IEICE TRANS. INF. & SYST., VOL.E96–D, NO.12 DECEMBER 2013

PAPER

Improving Text Categorization with Semantic Knowledge in Wikipedia*

Xiang WANG[†a)], **Yan JIA**[†], **Ruhua CHEN**[†], ***Nonmembers***, **Hua FAN**[†], ***Student Member***, ***and*** **Bin ZHOU**[†], ***Nonmember***

SUMMARY Text categorization, especially short text categorization, is a difficult and challenging task since the text data is sparse and multidimensional. In traditional text classification methods, document texts are represented with "Bag of Words (BOW)" text representation schema, which is based on word co-occurrence and has many limitations. In this paper, we mapped document texts to Wikipedia concepts and used the Wikipedia-concept-based document representation method to take the place of traditional BOW model for text classification. In order to overcome the weakness of ignoring the semantic relationships among terms in document representation model and utilize rich semantic knowledge in Wikipedia, we constructed a semantic matrix to enrich Wikipedia-concept-based document representation. Experimental evaluation on five real datasets of long and short text shows that our approach outperforms the traditional BOW method.

key words: *text categorization, Wikipedia, document representation, semantic matrix*

1. Introduction

With the rapid growth of online digital information, text categorization has become one of the key challenging tasks to data mining and machine learning communities for handling and organizing text data in automatic information retrieval systems. Especially in the last decade, with the explosion of applications in e-commerce, social networks, instant communication and RSS, short and sparse text classification becomes more and more important. Traditional text categorization algorithms are usually based on the BOW (Bag of Words) model in which a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order. The BOW model uses the co-occurrence frequency of each word as a feature for text categorization and ignores the semantic relationship among words. This technique breaks multi-word expressions into independent features, maps synonymous words into different features and considers polysemous words as one single feature. In short text classification there is not enough word co-occurrence or shared context to achieve high accuracy. Although some preprocessing technologies such as removing stop words and stemming are proposed to improve the representation, the effect is limited and background knowledge is needed to better understand the meanings of the documents.

Attempts have been made to utilize ontology such as WordNet to add background knowledge to document representation for classification. The most common way of applying ontologies for text categorization is to match ontology concepts to document terms. The matched ontology concepts are either used as replacement or additional features to the original document representation. The approach that replaces original content with ontology concepts may change the semantics of the original document and the method that adds ontology concepts to the original text may introduce noise especially when the coverage of ontology is limited. In this paper, we try to use the replacing method rather than the enriching approach that has been successfully used in an amount of research before [1]–[3]. In order to enhance text categorization, we have to make sure that the ontology is large enough to cover the topical domains in the dataset as completely as possible and the replacing method should not change the original meanings.

Wikipedia today is the largest encyclopedia in the world and a large amount of research utilizes common background knowledge in Wikipedia to improve document representation. It contains millions of articles with each one explaining a concept. In this paper, Wikipedia concept is designated by the title of a Wikipedia article. There are hyperlinks in the articles that link concepts with each other and the link structure represents the semantic relationship between concepts. On average, each article has 39.6 links out to other articles and receives another 39.6 links from them. The large amount of articles and links potentially make it to be a good ontology that can be used for improving text classification performance.

In this paper, we use the textual content and the link structure in Wikipedia to enhance text categorization. We overcome the drawbacks of the BOW model by replacing document terms with Wikipedia concepts and perform classification directly on the Wikipedia concepts vector. We build an inverted index from Wikipedia textual content like what has been done in ESA method [4] and perform mapping using it. We also utilize links between concepts to build a semantic matrix to better understand the semantic relationship between concepts. Our mapping approach can understand synonyms and perform implicit word sense dis-

Manuscript received February 4, 2013.
Manuscript revised July 23, 2013.

[†]The authors are with the School of Computer, National University of Defense Technology, Changsha, China.

*This work was supported by 973 Program (No.2013CB329601, 2013CB329602, 2013CB329604), NSFC (No. 60933005, 91124002), 863 Program (No. 2012AA01A401, 2012AA01A402), NTSF (No. 2012BAH38B04, 2012BAH38B06) of China.

a) E-mail: xiangwangcn@nudt.edu.cn

DOI: 10.1587/transinf.E96.D.2786

ambiguation for polysemous terms. Our method is evaluated not only on long text classification but also in short text classification.

The rest of this paper is organized as follows: Section 2 discusses some important related works. Section 3 introduces our Wikipedia-concept-based document representation method that represents document text with Wikipedia concept vector. In Sect. 4, semantic matrix is built from Wikipedia link structure to enrich semantic relations between terms in Wikipedia-concept-based document representation. Experimental results in short and long text classification are both presented and discussed in Sect. 5. Finally, conclusion and future work are provided in Sect. 6.

2. Related Work

Recently, a large amount of research utilizes background semantic knowledge in an external knowledge base such as Wikipedia to improve performance of text mining tasks like text categorization [1], [2], [5]–[7] and text clustering [3], [8], [9]. There are two common approaches that have been exploited: one is to enrich BOW (Bags of Words) document representation with new features derived from external knowledge base; the other is to replace BOW document representation with concept-based document representation. Our approach proposed in this paper belongs to the second one.

Gabrilovich et al. [1], [7] propose a method to enhance text categorization algorithms with features generated from Open Directory Project (ODP) and Wikipedia. They first build a feature generator that maps each document to ODP or Wikipedia concepts. Then they use a multi-resolution approach to perform the mappings, first at the level of individual words, followed by sentences, paragraphs, and finally the entire document. The feature generator generates a huge number of features and they use feature selection to eliminate the spurious ones. Finally, they use SVMs to construct a text classifier in the augmented feature space. Experimental results confirm that their methods can improve classification performance. The main difference between our method and theirs is that we directly use the mapped concepts to classify rather than using the mapped concepts to enrich document representation. The method that maps document to Wikipedia concepts is also different. They build an inverted index that maps terms into a list of concepts and use multi-resolution approach for feature generation which needs to scan each document many times. It introduces noise and it is time consuming. In our method, we directly map document words to Wikipedia concept using inverted index.

Pu Wang et al. [2] build a semantic kernel from Wikipedia for text classification. The semantic kernel is a Semantic Matrix which is formed by computing semantic relatedness between concepts based on Wikipedia articles and taxonomy. It utilizes semantic knowledge from Wikipedia to enhance understanding of natural language. The representation vector of document in this method contains not just Wikipedia concepts but also terms which are not in the semantic kernel. So if many topical terms used in a document can not be mapped to Wikipedia concepts, the performance will decrease. In our method, we use inverted index which can map all words to Wikipedia concepts and overcome this problem. The approach they use to build the semantic kernel requires high processing effort, because it utilizes not just Wikipedia articles but also taxonomy. In our method we only utilize the hyperlink structure and it is an effective and low cost measure [10].

Phan et al. [5] use large scale external data collection Wikipedia and MEDLINE to discover hidden topics depending on latent topic analysis model LDA. Then they use the hidden topics and labeled short text training data to build a classifier for classifying short texts. Empirical evaluation demonstrates that this method performs better than traditional methods which just train the classifier using labeled short text training data. Chen et al. [11] improve the method in [5] by introducing Multi-Granularity Topics method which sets different topic number in LDA and chooses the best one to discover hidden topics. Hu et al. [3] propose a text clustering method by enriching document representation with Wikipedia concept and category information. It computes the semantic similarity values between documents or between document and clustering centroid using document words vector, Wikipedia concept vector and Wikipedia category vector. They use two schemes named Exact-Match and Relatedness-Match to map document words to Wikipedia concepts and categories. In our method, the mapping scheme is similar to the Relatedness-Match, but we use all concepts rather than select top-k concept for each word to take full advantage of the semantic knowledge in Wikipedia.

Daniele Vitale et al. [6] represent document text with Wikipedia concepts using TAGME method [12] to perform the mapping for short text categorization. They utilize Wikipedia concept vector to represent the document text rather than enriching BOW representation. They select top-k Wikipedia concepts to characterize a specific category and compute semantic relatedness using WLM method [10] to evaluate which category does the input document belong to. In our method, we also use Wikipedia concept vector to represent the document text, but we utilize traditional approach SVM to perform text classification rather than utilizing WLM to measure relatedness between Wikipedia concept vectors for classification. Anna Huang et al. [13] utilize Wikipedia to text clustering without enriching the BOW representation. They first create a concept-based document representation by mapping the terms and phrases within documents to their corresponding articles (or concepts) in Wikipedia and then they develop a similarity measure for two documents based on Wikipedia articles rather than enriching BOW terms and phrases.

3. Wikipedia-Concept-Based Document Representation

In this Section, we introduce a document representation method which is based on Wikipedia concepts. We first build a weighted word-concept inverted index from Wikipedia articles that contains the relationships between each word and a list of related concepts. Then we map document terms in the BOW model to Wikipedia concepts using the inverted index. The mapped Wikipedia concept vector is used as the representation of the document text. The process of our document representation method is shown in Fig. 1.

In the first step, we build an inverted index from Wikipedia articles. In Wikipedia, each concept (or topic) is described by an article. In the BOW model an article can be represented as "bag of words" with TFIDF schema and relationships between words and related articles (or concepts) can be built from Wikipedia. A word may appear in a number of articles and then a word in the inverted index is related to a list of concepts. There is a weight value which is calculated from the articles of the related concepts using TFIDF scheme to denote the relatedness between words and Wikipedia concepts. Let $r_{c_j}^{w_i}$ be the weight value of relatedness between word w_i and Wikipedia concept c_j. Then $r_{c_j}^{w_i}$ can be calculated as Eq. (1).

$$r_{c_j}^{w_i} = tf_{article(c_j)}(w_i) \cdot idf(w_i) \tag{1}$$

where $article(c_j)$ is the article that explain concept c_j. $tf_{article(c_j)}(w_i)$ is the frequency value of word w_i in $article(c_j)$. $idf(w_i)$ is the inverse document frequency of word w_i in the whole Wikipedia articles.

We discard words whose total frequency in all articles is less than 5 because they are probably caused by spelling mistakes. We get 1,739,060 words in the inverted index which are big enough to cover almost all words in the input documents. We discard insignificant associations between words and concepts in the inverted index by removing concepts whose weight for a given word is too low.

In the inverted index we can get the relatedness between words and Wikipedia concepts. In the BOW model a document can be represented as "bag of words" and then we can map document text to Wikipedia concepts with the inverted index. A document d can be represented in the BOW model as Eq. (2):

$$vec_{word} = (tf(w_1), tf(w_2), \cdots, tf(w_n)) \in \mathrm{R}^n \tag{2}$$

where $tf(w_i)(i \in \{1, 2, \cdots, n\})$ is the frequency value of word $w_i(i \in \{1, 2, \cdots, n\})$ in document d and n is the size of the dictionary.

We can utilize the relatedness between words and Wikipedia concepts in the inverted index. For a word $w_i(i \in \{1, 2, \cdots, n\})$ in the BOW model, $S(w_i)(i \in \{1, 2, \cdots, n\})$ is the set that contains a number of concepts related to it in the inverted index. In order to improve performance and processing efficiency, we only choose top-k concepts in set $S(w_i)(i \in \{1, 2, \cdots, n\})$ for each word in the inverted index. The number of top concepts k using in our experiments is an empirical value. When the k is too small, there will be some useful concepts missed. When the k is too large, there will be a lot of noise. It's hard to find the proper value for k to keep useful concepts and do not bring in much noise. We tried the different values of k on long text datasets "Reuters-21578", "Movie Reviews" and short text datasets "Google Snippets", "Reuters-21578" to check the performance of the methods in this paper. We find that k was set to 5 in long text classification and 10 in short text classification can get best performance. In our experiment, we set k to 5 in long text classification and 10 in short text classification. All concepts in the Wikipedia-concept-based document representation will be in $\bigcup_{i=1}^{n} S(w_i)$. Document d can be represented as a vector with Wikipedia concepts as Eq. (3):

$$vec_{concept} = (\mathrm{wei}ght(c_1), weight(c_2), \cdots, weight(c_m)) \tag{3}$$

where $c_j \in \bigcup_{i=1}^{n} S(w_i)(j \in \{1, 2, \cdots, m\})$ and weight(c_j) $(j \in \{1, 2, \cdots, m\})$ is the weight value of concept $c_j(j \in \{1, 2, \cdots, m\})$ in the Wikipedia-concept-based document representation. Then the weight $\mathrm{wei}ght(c_j)(j \in \{1, 2, \cdots, m\})$ of the concept $c_j(j \in \{1, 2, \cdots, m\})$ can be calculated as Eq. (4):

$$\mathrm{wei}ght(c_j) = \sum_{i=1}^{n} tf(w_i) \cdot r_{c_j}^{w_i} \tag{4}$$

where $r_{c_j}^{w_i}$ is the relatedness between word $w_i(i \in \{1, 2, \cdots, n\})$ and concept $c_j(j \in \{1, 2, \cdots, m\})$ in the inverted index which is calculated in Eq. (1). $tf(w_i)(i \in \{1, 2, \cdots, n\})$ is the frequency value in document d. In Eq. (4), we calculate the weight value $\mathrm{wei}ght(c_j)$ of concept c_j to document d by summing up the product of the relatedness value $r_{c_j}^{w_i}$ of concept c_j to word w_i in the inverted index and frequency value $tf(w_i)$ of word w_i in the document.

For example, for a short document d "Machine learning is a branch of artificial intelligence", we can find that "machine", "leaning", "artificial" and "intelligence" are all related to concept "Machine learning". To compute the relatedness of concept "Machine learning" to document d, we first compute the frequency value of

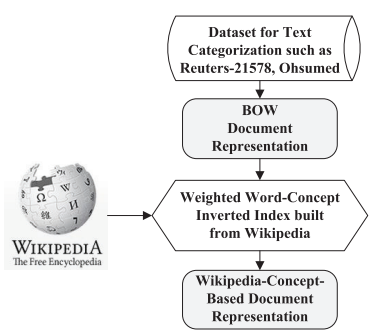

Fig. 1 Text classification using Wikipedia concept document representation.

word "machine", "leaning", "artificial" and "intelligence" in document d. Then we will compute the relatedness of "machine", "leaning", "artificial" and "intelligence" to concept "Machine learning" using Eq. (1). Finally, $weight('\text{Machine learning}')= \text{tf}('\text{machine}')\cdot r'^{\text{machine}}_{'\text{Machine learning}'} + \text{tf}('\text{leaning}')\cdot r'^{\text{leaning}}_{'\text{Machine learning}'} + \text{tf}('\text{artificial})\cdot r'^{\text{artificial}}_{'\text{Machine learning}'} + \text{tf}('\text{intelligence}')\cdot r'^{\text{intelligence}}_{'\text{Machine learning}'}$ is the relatedness weight of concept "Machine learning" to document d.

Although we only choose top-k concepts for a given word, noise is still introduced because the concepts associated to the word might have no relevance to the document text. For example, top-5 related concepts for word "iphone" are "IPhone", "iOS", "iPhone (original)", "IPhone 3G" and "IOS SDK". For a document text "I like iphone more than HTC", the associated concept "IOS SDK" will be a noise. In order to improve performance and processing efficiency, we only choose top-m concepts in a Wikipedia-concept-based vector to remove irrelevance concepts. For one thing, removing the concepts with lower weight in a Wikipedia-concept-based vector can prevent document representation from noise; for another thing, the processing efficiency is improved because of shorter size of document representation vector. The value m must be carefully chosen because it will introduce noise if the value m is too large and cause information loss if the value m is too small.

Finally, the generated Wikipedia concept vector is used as representation of the document. The semantic relationship between terms in traditional BOW model is ignored and it limits the performance. The terms in Wikipedia-Concept-based document representation method are Wikipedia concepts and we can utilize rich semantic knowledge between Wikipedia concepts to improve performance. In the next Section, we will introduce our method to overcome this weakness by building a semantic matrix using WLM method which is based on Wikipedia link structure [10].

Let sentence "Obama is the president of the United States" be an example for Wikipedia concept based document representation. There are four words after stemming and removing stop words: "Obama", "president", "United" and "State". Top-5 concepts of the sentence terms in the inverted index are shown in Table 1. In the Wikipedia-concept-based document representation schema, the concepts of the sentence terms in Table 1 are used to represent the sentence. "Barack Obama" and "United States" are both related to two words in the sentence. The weights of 'Barack Obama" and "United States" will be higher because the weights are added from two words. The method for computing weight of concepts is shown in Eq. (4).

4. Building Semantic Matrix Using WLM Method

Like traditional BOW model, the semantic relationship between concepts is ignored in the Wikipedia concept document representation model discussed in Sect. 3. In order to improve performance of text classification, semantic relationship between concepts must be considered. In this Section, we build a semantic matrix to enrich semantic relationship between Wikipedia concepts using effective WLM method [10].

David Milne and Ian H. Witten propose WLM method which is an effective and low cost measure for obtaining semantic relatedness between Wikipedia concepts. It only utilizes the hyperlink structure of Wikipedia rather than textual content or category taxonomy and gives excellent performance. Although ESA [4] remains the best measure in terms of robustness, WLM is able to match it's accuracy when the document representation is based on Wikipedia concepts. We choose WLM method to compute semantic relatedness for building semantic matrix because of it's good characteristics. First, our document representation method is based on Wikipedia concepts rather than common words, so WLM method can get the best performance as good as ESA method. Second, WLM method is more effective than ESA method. It's a low cost and effective method for building semantic matrix.

We can now define a semantic matrix P for Wikipedia concepts. The semantic matrix P is a symmetrical matrix represented in Fig. 2. The elements in P are defined as follows. For any concept $c_i (i \in \{1, 2, \cdots, n\})$ and $c_j (j \in \{1, 2, \cdots, n\})$, the semantic relatedness r_{ij} is calculated by the methods shown in Eq. (5).

$$\text{r}_{ij} = \begin{cases} 1 & if\ c_i\ and\ c_j\ are\ synonyms; \\ r_{WLM} & otherwise. \end{cases} \tag{5}$$

Semantic relatedness r_{WLM} is calculated with WLM method. In Wikipedia, synonyms are linked together by redirect links and we utilize the redirect links to check if

Concepts \ Concepts				
	1	r_{12}	$\cdots$	r_{1n}
	r_{21}	1	$\cdots$	r_{2n}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	r_{n1}	r_{n2}	$\cdots$	1

Fig. 2 Semantic matrix P.

Table 1 Top-5 concepts of document terms in the inverted index.

Word	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5
Obama	Barack Obama	Presidency of Barack Obama	Family of Barack Obama	Barack Obama presidential campaign, 2008	Obama, Fukui
president	President	President of the United States	President of France	George W. Bush	Barack Obama
United	United	United!	United Airlines	United States	United Ireland
State	State	U.S. state	State highway	United States	State (polity)

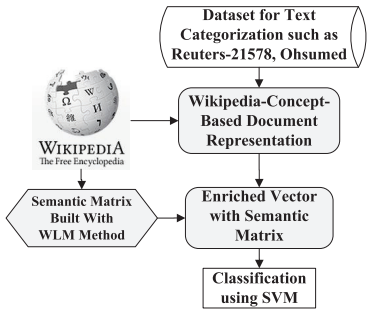

Fig. 3 Text classification using enriched document vector with semantic matrix.

two concepts are synonyms. For example, there is a redirect link from concept "USA" to concept "United States". In Wikipedia, there are more than five million redirect links, so we can get a large number of synonyms from the redirect links.

We enriched the Wikipedia-concept-based vector representation of documents with the semantic matrix P. Let $vec_{concept}$ be the representation of a document based on Wikipedia concepts and vec_{SM} be the vector enriched by the semantic matrix P. Then the extended vector vec_{SM} can be computed as Eq. (6).

$$vec_{SM} = vec_{concept} \cdot P \tag{6}$$

In vector vec_{SM}, semantic relations between elements are enriched with semantic matrix. In text classification, semantically similar documents should be mapped to nearby positions in feature space. Using this semantic transformation, the corresponding vector space kernel takes the form shown in Eq. (7) below:

$$\begin{aligned} \tilde{k}(d_1, d_2) &= vec_{concept}(d_1)PP^T vec_{concept}(d_2) \\ &= vec_{SM}(d_1)vec_{SM}(d_2) \end{aligned} \tag{7}$$

Thus, the inner product between document d_1 and d_2 in SVM feature space can be efficiently computed using the semantic matrix. The process that utilizes enriched Wikipedia-concept-based vector of a document to text classification is shown in Fig. 3. First, we build Wikipedia concept vector of a document using the method described in Sect. 3. Then, we utilize semantic matrix which is built from Wikipedia to generate enriched document representation vector. Finally, the enriched vector is normalized and input into SVM for classification.

5. Experimental Evaluation

5.1 Wikipedia Dump Data

Wikipedia database dumps are released periodically and the dumps can be downloaded from website http://dumps.wikimedia.org/. The version of Wikipedia dump that we used in this paper is enwiki-20120902. We imported the SQL and XML dump files to mysql database and obtained more than 130 GB of data. The statistics of Wikipedia data is shown in Table 2.

Table 2 Statistics of Wikipedia dump.

Content	Size
Concepts	9,618,661
Articles	4,090,633
Links in Articles	380,692,384
Redirect Links	5,658,860

We process the text of Wikipedia articles by removing stop words and rare words whose total frequency in all articles is less than 5. We stem all text with "Lucene Snowball". We remove links whose target page does not exist due to the fact that there are unedited concepts that are cited by other articles. Every concept is redirected to it's target concept which is described by an article using redirect links in all our experiments.

5.2 Datasets and Experimental Methodology

We used full text of the four real datasets (Reuters-21578, OHSUMED, 20 Newsgroups, and Movies Reviews) to evaluate our approaches in long text classification. Like what has been done in [1], to evaluate the performance of our methods in short text classification, only document titles of the datasets described above was taken to create short text datasets (with the exception of Movie Reviews, where documents have no titles). We also used a dataset available to community for short text classification named Google Snippets to evaluate our methods. A short description of each dataset is provided in the sequel.

1. Reuters-21578. This is a collection of documents that appeared on Reuters newswire in 1987 and it is one of the most widely used for text classification. We used the ModApte Split method to split the training and testing documents. We used 10 top-sized categories for evaluation and got 7156 documents for training and 3211 documents for testing. In this dataset, a document may belong to more than one category and the document will be in all categories it belongs to.

2. OHSUMED [14]. This collection includes titles and/or medical abstracts from 270 medical journals over a five-year period (1987-1991). Following Joachims [15], we used the 20,000 documents and took the first 10,000 documents for training and the following 10,000 for testing. In the 20,000 documents, they all contain abstracts. There are 23 diseases categories in OHSUMED dataset. In the 23 categories, the number of documents in different categories varies from each other. Category "Pathological Conditions, Signs and Symptoms" contains 1799 documents and category "Bacte-rial Infections and Mycoses" contains just 423 documents for training.

3. 20 Newsgroups [16]. It contains 20,000 articles for 20 categories (about 1000 documents each class). The articles are taken from the "Usenet" newsgroups collection. We used the subject and the body of each message only. For training and testing, a 4-fold cross-validation method was implemented.

Table 3 Precision results for long documents.

Datasets	*Baseline*		*Wiki-Replacing*		*Wiki-SM*		*Wiki-SK*	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
OHSUMED	0.4712	0.4181	0.5351	0.5705	0.5458	0.5887	0.5324	0.5613
Reuters-21578	0.8573	0.7351	0.8328	0.7012	0.8631	0.7412	0.8614	0.6221
Movie Reviews	0.8475	-	0.8445	-	0.8743	-	0.8635	-
20 Newsgroups	0.8091	-	0.8127	-	0.8235	-	0.8223	-

4. Movie Reviews [17]. This collection contains 2000 reviews of movies with 1000 reviews expressing a positive opinion and 1000 reviews expressing a negative opinion about the movies. It is released by Pang and Lee [18] in 2004. Like what has been done in [2], we performs 4-fold cross-validation in this corpora for training and testing.

5. Google Snippets [5]. This labeled collection was retrieved from Google search using JWebPro by [5] and it is composed of 12,000 (10,000 for training and 2,000 for testing) snippets. Snippets have length of about 13 terms (on average) and are labeled with 8 categories. This collection was used to evaluate the performance of our method on short text classification.

Six methods were implemented to evaluate their performance: the baseline method that is based on BOW model, the replacing method described in Sect. 3, the semantic matrix method shown in Sect. 4, Wiki-SK method proposed in [2], Daniele's method proposed in [6] and Phan's method proposed in [5].

1. Baseline method. The baseline method is based on the BOW model. TF-IDF weighting scheme is used for computing weight of terms. We use F-score [19] for feature selection. Some preprocessing methods such as discarding stop words, removing rare words, stemming and normalizing were used in our implementation.

2. Wiki-Replacing method. This method is described in Sect. 3. We are not the first who propose to utilize Wikipedia concept vector to represent document text. Daniele Vitale et al. used this method in [6], but we used different mapping method and classification method. We used inverted index to map document terms with Wikipedia concepts while they used TAGME. Compared to TAGME, the Inverted Index method is more time consuming, but it helps identify relevant Wikipedia concepts which are not explicitly present in a document. It is especially useful when Wikipedia concepts have less coverage for a dataset. We used SVM to classify rather than their semantic distance based method which does not rely on any learning method like SVM.

3. Wiki-SM method. This method is based on the Wikipedia-concept-based document representation in Wiki-Replacing method. The Wikipedia concept vector is enriched by the semantic matrix described in Sect. 4.

4. Wiki-SK method. This method is proposed in [2]. They build a semantic kernel of Wikipedia concepts for text classification, but their method for building semantic kernel is based on Wikipedia articles and taxonomy rather than Wikipedia link structure. In our document representation schema, there are only concepts rather than document terms.

5. Daniele's method. This method is proposed in [6] for short text classification. They use Wikipedia concepts to represent short text using TAGME method [12]. They select top-k Wikipedia concepts to characterize a specific category and compute semantic relatedness using WLM method [10] to evaluate which category the input document belongs to.

6. Phan's method. Phan et al. [5] use large scale external data collection Wikipedia and MEDLINE to discover hidden topics depending on latent topic analysis model LDA. Then they use the hidden topics and labeled short text training data to build a classifier for classifying short text.

Support Vector Machine (SVM) with a linear kernel was used to learn models for text categorization as it can get state of the art results [20]. SVM is a supervised learning model for classification and regression analysis. An open source implementation of SVM named LIBSVM [21] was used in all our experiments. LIBSVM is an integrated software for support vector classification and has been widely used in many papers like [2] and so on. Main features of LIBSVM include different SVM formulations, efficient multi-class classification, cross validation for model selection, Various kernels (including precomputed kernel matrix) and so on. LIBSVM is among the first SVM software to handle multi-class data. LIBSVM supports different kinds of multi-class svms and the default is 1-vs-all without additional options[†].

We evaluated text categorization performance of the methods described above using micro-averaged precision and macro-averaged precision [22]. Micro-averaged precision score gives equal weight to every document while macro-averaged precision score gives equal weight to every class. If the categories in data set differ in size substantially, it's meaningful to compute macro-averaged precision because the micro-averaged precision varies from macro-averaged precision. But if the categories have little difference in size, the micro-averaged precision and macro-averaged precision will nearly the same. In our experiments, data set Reuters-21578 and OHSUMED's categories differ in size substantially, so we have to compute macro-averaged precision and micro-averaged precision. For data set 20 Newsgroups and Movie Reviews, the categories' size is nearly the same. The micro-averaged precision and macro-averaged precision will nearly be equal to each other. So we did not calculate the macro-averaged precisions and let them to be null in Table 3 and Table 4.

[†]LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm

5.3 Experimental Results on Long Text Classification

Table 3 shows micro-averaged precision and macro-averaged precision results for the four methods on the four long text datasets. Method Wiki-Replacing gives higher micro and macro precision on dataset OHSUMED and 20 Newsgroups, but a little lower on Reuters-21578 and Movie Reviews. We use significant test to indicate statistically significant improvements over the other method. T-test is used in our paper. The p-value of Wiki-Replacing method and baseline is 0.6341. Wiki-Replacing method is true to not be always better than the BOW models and Wiki-Replacing method can give more or less the same results as the baseline method. Our method Wiki-SM gives higher micro and macro precision on all four datasets comparing with the baseline method. The p-value of Wiki-SM method and baseline is 0.1420. Although the p-value is not lower than the threshold chosen for statistical significance (usually the 0.10, the 0.05, or 0.01 level), it's quite close to the 0.10 level. Our method Wiki-SM is significantly better than the baseline method can be accepted with high probability. The results show the benefit of utilizing semantic knowledge in Wikipedia for text categorization. Wiki-SK method [2] is better than the baseline method on all the four datasets. The p-value of Wiki-SK method and baseline is 0.2156. Wiki-SK method gives worse performance than our Wiki-SM method on all the four datasets. The p-value of Wiki-SK [2] method and Wiki-SM is 0.3910. There is no significant improvement. Our method Wiki-SM is more effective and lower cost than the Wiki-SK method. Our Wiki-SM method utilize the hyperlink structure rather than Wikipedia articles and taxonomy for building semantic kernel.

5.4 Experimental Results on Short Text Classification

We conjectured that our methods described in Sect. 3 and Sect. 4 might be useful for short text classification. Like what has been done in [1], we derived several short document datasets from dataset Reuters-21578, OHSUMED and 20 Newsgroups. We replaced the full document text with only document title in these datasets to construct short text datasets. We removed the documents whose document title is less than five words. Table 4 shows the average length of all short document datasets.

Table 4 Average length of short document datasets.

Datasets	Average length
20 Newsgroups	5.73
OHSUMED	8.97
Reuters-21578	6.02
Google Snippets	17.99

Table 5 shows the precision results for the four methods on the four short text datasets. Method Wiki-Replacing gets higher precision than the baseline method and method Wiki-SM gets highest precision on all four datasets. The improvement is significant in the real short text dataset "Google Snippets" which is got from Google search results. The p-value of the baseline method and Wiki-Replacing method in t-test is 0.1121. Although it is larger than statistical significance 0.1, it's quite close to the 0.10 level. In the Wiki-Replacing method, the length of the Wikipedia-concept-based vector is longer than the BOW vector and the semantic knowledge can be fully represented. The semantic relationships between Wikipedia concepts are added to the document representation in the Wiki-SM method. Wiki-SM method gives higher precision values than the baseline method and Wiki-Replacing method on all datasets. The p-value of the baseline method and Wiki-Replacing method in t-test is 0.0543. Wiki-SM method shows statistically significant improvements over the baseline method. The p-value of Wiki-Replacing method and Wiki-SM method in t-test is 0.1209. It's also quite close to the statistical significance 0.10. The results show that the semantic knowledge in Wikipedia can be used to better understand document content for text classification. Our Wiki-SM method gives better performance than the Wiki-SK method on all the four datasets. The p-value of our Wiki-SM method and Wiki-SK method in t-test is 0.0730. So our method shows statistically significant improvements over the Wiki-SK method on short text classification when the statistical significance equals 0.1.

Figure 4 shows the micro-averaged precision results of Wiki-SM method, Wiki-SK method [2], Daniele's method [6] and Phan's method [5]. The results are changed with different sizes of labeled training data on "Google Snippets" dataset. We can find that our method Wiki-SM gives better performance than Wiki-SK method and Daniele's method in all different sizes of labeld trainning data. Phan's method gives the highest micro-averaged precision. But Phan's method is hard to construct large-scale data collections for training in a task. The practical usage of this method is limited. Our method is common and can be used in any text classification tasks.

5.5 Concept Number in Document Representation Vector

In Sect. 3, we choose top-m concepts in a Wikipedia-

Table 5 Precision results for short documents.

Datasets	*Baseline*		*Wiki-Replacing*		*Wiki-SM*		*Wiki-SK*	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
OHSUMED	0.4008	0.3262	0.4109	0.3464	0.4314	0.3513	0.4203	0.3321
Reuters-21578	0.6880	0.2170	0.6972	0.2890	0.7095	0.2896	0.6910	0.2832
20 Newsgroups	0.6924	-	0.7187	-	0.7242	-	0.7195	-
Google Snippets	0.6673	0.6341	0.7271	0.7001	0.7335	0.7153	0.7077	0.6828

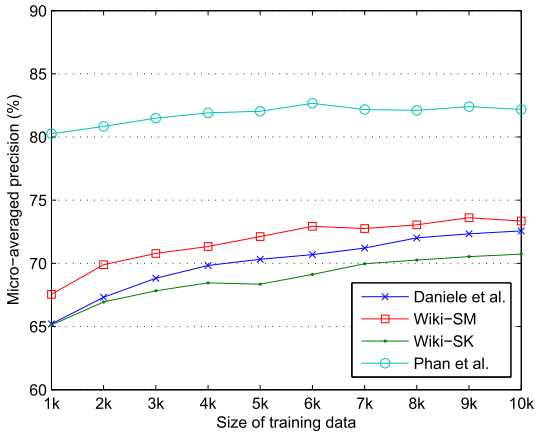

Fig. 4 Evaluation with different sizes of labeled training data on Google Snippet dataset.

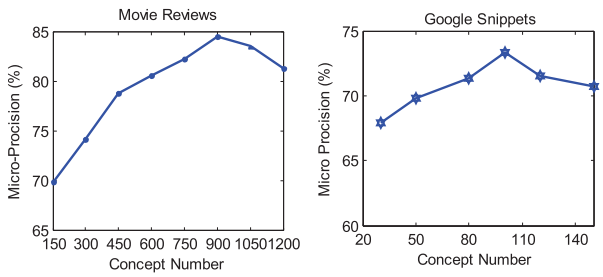

Fig. 5 Micro-Precision with different concept number in document representation vector.

concept-based document representation vector to remove irrelevance concepts. Figure 5 shows the number of Wikipedia-concept-based document representation vector influences the performance of text classification on long text dataset "Movie Reviews" and short text dataset "Google Snippets". In Wikipedia-concept-based document representation schema it will introduce noise if the number of concepts in the document representation vector m is too large and cause information loss if the value m is too small. If m is too large, a lot of noise will be introduced and the performance of text categorization method will decrease. If m is too small, it will cause information loss and the performance of text categorization method will also decrease. Experimental results in Fig. 5 confirms it. In Fig. 5, when the number of top concepts m equals 900 in dataset "Movie Reviews" and 100 in dataset "Google Snippets", the Wiki-Replacing method give the best performance. For different dataset, the best value of m is different. Experiments on other datasets and Wiki-SM method get the same conclusion. The number of document representation vector m must be carefully chosen and we chose it empirically in all our experiments.

6. Conclusion

In this paper, we present a new method for improving performance of text classification by leveraging semantic knowledge in Wikipedia. Based on the inverted index built from the content of Wikipedia articles, document text can be represented as Wikipedia concept vector. In Wikipedia, concepts are linked together with hyperlinks which show the semantic relationships between them. A semantic matrix is constructed using effective WLM method which is based on Wikipedia link structure to enrich semantic relationships between concepts in the Wikipedia-concept-based document representation. The enriched Wikipedia-concept-based document representation is then used for text classification in SVM. Experimental results on the five long and short text datasets shows that our method Wiki-SM gives better performance than the BOW method in all datsets. That means the enriched Wikipedia-concept-based document representation method that utilizes semantic knowledge in Wikipedia can get better performance than the traditional BOW model. Wiki-SM gives better performance than the recently developed Wiki-SK method on long text classification. Although Wiki-SM method gives lower performance than the state-of-the-art Phan's method on short text classification, the practical usage of Phan's method is limited and our Wiki-SM method can be used in common.

We believe that our Wikipedia-concept-based document representation method that represents document text with enriched Wikipedia concept vector can be used in other applications like document similarity measurement, document clustering and information retrieval. In future, we will try to further utilize semantic knowledge in Wikipedia for text classification and clustering.

References

[1] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge," Proc. 21st National Conference on Artificial intelligence, AAAI'06, pp.1301–1306, Boston, Massachusetts, 2006.

[2] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'08, pp.713–721, Las Vegas, Nevada, USA, 2008.

[3] X. Hu, X. Zhang, C. Lu, E.K. Park, and X. Zhou, "Exploiting Wikipedia as external knowledge for document clustering," Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, pp.389–396, Paris, France, 2009.

[4] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," Proc. 20th International Joint Conference on Artifical Intelligence, IJCAI'07, pp.1606–1611, Hyderabad, India, 2007.

[5] X.H. Phan, L.M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," Proc. 17th International Conference on World Wide Web, WWW '08, pp.91–100, Beijing, China, 2008.

[6] D. Vitale, P. Ferragina, and U. Scaiella, "Classification of short texts by deploying topical annotations," Proc. 34th European Conference on Advances in Information Retrieval, ECIR'12, pp.376–387, Barcelona, Spain, 2012.

[7] E. Gabrilovich and S. Markovitch, "Feature generation for text categorization using world knowledge," Proc. 19th International Joint Conference on Artificial Intelligence, IJCAI'05, pp.1048–1053, Edinburgh, Scotland, 2005.

[8] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using wikipedia," Proc. 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, pp.787–788, Amsterdam, The Netherlands, 2007.
[9] J. Tang, X. Wang, H. Gao, X. Hu, and H. Liu, "Enriching short text representation in microblog for clustering," Front. Comput. Sci. China, vol.6, no.1, pp.88–101, Feb. 2012.
[10] D. Milne and I.H. Witten, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," Proc. AAAI 2008, 2008.
[11] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," Proc. Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three, IJCAI'11, pp.1776–1781, Barcelona, Catalonia, Spain, 2011.
[12] P. Ferragina and U. Scaiella, "Tagme: On-the-fly annotation of short text fragments (by wikipedia entities)," Proc. 19th ACM International Conference on Information and Knowledge Management, CIKM '10, pp.1625–1628, 2010.
[13] A. Huang, D. Milne, E. Frank, and I.H. Witten, "Clustering documents using a wikipedia-based concept representation," Proc. 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD '09, pp.628–636, Bangkok, Thailand, 2009.
[14] W. Hersh, C. Buckley, T.J. Leone, and D. Hickam, "Ohsumed: an interactive retrieval evaluation and new large test collection for research," Proc. 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '94, pp.192–201, Dublin, Ireland, 1994.
[15] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," Proc. 10th European Conference on Machine Learning, ECML '98, pp.137–142, 1998.
[16] K. Lang, "Newsweeder: Learning to filter netnews," Proc. 12th International Machine Learning Conference (ML95), 1995.
[17] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," Proc. ACL-02 Conference on Empirical Methods in Natural Language Processing - vol.10, EMNLP '02, pp.79–86, 2002.
[18] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," Proc. 42nd Annual Meeting on Association for Computational Linguistics, ACL '04, Barcelona, Spain, 2004.
[19] Y.W. Chen and C.J. Lin, "Combining svms with various feature selection strategies," in Feature Extraction, ed. I. Guyon and M. Nikravesh, Studies in Fuzziness and Soft Computing, vol.207, ch. 13, pp.315–324, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
[20] F. Sebastiani, "Machine learning in automated text categorization," ACM Comput. Surv., vol.34, no.1, pp.1–47, March 2002.
[21] C.C. Chang and C.J. Lin, "Libsvm: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol.2, no.3, pp.27:1–27:27, May 2011.
[22] Y. Yang, T. Ault, T. Pierce, and C.W. Lattimer, "Improving text categorization methods for event tracking," Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00, pp.65–72, Athens, Greece, 2000.

Xiang Wang is a Ph.D. candidate in School of Computer Science, National University of Defense Technology. His main research interests are Natural Language Processing and Text Mining.

Yan Jia was born in 1960. Professor in the School of Computer Science, National University of Defense Technology. Her main research interests include data mining and information security.

Ruhua Chen is a graduate student of Software Engineering in School of Computer of NUDT. He received a bachelor's degree from the Central South University. His current research is in text mining, data storage.

Hua Fan received the B.S. and M.S. degrees in Computer Science from National University of Defense Technology. He is currently a Ph.D. student at National University of Defense Technology. His research interests include stream data management and Sensor network.

Bin Zhou was born in 1971. Professor in the School of Computer Science, National University of Defense Technology. His main research interests include text mining and information security.