texts. Experimental results showed that NMFFE algorithm performed better than Word2Vec algorithm and Char-CNN algorithm in accuracy of classification. However, the data sets used in this paper were all open data sets which actually had been pre-processed. However, the main challenge of short-text feature expansion and classification is the online and real-time data processing. So, we will adjust our method to adapt the real-time online environments in the future.

ACKNOWLEDGMENTS

This paper was funded by Scientific Project of Guangdong Provincial Transport Department (No. Tec-2016-02-30), Natural Science Foundation of Guangdong Provinceunder Grant 2018A030313061, in part by the Guangdong Science and Technology Planunder Grants 2017B010124001, 201902020016, and 2019B010139001. The last author is also partially supported by the European Horizon 2020 research and innovation programme by the ENVRI-FAIR project (824068), the BLUECloUD project (862409), and the ARTICONF project (825134).

REFERENCES

1. Rafeeque PC, Sendhilkumar S. A survey on short text analysis in Web. Third International Conference on Advanced Computing, Chennai, India. https://doi.org/10.1109/ICoAC.2011.6165203

2. Heap B, Bain M, Wobcke W. Word vector enrichment of low frequency words in the bag-of-words model for short text multi-class classification problems. e-print, 2017arXiv170905778H
3. Tommasel A, Godoy D. Short-text feature construction and selection in social media data: a survey. *Artif Intell Rev*. 2018;49(3):301-338.
4. Kang W, Qiu HZ, Jiao DD. Search-based short-text classification. *Appl Electron Tech*. 2018;44(1):122-128. https://doi.org/10.16157/j.issn.0258-7998.181392
5. Li XH, Su Y, Ma HF. Combining statistical information and semantic similarity for short text feature extension. International Conference on Intelligent Information Processing; 2016; 486:205-210.
6. Li JZ, Cai Y, Cai ZW. Wikipedia based short text classification method. International Conference on Database Systems for Advanced Applications; 2017; Vol. 3, Suzhou:275-286.
7. Li P, He L, Wang H, et al. Learning from short text streams with topic drifts. *IEEE Trans Cybern*. 2018; 48(9):2697-2710.
8. Vo DT, Ock CY. Learning to classify short text from scientific documents using topic models with various types of knowledge. *Exp Syst Appl*. 2015;42(3):1684-1698.
9. Zhang H, Zhong GQ. Improving short text classification by learning vector representations of both words and hidden topics. *Knowl-Based Syst*. 2016;102(C):76-86.
10. Sikdar UK, Gambac B. Named entity recognition for Amharic using stack-based deep learning. 18th International Conference on Computational Linguistics and Intelligent Text Processing; 2017; Vol 4, Budapest. 10761:276-287.
11. Zhang X, Zhao JB, Yann LC. Character-level convolutional networks for text classification. 29th Annual Conference on Neural Information Processing Systems (NIPS); 2015; Montreal:649-657.
12. Quan XJ, Kit CY, Ge Y. Short and sparse text topic modeling via self-AGGREGation. 1st International Workshop on Social Influence Analysis /24th International Joint Conference onArtificial Intelligence (IJCAI); 2015; Vol 7, Buenos Aires:2270-2276.
13. Pennington J, Socher R, Manning CD. GloVe: global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing; 2014; Vol 10, Doha:1532-1543.
14. Mikolov T, Chen K, Corrado G. Efficient estimation of word representations in vector space; 2013:9. https://arxiv.org/abs/1301.3781v3
15. Li ZH, Yang ZH, Shen C. Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text. *BMC Med Inform Decis Mak*. 2019;19(1):22.
16. wang DQ, Lu CW, Wu JJ. Softly associative transfer learning for cross-domain classification. *IEEE Trans Cybern*. 2019;1:1-13. https://doi.org/10.1109/TCYB.2019.2891577
17. Xia W, He YX, Tian Y. Feature expansion for microblogging text based on latent Dirichlet allocation with user feature. The 6th IEEE Joint International Information Technology and Artificial Intelligence Conference; 2011; Vol 8, Chongqing:228-232.
18. Xu Y. *Research on Short Text Classification Based on Word Vectors and Topics*. Vol 5. Huazhong University of Technology, Wuhan; 2018: 514.
19. Yu J, Qiu LR. ULW-DMM: an effective topic modeling method for microblog short text. *IEEE Access*. 2019; 7:884-893.
20. Hassan A, Mahmood A. Deep learning approach for sentiment analysis of short texts. The 3rd IEEE International Conference on Control, Automation and Robotics; 2017; Nagoya:705-710.
21. Wang ZL, Li S, Chen G. Deep and shallow features learning for short texts matching. The 5th IEEE International Conference on Progress in Informatics and Computing; 2017; Vol 12, Nanjing:51-55.
22. Severyn A, Moschitt A. Learning to rank short text pairs with convolutional deep neural networks. The 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2015; Vol 8, Santiago:373-382.
23. Al-Azani S, El-Alfy ESM. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short Arabic text. The 8th International Conference on Ambient Systems, Networks and Technologies/7th International Conference on Sustainable Energy; 2017; Vol 5, Madeira, 109:359-366.
24. De Boom C, Van Canneyt S, Demeester T. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recogn Lett*. 2016;80:150-156.
25. Wang P, Xu B, Xu JM. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*. 2016;174:806-814.

26. Sang L, Xie F, Liu XJ. WEFEST: Word embedding feature extension for short text classification. The 16th IEEE International Conference on Data Mining; 2016; Vol 12, Barcelona:677-683.
27. Jinarat S, Manaskasemsak B, Rungsawang A. Short text clustering based on word semantic graph with word embedding model. *Joint 10th International Conference on Soft Computing and Intelligent Systems/19th International Symposium on Advanced Intelligent Systems;* 2018; Vol 12, Toyama:1427-1432.
28. Pascual AJ, Fujita S. Text similarity function based on word embeddings for short text analysis. The 18th International Conference on Computational Linguistics and Intelligent Text Processing; 2018; Vol 4, Budapest. 10761:391-402.
29. Xun GX, Gopalakrishnan V, Ma FL. Topic Discovery for Short Texts Using Word Embeddings. *The 16th IEEE International Conference on Data Mining*. 2016, 12. Barcelona, SPAIN: 1299-1304.
30. Liang WX, Feng R, Liu LXY. GLTM: a global and local word embedding-based topic model for short texts. *IEEE Access*. 2018;6:43612-43621.
31. Naili M, Chaibi AH, Ben G. Comparative study of word embedding methods in topic segmentation. *Procedia Comput Sci.*2017;112:340-349.
32. Jameel S, Bouraoui Z, Schockaert S. Unsupervised learning of distributional relation vectors. The 56th Annual Meeting of the Association for Computational Linguistics; 2018; Vol 7, Melbourne:1-11.
33. Cheng X, Guo J, Liu S. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. 13th SIAM International Conference on Data Mining; 2013; Vol 5, TX:749-757.
34. Borg I. A note on the positive manifold hypothesis. *Pers Individ Dif*. 2018;134(1):13-15.
35. Shang FH, Jiao LC, Wang F. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recogn*. 2012;45(6):2237-2250.
36. Boyd S, Vandenberghe L. *Convex Optimization*. Vol 3. New York: Cambridge University Press; 2004:63-107.
37. Phan XH, Nguyen LM, Horiguchi S. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. Proceeding of the 17th international conference on World Wide Web*;* 2008:91-100.
38. Hu YT, Zheng L, Yang Y. Twitter100k: a real-world dataset for weakly supervised cross-media retrieval. *IEEE Trans Multimedia*. 2018;20(4):927-938.
39. Zhang X. AG's News Topic Classification Dataset Version 3; 2015.
40. Hassani Marwan, Seidl Thomas. Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. *Vietnam J Comput Sci*. 2017;4(3):171-183.
41. Yang S, Zhang LJ. Non-redundant multiple clustering by nonnegative matrix factorization. *Mach Learn*. 2017;106(5):695-712.
42. Robert V, Vasseur Y, Brault V. Comparing high dimensional partitions, with the Coclustering Adjusted Rand Index; 2017.

How to cite this article: Zhang L, Jiang W, Zhao Z. Short-text feature expansion and classification based on nonnegative matrix factorization. *Int J Intell Syst*. 2020;1-15. https://doi.org/10.1002/int.22290