A New Vector Representation of Short Texts for Classification

Yangyang Li and Bo Liu
College of Information Science and Technology, Jinan University, China

Abstract: *Short and sparse characteristics and synonyms and homonyms are main obstacles for short-text classification. In recent years, research on short-text classification has focused on expanding short texts but has barely guaranteed the validity of expanded words. This study proposes a new method to weaken these effects without external knowledge. The proposed method analyses short texts by using the topic model based on Latent Dirichlet Allocation (LDA), represents each short text by using a vector space model and presents a new method to adjust the vector of short texts. In the experiments, two open short-text data sets composed of google news and web search snippets are utilised to evaluate the classification performance and prove the effectiveness of our method.*

Keywords: *Text representation, short-text classification, Latent Dirichlet Allocation, topic model.*

Received January 30, 2019; accepted July 2, 2019
https://doi.org/10.34028/iajit/17/2/12

1. Introduction

The popularity of various platforms, such as the Internet, Twitter, Weibo and e-commerce, and their integration into people's daily lives have enabled people to express their opinions on these platforms at any time and any place. As a result, hundreds of millions of short texts are generated every day. Short texts, such as search snippets, comments, titles and opinions posted, are much more concise and sparser on major social platforms. Titles of news or products and search snippets are usually less than 20 words, while the maximum length of a tweet was only 140 characters before 2017. Short texts have posed new challenges because of their inherent characteristics, including

1. Being short and sparse.
2. Having different expressions produced by uncertain sources, which result in a low co-occurrence rate of words.

Text representations based on bag-of-word, such as Term Frequency-Inverse Document Frequency (TF-IDF) [13] and its variants [12, 24], have been developed and have achieved significant results in long text classification. These approaches focused on feature enrichment regarding the correlation of words occurring in documents and didn't consider the word order and semantics [32].

Many studies have been performed for providing more short text information, and they can be summarized into two categories. The first category provides a context for short texts by using web search results. Every short text in a search engine (e.g. Google) is searched [22, 29], the results are ranked by similarity scores and the top *k* results are chosen as a semantic background of the short text. This approach is disadvantageous because each short text needs to be queried from search engines, which is time consuming. Thus, this method is unsuitable for real-time applications. The other is to extend the short text through an external knowledge (e.g., wiki) [2, 32]. However, this approach cannot guarantee the validity of the external data set [21, 31].

In this study, we propose a new method by adjusting the vector of short texts. Many words have the same meaning but are completely different, such as 'Apple' and 'iPhone'. Moreover, many words have more than one meaning. For example, 'Apple' in 'Apple Watch' represents a watch released by Apple Inc., but 'Apple' in 'Apple is my favourite kind of fruit' represents a kind of fruit. These words are called synonyms and homonyms. In this work, we focus on weakening the influence of synonyms and homonyms on short-text classification.

The rest of this paper is organized as follows. Section 2 briefly introduces related works on text expression. Section 3 provides an overview of the proposed framework and introduces the Latent Dirichlet Allocation (LDA) topic model. Section 4 introduces the proposed algorithm for the adjusting vector representation of short texts. Section 5 evaluates the effectiveness of the algorithm from various aspects via experiments. Section 6 discusses advantages of our method. Section 7 summarizes the core work content of this study and proposes direction for further research.

2. Related Works

This section, we briefly introduce the related works on text representation:

1. Statistical model.
2. Probability model.
3. Neural network language model.
4. Text extension model for short texts.

2.1. Statistical Model

As the name suggests, the statistical model calculates the weight of words in a text by counting the frequency of occurrence of words. The statistical model does not consider word meaning and word order. The most widely used statistical model is TF-IDF, which calculates the weight of each word through term weighting scheme TF-IDF [13]. Each text in the TF-IDF model is represented by an N-dimensional vector, where N is the total number of words in the data set. The weight of TF-IDF is calculated as follows:

$$w_i = tf_{i,j} \times log\frac{|D|}{|\{j: t_i \in d_i\}|}, \qquad (1)$$

Where $tf_{i,j}$ is the number of occurrences of a specific word in the document, $log\frac{|D|}{|\{j: t_i \in d_i\}|}$ is a measure of the general importance of words, $|D|$ is the total number of documents in the corpus and $|\{j: t_i \in d_i\}|$ is the number of documents containing the word t_i.

2.2. Probability Based Model

The naive Bayesian model is the simplest probabilistic model, but it cannot accurately describe semantic information in text tasks. Probabilistic latent semantic analysis (PLSA) [10], which is a topic related model proposed to overcome the shortcomings of the semantic expression of the naive Bayesian model. Blei *et al.* [4] added a layer of Bayesian framework to PLSI, which is the LDA [4] model. The LDA model is a more complete theory of probability generation. It can produce a document-topic probability distribution matrix, where each text corresponds to a row in the matrix .

2.3. Neural Network Based Model

Neural network-based models have emerged with the development of deep learning technology [7]. Word2vec [15] and glove [17], are widely recognized. Bengio proposed a method of training a language model using a neural network [3], specifically, an N-gram model that predicts the probability distribution of the n th word on the basis of the input n−1 words. The word vector is obtained in the process of training the language model. Many researchers have conducted in-depth research on word vector because it contains very rich information. Most of them focus on improving the complexity of word vectors. In these studies, word2vec [15] is the most typical and contains two training methods: continuous bag-of-words model and skip-gram model. Wang *et al.* [28] made a comparative analysis of the three classical expressions in this model.

2.4. Text Extension Model for Short Texts

Short text extension methods can be divided into three categories:

- Methods that provide a large semantic background for each short text by accessing the search engine [5, 14].
- Methods that use the existing semantic base (e.g. WordNet) to extend the short text [11, 25, 26].
- Methods that extend the short text with hidden topics , which are learned from external data sets (e.g., Wikipedia) [19, 27, 30]. Phan *et al.* [19] proposed a method of learning hidden topics from a large data set in 2008. Vo and Ock [27] used the topic model to train multiple common data sets for enriching short texts based on the topic-document probability distribution. Zhang and Zhong [30] utilized the vector representation of both words and hidden topics.

The current text representation methods introduced above have some disadvantages as follows. The probability of co-occurrence of text words is very small, and the statistical-based model has difficulty in expressing the meaning of a short text because of the short and noisy nature of short texts. Probability-based topic text model assumes that each document belongs to multiple topics and analyses the text set to obtain the distribution probability on the topic. The probability matrix can be used as a vector expression of the text, but it may be very sparse. The neural network language model is initially based on word vector representation, which can be realized by word connection or by averaging word vectors in the text, but the interpretability is relatively poor. Zheng *et al.* [31] also mentioned that extra resources and might lead to possible inconsistency. In addition, short texts expansion method is based on the external data set, which requires more time to process additional data and cannot guarantee its availability.

This study provides a new solution and proposes a specific implementation algorithm. Our contributions are as follows:

- We propose a method of adjusting the vector presentation using the LDA topics of the short text itself.
- We propose the concept of jumping point for the selecting topic words.
- We verified the validity of the proposed algorithm on two public data sets.

3. Overview of Proposed Classification Framework and Topic Model Analysis

3.1. Proposed Basic Framework

The proposed classification framework is depicted in Figure 1. This framework consists of the following three steps:

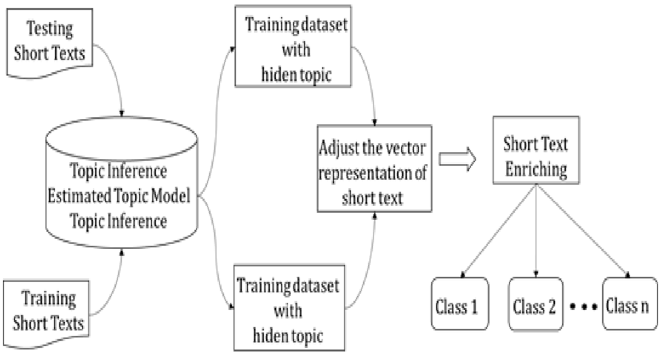

Figure 1. Short text classification by adjusting the vector representation of short texts.

- Use short texts to build LDA [4] topic model and then estimate the parameters of the topic model.
- Use the vector space model (VSM) [23] to represent short texts and calculate the weight of each word through the term weighting scheme TF-IDF. Adjust the vector representation of short texts to weaken the effect of synonyms and homonyms on short-text classification.
- Classify short texts.

The issues that need to be solved in this framework are as follows: setting hyperparameters and estimating parameters, selecting words in the topic thesaurus and adjusting vector representation of short texts to reduce the effect of synonyms and homonyms on short texts.

3.2. Topic Model Analysis and Gibbs Sample

LDA is a document topic generation model that contains three layers of words, topics and documents. Generating documents can be divided into two parts Figure 2. The symbols used in the following parts are described in Table 1.

Table 1. Symbols for LDA.

Symbols	Description
M	The total number of documents
K	The number of topics
V	The total number of words
$\vec{\boldsymbol{\theta}}$	A matrix of M ∗ K , $\overrightarrow{\theta_m}$ is the topics distribution of the m_{th} document
$\vec{\boldsymbol{\varphi}}$	A matrix of K*V, $\overrightarrow{\varphi_k}$ is the words distribution of the k_{th} topic
$\boldsymbol{Z_{m,n}}$	The topic number of the n_{th} word of the m_{th} document
$\boldsymbol{w}$	A word that can be observed, $w_{m,n}$ is the n_{th} word of the m_{th} document
$\boldsymbol{word_i^k}$	The word ranked i in the relevance to the k_{th} topic

The LDA topic model assumes that the document is composed of multiple topics, in which the words in the document are generated separately by a specific topic, and Blei *et al.* [4] defined the topic as the distribution of words. Baker and McCallum [1] proposed that a word can be considered as a distribution of document categories [17] provided by Pereira *et al.* [18]. The concept of 'distributed clustering' reveals the complex relationship between words and topics. To more easily understand the complex relationship between words and topics, the LDA model can be used to decompose the corpus and obtain the distribution of the topics on the words and the distribution of the documents on the topics. Determining the relationship between words, topics and documents enables us to better understand semantics.

In Figure 2, the first part (i.e. $\vec{\alpha} \rightarrow \overrightarrow{\vartheta_m} \rightarrow z_{m,n}$) assumes that M documents are present, selects a document based on $Dirichlet(\vec{\alpha})$ distribution and generates the topic of this document based on the $Multinomial(\vartheta_m)$ through the $Multinomial(\varphi m,n)$ distribution. This part is also a $Multinomial\text{–}Dirichlet$ conjugate distribution. The probability distribution function is as follows [4]:

$$p(\vec{z}|\theta) = \prod_{i=1}^{W} p(z_i = k|d_i = m) = \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_m^n \quad (2)$$

In Figure 2, the second part (i.e. $\vec{\beta} \rightarrow \overrightarrow{\varphi_k} \rightarrow w_{m,n}|k = z_{m,n}$) supposes that multiple words are present for each topic and selects one word from all words in topic $z_{m,n}$ by ($Multinomial(\varphi_{m,n})$ distribution. This part is also a *Multinomial-Dirichlet* conjugate distribution. The probability distribution function is as follows [4]:

$$p(\vec{w}|\vec{z},\varphi) = \prod_{i=1}^{W} p(w_i|z_i) = \prod_{k=1}^{K} \prod_{\{i:z_i=k\}} p(w_i = t|z_i = k) = \prod_{k=1}^{K} \prod_{t=1}^{V} \varphi_{k,t}^{n_t^{(k}} \quad (3)$$

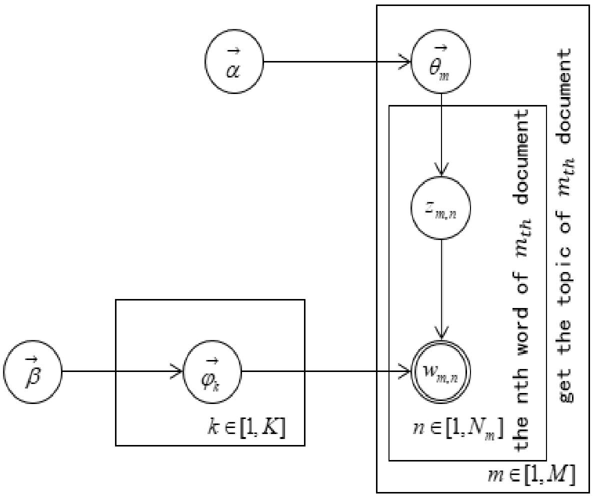

Figure 2. Latent dirichlet allocation model.

LDA is a relatively simple model, but estimating its parameters is difficult. Three methods, namely, variational method [4], expectation propagation [16] and Gibbs sampling [8, 9], are currently used to estimate parameters. In this study, we choose Gibbs sampling that was firstly used by Griffiths [8] for parameter estimation of LDA. We estimate the parameters of LDA with Gibbs sampling (a package named GibbsLDA++ [1] is used to implement this estimation). Gibbs sampling is a method based on Markov chain Monte Carlo. The Gibbs sampling formula of LDA model [4] is as follows:

$$p(z_i = k|\overrightarrow{z_{\neg i}}, \vec{w}) = \frac{n_{k,\neg i}^{t}+\beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{t}+\beta_t} \cdot \frac{n_{m,\neg i}^{(k)}+\alpha_k}{\left[\sum_{k=1}^{K} n_m^{(k)}+\alpha_k\right]-1}, \quad (4)$$

Where $\neg i$ indicates that the index is not i and $n_{k,\neg i}^{t}$ is the number of words with an index that is not i and is assigned to topic k. $n_{m,\neg i}^{(k)}$ is the number of the documents assigned to topic k after removing the word with the index i.

After sampling, we can calculate the document-topic matrix and the topic-word matrix [4]:

$$\varphi_{k,t} = \frac{n_k^t+\beta_t}{\sum_{t=1}^{V} n_k^t+\beta_t} \quad (5)$$

$$\theta_{m,k} = \frac{n_m^k+\alpha_k}{\sum_{k=1}^{K} n_m^k+\alpha_k} \quad (6)$$

4. New Method for Adjusting Vector Representation of Short Texts

4.1. Text Observation and Analysis

An observation of the texts reveals three phenomena:

1. Synonym.
2. Two different type words with the same topic.
3. Homomorphic.

The algorithm presented in section 4.2 intends to solve these phenomena.

A data snippet represents a short pre-processed text. For Example, Table 2 shows three groups of sentences extracted from the data set composed of web search snippets. These three groups correspond to the three phenomena mentioned above.

Table 2. Examples of web search snippets.

Education-science 1: school library media school library media activity monthly school library media activity monthly magazine school library media specialist plan collaborative lessons unit teacher **Education-science 2:** scholarship scholar ship com college scholarship search student loan college scholarship scholar ship com college scholarship info college scholarship search engine financial aid student loan
Business 1: market commodity commodity news market data com commodity market news financial market price investor analysis com **Business 2:** money central msn detail stock quote stock quote msn money stock quote stock market quotes stock price fundamental investing data price chart news
Computer: computer how stuff work operating system how stuff work operating system operating system control task computer carry manage system resource optimize performance learn operating system **politics-society:** new america publication article downside our presidential system downside presidential system america foundation presidential system mechanism grapple election united live

- In the first group, the word ‘school’ appears multiple times in Education-science 1, and the word ‘college’ appears multiple times in Education-science 2. The two words are synonymous, but this association cannot be obtained by the word vectors derived through statistical methods.
- In the second group, ‘market’ appears multiple times in Business 1, and ‘stock’ appears multiple times in Business 2. ‘Market’ and ‘stock’ are different types but highly related words that exist in the same domain. Given that they are different words, the statistics-based VSM document representation treats them as completely unrelated words. Accordingly, the distance of texts of the same category increases.
- The third group includes two sentences drawn from two categories. They use the word ‘system’, which refers to different meanings in both sentences. We need to adjust its representation according to the topic of the text it belongs to, thus making the categories of the two texts more distinguishable.

4.2. Algorithm for Adjusting Vector Representation of Short Texts

4.2.1. Main Idea of the Algorithm

In view of the characteristics of short texts, we propose a method for adjusting vector representation of short texts using topic models (AVR hereinafter). The AVR algorithm uses the information obtained by the LDA topic model to adjust the textual representation of TF-IDF using the following steps:

- Build original vector representation for each short text by calculating the weight of each word using TF-IDF.
- Use the short text set to build the topic model based on LDA.
- Estimate the parameters of LDA with Gibbs sampling and obtain document-topic probability distribution matrix and a topic-related vocabulary.
- Sort the words in the topic-related vocabulary by the relevance weight between the word and the topic.
- Select the words in the topic-related vocabulary based on the jumping points defined in Definition 1.
- Adjust original vector representation for each short text.

We cannot accurately set the number of words retained in each topic when constructing LDA models, so in the fifth step, we will filter some topic-related words. The criteria for selecting is based on the definition of jumping point. Words with a weight lower than that of the jumping point are filtered out. The introduction of jump points ensures that the selected words are highly relevant to the topic.

[1] http://gibbslda.sourceforge.net/

- *Definition 1 (jumping point): Suppose that topic k contains the following words:* $\{word_1^k, \ldots, word_{i-1}^k, word_i^k, word_{i+1}^k, \ldots, word_n^k\}$. *The words are arranged in ascending order of relevance to the topic. When* $(word_i^k - word_{i-1}^k) \div (word_{i+1}^k - word_i^k) \geq \gamma$ *occurs firstly, then* $word_i^k$ *is the jumping point.*

In Definition 1, γ is a hyper-parameter. Generally, the threshold >1.

Table 3 and Figure 3 illustrate an example of the jumping point concept. A topic and its topic words sorted by relevance weights obtained by the LDA model are listed in Table 3. Figure 3 shows the weight change, where the coordinate scale of weight is enlarged to 1,000 times.

Table 3. Example of a topic and topic words.

Index	⋯	$word_{i-3}$	$word_{i-2}$	$word_{i-1}$	$word_i$	$word_{i+1}$	$word_{i+2}$	$word_{i+2}$	⋯
Word	⋯	protest	baltimor	concuss	stream	kiko	alonso	watch	⋯
Weight	⋯	0.0063	0.0071	0.0079	0.0152	0.0165	0.0173	0.0189	⋯

In Figure 3, each turning point in the curve represents the weight of a word. Let *d*d_{i-1} represent the difference value between $word_i$ and $word_{i-1}$, that is, $d_{i-1}=word_i-word_{i-1}$. In Figure 3, suppose that $word_{i-1}$= 'concuss', $word_i$='stream', $word_{i+1}$ = 'kiko', and γ=1.8, then $d_{i-1} \div d_i$ is greater than the threshold γ. Thus $word_i$ (i.e., 'stream') is the jumping point , and the words whose weights are lower than that of the $word_i$ are filtered out. It is worth noting that, we only look for $word_i$, which satisfies the first occurrence of $(word_i - word_{i-1}) \div (word_i - word_{i-1}) \geq \gamma$.

After determining the topic words, we adjust the short text vector by combining the topic distribution matrix of the text with the topic distribution of the words in the short text for reducing the occurrence of the three phenomena mentioned in section 4.1. A word (as follows: w_i, w_j) whose weight is to be adjusted meets one of the following two conditions:

- $w_i \in text_x, w_j \in text_y, w_i \in topic_k, w_j \in topic_m$, $w_i \neq w_j, text_x \neq text_y,\ topic_k = topic_m$;
- $w_i \in text_x,\ w_j \in text_y,\ w_i \in topic_k, w_j \in topic_m$, $w_i = w_j, text_x \neq text_y,\ topic_k \neq topic_m$;

w_i and w_j are words; $text_x$ and $text_y$ are short texts; $topic_k$ and $topic_m$ represent the k_{th} and m_{th} topics, respectively.

In Algorithm 1, firstly, input the TextSet and represent each short text as a vector by TF-IDF. Afterwards, build LDA model on short texts, and obtain the topic-related vocabulary (trv) and topic distribution matrix of the document (θ). In the next steps, filter topic-related vocabularies based on the definition of jumping point. In the last two loops, if the word x of the short text meets one of the two conditions mentioned in section 4.2.1, then its TF-IDF weight is replaced with the topic-related weight.

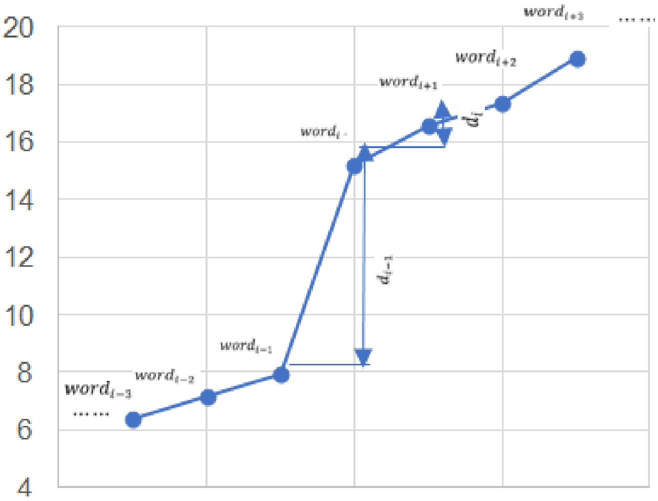

Figure 3. A sketch of the jumping point.

4.2.2. Algorithm Description

Our algorithm AVR is described as follows.

```
Algorithm 1 AVR(TextSet)
$W_x^y$ is the weight of the word x in the short text y
$\vartheta_y^k$ is the probability that the text y belongs to the topic k
$trv_k^x$is relevance measure of word x in topic k
Input TextSet
 for each text y in TextSet TF-IDF(y)
 LDA_M(TextSet)
GetTopicVoc ($trv$)
GetDisMat ($\theta$)
Filter ($trv$)
for each text y in TextSet
{
for each word x in y
{
        if the word meets one of the two conditions
described in 4.2.1
        then $W_x^y = \vartheta_y^k \times trv_k^x$;
    }
}
Return adjusted TextSet vector representation matrix
```

Corresponding to Equations (5) and (6), the replaced weight can be calculated using Equation (7):

$$W_x^y = \vartheta_y^k \times \mathrm{trv}_k^x = \frac{n_y^k+\alpha_k}{\sum_{k=1}^K n_y^k+\alpha_k} \times \frac{n_k^x+\beta_x}{\sum_{t=1}^V n_k^x+\beta_x} \qquad (7)$$

For example, the vector representation of the short texts in Table 2 is processed using AVR, and the results are shown in Table 4. In the table, w_x is the weight of the word *x* calculated using TF-IDF.

Table 4. Vector representation of short text processed by the AVR algorithm.

Education-science 1: $\vartheta^{k}_{Education-science1} \times trv^{school}_{k}$、$W_{library}$、$W_{media}$、$\vartheta^{k}_{Education-science1} \times trv^{school}_{k}$、$W_{library}$、$W_{media}$、$W_{activity}$、$W_{monthly}$、$dtm^{k}_{Education-science1} \times trv^{school}_{k}$、$W_{library}$、$W_{media}$、$W_{activity}$、$W_{monthly}$、$W_{magazine}$、$\vartheta^{k}_{Education-science1} \times trv^{school}_{k}$、$W_{library}$、 **Education-science 2:** $W_{scholarship}$、W_{com}、$\vartheta^{k}_{Education-science2} \times trv^{college}_{k}$、$W_{scholarship}$、$W_{seaerch}$、$W_{student}$、$W_{loan}$、$\vartheta^{k}_{Education-science2} \times trv^{college}_{k}$、$W_{scholarship}$、$W_{scholarship}$、$W_{com}$、$\vartheta^{k}_{Education-science2} \times trv^{college}_{k}$、$W_{scholarship}$、$W_{info}$、$\vartheta^{k}_{Education-science2} \times trv^{college}_{k}$、$W_{scholarship}$、$W_{search}$、$W_{engine}$、$W_{financial}$、$W_{student}$、$W_{loans}$
Business 1: $\vartheta^{k}_{Business1} \times trv^{maket}_{k}$、$W_{commodities}$、$W_{news}$、$\vartheta^{k}_{Business1} \times trv^{maket}_{k}$、$W_{data}$、$W_{com}$、$W_{commodity}$、$\vartheta^{k}_{Business1} \times trv^{maket}_{k}$、$W_{news}$、$W_{financial}$、$\vartheta^{k}_{Business1} \times trv^{maket}_{k}$、$W_{price}$、$W_{investor}$、$W_{analysis}$、$W_{com}$ **Business 2:** W_{money}、$W_{central}$、W_{msn}、W_{detail}、$\vartheta^{k}_{Business2} \times trv^{stock}_{k}$、$W_{dnote}$、$\vartheta^{k}_{Business2} \times trv^{stock}_{k}$、$W_{quote}$、$W_{msn}$、$W_{money}$、$\vartheta^{k}_{Business2} \times trv^{stock}_{k}$、$W_{quote}$、$\vartheta^{k}_{Business2} \times trv^{stock}_{k}$、$W_{market}$、$W_{quote}$、$\vartheta^{k}_{Business2} \times trv^{stock}_{k}$、$W_{price}$、$W_{fundamental}$、$W_{investing}$、$W_{data}$、$W_{price}$、$W_{chart}$、$W_{news}$
Computer: $W_{computer}$、W_{how}、W_{stuff}、W_{work}、$W_{operating}$、$\vartheta^{k}_{computer} \times trv^{system}_{k}$、$W_{how}$、$W_{stuff}$、$W_{work}$、$W_{operating}$、$\vartheta^{k}_{computer} \times trv^{system}_{k}$、$W_{operating}$、$\vartheta^{k}_{computer} \times trv^{system}_{k}$、$W_{control}$、$W_{task}$、$W_{computer}$、$W_{carry}$、$W_{manage}$、$W_{system}$、$W_{resource}$、$W_{optimize}$、$W_{performance}$、$W_{learn}$、$W_{operating}$、$\vartheta^{k}_{computer} \times trv^{system}_{k}$ **politics-society:** $W_{newamerica}$、$W_{publication}$、$W_{article}$、$W_{downside}$、W_{our}、$W_{presidential}$、$\vartheta^{k}_{politics-society} \times trv^{system}_{k}$、$W_{how}$、$W_{downsides}$、$W_{presidential}$、$\vartheta^{k}_{politics-society} \times trv^{system}_{k}$、$W_{america}$、$W_{foundation}$、$W_{presidential}$、$\vartheta^{k}_{politics} \times trv^{system}_{k}$、$W_{mechanism}$、$W_{grapple}$、$W_{election}$、$W_{united}$、$W_{live}$

5. Experiments and Result

5.1. Data Sets

Two typical short-text data sets are selected, namely web search snippets and Google News. These two data sets are well suited to the characteristics of short texts. The web search snippet data set is composed of search snippets, and each text is no longer than 30 words. The Google News data set is composed of Google News headlines, and each text is no longer than 10 words.

5.1.1. Web Search Snippets

The data set composed of web search snippets[2] of different domains can be obtained directly from the Internet using search engines. For example, in the business domain, we select 60 phrases that belong to the business domain, select the top 20 in each search result and finally obtain 1,200 web search snippets [21]. These web search snippets are short, sparse, noisy, less topic-focused and very representative. The details are shown in Table 5.

Table 5. Number of web search snippets as training and test data.

Domain	Training data	Test data
Business	1200	300
Computer	1200	300
Culture-Arts-Ent	1800	330
Education-Science	2360	300
Engineering	220	150
Health	880	300
Politics-Society	1200	300
Sport	1120	300
Total	10060	2280

5.1.2. Google News

All news (reports in Google news[3]) are grouped according to different fields. We use the method described by Banerijee and Ramanathan 2] to crawl news titles, which belong to the six domains of business, health, science, sport, technology and world, on October 20, 2017 and October 29, 2017. The details are shown in Table 6.

Table 6. Crawled google news number in various domains.

Domain	Business	Health	Science	Sport	Technology	World	Total
Dataset	2057	1572	797	2813	2343	2850	12432

5.2. Data Pre-Processing and Classification Evaluation

The data set composed of web snippets is already processed and can be used directly. We deal with Google News as follows:

1. Letters are converted to lowercase.
2. Numbers, symbols and stop words are removed.
3. Sentences with fewer than five words are deleted.
4. Word stems are extracted. After pretreatment, 8,051 Google news titles are obtained.

5.3. Evaluation

Support Vector Machine (SVM) or K-Nearest Neighbour (KNN) classification method is conducted on the two text data sets represented by our approach to verify the effect of our text representation method on classification results.

Accuracy is used to evaluate the effectiveness of the classification, which proves the effectiveness of the proposed algorithm. Accuracy rate is defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

Where True Positive (TP) is the number of records from class A predicted as class A, True Positive (FP) is the number of records from class B predicted as class A, False Negative (FN) is the number of records from class A predicted as class B and True Negative (TN) is the number of records from class B predicted as class

[2] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz

[3] https://news.google.com/news

B. Original vector representation of the short text data set is performed, and the validity of this expression is verified by classification.

5.4. Parameter Selection of LDA Model for Web Search Snippets

The experimental results are compared with the different number of topics and the different iterations of the LDA model. The data set used in this section is the web search snippets. The classification algorithm used in this section is KNN. The results are shown in Figures 4 and 5.

Figure 4 shows the change in classification accuracy with the number of topics. In this experiment, we use different numbers of topics (from 10 to 100) to estimate the LDA model, use the distribution matrix to adjust the vector representation of short texts in web search snippets and classify the adjusted short texts. Accuracy rate gradually increases with the number of topics from 10 to 60 and gradually decreases after 60. This finding indicates that classification performance is relatively stable with the number of topics, and we can carry out a test to obtain the optimal number of topics.

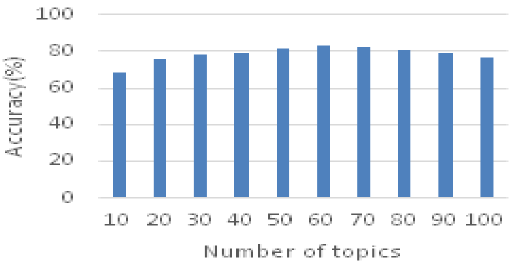

Figure 4. Classification accuracy with different topic number for web search snippets.

Figure 5 shows the change in classification accuracy with the number of topics and Gibbs iterations. The figure shows that classification accuracy is the highest when the number of topics is 60 and the number of iterations is 800.

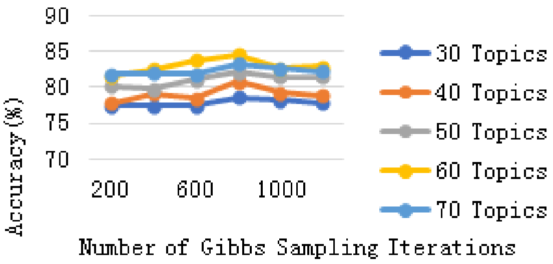

Figure 5. Classification accuracy with different topic number and Gibbs iteration for web search snippets.

5.5. Results of Web Search Snippets

In this section, the experimental results of web search snippets are compared with those of other related works on web search snippets. The results are shown in Table 7. A comparative experiment is conducted with the four types of textual expressions introduced in the relevant works in section 2. Expressions of statistical model, probability-based model, text extension model for short text and our model are classified by SVM classifier. Expression of neural network language mode is classified by convolutional neural network (CNN) classifier.

Table 7. The classification accuracy of proposed method against other models.

Domain	Expression	Accuracy
Statistical model	TF-IDF	68.62%
Probability based model	LDA	76.10%
Neural network language model [29]	Senna	83.6%
	GloVe	84.4%
	Word2Vec	85.1%
Text extension model for short texts	Phan *et al.* [20]	82.7%
	Chen *et al.* [6]	85.31%
	Zhang and Zhong [30]	86.57%
Our model	AVR	86.35%

Table 7 shows that our algorithm accuracy is higher than those of other algorithms, except for the approach proposed by Zhang and Zhong [30]. Our method does not require the introduction of external knowledge.

5.6. Results of Google News

We conduct experiments on the Google News data set to further illustrate the validity of our model. Similar to section 5.3, we obtain the best parameters for the LDA model on the Google News data set: the number of topics is 45, and the number of samples is 600.

We compare unprocessed TF-IDF text expression and AVR text expression. We randomly divide the data set into five equal parts and perform five cross-validation tests to examine the accuracy of our proposed algorithm on the Google News data set. The classification algorithm used in this section is KNN. The experimental results are shown in Figure 6. The figure shows that classification accuracy by AVR expression is higher than that of TF-IDF expression.

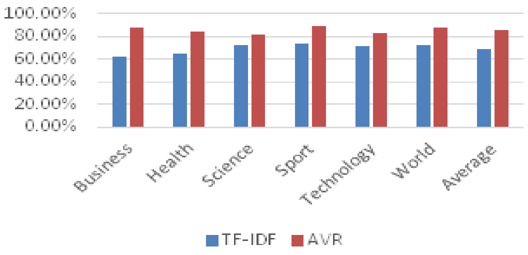

Figure 6. Classification accuracy of two text expressions of TF-IDF and AVR for Google news.

The Wilcoxon signed rank test method is used to analyse the experiment and compare the difference between the two text expressions of TF-IDF and AVR. The probability of no difference between the two methods is 0.018, which is less than 0.05. Thus, the AVR expression pattern is effective and is considerably improved relative to the original TF-IDF expression pattern.

6. Discussion

The proposed text expression method is verified through public data sets, and the experimental results show that the text classification accuracy is greatly improved without extending the number of text representation words. The proposed method is simpler and easier to operate compared with augmenting short texts and can achieve a very good result.

The original words in the text are not processed by other existing methods [19, 27, 30] after the topic words are inserted, although adding the topic words will reduce the influence of homonyms and synonyms. This study examines the processing of short text from a new angle by using carefully processed topic information to adjust the weights of the homographs and synonyms, such that several original words are changed. Several homographs and synonyms in the original texts are replaced by topics or topic words.

The largest advantage of this work is in solving the problem of unavailability of external data. In addition, the method of this research greatly improves the efficiency of short-text representation and does not take time to crawl external data and clean up crawled data.

7. Conclusions and Future Work

This study proposes an algorithm based on the topic model for adjust vector representation of short texts. The proposed algorithm reduces the influence of synonyms and homonyms on short-text classification. We build an LDA model of the short-text data set, establish a document–topic probability distribution matrix and a topic-related vocabulary, and propose the AVR algorithm. The problems of synonyms and homonyms are solved, and the performance of the classification is improved. In the future, further research will be conducted from the following aspects:

1. Optimizing the proposed AVR algorithm.
2. Combining our approach with short text enriching methods to improve the stability of classification.

Acknowledgements

The work was supported by the National Natural Science Foundation of China under Grant No. U1431227; the Foundation of Guangzhou Science and Technology Planning Project under Grant No. 201604010037.

References

[1] Baker L. and McCallum A., "Distributional Clustering of Words for Text Classification," *in Proceedings of The 21st Annual International ACM C Conference on Research and Development in Information Retrieval*, Melbourne, pp. 96-103, 1998.

[2] Banerjee S., Ramanathan K., and Gupta A., "Clustering Short Texts Using Wikipedia," *in Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, pp. 787-788, 2007.

[3] Bengio Y., Ducharme R., Vincent P., and Jauvin C., "A Neural Probabilistic Language Model," *Journal of Machine Learning Research*, vol. 3, no. 2, pp. 1137-1155, 2003.

[4] Blei D., Ng A., and Jordan M., "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 993-1022, 2003.

[5] Bollegala D., Matsuo Y., and Ishizuka M., "Measuring Semantic Similarity Between Words Using Web Search Engines" *in Proceedings of the 16th International Conference on World Wide Web*, Banff, pp. 757-766, 2007.

[6] Chen M., Jin X., and Shen D., "Short Text Classification Improved by Learning Multi-Granularity Topics," *in Proceedings of International Joint Conference on Artificial Intelligence*, Barcelona, pp. 1776-1781, 2011.

[7] Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., and Kuksa p., "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, no. 8, pp. 2493-2537, 2011.

[8] Griffiths T., "Gibbs Sampling in The Generative Model of Latent Dirichlet Allocation," *Standford University*, vol. 518, no. 11, pp. 1-3, 2002.

[9] Griffiths T. and Steyvers M., "Finding Scientific Topics," *in Proceedings of the National Academy of Sciences*, United States of America, pp. 5228-5235, 2004.

[10] Hofmann T., "Probabilistic Latent Semantic Analysis," *in Proceedings of the 5th conference on Uncertainty in Artificial Intelligence*, Stockholm, pp. 289-196, 1999.

[11] Hu X., Sun N., Zhang C., and Chua T., "Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge," *in Proceedings of the 18th ACM Conference on Information and Knowledge Management*, Hong Kong, pp. 919-928, 2009.

[12] Ko Y., "A Study of Term Weighting Schemes Using Class Information for Text Classification," *in Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Portland, pp. 1029-1030, 2012.

[13] Martineau J. and Finin T., "Delta TFIDF: An Improved Feature Space for Sentiment Analysis," *in Proceedings of the 3th International Conference on Weblogs and Social Media*, California, pp. 17-20, 2009.

[14] Metzler D., Dumais S., and Meek C., "Similarity Measures for Short Segments of Text," *in Proceedings of European Conference on Information Retrieval*, Rome, pp. 16-27, 2007.

[15] Mikolov T., Chen K., Corrado G., and Dean J., "Efficient Estimation of Word Representations in Vector Space," *in Proceedings of the International Conference on Learning Representations*, Scottsdale, pp. 1-12, 2013.

[16] Minka T. and Lafferty J., "Expectation-Propagation for The Generative Aspect Model," *in Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Alberta, pp. 352-359, 2002.

[17] Pennington J., Socher R., and Manning C., "Glove: Global Vectors for Word Representation," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, pp. 1532-1543, 2014.

[18] Pereira F., Tishby N., and Lee L., "Distributional Clustering of English Words," *in Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, Columbus, pp. 183-190, 1993.

[19] Phan X., Nguyen L., and Horiguchi S., "Learning to Classify Short and Sparse Text and Web with Hidden Topics from Large-Scale Data Collections," *in Proceedings of the 17th International Conference on World Wide Web*, Beijing, pp. 91-100, 2008.

[20] Phan X., Nguyen C., Nguyen L., Horiguchi S., and Ha Q., "A Hidden Topic-Based Framework toward Building Applications with Short Web Documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 961-976, 2011.

[21] Pinto D., Rosso P., and Jiménez-Salazar H., "A Self-Enriching Methodology for Clustering Narrow Domain Short Texts," *The Computer Journal*, vol. 54, no. 7, pp. 1148-1165, 2011.

[22] Sahami M. and Heilman T., "A Web-Based Kernel Function for Measuring The Similarity of Short Text Snippets," *in Proceedings of the 15th International Conference on World Wide Web*, Edinburgh, pp. 377-386, 2006.

[23] Salton G., Wong A., and Yang C., "A Vector Space Model for Automatic Indexing," *Communications of the Association for Computing Machinery*, vol. 18, no. 11, pp. 613-620, 1975.

[24] Singh S. and Siddiqui T., "Utilizing Corpus Statistics for Hindi Word Sense Disambiguation," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 755-763, 2015.

[25] Sinoara R., Collados J., Rossi R., Navigli R., and Rezende S., "Knowledge-Enhanced Document Embeddings for Text Classification," *Knowledge-Based Systems*, vol. 163, pp. 955-971, 2019.

[26] Song Y, Wang H, Wang Z, Li H., and Chen W., "Short Text Conceptualization Using A Probabilistic Knowledgebase," *in Proceedings of the 22th International Joint Conference on Artificial Intelligence*, Barcelona, pp. 16-22, 2011.

[27] Vo D. and Ock C., "Learning to Classify Short Text from Scientific Documents Using Topic Models with Various Types of Knowledge," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1684-1689, 2015.

[28] Wang P., Xu J., Xu B., Liu C., Zhang H., Wang F., and Hao H., "Semantic Clustering and Convolutional Neural Network for Short Text categorization," *in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, pp. 352-357, 2015.

[29] Yih W. and Meek C., "Improving Similarity Measures for Short Segments of Text," *in Proceedings of the 20nd AAAI Conference on Artificial Intelligence*, Vancouver, pp. 1489-1494, 2007.

[30] Zhang H. and Zhong G., "Improving Short Text Classification by Learning Vector Representations of Both Words and Hidden Topics," *Knowledge-Based Systems*, vol. 102, pp. 76-86, 2016.

[31] Zheng C., Liu C., and Wong H., "Corpus-Based Topic Diffusion for Short Text Clustering," *Neurocomputing*, vol. 275, pp. 2444-2458, 2018.

[32] Zhu Y., Li L., and Luo L., "Learning To Classify Short Text with Topic Model and External Knowledge," *in Proceedings of International Conference on Knowledge Science, Engineering and Management*, Dalian, pp. 493-503, 2013.

Yangyang Li received a master's degree in computer science and technology from Jinan University in 2019. Her main research contents are: data mining, machine learning, natural language processing.

Bo Liu received her master's degree in computer application from Central South University in 1991. Now she is a professor at Jinan University. Her research interests include data mining, information retrieval, natural language processing, etc.