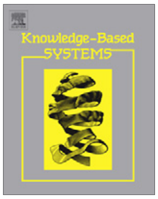
Knowledge-Based Systems xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Text classification based on multi-word with support vector machine

Wen Zhang [a,*], Taketoshi Yoshida [a], Xijin Tang [b]

[a] *School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Ashahidai, Tatsunokuchi, Ishikawa 923-1292, Japan*
[b] *Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, PR China*

ARTICLE INFO

Article history:
Received 12 December 2007
Accepted 30 March 2008
Available online xxxx

Keywords:
Text mining
Text representation
Multi-word
Feature selection
Information gain
Support vector machine

ABSTRACT

One of the main themes supporting text mining is text representation, i.e., looking for the appropriate terms to transfer the documents into numerical vectors. Recently, many efforts have been invested on this topic to enrich text representation using vector space model (VSM) to improve the performances of text mining techniques such as classification, clustering, etc. The main concern of this paper is to investigate the effectiveness of using multi-words for text representation on the performances of classification. Firstly, a practical method is proposed to implement the multi-word extraction from documents based on the syntactical structure. Secondly, two strategies as general concept representation and subtopic representation are presented to represent the documents using the extracted multi-words. Especially, the dynamic *k*-mismatch is proposed to determine the presence of a long multi-word which is a subtopic of the content of a document. Finally, we carried out a series of experiments on classifying the Reuters-21578 documents using the representations with multi-words, respectively. We used the performance of representation in individual words as the baseline, which has the largest dimension of feature set for representation without linguistic preprocessing. Moreover, linear kernel and non-linear polynomial kernel in support vector machines (SVM) are examined comparatively for classification to investigate the effect of kernel type on the performance of classification. And the index terms with low information gain (IG) are removed from the feature set at different percentage to observe the robustness of each classification method. Our experiments demonstrate that in multi-word representation, subtopic of general concept representation outperforms the general concept representation and the linear kernel outperforms non-linear kernel of SVM in classifying the Reuters data. And the effect of applying different representation strategies is greater than the effect of applying the different SVM kernels on classification performance. Furthermore, the representation using individual words outperforms any representation using multi-words. This is consistent with the most opinions concerning the role of linguistic preprocessing on documents' features when using SVM for classification.

1. Introduction

With the rapid growth of online information, text classification has become one of the key techniques for handling and organizing text data. Automated text classification utilizes a supervised learning method to assign predefined category labels to new documents based on the likelihood suggested by a trained set of labels and documents. Text representation, which is the process of transforming the unstructured texts into structured data as numerical vectors which can be handled by data mining techniques, is of strong impact on the generalization accuracy of a learning system. Usually, bag of words (BOW) in vector space model [1] is used to represent the text using individual words obtained from the given text data set. As a simple and intuitive method, BOW method makes the representation and learning easy and highly efficient as it ignores the order and meaning of individual words. But it is also criticized that the information patterns discovered by BOW are not interpretable and comprehensible because the linguistic meaning and semantics are not integrated into representation of documents. To address this problem, three types of representation methods at different semantic levels are proposed recently as follows.

- Ontology enhanced representation. That is, using ontology to capture the concepts in the documents and integrate the domain knowledge of individual words into the terms for representation. For instance, Hotho et al. developed different types of methods to compile the background knowledge embodied in ontologies into text documents representation and improved the performance of document clustering [2]. Such kind of works also can be found in [3,4].

[*] Corresponding author. Tel.: +81 80 3049 6798.
E-mail addresses: zhangwen@jaist.ac.jp (W. Zhang), yoshida@jaist.ac.jp (T. Yoshida), xjtang@amss.ac.cn (X. Tang).

0950-7051/$ - see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.knosys.2008.03.044

- Linguistic unit enhanced representation. This method makes use of lexical and syntactic rules of phrases to extract the terminologies, noun phrases and entities from documents and enrich the representation using these linguistic units. For instance, Lewis compared the phrase-based indexing and word-based indexing for representation for the task of document categorization [5]. His result showed that the phrase indexing can not improve the categorization in most cases because of the low frequencies of most phrases. Such kind of work can also be found in [6] which used multi-words to improve the effectiveness of text-retrieval system.
- Word sequence enhanced representation. This method ignores the semantics in documents and treats the words as string sequences. Text representation using this method is either on words' group based on co-occurrence or a word sequence extracted from documents by traditional string matching method. In this aspect, Li used the generalized suffix tree to extract the frequent word sequences from documents and used the frequent word sequences for text representation to propose the CFWS clustering algorithm [7]. Similar work can be found in [8–10]. Particularly, the *N*-gram-based representation [11] can also be categorized as this type for it also ignores the semantics and meaning of individual words.

In this paper, our primary concern is on the second one mentioned above, that is, using the linguistic unit to enhance text representation. Briefly, we investigate which beneficial effects can be achieved for text documents classification if the multi-word, which is regarded as including contextual information of individual in, is used as the feature for representation. In details, a multi-word extraction method is developed based on the syntactical rules of multi-word firstly. Then, documents are represented with these multi-words using different strategies. Finally, a series of experiments are designed to examine the performances of text classification methods in order to evaluate the effectiveness of multi-word representation.

The rest of this paper is organized as follows. Section 2 is the preliminaries of the techniques used in this paper and related work. Section 3 describes the developed multi-word extraction method and the details of using multi-words for text representation. Section 4 is the experiments we carried out to examine the multi-word representation methods on classifying the Reuters-21578 data set. Finally, concluding remarks and further research plans are indicated in Section 5.

2. Preliminaries and related work

The basic ideas of the techniques used in this paper are specified and the related work of each topic is introduced.

2.1. Multi-word

A word is characterized by the company it keeps [12] and the closer a set of interesting terms, the more likely they are to indicate relevance [13]. That means not only the individual word but also its contextual information embodied in its close terms should be emphasized for further processing. This simple and direct idea motivates the research on multi-words, which is expected to capture the context information of the individual words. Although multi-word has no satisfactory formal definition, it can be defined as a sequence of two or more consecutive individual words, which is a semantic unit, including steady collocations (e.g. proper nouns, terminologies, etc.) and compound words [14]. Usually, it is made up of a group of individual words, and its meaning is either changed to be entirely different from (e.g. collocation) or derived from the straight-forward composition of the meanings of its parts (e.g. compound phrase). In fact, there are some overlappings between multi-word, collocation, terminology and similar concepts for describing unique lexical units in natural language. For this reason, the definition of multi-word varies according to different purposes [15–19], while the fundamental idea behind these concepts is the same, that is, to find a more meaningful and descriptive lexical unit than the individual word from documents.

Multi-word has many potential applications in language engineering. For instance, in natural language generation by computer, multi-words are used to make the output of computer sound as natural as human utterance. In practical applications of speech recognition and optical character recognition, multi-words can be used to disambiguate the vagueness of recognized characters through constructing statistical language model for character prediction. Also multi-words can be used in syntactical parsing, computational lexicography, etc. for various applications.

2.2. Text classification

Text classification, namely text categorization, is defined as assigning predefined categories to text documents, where documents can be news stories, technical reports, web pages, etc., and categories are most often subjects or topics, but may also be based on style (genres), pertinence, etc. Whatever the specific method employed, a text classification task starts with a training set $D = (d_1, \ldots, d_n)$ of documents that are already labeled with a category $L \in C$ (e.g. sports, politics). The task is then to determine a classification model as Eq. (1) which is able to assign the correct class to a new document d of the domain.

$$f : D \rightarrow C \quad f(d) = L \tag{1}$$

To measure the performance of a classification model, a random fraction of the labeled documents is set aside and not used for training. We may classify the documents of this test set with the classification model and compare the estimated labels with true labels. The fraction of correctly classified documents in relation to the total number of documents is called accuracy, and is a basic performance measure.

Recently, various kinds of research on text classification have been conducted regarding its applications. For instance, Adeva and Atxa applied Naive bayes (NB), *k*-nearest neighbour (KNN) and Racchio classifiers to learn the characteristics of both normal and malicious user behaviors from the log entries generated by the web application server and the performance of each classifier was compared [20]. They reported that NB outperformed the other two by more than one percent on both the macro- and micro-average *F*-measure. Zhang and Jiao developed an associative classification-based recommendation system for customer profile personalization in B2C e-commerce to predict customer requirements according to the sales records stored in database by evolving the traditional association rule [21]. The regular linear least-squares fit (LSSF) algorithm was used in Hissa et al. for the automatic classification of texts whose contents concern the nursing care narratives of some diseases [22]. Their results indicated that the free text in nursing documentation can be automatically classified, and this can offer a way to develop electronic patient records. Yang and Liu used many kinds of statistical learning methods such as SVM, neural network (NNet), etc., on the Reuters-21578 text classification [23]. They reported that SVM, KNN and LLSF outperform NNet and NB when the number of positive training instances per category is small (less than 10), and that all the methods perform comparably when the categories are sufficiently common (over 300 instances per category).

Please cite this article in press as: W. Zhang et al., Text classification based on multi-word with support vector machine, Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.03.044

2.3. Feature selection with information gain

A major difficulty of text categorization is the high dimensionality of the feature space and most of the features (i.e., terms) are irrelevant to the classification task or redundant. This makes it is highly desirable that to reduce the native space without sacrificing classification accuracy. Generally, automatic feature selection methods include the removal of non-informative terms according to corpus statistics, and the construction of new features which combine lower-level features into higher-level orthogonal dimensions. Yang and Pedersen [24] compared five methods for feature selection as information gain (IG), document frequency (DF), term strength (TS), mutual information (MI) and χ^2-test (CHI) on the task of text categorization using k-nearest-neighbor (KNN) and linear least-squares fit mapping (LLSF). They reported that IG and CHI are the most effective methods in feature selection. Especially for IG, 98% removal of unique terms yields text classification accuracy up to 89.2% with the Reuters-22173 data set. IG is defined as the expected reduction in entropy caused by partitioning the objects according to an attribute. The formula of IG is as Eq. (2) and the formula of entropy is as Eq. (3).

$$\mathrm{Gain}(S,A) = \mathrm{Entropy}(S) - \sum_{v \in \mathrm{Value}(A)} \frac{|S_v|}{|S|} \mathrm{Entropy}(S_v) \tag{2}$$

$$\mathrm{Entropy}(S) = \sum_{i=1}^{c} -p_i \log p_i \tag{3}$$

S is the collection of the labels of all objects, Value(A) is the set of all possible values for attribute A, S_v is the subset of S for which A has value v, c is the number of categories of all objects, and p_i is the proportion of the objects which belong to category i.

IG is employed in this paper to rank the terms according to their discriminative power to categorize the documents. In order to observe the effectiveness of different representation methods, the features set for representation is established at different removal percentage of the low IG value terms. If the performance of a classification based on one of representation methods is kept stable when more and more terms with low IG value is removed from the feature set, the representation method being used can be regarded as more reliable and robust than those can not keep the performance stable. That is, the current representation method has more powerful ability to capture the features with high statistical quality from the documents.

2.4. Support vector machine

SVM is a relatively new learning approach introduced by Vapnik in 1995 for solving two-class pattern recognition problem [25,26]. The method is defined over a vector space where the problem is to find a decision surface that "best" separates the data into two classes. For linearly separable space, the decision surface is a hyperplane which can be written as

$$wx + b = 0 \tag{4}$$

where x is an arbitrary objects to be classified; the vector w and constant b are learned from a training set of linearly separable objects. SVM was proposed that it is equivalent to solve a linearly constrained quadratic programming problem as Eq. (5) so that the solution of SVM is always globally optimal.

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + C\sum_i \xi_i \tag{5}$$

with constraints

$$y_i(x_i w + b) \geqslant 1 - \xi_i \quad \xi_i \geqslant 0,\ \forall i \tag{6}$$

For linearly inseparable objects, the original input data is transformed into a higher dimensional space using a non-linear mapping and the linearly separating hyperplane can also be found in the new space without increasing the computation complexity of the quadratic programming problem by employing kernel function [27]. That is, for the linearly inseparable problem, in order to compute the similarities between the vectors in higher dimensional space, kernel function is used to derive these similarities in the original lower dimensional space.

Considering the multi-class classification in this paper, the One-Against-the-Rest approach was adopted. With this method, k-class pattern recognition was regarded as a collection of $\frac{k(k-1)}{2}$ binary classification problems. The kth classifier constructs a hyperplane between class n and the other $k-1$ classes. A majority vote across the classifier or some other measures can be applied to classify a new point. In addition, other methods for k-classes ($k > 2$) classification are also discussed in [28] such as error-correcting output codes, SVM decision tree, etc.

3. Text representation using multi-word

A method for multi-word extraction is implemented based on syntactical structure of technical terminology proposed by Justeson [14]. Two strategies are developed to use the extracted multi-words to represent the documents at different semantic level.

3.1. Multi-word extraction

Generally speaking, there are mainly two types of methods developed for multi-word extraction. One is the linguistic method, which utilizes the structural properties of phrases in sentence to extract the multi-words from documents [14,29,30]. For instance, Smadja used the relative offset of two words' positions occurring in all the documents of a corpus to determine whether or not they constitute a multi-word [16]. And the results showed that his method works well for fixed phrases but it cannot cope with the long multi-word. The other is the statistical method, based on corpus learning with MI for word occurrence pattern discovery. For instance, Zhang et al. propose a method based on the adaptation of MI and context dependency for compound words extraction from very large Chinese Corpus, and they report that their method is efficient and robust for Chinese compounds extraction [17]. But, their method involves many heuristics involved to determine the context dependency of a word pair, and the parameters in their setting are so complex so that their model and not produce a robust performance. Some other methods also combine both linguistic knowledge and statistical computation for multi-word extraction such as in [31,32,18].

For simplicity, the multi-word extraction method used in this paper is as Justeson and Katz's regular expression for multi-word noun phrases as follows [14].

$$((\mathrm{A} \mid \mathrm{N})^{+} \mid (\mathrm{A} \mid \mathrm{N})^{*}(\mathrm{NP})^{?}(\mathrm{A} \mid \mathrm{N})^{*})\mathrm{N} \tag{7}$$

where A is an adjective, N is a noun and P is a preposition. Taking the following English sentence for example, by this method "*U.S. agriculture department*" (NNN), "*U.S. agriculture*" (NN), "*agriculture department*" (NN), "*last December*" (AN), "*sugar import quota*" (NNN), and "*short tons*" (AN) will be extracted from this sentence.

- The U.S. agriculture department last December slashed its 12 month of 1987 sugar import quota from the Philippines to 143,780 short tons from 231,660 short tons in 1986.

But a problem existing here is that there are too many word sequences satisfying the criterion of the above regular expression.

Please cite this article in press as: W. Zhang et al., Text classification based on multi-word with support vector machine, Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.03.044

```
Input:
  s1, the first sentence
  s2, the second sentence
Output:
  Multi-word extracted from s1 and s2.
Procedure:
  s1 = {w1,w2,…,wn}, s2 = {w1',w2',…,wm'}, k=0
  For each word wi in s1
      For each word wj in s2
            While(wi equal to wj)
              k++
            End while
            If k>1
              extract the words from wi to wi+k to form a multi-word candidate
              k = 0
            End if
      End for
   End for
```

Algorithm 1. Repetition pattern Extraction from Sentences

For this reason, Justeson and Katz proposed another criterion as repetition of the multi-word candidates because it is unusual for the multi-word which is not about the topic of a document to repeat more than two times in the document. In addition, from stand of view of text categorization, if the frequency of a term is too small, this term will have no discriminative power to categorize the documents. To reduce the computation complexity, our trick is to extract the repetition pattern of two sentences in a documents firstly and then using lexical tool to conduct the part of speech[1] for the extracted pattern. For example, if we have two sentences as follows:

- Standard oil co and bp north America inc said they plan to form a venture to manage the money market borrowing and investment activities of both companies.
- The venture will be called bp/standard financial trading and will be operated by standard oil under the oversight of a joint management committee.

From the above two sentences, "*standard oil*" will be extracted as the repletion pattern. After the part of speech processing, it will be extracted as a multi-word to represent the documents in the text collection. The basic idea of finding the repetition pattern from two sentences is string matching. For example, assuming we have two sentences as s_1 is {A B C D E F G H} and s_2 is {F G H E D A B C} where a capital character represents an individual word in a sentence as is shown in Fig. 1, the individual words in s_2 will be used to match the individual words in s_1 one by one and the same patterns in these two sentences will be fetched out for simplicity. The computation complexity of this algorithm is O(mn) where m and n is the length of s_1 and s_2, respectively. We also know that this algorithm could be improved using Knuth-Morris-Pratt's idea as KMP flowchart [33] with complexity as O($m + n$).

The algorithm we implemented to extract the repetition pattern from two sentences in a document is shown in Algorithm 1.

3.2. Text representation

After extracting multi-words from documents, we need to represent the documents using these multi-words. Actually, there are some overlappings among the extracted multi-words such as "*U.S. agriculture*", "*agriculture department*" and "*U.S. agriculture department*" and they refer to the concepts at different semantic level such as "*U.S. agriculture department*" is a more specific concept than the general concepts as "*U.S. agriculture*" and "*agriculture department*". Usually, short multi-words refer to the general concepts in the documents and long multi-word is a subtopic of these general topics. Following this idea, two strategies are proposed to further process the extracted multi-words and represent the documents using multi-words.

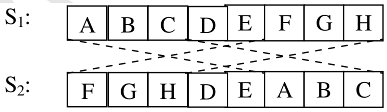


Fig. 1. Two sentences and their corresponding repetition patterns.

3.2.1. Decomposition strategy

In this strategy, only the general concepts embodied in the short multi-words are used for representation. That is, a long multi-word will be eliminated from the feature set if it can be produced by merging the short multi-words extracted from the corpus. For example, "*U.S. agriculture department*" will be eliminated from the feature set because it can be replaced by "*U.S. agriculture*" and "*agriculture department*". After normalize the multi-words into features with this strategy, the documents will be represented with term weights using the document frequencies (DF) of the multi-words because frequency is an important clue to determine the degree of relevance of a multi-word to the topic of a document, i.e., the category of a document [34].

3.2.2. Combination strategy

In this strategy, only the subtopics of the general concepts embodied in the long multi-words will be used for representation. That is, short multi-words will be eliminated from the feature set they are included in long multi-words. For example, "*U.S. agriculture*" and "*agriculture department*" will be eliminated from the feature set because they are included in "*U.S. agriculture department*".

After feature normalization for the originally extracted multi-words with this method, a crucial problem confronted with us is that how to use the long multi-word for representation. In fact, the long multi-words occur very few times in documents and they will have very small occurrence if the simple string full matching is used to match the long multi-words with the words in the

[1] The part of speech of English word is determined by WordNet2.0 which is available online: http://wordnet.princeton.edu/ obtain and Java WordNet library which is online: http://sourceforge.net/projects/jwordnet.

Please cite this article in press as: W. Zhang et al., Text classification based on multi-word with support vector machine, Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.03.044

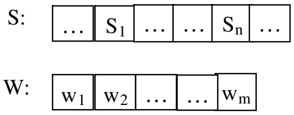


Fig. 2. The subject sentence and the word sequence of individual words which are found in the subject sentence.

$$
\begin{aligned}
&L_1 : a_{1,1}, a_{2,*1}, \ldots, a_{m,*1}\\
&L_2 : a_{1,2}, a_{2,*2}, \ldots, a_{m,*2}\\
&\ldots\\
&L_{t1} : a_{1,t1}, a_{2,*t1}, \ldots, a_{m,*t1}
\end{aligned}
$$

Fig. 3. The generated $t1$ lists using the rule of minimum difference.

documents. Nevertheless, how to deal with the occurrence of the short multi-words those are included in the long multi-words is another problem. For example, if "*U.S. agriculture department*" is used for the feature and "*U.S. agriculture*" or "*agriculture department*" are eliminated, should we regard "*U.S. agriculture department*" is approximately the same as "*U.S. agriculture*" or "*agriculture department*" for representation?

To overcome the problem in representation mentioned above, dynamic *k*-mismatch is proposed. The original *k*-mismatch is about characters and strings: given a pattern *p*, a text *t*, and a fixed number *k* that is independent of the lengths of *p* and *t*, a *k*-mismatch of *p* in *t* is a $|p|$-substring of *t* that matches $(|p| - k)$ characters of *p*. That is, it matches *p* with *k* mismatches [7]. The main idea of dynamic *k*-mismatch is to set *k* as a dynamic threshold cutoff value according to the length of pattern *p* and look for the minimum scope for the occurrences of the words in pattern *p* in *t*. In details, two heuristic parameters are set to determine whether a long multi-word occurs in a document. The first one is occurrence ratio (OR) which is the number of present individual words (a multi-word comprises at least two individual words) of pattern *p* to the total number of words of *p*, i.e., the multi-word. The second one is the minimum scope (MS) which is the minimum number of words included in a sequence which contains all the presenting individual words. For example, assuming we have a concept expressed as "*Mickey mouse*" and a sentence as follows:

- Mickey is a mouse whose name is Mickey.

We will make out that OR is equal to 1.0 because both "*Mickey*" and "*mouse*" are present in this sentence. And MS is 4 because the sequence with minimum number of words is "*Mickey is a mouse*", not "*mouse whose name is Mickey*" which has 5 words.

By the method of dynamic *k*-mismatch, the presences of long multi-words in a document are determined with the heuristic parameter setting as OR and MS. The basic idea of finding the MS of sequence as $\{w_1,w_2,\ldots,w_m\}$ in sentence $\{s_1,s_2,\ldots,s_n\}$ is based on the minimum position difference discovery for the individual words in the sequence. Assuming the subject sentence and the word sequence as shown in Fig. 2, the details of this algorithm are explained as follows. Firstly, the positions of each word w_j will be identified as $w_1 = \{a_{1,1},a_{1,2},\ldots,a_{1,t1}\}$, $\ldots,w_m = \{a_{m,1},a_{m,2},\ldots,a_{m,tm}\}$, where $a_{m,1}$ means the first occurrence position of w_m in sentence *s*. Secondly, *t*1 lists will be generated where *t*1 is the number

```
Input:
  S = {s_1, …, s_n} // S is a word sequence which combines all sentences in a text and s_i is the ith individual word;
  W = {w_1, …, w_m} // W is a multi-word and w_i is the ith individual word in W;
Output:
  //In order to describe the program clearly, we adopt W' that refers to a set of individual words in W and theses
words occur in S.
  OR --- the ratio of the number of individual words present in S to the number of the individual words included in
a multi-word, i.e., |W'|/|W|;
  MS --- minimum number of words of a sub-sequence of S that contains all the individual words in W';
Procedure:
  W' = ∅;
  For each w_i in W
       If w_i exists in S
             W' = W' ∪ {w_i}
       End if
  End for
  OR = |W'|/|W|;
  For each w_j in W'
       L_j = {a_j,k | a_j,k is the kth occurrence position for w_j occurring in S};
  End for
  // the total number of L_j is |W'|;
  For each a_i,k in L_1 // search from the first present word's positions, |W'| = |L_1|;
       L'_k = {a_i,k};
       While |L'_k| < |W'|
             next = |L'_k| +1;
             Find a_next,* in L_next which has the minimum difference from any one element in L'_k currently
             L'_k = L'_k ∪ { a_next,* }
       End while
  End for
  For each L'_k where k = 1 to |L_1|
       b_k = maximum value in L'_k – minimum value in L'_k
  End for
  MS = minimum b_j (j=1,…,|L_1|)
```

Algorithm 2. Dynamic k-mismatch to determine OR and MS of a multi-word in a document

Please cite this article in press as: W. Zhang et al., Text classification based on multi-word with support vector machine, Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.03.044

Table 1
Percentage of high IG value features corresponding to each test

Test No.	1	2	3	4	5	6	7	8	9	10	11
Removal percentage (%)	0	50	70	75	80	85	90	92	95	98	99

of occurrences of w_1 in sentence s using the minimum difference rule. For example, for the first list L_1, it is produces the following steps: firstly, $a_{1,1}$ is added into L_1, next, we added the one $a_{2,2}$ in w_2 which has the minimum $|a_{1,1} - a_{2,*2}|$ into L_1, then the one $a_{3,*3}$ in w_3 which minimize $\{|a_{1,1} - a_{3,*3}|\}$ or minimize $\{|a_{1,1} - a_{2,*2}|\}$ will be added into L_1. This process continues until all the $t1$ lists are produced as shown in Fig. 3.

Then, we define the length of each list as Eq. (8). And at last the MS is derived as $\min_{1\leqslant i\leqslant t1}|L_i|$.

$$| L_i |= \max\{a_{1,i}, a_{2,*i}, \ldots, a_{m,*i}\} - \min\{a_{1,i}, a_{2,*i}, \ldots, a_{m,*i}\} \tag{8}$$

The computation complexity of this algorithm is $O(t1t2\ldots tm)$ and we know that this algorithm can be further improved as $O(m * t1)$ if $\{a_{i,1},a_{i,2},\ldots,a_{i,ti}\}$ is sorted before the list construction. Following is the algorithm of dynamic k-mismatch.

Finally, all the documents are represented with binary weight, i.e., ignoring the frequency, because long multi-word is usually a subtopic in the document and it has greater discriminative power than short multi-word. (Algorithm 2)

4. Experiments

Experiments are conducted on the task of text categorization on Reuters-21578 to examine the performance of multi-word for representation. We also compare the performance of linear kernel and non-linear kernel of SVM to investigate which type of kernel can better project the feature space characterized by multi-word representation to the geometrical classification space among the categories.

4.1. Text collection and preprocessing

Reuters-21578 text collection (http://www.daviddlewis.com/resources/testcollections/reuters21578/) was applied as our experimental data. It appeared as Reuters-22173 in 1991 and was indexed with 135 categories by personnel from Reuters Ltd. in 1996. By our statistics, it contains in total 19403 valid texts with average 5.4 sentences for each text. For convenience, the texts from 4 categories, "grain", "crude", "trade" and "interest" were assigned as our target data set, on the condition that the number of sentences for each text in these categories is between 4 and 7. With this method, 252 texts from "grain", 208 texts from "crude", 133 texts from "interest" and 171 texts from "trade" were assigned as our target data set and 2/3 of these texts from each category were used as training data and 1/3 of them as test data by random sampling.

The preprocessing we carried out for the assigned data includes stop word elimination, stemming and sentence boundary determination. Stop word elimination is to filter out the words in a text which are generally regarded as 'functional words' and do not carry meaning. On the other side, the computation complexity for multi-word extraction can be reduced by stop word elimination because usually the stop words have high frequency and are prone to comprise an undesired repetition pattern.[2] For stemming, the singular/plural regularization was conducted to transform the singular noun into plural noun, because what is concerned in this paper is noun multi-word and it mostly occurs in plural form such as "*mln dlrs*" to "*mln dlr*". In order to extract the repetition pattern from documents, we need to split a document into single sentences so that Algorithm 1 can be used. The method of sentence boundary determination in this paper is from [35].

Thus, 1514 multi-words were extracted from the training document set and with the processing in decomposition strategy, 984 short multi-words were obtained and used as the complete feature set in strategy 1. In combination strategy, 1037 long multi-word were obtained and used as the complete feature set in strategy 2. OR is set as 0.5 and MS is set as the number as two times of the number of words included in a multi-word for matching.

4.2. Experiment design and setting

In order to evaluate the effectiveness of multi-words for representation in different strategies, the feature sets used for representation vary at different proportion of the whole feature set under two strategies, respectively, according the ranking of IG values of the multi-words. In details, for each classification task, 11 tests were conducted at different removal percentage of the whole feature set as shown in Table 1. The features as individual words (the dimension is 10,116) are also used for comparison with the weight as document frequency for representation.

In SVM setting, the linear kernel used is as $(u * v)^1$ and the non-linear kernel is as $(u * v + 1)^2$ in the experiments.[3] We conjecture that only if the polynomial kernel is proved superior to linear kernel in classification can more complex non-linear kernel such as radial basis function (RBF) can be further attempted.

Thus, 6 text classification tasks were designed to study the performance of multi-word representation as shown in Table 2.

In order to guage the performance of the classification methods objectively, threefold cross validation is employed. That is, after all the training and test documents are represented using selected feature set, 2/3 of all the data objects are used as training the SVM learner and 1/3 for testing the trained SVM. One-Against-the-Rest approach is also used for multi-class classification and the accuracy is calculated out based on 4 iterations on average, of them each category was assigned as positive class once.

4.3. Results and evaluation

Fig. 4 is the accuracy graph of the above 6 classification tasks. It can be seen that individual words with linear kernel is the best classification method and individual words with non-linear kernel is in the second place. This point is easy to understand because the dimension of the individual words is about 10 times as the dimension of the multi-words and SVM has the potential to manage high dimensional input spaces effectively [36–38]. Table 3 shows the average performances of the 6 classification methods and their standard deviations. We can draw that on average performance for the designed classification tasks, IL > IN > MCL > MCN > MDL > MDN and on stand deviation with indicates the stableness of the classification methods: IL < IN MCN < MCL < MDL < MDN,

[2] We obtain the stop-words from United States Patent and Trademark Office (USPTO) patent full-text and image database at http://ftp.uspto.gov/patft/help/stopword.htm. It includes about 100 usual words as stop-words.

[3] Here, libsvm is used to conduct the work of text classification which is online and can be download freely: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Table 2
Classification tasks designed under 3 representation strategies and 2 kernels functions in SVM

Classification Task No.	1	2	3	4	5	6
Representation Features	Individual words	Individual words	Multi-words with strategy 1	Multi-words with strategy 1	Multi-words with strategy 2	Multi-words with strategy 2
Kernel function	Linear kernel	Non-linear kernel	Linear kernel	Non-linear kernel	Linear kernel	Non-linear kernel
Abbreviation	IL	IN	MDL	MDN	MCL	MCN

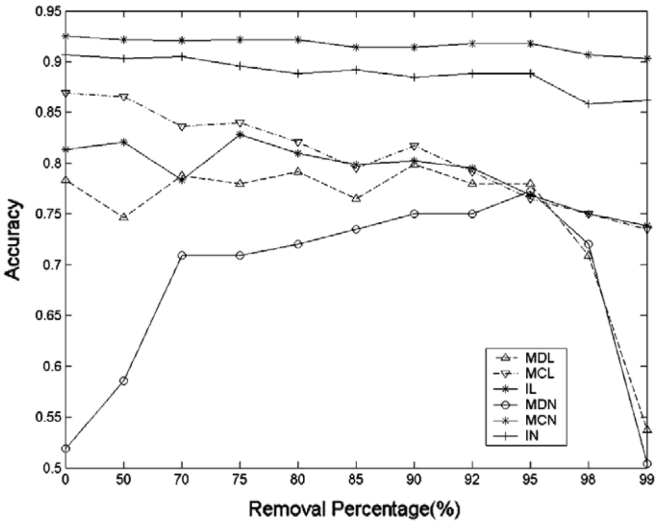


Fig. 4. The overall performances of the designed 6 classification methods.

Table 3
The average and stand deviation of each classification method

Classification method	Average accuracy	Standard deviation
IL	0.9168	0.0068
IN	0.8883	0.0159
MDL	0.7507	0.0751
MDN	0.6794	0.0960
MCL	0.8077	0.0488
MCN	0.7917	0.0288

Table 4
Results of *t*-test on the performance of each classification method on text classification

Method	IL	IN	MDL	MDN	MCL	MCN
IL		≫	≫	≫	≫	≫
IN			≫	≫	≫	≫
MDL				≫	≪	<
MDN					≪	≪
MCL						>
MCN						

which means IL is the most stable one for representation and MDN is not so good in robustness of classification performance. To better illustrate the effectiveness of each classification method, the classic *t*-test is also employed for the analysis [23,39]. Table 4 shows the results of *t*-test and the following codification of the *P*-value in ranges was used: “≫” and “≪” mean that the *P*-values is lesser than or equal to 0.01, indicating a strong evidence of that a system generates a greater or smaller classification error than another one, respectively; “<” and “>” mean that *P*-value is bigger than 0.01 and minor or equal to 0.05, indicating a weak evidence that a system generates a greater or smaller classification error than another one, respectively; “∼” means that the *P*-value is greater than 0.05 indicating that it does not have significant differences in the performances of the systems.

Combing the results in Tables 3 and 4, it can be concluded that linear kernel outperforms non-linear kernel on whatever kind of representation method and in the multi-word representation, the combination strategy is superior to the decomposition strategy. Moreover, the effect of different representation strategies is more then the effect of different kernel functions on text classification because the difference in performance of MCL and MCN is less significant than the difference of MCL and MDL or MCN and MDN. This outcome proves that representation using subtopics of general concepts can obtain better performance than representation using general concepts in text classification. That is, representation in more specific concepts can produce more desirable performance than general concepts can produce.

5. Concluding remarks

Multi-word is a newly exploited feature for text representation in the filed of information retrieval and text mining. In this paper, we implemented the multi-word extraction based on the syntactical structure of the noun multi-word phrases. For the sake of reduction on computation, repletion pattern identification is proposed to be extracted from sentences firstly and then use the extracted repetition patterns for regular expression matching to extract the multi-words. In order to use the multi-words for representation, two strategies were developed based on the different semantic level of the multi-words: the first is the decomposition strategy using general concepts for representation and the second is combination strategy using subtopics of the general concepts for representation. Moreover, IG method was employed as a scale to remove the multi-word from the feature set to study the robustness of the classification performance. Finally, a series of text classification tasks were carried out with SVM in linear and non-linear kernels, respectively, to analyze the effect of different kernel functions on classification performance. That is, to study the problem of what kind of vector mapping method is more preferred to project the document vector space to the category space.

Our experimental results demonstrate that the combination strategy for multi-word representation outperforms the decomposition strategy, and linear kernel outperforms non-linear kernel with SVM. In addition, it also appears that the combination strategy has poorer robustness than the decomposition strategy when the low IG value features are removed from the feature set. Nevertheless, the effect of using different representation strategies is greater than using different kernel functions in SVM on the classification performances.

The benefits of multi-word representation include at least three aspects. Firstly, it has lower dimension than individual words but its performance is acceptable. For instance, MCL can attain the accuracy up to 0.8673. Secondly, multi-word is easy to acquire from documents by corpus learning without any support of thesaurus, dictionary or ontology. Thirdly, multi-word includes more semantics and is a larger meaningful unit than individual word.

Please cite this article in press as: W. Zhang et al., Text classification based on multi-word with support vector machine, Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.03.044

For this reason, if multi-word is used for knowledge discovery, it can produce more interpretability and comprehensibility for the discovered patterns.

Although the experiment results have provided us with some clues on text classification with multi-words, unfortunately, a generalized conclusion was not obtained from this examination because of the lack of theoretical proof. A critical problem we need further advance is to provide a mathematical analysis on the statistical properties of multi-words on text classification. To attain this, more examination and investigation should be undertaken for more convincing work.

One of the promising directions in the text mining field concerns predictive pattern discovery from large amounts of documents. In order to achieve this goal, many kinds of work are involved in this field such as algorithm optimization, linguistics and machine learning. As for the further research, we would like to develop more practical algorithms for text mining using multi-word representation and explore more compressible methods for knowledge discovery such as concept clustering. Another interesting aspect in the future is concerned with improving the performance by integrating the learning method with the characteristics of the document vectors. Especially, the semi-supervised learning will be employed to boost the performance with semantic supervision from background knowledge.

Acknowledgments

This work is supported by Ministry of Education, Culture, Sports, Science and Technology of Japan under the ‘‘Kanazawa Region, Ishikawa High-Tech Sensing Cluster of Knowledge-Based Cluster Creation Project” and partially supported by the National Natural Science Foundation of China under Grant Nos.70571078 and 70221001.

References

[1] G. Salton, C.S. Yang, On the specification of term values in automatic indexing, Journal of Documentation 29 (4) (1973) 351–372. Q2

[2] A. Hotho et al., Ontologies Improve Text Document Clustering, in: Proceedings of the 3rd IEEE International Conference on Data Mining, 2003, pp. 541–544.

[3] S. Scott, S. Matwin, Text classification using WordNet Hypernyms, in: Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, pp. 45–52.

[4] M.B. Rodriguez et al., Using WordNet to complement training information in text categorization, in: Proceedings of 2nd International Conference on Recent Advances in Natural Language Processing II: Selected Papers from RANLP'97, vol. 189 of Current Issues in Linguistic Theory (CILT), 2000, pp. 353–364.

[5] D.D. Lewis, An evaluation of phrasal and clustered representation on a text categorization task, in: SIGIR'92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1992, pp. 37–50.

[6] R. Papka, J. Allan, Document classification using multiword features, in: Proceedings of the Seventh International Conference on Information and Knowledge Management Table of Contents, Bethesda, Maryland, United States, 1998, pp. 124–131.

[7] Y.J. Li et al., Text document clustering based on frequent word meaning sequences, Data & Knowledge Engineering 64 (2008) 381–404.

[8] B.C.M. Fung et al., Hierarchical document clustering using frequent itemsets, in: Proceedings of SIAM International Conference on Data Mining, 2003, pp. 59–70.

[9] T.B. Ho, K. Funakoshi, Information retrieval using rough sets, Journal of the Japanese Society for Artificial Intelligence 13 (3) (1998) 424–433.

[10] T.B. Ho, N.B. Nguyen, Non-hierarchical document clustering based on a tolerance rough set model, International Journal of Intelligent Systems 17 (2000) 199–212.

[11] W.B. Cavnar, J.M. Trenkle, *N*-Gram based text categorization, in: Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994, pp. 161–169.

[12] J.R. Firth, A synopsis of linguistic theory 1930–1955. Studies in linguistic analysis, Philological Society, Blackwell, Oxford, 1957.

[13] D. Hawking, P. Thistlewaite, Proximity operators – so near and yet so far, in: Proceedings of TREC-4, 1996, pp. 131–144.

[14] J.S. Justeson, S.M. Katz, Technical terminology: some linguistic properties and an algorithm for identification in text, Natural Language Engineering 1 (1) (1995) 9–27.

[15] D. Bourigault, Surface grammatical analysis for the extraction of terminological noun phrases, in: Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992, pp. 977–981.

[16] F. Smadja, Retrieving collocations from text: Xtract, Computational Linguistics 19 (1) (1993) 143–177.

[17] Y.J. Park, R.J. Byrd, K.B. Boguraev, Automatic glossary extraction: beyond terminology identification, in: Proceedings of the 19th International Conference on Computational linguistics, Taiwan, 2002, pp. 1–17.

[18] B. Daille et al., Towards automatic extraction of monolingual and bilingual terminology, in: Proceedings of the International Conference on Computational Linguistics, Kyoto, Japan, 1994, pp. 93–98.

[19] J. Zhang, J.F. Gao, M. Zhou, Extraction of Chinese compound words: an experiment study on a very large corpus, in: Proceedings of the Second Chinese Language Processing Workshop, HongKong, 2000, pp. 132–139.

[20] J.J.G. Adeva, J.M.P. Atxa, Intrusion detection in web applications using text mining, Engineering Applications of Artificial Intelligence 20 (1) (2007) 555–566.

[21] Y.Y. Zhang, J.X. Jiao, An associative classification-based recommendation system for personalization in B2C e-commerce application, Expert Systems with Applications 33 (1) (2007) 357–367.

[22] M. Hiissa et al., Towards automated classification of intensive care nursing narratives, International Journal of Medical Informatics, in press. Q1

[23] Y.M. Yang, X. Liu, A re-examination of text categorization methods, in: Proceedings on the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, 1999 pp. 42–49.

[24] Y.M. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 412–420.

[25] V. Vapnic, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[26] J.W. Han, M. Kamber, Data Mining Concepts and Techniques, second ed., Morgan Kaufmann Publishers, 2006.

[27] M.A. Aizerman et al., Theoretical Foundations of the potential function method in pattern recognition learning, Journal of Machine Learning Research (2000) 113–141. Available from: <http://www.jmlr.org/>.

[28] J. Weston, C. Watkins, Multi-class Support Vector Machines, Technical Report CSD-TR-98-04, Royal Holloway, University of London, Department of Computer Science, 1998.

[29] D. Bourigault, Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases, in: Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992, pp. 977–981.

[30] F. Jelinek, Self-organized language modeling for speech recognition, in: A. Waibel, K.F. Lee (Eds.), Readings in Speech Recognition, Morgan Kaufmann Publishers, 1990, pp. 450–506.

[31] J.S. Chang et al., A multiple-corpus approach to recognition of proper names in chinese texts, Computer Processing of Chinese and Oriental Languages 8 (1) (1994) 75–85.

[32] I. Fahmi, C Value Method for Multi-word Term Extraction, Seminar in Statistics and Methodology, Alfa-informatica, RuG, May 23, 2005. Available from: <http://odur.let.rug.nl/fahmi/talks/statistics-c-value.pdf/>.

[33] S. Baase, Computer Algorithms: Introduction to Design and Analysis, Addison-Wesley Publishing Company, 1978. pp. 173–185.

[34] S. Katz, Distribution of content words and phrases in texts and language modeling, Natural Language Engineering 2 (1) (1996) 15–59.

[35] S.M. Weiss et al., Text Mining: Predictive Methods for Analyzing Unstructured Information, Springer-Verlag, New York, 2004.

[36] E. Leopold, J. Kindermann, Text categorization with support vector machines. How to represent texts in input space?, Machine Learning 46 (2002) 423–444

[37] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of ECML-98, 10th European Conference on Machine Learning, pp. 137–142.

[38] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (1) (2002) 11–12. 32–33.

[39] R.F. Correa, T.B. Ludermir, Improving self-organization of document collection by semantic mapping, Neurocomputing 70 (2006) 62–69.

Please cite this article in press as: W. Zhang et al., Text classification based on multi-word with support vector machine, Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.03.044