Expert Systems with Applications 38 (2011) 2758–2765

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A comparative study of TF*IDF, LSI and multi-words for text classification

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

[a] Laboratory for Internet Software Technologies, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PR China
[b] School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Ashahidai, Nomi, Ishikawa 923-1292, Japan
[c] Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Keywords:
Text representation
TF*IDF
LSI
Multi-word
Text classification
Information retrieval
Text categorization

ABSTRACT

One of the main themes in text mining is text representation, which is fundamental and indispensable for text-based intellegent information processing. Generally, text representation inludes two tasks: indexing and weighting. This paper has comparatively studied TF*IDF, LSI and multi-word for text representation. We used a Chinese and an English document collection to respectively evaluate the three methods in information retreival and text categorization. Experimental results have demonstrated that in text categorization, LSI has better performance than other methods in both document collections. Also, LSI has produced the best performance in retrieving English documents. This outcome has shown that LSI has both favorable semantic and statistical quality and is different with the claim that LSI can not produce discriminative power for indexing.

1. Introduction

Any text-based system requires some representation of documents, and the appropriate representation depends on the kind of task to be performed (Lewis, 1992a). Moreover, the ability to accurately perform a classification task depends on the representation of documents to be classified (Quinlan, 1983). Different from data mining that handles the well-structured data, text mining deals with a collection of semi-structured, even unstructured documents. This makes that one of the main themes supporting text mining is transforming text into numerical vectors, i.e., text representation.

In information retrieval, documents are generally identified by sets of terms or keywords that are collectively used to represent their contents. Vector space model (VSM) (Salton & Yang, 1973) is one of the mostly used models for representation, because of its conceptual simplicity and the appeal of the underlying metaphor of using spatial proximity for semantic proximity. Generally, there are two kinds of works included in text representation: indexing and term weighting (Lewis, 1992a). Indexing is the job to assign indexing terms for documents. Term weighting is the job to assign the weight for each term, which measures the importance of a term in a document. We should clarify here that, this paper will regard indexing and term weighting as two components of text representation scheme, and will not discuss the effectiveness of indexing and term weighting individually.

Currently, there are many term weighting methods, which are derived from the different assumptions for terms' characteristics in texts. For instance, IDF (inverse document frequency) assumes that the importance of a term relative to a document is inversely proportional to the frequency of occurrence of this term in all the documents, while RIDF (residual inverse documents frequency) holds the assumption that the importance of a term should be measured by the difference between its actual frequency of occurrence in documents and the predicted frequency of occurrence by Poisson distribution (random occurrence).

Essentially, in the task of text classification, which includes information retrieval (IR) and text categorization (TC) (Lewis, 1992a), we are mainly concerned with two kinds of properties of the indexing term: semantic quality and statistical quality (Jose, 2003). Semantic quality is related to a term's meaning, i.e., to how much extent the index term can describe text content. Statistical quality is related to the discriminative (resolving) power of the index term to identify the category of a document in which the term occurs. Table 1 lists the index terms currently used for text representation. We can see that more and more semantic meaning of index terms are used for representation.

The purpose of this research is to study the effectiveness of different representation methods in text classification. Here, we would like to reaffirm that text classification includes both information retrieval and text categorization though many researchers regard text categorization is the same as text classification. This point can refer to (Lewis, 1992a). Although TF*IDF, LSI and multi-word have been proposed for a long time, there is no comparative study on these indexing methods, and no results are reported concerning their classification performances. Despite that some

* Corresponding author.
E-mail addresses: zhangwen@itechs.iscas.ac.cn (W. Zhang), yoshida@jaist.ac.jp (T. Yoshida), xjtang@amss.ac.cn (X. Tang).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.08.066

indexing methods are accepted as having superior qualities, such as LSI and multi-word with better semantic quality, there is no clear evidence to show that to how much extent their preferred quality will produce better performances in text classification. Based on the above facts, text classification using TF∗IDF, LSI and multi-words is conducted in this paper. To make the outcomes more convincing, we conducted many experiments on our corpora.

The rest of this paper is organized as follows. Section 2 presents preliminaries of this paper, including the concept of information retrieval, text categorization, entropy computation and support vector machine. Section 3 describes the rationales of TF∗IDF, LSI and multi-word in text representation. Their advantages and disadvantages are discussed. In particular, Justeson and Katz's method (Justeson & Katz, 1995) for multi-word extraction is introduced in this section. Section 4 is experiments and evaluation for the methods. The corpora, experiment design, evaluation method and experimental results are specified in this section. Section 5 concludes the whole paper and indicates our future work.

2. Preliminaries

This section introduces the basic techniques used in this paper, including the concept of information retrieval, text categorization, and information gain and support vector machine.

2.1. Information retrieval systems

Information retrieval systems (Ho & Funakoshi, 1998) can be formulated as a quadruple $\delta = (J, D, Q, \alpha)$, where $J = \{t_1, t_2, \ldots, t_M\}$ is a set of index terms; $D = \{d_1, d_2, \ldots, d_N\}$ is a set of documents for each $d_j \subseteq J$; $Q = \{Q_1, Q_2, \ldots, Q_P\}$ is a set of queries for each $Q_k \subseteq J$; and $\alpha: Q \times D \to R^+$ is a ranking function that evaluates the relevance between a query and a document. Given a query $q \in Q$, for any documents $d_{j1}, d_{j2} \in D$, if $\alpha(q, d_{j1}) > \alpha(q, d_{j2})$ then d_{j1} is considered more relevant to q than d_{j2}. In a general form, a document can be denoted as a set of index term-weight pairs $d_j = (t_{j1}, w_{j1}; t_{j2}, w_{j2}; \cdots; t_{jn}, w_{jn})$, where $t_{jk} \in J$ and $w_{jk} \in [0, 1]$ reflects the relative importance of index term t_{jk} in d_j. A query $q \in Q$ can also be denoted as a set of index term-weight pairs $q = (q_1, w_{q1}; q_2, w_{q2}; \cdots; q_l, w_{ql})$, where $q_k \in J$ and $w_{qk} \in [0, 1]$. The information retrieval task is to yield a set $A = \{d_{j1}, d_{j2}, \cdots, d_{jm}\} \subseteq D$ to the query q with a ranking order of $\alpha(q, d_{jk})$.

2.2. Text categorization

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 objective is to train a classification model (classifier) as Eq. (1) which is able to assign correct class label(s) to a new document d of the domain.

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

To measure the performance of a classification model, a random fraction of the labeled documents is set aside as a test set and not used for training. We may classify the documents in the test set with the trained classification model and compare the predicted labels with true labels. Thus, performance measures as precision and recall can be produced by this comparison.

Table 1
Existing methods to assign the index term.

Index term	Description	References
N-grams	Character pattern in words	Edda and Jorg (2002), Caropreso et al. (2001)
Individual words	Used for lexical matching	Salton and Yang (1973), Salton (1989)
A set of individual words	To characterize co-occurrence of individual words	Ho and Funakoshi (1998), Ho and Nguyen (2000)
Multi-words, word sequences, phrases	To capture contextual information of individual words	Li, Chung, and Holt (2008), Papka and Allan (1998), Zhang, Yoshida, and Tang (2007), Zhang et al. (2008), Zhou, Hu, and Zhang (2007), Lewis (1992b)
Ontology	To compile background knowledge into representation	Zhou et al. (2007), Scott (xxxx), Hotho, Staab, and Stumme (2003)

2.3. Information gain

A major difficulty of text categorization is the high dimensionality of the feature space, and most of the terms are redundant to the categorization task. So it is highly desirable to find some methods which can reduce dimensions of the feature space without sacrificing categorization performance. For this purpose, information gain (IG) is proposed, which is defined as the expected reduction in entropy caused by partitioning the texts according to a term. The formula of IG is Eq. (2) and the formula of entropy is Eq. (3).

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

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

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

In this paper, IG is employed to rank multi-words for the task of text categorization. In order to observe the performances of multi-word on categorization continuously, the feature set of multi-word is constructed at different removal percentages.

2.4. Support vector machine

SVM is a relatively new learning approach introduced by Vapnik in 1995 for solving two-class pattern recognition problem (Han & Kamber, 2006; Vapnic, 1995). The method is originally 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 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 the linearly inseparable problem, kernel function (Aizerman, Braverman, & Rozoner, 1964) is used to derive the similarities in the original lower dimensional space.

Considering the multi-class classification in this paper, the One-Against-the-Rest approach was adopted. Other methods for k-classes ($k > 2$) classification are also discussed in Weston and Watkins (1998) such as error-correcting output codes, SVM decision tree, etc.

3. Text representation with TF∗IDF, LSI and multi-word

This section describes the rationales of TF∗IDF, LSI and multi-word. The theoretical analysis behind each method is also analyzed in details.

3.1. TF∗IDF for text representation

TF∗IDF is evolved from IDF which is proposed by Sparck Jones (1972, 2004) with heuristic intuition that a query term which occurs in many documents is not a good discriminator, and should be given less weight than one which occurs in few documents. Eq. (7) is the classical formula of TF∗IDF used for term weighting.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{7}$$

where $w_{i,j}$ is the weight for term i in document j, N is the number of documents in the collection, $tf_{i,j}$ is the term frequency of term i in document j and df_i is the document frequency of term i in the collection.

The basic idea of TF∗IDF is from the theory of language modeling that the terms in a given document can be divided into two categories: those words with eliteness and those words without eliteness (Roberston, 2004), i.e., whether or not a term is relevant with the topic of a given document. Further, the eliteness of a term for a given document can be evaluated by TF and IDF and in TF∗IDF formulation, it is used to measure the importance of a term in the document collection.

However, there are some criticisms of using TF∗IDF for text representation. The first one is that TF∗IDF is too 'ad hoc' because it is not directly derived from a mathematical model, although usually it is explained by Shannon's information theory (Caropreso, Matwin, & Sebastiani, 2001). The second criticism comes from that the dimensionality (size of feature set) in TF∗IDF for textual data is the size of the vocabulary across the entire dataset, resulting in that it brings about a huge computation on weighting all these terms (Christopher & Hinrich, 2001).

3.2. LSI for text representation

A fundamental deficiency of current information retrieval is that the words searchers use often are not the same as those words, by which the information they seek has been indexed (Berry, Dumais, & O'Brien, 1995). There are actually two sides to this issue: synonymy and polysemy. Users in different contexts or with different needs, knowledge, or linguistic habits will describe same information using different terms. Polysemy is the fact that most words have more than one distinct meaning. Thus, the use of a term in a search query does not necessarily mean that a document is containing or labeled by the same term.

LSI (Latent Semantic Indexing) (Deerwester, Dumais, Landauer, Furnas & Harshman, 1990) is a popular linear algebraic indexing method to produce low dimensional representations by word co-occurrence. The basic idea behind LSI is to take advantage of implicit higher-order structure in the association of terms with documents ("semantic structure") in order to improve the detection of relevant documents, on the basis of terms found in queries.

LSI aims to find the best subspace approximation to the original document space in the sense of minimizing the global reconstruction error (the difference of Frobenius norm between the original matrix and its approximation matrix). It is based on SVD (Singular Value Decomposition) and projects the document vectors into an approximated subspace, so that cosine similarity can accurately represent semantic similarity.

Given a term-document matrix $X = [x_1, x_2, \dots, x_n] \in R^m$ and suppose the rank of X is r, LSI decomposes the X using SVD as follows:

$$X = U\Sigma V^T \tag{8}$$

where $\Sigma = diag(\sigma_1, \dots, \sigma_r)$ and $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r$ are the singular values of X. $U = [u_1, \dots, u_r]$ and u_i is called the left singular vector. $V = [v_1, \dots, v_r]$ and v_i is called the right singular vector. LSI uses the first k vectors in U as the transformation matrix to embed the original documents into a k-dimensional space.

There are some deficiencies of LSI method. The first one is that some negative values are in the approximation matrix which we cannot give a plausible explanation. The second one is its huge computation as $O(n^2r^3)$, where n is the smaller of the number of documents and the number of terms, r is the rank of X Christopher and Hinrich (2001).

3.3. Multi-word for text representation

A word is characterized by the company it keeps (Firth, 1957). That means not only the individual word but also the context of the individual word should be laid on great emphasis for word sense disambiguation. This simple and direct idea motivates the researches on multi-word, which is expected to capture the contextual information of individual words. Currently, multi-word has no satisfactory formal definition. Usually, it can be defined as a sequence of two or more consecutive individual words with meaningful contents, including collocations and compounds (Chen, Yeh, & Chau, 2006; Weiss, Indurkhya, Tang Zhang, & Damerau, 2005). In this paper, we use compound s in documents as multi-word index terms.

There are mainly two categories of methods for multi-word extraction. The one is statistical methods based on mutual information and the other is linguistic methods based on grammatical and syntactical rules of phrases (Church & Hanks, 1990; Zhang, Yoshida, & Tang, 2009b; Zhang, Yoshida, Tang, & Ho, 2009a).

For simplicity, we adopt the idea of Justeson and Katz (1995) concerning the syntactical properties of terminology to extract multi-words from documents. Thus, our rule set for a multi-word is that, the length of the multi-word should be between 2 and 6, and the multi-word should meet the regular expression, and its occurrence frequency should be at least twice in a document. The regular expression is Eq. (9) for Chinese multi-word and Eq. (10) for English multi-word, respectively.

$$((A|N)^*)N \tag{9}$$

$$((A|N)^+|(A|N)^*(NP)^?(A|N)^*)N \tag{10}$$

where A is an adjective, N is a noun and P is a preposition. We produce multi-word candidates by matching any two sentences in a document to look for the repetitive patterns, as described in Fig. 1.

After repetition extraction from documents, we should normalize these repetitions to meet the regular expression as described in Eqs. (9) and (10). The procedure we used to normalize the repetitions into multi-words is shown in Fig. 2.

However, there are some disadvantages with multi-word for indexing. On one hand, multi-word is neither derived from a classical mathematic model nor a formally defined linguistic model. Although it is superiority in text classification could be explained using N-Grams, there is also no established theory for N-Grams, despite that N-Gram is validated in some practical applications.

Input:
S_1, the first sentence
S_2, the second sentence
Output:
Multi-word extracted from S_1 and S_2.
Procedure:
$S_1 = \{w_1, w_2, \ldots, w_n\}$, $S_2 = \{w_1', w_2', \ldots, w_m'\}$, k=0
For each word w_i in S_1
For each word w_j in S_2
While(w_i equal to w_j)
k++
End while
If k>1
extract the words from w_i to w_{i+k} to form a multi-word candidate
$k = 0$
End if
End for
End for

Fig. 1. Repetitive pattern extraction from two sentences.

On the other hand, the effectiveness of multi-word is strongly dependent on the types of literature (genres). For instance, multi-word indexing is effective for documents, in which fixed expressions (terminologies, collocations, etc.) are usually used, such as academic papers, but may be not effective for the documents with extensive topics, in which fixed expressions are not usually used such as essay.

4. Experiments and evaluation

In this Section, we carried out experiments using TF∗IDF, LSI and multi-word as indexing methods, to examine their performances on text classification.

4.1. The datasets

As for the Chinese corpus, TanCorpV1.0 is used in this research which is available online (http://www.searchforum.org.cn/tansongbo/corpus.htm). On the whole, this corpus has 14,150 documents with 20 categories from Chinese academic journals concerning computer, agriculture, politics, etc. In this paper, documents from four categories as "agriculture", "history", "politics" and "economy" are selected as target Chinese document collection. For each category, 300 documents were selected randomly from original corpus so that totally 1200 documents were used, which have 219,115 sentences and 5,468,301 individual words in sum after morphological analysis.[1]

For the English corpus, Reuters-21578 distribution 1.0 is used in this paper which is also available online (http://www.research.att.com/~lewis). It collects 21,578 news from Reuters newswire in 1987. Since 1991, it appeared as Reuters-22173 and was assembled and indexed with 135 categories by the personnel from Reuters Ltd in 1996. In this research, the documents from four categories as "crude" (520 documents), "agriculture" (574 documents), "trade" (514 documents) and "interest" (424 documents) are assigned as the target English document collection. That is, we select totally 2042 English documents, which have 50,837 sentences and 281,111 individual words after stop-word elimination.[2]

[1] Because Chinese is character based, we conducted the morphological analysis using the ICTCLAS tool. It is a Chinese Lexical Analysis System. Online: http://nlp.org.cn/~zhp/ICTCLAS/codes.html.

[2] We obtain the stop-words from USPTO (United States Patent and Trademark Office) patent full-text and image database at http://ftp.uspto.gov/patft/help/stopword.htm. It includes about 100 usual words as stop-words. The part of speech of English word is determined by QTAG which is a probabilistic parts-of-speech tagger and can be downloaded freely online: http://www.english.bham.ac.uk/staff/omason/software/qtag.html.

Input:
S: the extracted repetitive pattern using the algorithm depicted in Figure 1
P: the regular pattern; Eq.10 is for Chinese and Eq.11 is for English
Output:
L: a list contains multi-words satisfying P regular expression
Procedure:
$S = \{w_1, w_2, \ldots, w_m\}$; $L = \phi$
For i = 0, i<m , i++
For j = m, j>i, j--
$S' = \{w_i, \ldots, w_j\}$;
If (S' meets P and $S' \notin L$)
$L = L \cup S'$
End If
End For
End For

Fig. 2. The procedure used for multi-word normalization.

4.2. Experiment setup

In TF∗IDF method, firstly, individual words in each document are collected to construct a feature set for each document. Secondly, for each individual word, we compute its TF∗IDF score in each document. Thirdly, all the individual words in a document are sorted by their TF∗IDF scores. Then different percentages of individual words with top TD∗IDF scores are retained to construct the feature set (vocabulary) for representation. The feature set for the whole document collection is produced by uniting the retained individual words of each document. Finally, each document in the corpus is represented using the constructed feature set. The term weight of each individual word in a document is the TF∗IDF score it has in that document. Otherwise, its term weight will be set as 0, if the individual word does not occur in the document. By this way, the term-document vector is produced for each document.

In LSI method,[3] original terms for the collection are those individual words whose term frequency in the document is more than two. The original term weight of an individual word in a document is set as the corresponding term frequency of that individual word in that document. Then, SVD is used to decompose the original term-document matrix. Next, we retain a certain percentage of singular values in Σ of Eq. (8) to produce the approximation matrix, which has lower dimensions than the original term-document matrix. The term percentage is varied from 0 to 1 with increment of 0.1.

For multi-word representation, firstly, all the multi-words produced in a document collection are used to represent the documents using Boolean method. That is, the weight of a multi-word in a document will be set as 1 if the multi-word occurs in the document. Otherwise, its weight will be set as 0. Secondly, information gain of each multi-word is computed according to the multi-words' occurrence in the documents of each category. Thirdly, multi-words are sorted descending by their information gain. Then, different percentages of multi-words with top information gain are retained, to construct the feature set for the document collection. Finally, each document is represented using the multi-word feature set and the term frequency of each multi-word is used as term weight.

Tables 2 and 3 are the dimensions (the number of indexing terms) of each indexing method at different term percentages for Chinese and English documents, respectively. For TF∗IDF, term percentage means the percentages of individual words with top TF∗IDF values will be retained to construct the feature set for the whole document collection. For LSI, term percentage means the percentages of top singular values in Σ, which will be retained to construct the approximation matrix. For multi-word, term

[3] LSI is carried out with JAMA (A Java Matrix Package) which is online and can be download freely: http://www.math.nist.gov/javanumerics/jama/.

percentage means the percentages of multi-words with top IG values, which will be retained to construct the feature set for representation.

In order to evaluate the performances of the above methods in information retrieval, 50 queries uniformly distributed on the four categories of Chinese document collection and 25 queries uniformly distributed on the four categories of English document collection are purposely developed to conduct the tasks of information retrieval. For each of these queries, we manually checked the whole Chinese and English document collection to identify corresponding set of relevant documents. Then, the query terms are transformed to query vectors using the same index terms as used in the indexing methods. For LSI, the query vectors are projected to the same subspace as approximation matrix by multiplying the left single vectors U^T. Then, cosine similarity is computed between a query vector and each document vector to retrieve the relevant documents from document collection. Finally, documents having similarities more than 0 with a query vector will be regarded as relevant with the query.

In order to evaluate the performances of the above methods in text categorization, support vector machine is used to categorize the documents. The linear kernel $(u*v)^1$ is used for SVM training, because it is superior over the non-linear kernel in text categorization, which is validated by our prior research (Zhang, Yoshida, & Tang, 2008). For multi-class categorization, the One-Against-the-Rest approach is employed.

4.3. Evaluation method

Salton and McGill (1983) and Yan, Grosky, and Fotouhi (2008) describe the Interpolated Recall Precision Curve (IRP curve), which depicts how precision changes over a range of recall, e.g. [0, 1]. In this paper, a single numerical value of T shown in Fig. 3, measuring the area covered between the IRP curve and the horizontal axis of recall and representing the average interpolated precision over the full rage ([0, 1]) of recall, is used to indicate the performance of a particular query or classifier. In details, for both retrieval and categorization tasks, the 10-point recall-precision is employed to evaluate the performance. That is, we set recall as 0.1, 0.2, ..., 1.0 with interval of 0.1 and precision is measured under these 10-point recalls. For English retrieval, T is averaged across the 25 queries and for Chinese retrieval, T is averaged across the 50 queries. In text categorization, 3-fold cross validation is used. Each time, we randomly set 1/3 samples in one category as positive labels and 1/3 samples in other categories as negative labels for training, and all the remaining samples are used for testing. Then, categorization performance is averaged across total four classifiers (we have four categories) with 10 times repeating under 10-point recalls.

The method for query (classifier) evaluation under a predefined recall is from Kolda and O'Learly (1998). When we evaluate a query (classifier), we received an ordered list of documents. Let r_i denote the number of relevant documents (correctly labeled documents) up to position i in the ordered list. For each document, we compute two values: recall and precision. The recall at ith document is the proportion of relevant documents (correctly labeled documents) returned so far, that is, $\frac{r_i}{r_n}$. r_n is the total number of relevant documents. The precision at the ith document, p_i, is the proportion of relevant documents returned, that is, $p_i = \frac{r_i}{i}$.

Table 2
Dimensions of TF∗IDF, LSI and multi-word of Chinese document collection at different term percentages.

Indexing method Term percentage	TF∗IDF	LSI	Multi-word
1.0	21,624	1200	33,661
0.9	21,624	1080	30,295
0.8	21,624	960	26,929
0.7	21,624	840	23,563
0.6	21,618	720	20,197
0.5	21,615	600	16,831
0.4	21,562	480	13,464
0.3	21,101	360	10,098
0.2	18,580	240	6732
0.1	11,640	120	3366

Table 3
Dimensions of TF∗IDF, LSI and multi-word of English document collection at different term percentages.

Indexing method Term percentage	TF∗IDF	LSI	Multi-word
1.0	4924	2042	3112
0.9	4922	1838	2799
0.8	4911	1634	2488
0.7	4889	1430	2177
0.6	4856	1226	1866
0.5	4773	1022	1555
0.4	4619	818	1244
0.3	4189	614	933
0.2	3353	410	622
0.1	2181	206	311

4.4. Results

Figs. 4 and 5 are the experimental results of text classification in Chinese document collection. We can see that in Chinese information retrieval, the effectiveness of multi-word is the poorest one of the three, but with the most robust performance. When more and more dimensions removed from the feature set, the performances of TF∗IDF and LSI are also declining: firstly slowly and then speedy after term percentage 0.5. In Chinese categorization, the performances of all the three methods are not kept stable. However, their fluctuation magnitudes are different from each other: LSI has the overall best performance; TF∗IDF and multi-word are comparable.

Figs. 6 and 7 are the experimental results of text classification in English corpus. We can see that in English information retrieval, the effectiveness of LSI is the best; TF∗IDF is superior to multi-word before term percentage 0.3, but with the opposite case after term percentage 0.3. Moreover, the performances of LSI and multi-word are more robust than TF∗IDF. In English categorization, the performances of all three methods are stable. LSI has the overall best performance. Multi-word is better than TF∗IDF. In addition, the performances of LSI and multi-word are more robust than TF∗IDF.

To better illustrate the effectiveness of each classification method, the classic t-test is employed (Correa & Ludermir, 2006; Yang & Liu, 1999). Tables 4 and 5 show the results of t-test of the performances of the three methods. The following codification

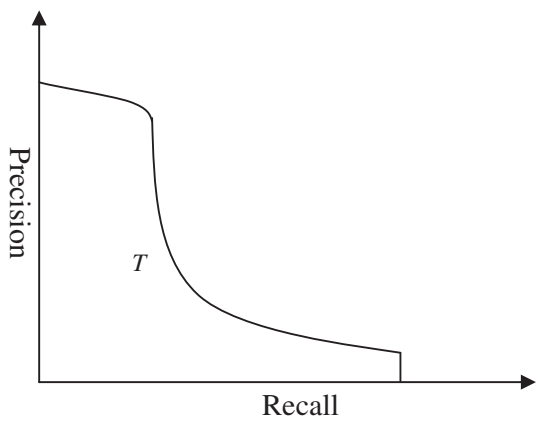

Fig. 3. Area covered by the IRP curve.

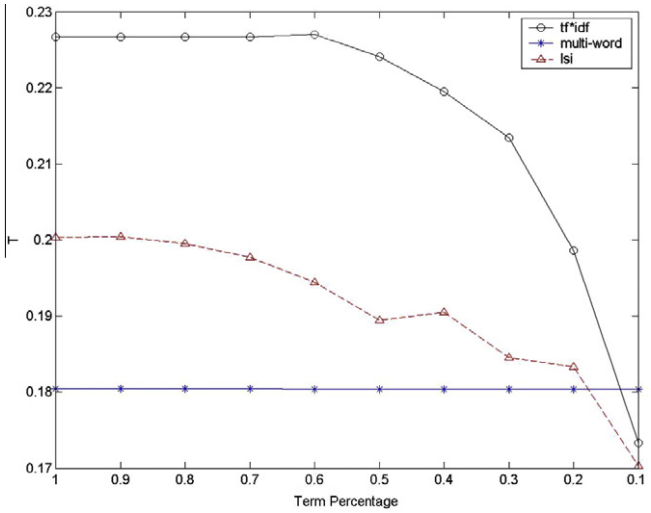

Fig. 4. The performances of TF∗IDF, LSI and multi-word representation in Chinese information retrieval.

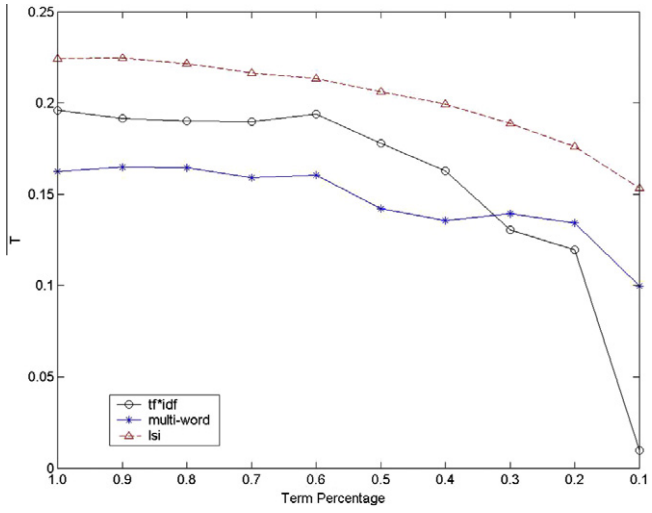

Fig. 6. The performances of TF∗IDF, LSI and multi-word representation in English information retrieval.

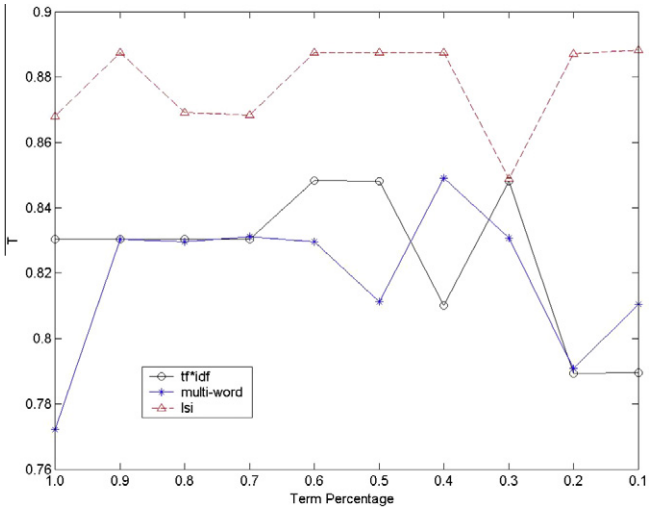

Fig. 5. The performances of TF∗IDF, LSI and multi-word representation in Chinese text categorization.

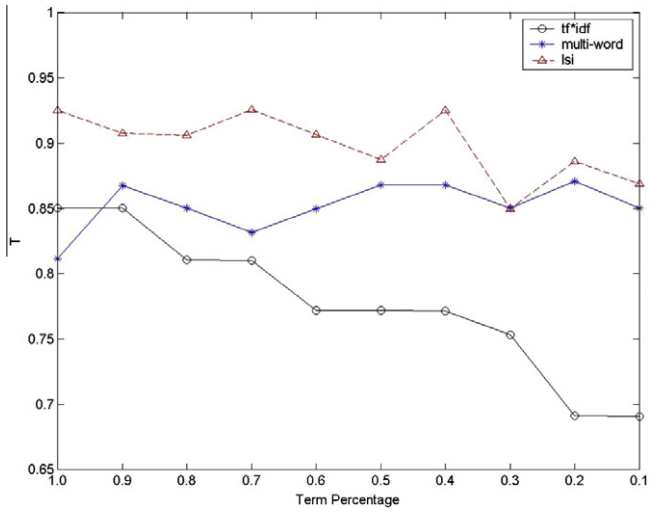

Fig. 7. The performances of TF∗IDF, LSI and multi-word representation in English text categorization.

of the
P-value in ranges was used: "$\gg$" ("$\ll$") means that *P*-value is lesser than or equal to 0.01, indicating a strong evidence of that a system generates a greater (smaller) classification error than another one; "<" (">") means 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 (smaller) classification error than another one; "$\sim$" means that *P*-value is greater than 0.05, indicating that the compared systems do not have significant differences in performances.

We can see from Table 4 that, in Chinese information retrieval, TF $*$ IDF $\gg$ LSI $\gg$ multi-word and in Chinese text categorization, LSI $\gg$ TFIDF $\sim$ multi-word. It can be seen in Table 5 that, LSI $\gg$ TFIDF $\sim$ multi-word in English information retrieval and LSI $\gg$ TFIDF $\gg$ multi-word in English text categorization.

We conjecture that TF∗IDF is better in Chinese information retrieval than LSI, because the number of dimensions (21,624) in TF∗IDF is overwhelmingly larger than that of LSI (1200). With such a huge amount of dimensions, TF∗IDF can produce ideal performance in information retrieval by simple lexical matching, not necessarily to make use of term associations. This hypotheses can be illustrated in English information retrieval, where the number of dimensions (4924) in TF∗IDF is at the same level as the number of dimensions (2042) used in LSI. However, with the utilization of term association, LSI produce better performance than TF∗IDF.

In both Chinese and English text categorization, LSI has the best performance among the three methods. This outcome shows that LSI can produce better indexing in discriminative power. This conclusion is different with that was discussed in Cai, He, and Han (2005), which argue that LSI is representative but not discriminative.

When using multi-word representation for information retrieval, the queries always produces close results at all term percentages, because we use Boolean vectors and lexical matching. In text categorization, multi-word can produce good performance, even with small term percentages, because of the power of IG method for text categorization.

5. Discussion and concluding remarks

In this paper, experiments are conducted to examine the performances of three document representation methods: TF∗IDF, LSI and multi-word in text classification. Basically, two kinds of prop-

Table 4
Results of *t*-test on the performances of the three methods in Chinese text classification.

Task	Chinese information retrieval			Chinese text categorization		
Method	TF∗IDF	LSI	Multi-word	TF∗IDF	LSI	Multi-word
TF∗IDF		$\gg$	$\gg$		$\ll$	$\sim$
LSI			$\gg$			$\gg$

Table 5
Results of *t*-test on the performances of the three methods in English text classification.

Task	English information retrieval			English text categorization		
Method	TF∗IDF	LSI	Multi-word	TF∗IDF	LSI	Multi-word
TF∗IDF		$\ll$	$\sim$		$\ll$	$\ll$
LSI			$\gg$			$\gg$

erties of indexing terms should be considered: semantic quality and statistical quality.

Usually, we regarded that LSI and multi-word have better semantic quality than TF∗IDF, and TF∗IDF has better statistical quality than the other two methods. However, from our experimental results, we can see that, the number of dimension is still a decisive factor for indexing when we use different indexing methods for classification. Furthermore, we show that LSI has better performance in categorization, which comes from good discriminative power. This point is more often than not overlooked by most researchers in text mining field.

Moreover, what is also worth our noticing is the computation complexity of the methods. The computation complexity of TD∗IDF is $O(nm)$, where n is the total number of individual words and m is the total of number of documents in the document collection. The huge computation of LSI is discussed in Section 3.2. For multi-word, most of the computation is cost on multi-word extraction, which is $O(ms^2)$, where s is average number of sentences in a document.

Although some conclusions are drawn from our theoretical analysis and experiments in this paper, there are still some questions as follows:

1. In this paper, we present some experimental evaluations of indexing methods on text classification. However, how to evaluate the performances of indexing methods theoretically is a problem.
2. Our multi-word extraction is a simple and intuitive linguistic method. We should validate whether or not an improvement in multi-word extraction will produce an improvement in indexing using multi-word.
3. The basic criterion of text representation is the semantic quality and statistical quality. Unfortunately, we do not have a standard measure to gauge these two kinds of qualities mathematically. These two qualities are considered merely by our intuition instead of theory.
4. We discuss two different aspects concerning text representation as term weighting and index term selection comprehensively. Their individual effects on representation are destined to be different from each other.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant Nos. 90718042, 60873072, and 60803023; the National Hi-Tech R&D Plan of China under Grant Nos. 2007AA010303 and 2007AA01Z179; the National Basic Research Program under Grant No. 2007CB310802. This work is also partially supported by the Foundation of Young Doctors of Institute of Software, Chinese Academy of Sciences, under Grant No. ISCAS2009-DR03.

References

Aizerman, A., Braverman, E. M., & Rozoner, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control, 25*, 821–837.

Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37*(4), 573–595.

Cai, D., He, X. F., & Han, J. W. (2005). Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering, 17*(12), 1624–1637.

Caropreso, M. F., Matwin, S., & Sebastiani, F. (2001). A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Amita G. Chin (Ed.), *Text databases and document management: theory and practice* (pp. 78–102). Hershey, U.S: Idea Group Publishing.

Chen, J. Yeh, C., & Chau, R. (2006). Identifying multi-word terms by text-segments. In *Proceedings of the seventh international conference on web-age information management workshops* (pp. 10–19). Hongkong.

Christopher, D. M., & Hinrich, S. (2001). *Foundations of statistical natural language processing* (pp. 529–574). Cambridge, Massachusetts: MIT Press.

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics, 16*(1), 22–29.

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

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of American Society of Information Science, 41*(6), 391–407.

Edda, L., & Jorg, K. (2002). Text categorization with support vector machines. How to represent texts in input space. *Machine Learning, 46*, 423–444.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. *Studies in linguistic analysis*. *Philological society*. Oxford: Blackwell.

Han, J. W., & Kamber, M. (2006). *Data mining concepts and techniques* (2nd ed.). Morgan Kaufmann Publishers.

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

Ho, T. B., & Nguyen, N. B. (2000). Non-hierarchical document clustering based on a tolerance rough set model. *International Journal of intelligent systems, 17*, 199–212.

Hotho, A., Staab, S., & Stumme, G. (2003). Ontologies improve text document clustering. In *Proceedings of the 3rd IEEE international conference on data mining* (pp. 541–544).

Jose, M. G. H. (2003). *Text representation for automatic text categorization*. Online: http://www.esi.uem.es/~jmgomez/tutorials/eacl03/slides.pdf.

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

Kolda, T. G., & O'Learly, D. P. (1998). A semidiscrete matrix decomposition for latent semantic indexing in information retrieval. *ACM Transactions on Information Systems, 16*(4), 322–346.

Lewis, D. D. (1992a). Text representation for intelligent text retrieval: A classification-oriented view. In S. J. Paul (Ed.), *Text-based intelligent systems: current research and practice in information extraction and retrieval* (pp. 179–197). Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates, Inc., Publishers.

Lewis, D. D. (1992b). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of SIGIR'92* (pp. 37–50).

Li, Y. J., Chung, S. M., & Holt, J. D. (2008). Text document clustering based on frequent word meaning sequences. *Data & Knowledge Engineering, 64*, 381–404.

Papka, R., & Allan, J. (1998). Document classification using multiword features. Document classification using multi-word features. In *Proceedings of the seventh international conference on information and knowledge management* (pp. 124–131). Bethesda, Maryland, United States.

Quinlan, J. R. (1983). Learning efficient classification procedures and their applications to chess and games. In Ryszard S. Michalski, Jaime G. Carbonell, & Tom M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (pp. 463–482). Los Altos, CA: Morgan Kaufmann.

Roberston, S. (2004). Understanding inverse document frequency: On theoretical argument for IDF. *Journal of Documentation, 60*(5), 503–520.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison Wesley.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill Book Company.

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

Scott, S., & Matwin, S. (xxxx). Text classification using WordNet Hypernyms. In *Proceedings of the COLING/ACL 98 workshop on usage of WordNet in natural language processing systems* (pp. 45–52).

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21.

Sparck Jones, K. (2004). IDF term weighting and IR research lessons. *Journal of Documentation, 60*(6), 521–523.

Vapnic, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Weiss, S., Indurkhya, Tong Zhang, T., & Damerau, F. (2005). *Text mining: predictive methods for analyzing unstructured information* (pp. 1–45). Springer Science and Business Media, Inc..

Weston, J., & Watkins, C. (1998). Multi-class support vector machines. *Technical report CSD-TR-98-04*. Royal Holloway, University of London, Department of Computer Science.

Yan, H., Grosky, W. I., & Fotouhi, F. (2008). Augmenting the power of LSI in text retrieval: Singular value rescaling. *Data & Knowledge Engineering, 65*(1), 108–125.

Yang, Y. M., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings on the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 42–49). Berkeley, CA.

Zhang, W., Yoshida, T., & Tang, X. J. (2007). Text classification with multi-words. In *Proceedings of 2007 IEEE international conference on system, man, and cybernetics* (pp. 3519–3524). Montreal Canada.

Zhang, W., Yoshida, T., & Tang, X. J. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems, 21*(8), 879–886.

Zhang, W., Yoshida, T., & Tang, X. J. (2009b). Using ontology to improve precision of terminology extraction from documents. *Expert Systems with Applications, 36*(5), 9333–9339.

Zhang, W., Yoshida, T., Tang, X. J., & Ho, T. B. (2009a). Augmented mutual Information for multi-word extraction. *International Journal of Innovative Computing, Information and Control, 5*(2), 543–554.

Zhou, X. H., Hu, X. H., & Zhang, X. D. (2007). Topic Signature language models for ad hoc retrieval. *IEEE transactions on Knowledge and Data Engineering, 19*(9), 1–12.