Knowl Inf Syst (2010) 25:473–491
DOI 10.1007/s10115-009-0250-y

REGULAR PAPER

Short text similarity based on probabilistic topics

**Xiaojun Quan · Gang Liu · Zhi Lu ·
Xingliang Ni · Liu Wenyin**

Received: 20 November 2008 / Revised: 19 May 2009 / Accepted: 21 August 2009 /
Published online: 17 September 2009

Abstract In this paper, we propose a new method for measuring the similarity between two short text snippets by comparing each of them with the probabilistic topics. Specifically, our method starts by firstly finding the distinguishing terms between the two short text snippets and comparing them with a series of probabilistic topics, extracted by Gibbs sampling algorithm. The relationship between the distinguishing terms of the short text snippets can be discovered by examining their probabilities under each topic. The similarity between two short text snippets is calculated based on their common terms and the relationship of their distinguishing terms. Extensive experiments on paraphrasing and question categorization show that the proposed method can calculate the similarity of short text snippets more accurately than other methods including the pure TF-IDF measure.

Keywords Text similarity measures · Information retrieval · Query expansion · Text mining · Question answering

1 Introduction

Calculating the similarity between two very short text snippets is increasingly an urgent task that needs to be well solved in text-related research and applications, such as text mining,

X. Quan · G. Liu · Z. Lu · X. Ni · L. Wenyin (✉)
Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China
e-mail: csliuwy@cityu.edu.hk

X. Quan
e-mail: xiaoquan@cityu.edu.hk

X. Ni
Department of Computer Science and Technology, University of Science and Technology of China,
Hefei, China

X. Ni · L. Wenyin
Joint Research Lab of Excellence, CityU-USTC Advanced Research Institute, Suzhou, China

web information retrieval and question answering. In these areas, evaluation of the similarity between two short text snippets or sentences is a key step and different short text similarity measures can greatly affect the accuracy of these systems. For example, in BuyAns [1], which is a user-interactive question answering system, evaluation of the similarity of two short text snippets, such as two questions, is an essential step for question categorization and question recommendation. In the area of web document retrieval, computing similarities between short text snippets has been proved to be an important technique, when the titles of pages can be used to represent documents in the named page finding task [2]. In the area of text mining, short text similarity is a useful measure for discovering hidden knowledge from textual databases [3]. Short text similarity has also wide applications in areas such as query reformulation and sponsored search [4].

Most traditional techniques for measuring the similarity of two documents mainly focus on comparing the word co-occurrences in them. The methods employing this strategy for documents can usually achieve good results, because they may share more common words than short text snippets. Because of the sparseness of short text snippets, state-of-the-art techniques usually fail to achieve desired accuracy [5]. Take the question "what is the price of apple" as an example, after removing stop words, only two words are left. The vector of this question in the vector space model (VSM) [6] may contain hundreds of or even thousands of dimensions but only two of them contribute to similarity calculation. High dimensional sparse vectors in the VSM may provide very limited common components and accordingly affect the similarity accuracy.

Much research work has been reported attempting to overcome the short text similarity problem. Most of them adopt one or more of the following methods. One way is to expand and enrich the short text snippets with a search engine (e.g., Google) [7–9]. For each short text snippet or sentence, they employ a search engine to find a set of related pages, which are then used to represent the original short text snippet or sentence. The similarity of two short text snippets is usually decided by the correlation of their corresponding search pages. In another method, in order to find the precise relationship between two short snippets, an external lexical database, such as WordNet [10], is employed to mine the relationships among words [11]. Although a large number of words and their semantic relationship can be found in the lexical database, there are still a few new words not included yet. For example, "Wikipedia" is a very famous free encyclopedia on the Web, but WordNet does not include this word. Hence, it is difficult to measure the semantic similarity between "Wikipedia" and "encyclopedia" by using a dictionary-based method. Take two questions "Who is the first President of USA?" and "Who is George Washington?" as an example, the above method will not detect their relationship although they are both talking about the same person actually.

The short text snippets usually contain very limited common words. Hence, it is usually difficult to detect their intrinsic relationship based on traditional similarity measures. If we can find several common topics, which are the topics related to both of the two short text snippets, and use these common topics as the "third party" to compare each of the snippets indirectly, it is possible to obtain their intrinsic relationship. That is, the two short text snippets are not compared to each other directly. Instead, each of them is compared with the "third-party" topics related to both. They are considered as similar if each is similar to the "third-party" topics. For example, given two questions "What is the price of apple?" and "How much is a pear?", obviously, they are both related to *fruit* although they share no meaningfully common words. Therefore, if there is a third-party topic *fruit* which contains the knowledge about both *apple* and *pear*, it is very likely to detect the intrinsic relationship between the two questions based on this topic.

In this paper, we propose a novel method for calculating the similarity between two short text snippets. The proposed method is built aiming to mine the relation of the non-common terms between the short text snippets based on a series of third-party topics. After that, the vectors of the two short text snippets in the VSM are modified according to the discovered relation, and then the cosine metric is employed to calculate the final similarity. Specifically, we firstly finds a set of distinguishing terms, which are terms occurring in one piece of short text snippet but not in the other (formally defined in Eq. (8)), for each of the two short text snippets. Since for a pair of short text snippets there are few common words, the distinguishing terms usually indicate the relationship of the two short text snippets. The distinguishing terms are examined in a series of probabilistic topics (serve as the third-party topics) extracted by the Gibbs sampling algorithm [12]. The relationship between the two short text snippets is determined based on their common words, as well as the probabilities of the distinguishing terms under each probabilistic topic. We evaluate the proposed method on the tasks of paraphrasing and question categorization. Experimental results show that the proposed method can calculate the similarity between short text snippets more accurately than previous techniques including the pure Term Frequency-Inverse Document Frequency (TF-IDF [13]) measure, and it can also improve the performance of short text categorization.

The rest of this paper is organized as follows. We briefly review related work in Sect. 2. Section 3 introduces the Probabilistic Topic Model involved in this paper. Our short text similarity method is presented in Sect. 4, and the application of this model on short text categorization is discussed. Our experimental evaluation is exhibited in Sect. 5, and finally, we draw a conclusion and discuss future work in Sect. 6.

2 Related work

Previous research work on text-related information retrieval (IR) mainly focuses on document processing since the involved text contains many words. However, due to the emergence of more and more short text applications, e.g., email, bulletin board system (BBS), and question answering, the need for short text snippet processing is discovered. Intuitively, many traditionally effective techniques for document similarity assessment can be directly applied to calculate the similarity between short text snippets. Such solutions, which usually rely on discovering word co-occurrence, are usually effective when dealing with documents because the documents with related topic usually contain many co-occurring words. However, short text snippets with similar meanings do not necessarily share a lot of words, and the above methods may fail to perform well in this case.

Certain extensions of the word co-occurrence method have been made to improve the effectiveness of the traditional techniques on short text snippets. Hatzivassiloglou et al. [14] propose a method combining primitive features (including word co-occurrence, noun phrase matching, WordNet synonyms, common semantic classes for verbs and shared proper nouns) and composite features (ordering, distance and primitive). However, it is just a straightforward synthesis of a number of various former approaches and its experiment only shows that it works well on paragraph-length text. Okazaki et al. [15] propose a method to compute the similarity between two sentences by using a lexical database. In their method, sentence similarity is simply obtained by aggregating similarity values of all pairs of words. However, this method has two shortcomings: firstly a word with more synonyms will have more impact on the resultant similarity; secondly it does not deal with how to choose the meaning of a word with polysemy from the lexical database. Similar to Okazaki, Li et al. [11] also employ a lexical knowledge base and propose a sentence similarity measurement based on lexical database and word ordering. However, the relationships among words are not fixed and usually rely on the particular scenario of application. Moreover, many words usually cannot be found in the lexical base, such as the names of persons and places. Mihalcea et al. [16] introduce another method for measuring the similarity of short text snippets, which use both corpus-based and knowledge-based measures when acquiring words similarity. In their method, they also consider the *specificity* of a word, and they use the inverse document frequency (*idf*) [13] to calculate the *specificity*.

Query expansion, which automatically enriches the original query with terms that are semantically related, has long been suggested as an effective solution to resolve the mismatch issue in IR [4,17,18]. Sahami and Heilman [8] present a novel technique for measuring the similarity of short text snippets, which utilizes search engines, such as Google, to provide a greater context for the given short text snippets, just like certain query expansion techniques. Because of its great dependency on the search engine's database, the results returned by this method are unstable. Similar to the work of Sahami and Heilman [8], Bollegala et al. [7] also take advantage of search engines to get the semantic relationship between words.

Phan et al. [5] have introduced a new method to classify short text snippets based on Latent Dirichlet Allocation [19], which is probably the most similar work to ours. They initially execute Gibbs sampling on a large external knowledge base, such as Wikipedia, as background data to extract a number of hidden topics. Gibbs sampling is then run again on the particular dataset of interest by incorporating into the word-topic assignments derived from the previous topic extraction procedure. Finally the original text is converted to a new format that is convenient for processing by a classifier. However, the optimal topics to be extracted are usually determined by the specific dataset and application instead of the external knowledge base. Since their method has to run Gibbs sampling again on the test dataset with sufficient data, it is not suitable for calculating the similarity of a pair of documents. The method proposed in this paper also extracts probabilistic topics from a collection of background data (such as the questions in BuyAns [1] and Yahoo! Answers [20]) but focuses on discovering the intrinsic relationship of words by examining their probabilities under the extracted probabilistic topics, which is the basis for calculating the similarity of a pair of short text snippets.

3 Probabilistic topic model

The Probabilistic Topic Model [19] is a generative model for documents, based upon the assumption that a document is composed of multiple topics, and each topic is treated as a probability distribution over words. The Probabilistic Topic Model makes explicit assumptions about how to generate a document, and identifies the latent structure that underlies the document with sophisticated statistical methods. We first introduce the notation used in this paper and the basic model.

3.1 Notation and the basic model

Given a collection of M documents $D = \{d_1, d_2, \ldots, d_M\}$ with N unique words $W = \{w_1, w_2, \ldots, w_N\}$, we assume the collection contains Z hidden topics $T = \{t_1, t_2, \ldots, t_Z\}$. For each word token i in D, we use $W^{(i)}$ to represent the corresponding word of i, $D^{(i)}$ its document, and $T^{(i)}$ its topic.

When generating a new document with the Probabilistic Topic Model, a distribution over topics is firstly chosen. Each word in the document is then iteratively sampled from this distribution of topics. The probability of word $W^{(i)}$ in the new document d is

$$P\left(W^{(i)}\right) = \sum_{j=1}^{Z} P\left(W^{(i)}|t_j\right) P(T^{(i)} = t_j), \tag{1}$$

where, $P(T^{(i)} = t_j)$ represents the probability of t_j for generating document d, and $P(W^{(i)}|t_j)$ is the probability of word $W^{(i)}$ under t_j.

A variety of probabilistic topic models have been investigated during the past few years, and a large amount of them are implemented based on the above idea but slightly different in the format of statistical assumptions. Hofmann [21] introduces a new probabilistic topic approach, known as probabilistic latent semantic indexing (PLSI), which is an alternative to Latent Semantic Indexing. The PLSI method assumes that each word in a document corresponds to only one topic. Based on this model, Blei et al. [19] integrate into a Dirchlet prior and accordingly propose the latent Dirchlet allocation (LDA), which has been pointed out to be more attractive than PLSI since LDA is a well-defined generative model and it can overcome several problems of PLSI [19]. In the following section, we will briefly introduce the principle of LDA.

3.2 Latent Dirichlet allocation

LDA has broad applications for general discrete datasets, while text is a typical example to which this model can be applied. Specifically, the process of generating a document with n words by LDA can be described as follows [19]:

1. Choose the number of words, n, according to Poisson Distribution;
2. Choose the distribution over topics, θ, for this document by Dirchlet Distribution;
3. To characterize each of the n words $W^{(i)}$:
   (a) Choose a topic $T^{(i)} \sim$ Multinomial (θ);
   (b) Choose a word $W^{(i)}$ from $P(W^{(i)}|T^{(i)}, \beta)$.

From the above description of creating a new document with n words, we can obtain the marginal distribution of the document:

$$P(d) = \int\limits_{\theta} \left(\prod_{i=1}^{n} \sum_{T^{(i)}} P(W^{(i)}\big|T^{(i)}, \beta) P(T^{(i)}\big|\theta) \right) P(\theta\big|\alpha) d\theta, \tag{2}$$

where, $P(\theta|\alpha)$ is derived by Dirichlet Distribution parameterized by α, and $P(W^{(i)}|T^{(i)}, \beta)$ is the probability of $W^{(i)}$ under topic $T^{(i)}$ parameterized by β. The parameter α can be viewed as a prior observation counting on the number of times each topic is sampled in a document, before we have actually seen any word from that document. The parameter β is a hyperparameter determining the number of times words are sampled from a topic [19], before any word of the corpus is observed. Finally, the probability of the whole corpus D can be derived by taking the product of all documents' marginal probabilities:

$$P(D) = \prod_{i=1}^{M} P(d_i). \tag{3}$$

3.3 Topic extraction

The topic-word distribution $P(W^{(i)}|T^{(i)}, \beta)$ in Eq. (2) is an important factor for the implementation of LDA. Among many approaches to estimating the topics, Griffiths and Steyvers [12] introduce a method using Gibbs Sampling to extract topics in the corpus, which is also adopted in this paper for topics extraction.

The Gibbs sampling method estimates the probability of assigning each word token in the collection to each topic, conditioned on the topic assignments to all other word tokens. We represent this conditional distribution as $P(T^{(i)} = t_j|T^{(-i)},\ W^{(i)},\ D^{(i)}, \cdot)$, where $T^{(i)} = t_j$ means the assignment of token i to topic t_j, $T^{(-i)}$ refers to the assignments of all other word tokens, and "$\cdot$" refers to all other known or observed information such as $W^{(-i)}$ and $D^{(-i)}$, and hyperparameters α and β. Griffiths and Steyvers [12] show this probability $P(T^{(i)} = t_j|T^{(-i)}, W^{(i)}, D^{(i)}, \cdot)$ can be calculated by:

$$P(T^{(i)} = t_j \big| T^{(-i)}, W^{(i)}, D^{(i)}, \cdot) \propto \frac{C_{T^{(i)}}^{W^{(i)}} + \beta}{\sum_{W^{(i)}} C_{T^{(i)}}^{W^{(i)}} + N \cdot \beta} \cdot \frac{C_{D^{(i)}}^{T^{(i)}} + \alpha}{\sum_{T^{(i)}} C_{D^{(i)}}^{T^{(i)}} + Z \cdot \alpha}, \tag{4}$$

where $C_{T^{(i)}}^{W^{(i)}}$ represents the number of times word $W^{(i)}$ is sampled from topic $T^{(i)}$, not including the current token i, and $C_{D^{(i)}}^{T^{(i)}}$ is the number of times topic $T^{(i)}$ is assigned to any word token in document $D^{(i)}$, not including the current instance token i. Eq. (4) is composed of two parts, while the first part is the probability of word $W^{(i)}$ under topic t_j and the second part is the probability of topic t_j under the current topic distribution for document $D^{(i)}$. The Gibbs sampling algorithm begins with the assignment of each word token to a random topic in T, determining the initial state of the Markov chain. This chain is then executed for a number of iterations, each time finding a new state by sampling each $T^{(i)}$ from the distribution specified by Eq. (4). After enough iterations for the chain to approach the target distribution, the final values of the $T^{(i)}$ variables are recorded. Once the iterative sampling is complete, the actual probability of word w_i under topic t_j is calculated as follows:

$$P(w_i \big| t_j) = \frac{C_{t_j}^{w_i} + \beta}{\sum_{i=1}^{N} C_{t_j}^{w_i} + N \cdot \beta}. \tag{5}$$

Griffiths and Steyvers [12] provide more details of the above procedure in their paper.

3.4 Important parameters

Three important parameters are mentioned in the LDA theory: α, β and the number of topics Z. The optimal parameters of α and β depend on the topic number and the size of vocabulary in the document collection, and they are typically set as $\alpha = 50/Z$ and $\beta = 0.01$ [12]. There is still an important parameter, the number of topics Z, needed to be optimized to use the probabilistic topics for our short text similarity algorithm. The number of topics can affect the interpretability of the extracted topics and accordingly influence the performance of calculating the similarity between two short text snippets. In case too few numbers of topics is chosen, the resultant topics will be too general to describe the collection in detail. However, a solution with too many topics will result in very narrow topics. Griffiths and Steyvers [12] introduce a strategy using the Bayesian model to find the optimal number of topics. The idea underlying their method is to estimate the posterior probability, $P(W|Z)$, of the model after having integrated over all possible parameter settings, which contains complex computation.

In this paper, we fix the hyperparameters of α and β as $\alpha = 50/Z$ and $\beta = 0.01$, and choose the optimal number of topics that leads to best performance for our task.

4 Topic based similarity

One of the most important reasons that traditional text similarity techniques fail to perform well on short text snippet lies in that there are usually very few co-occurred words in two short text snippets. Many techniques aiming to solve this problem focus on expanding the short text snippet through a huge thesaurus, such as WordNet. Intuitively, this kind of solutions can explore well the relationship between two short text snippets since with such thesaurus the intrinsic relationship among words may be discovered. However, as mentioned above, the relationships among words usually rely on the particular scenario of application. In this paper, we propose a novel method to discover the intrinsic relationships among words, based on the statistical analysis of background dataset, and accordingly derive the similarity between two short text snippets.

4.1 Comparison of two short text snippets through third-party topics

Intuitively, the distinguishing terms between two pieces of short text snippets usually contain useful information that can indicate their implicit relationships. Hence, if we can find certain common "third-party" topics for both short text snippets, and calculate the similarity between the two snippets by comparing the distinguishing terms with these topics, we may discover the real relationship underlying the two short texts. In most applications, such as question answering, there is usually sufficient data, e.g., accumulated questions. We assume that the collection of data, as the background data, contains sufficient knowledge about the relationships among words, from which we can discover the intrinsic relationship among short text snippets with the statistical method. Take the pair of sentences "The price of apple is high" and "The price of banana is low" as an example, the traditional methods can only find one co-occurred word price in the two sentences after stop words are removed. However, if compared them with the third-party topic "fruit", they will exhibit closer relationship as the words apple and banana are similar within this topic. We will describe our method in detail in the following passages.

4.2 Topics based similarity method

The short text snippets in present information services are usually organized as categories or related forms, such as boards in BBS and question answering systems. Our method starts with extracting probabilistic topics using the Gibbs sampling algorithm without taking the categories information into consideration. That is, we treat all short text snippets contained in the collection equally and perform Gibbs sampling on them, and accordingly derive a set of T probabilistic topics. Even though these topics are extracted with the optimal parameters, the problem of how to make use of these topics effectively is still a great challenge.

To measure the similarity between two short text snippets d_1 and d_2 in a data collection D, we firstly represent them as vectors:

$$\begin{aligned} V^{(1)} &= \left\{ v_1^{(1)}, v_2^{(1)}, \ldots, v_N^{(1)} \right\}, \\ V^{(2)} &= \left\{ v_1^{(2)}, v_2^{(2)}, \ldots, v_N^{(2)} \right\}, \end{aligned} \tag{6}$$

where $v_i^{(j)}$ is the weight of the ith feature in the vector of d_j and is defined by the TF-IDF measure as follows.

$$V_i^{(j)} = tf_{ij} \times \log(M/df_i), \tag{7}$$

where M is the total number of documents in the collection and df_j is the document frequency, i.e., the number of documents in which term w_i occurs. tf_{ij} is the term frequency of term w_i in document d_j. In this paper, tf_{ij} is simply the number of occurrences of term w_i in document d_j.

The short text snippet usually contains a few words and the unimportant words (such as the words appear in many short text snippets) may affect the similarity calculation greatly. Hence, the employment of TF-IDF is necessary in this case.

After performing Gibbs sampling with appropriate parameters on D, we derive a set of probabilistic topics $T = \{t_1, \ldots, t_i, \ldots, t_Z\}$, each of which has the following form:

$$t_i = \{t_{i1}, t_{i2}, \ldots, t_{iN}\}, \tag{8}$$

where t_{ij} is a probability weighing how the jth word is possibly related to topic t_i. For the given two short text snippets d_1 and d_2, we derive their distinguishing term sets as follows:

$$\begin{aligned} Dist(d_1) &= \left\{w \middle| w \in d_1, w \notin d_2\right\}, \\ Dist(d_2) &= \left\{w \middle| w \in d_2, w \notin d_1\right\}. \end{aligned} \tag{9}$$

An extracted probabilistic topic can be viewed as a kind of "concept" which is represented by a set of words and their corresponding probabilities. If terms in $Dist(d_1)$ and $Dist(d_2)$ co-occur under some topic with high probability scores, it is rational to conclude that the two distinguishing sets are correlated with each other to certain extent. For each topic t_i, the term with the highest probability is selected from $Dist(d_1)$ and the highest probability term from $Dist(d_2)$ is also selected. If the probabilities for the two selected terms are both higher than a predefined threshold λ under the same topic t_i, we think the two terms are related to each other in this topic. Assume the index of the selected term for $Dist(d_1)$ is m, and n for $Dist(d_2)$, we modify the vector of d_1 and d_2, $V^{(1)}$ and $V^{(2)}$, as follows:

$$\begin{aligned} v_n^{(1)} &= v_n^{(1)} + v_n^{(2)} \times P(w_n | t_i), \\ v_m^{(2)} &= v_m^{(2)} + v_m^{(1)} \times P(w_m | t_i). \end{aligned} \tag{10}$$

With the relationship of the two distinguishing term sets on a variety of extracted topics, we can renew the vectors of d_1 and d_2, and calculate their similarity with the cosine method described in Eq. (11). The proposed similarity method is described in Fig. 1.

$$Cosine\left(V^{(1)}, V^{(2)}\right) = \frac{V^{(1)} \cdot V^{(2)}}{\left|V^{(1)} \parallel V^{(2)}\right|}. \tag{11}$$

5 Experiments

To evaluate the proposed method, two experiments are designed and conducted in this section:

1. Mihalcea et al. evaluate their short text method by identifying if two given text segments are paraphrases [16]. In the first experiment, we use the same paraphrasing method with the same data collection as Mihalcea et al. [16] do and we compare the performance of our method with theirs.

```
**ShortTextSimilarity** $(d_1, d_2, T, \lambda)$
    **Input**:
        $d_1, d_2$: the input short text snippets
        $T$: the set of extracted topics
        $\lambda$: the threshold for distinguishing terms
    **Output**:
        $Sim$: The similarity between $d_1$ and $d_2$
1. Init the text vectors $V^{(1)}$, $V^{(2)}$ for $d_1$ and $d_2$ respectively
2. Get the distinguishing term sets of $d_1$ and $d_2$, $Dist(d_1)$
   and $Dist(d_2)$, according to Eq. (9)
   **For** each topic $t_i$ in $T$
3. Let $m = \arg\max_j P(w_j \mid t_i),\ w_j \in Dist(d_1)$
4. Let $n = \arg\max_j P(w_j \mid t_i),\ w_j \in Dist(d_2)$
     **If** $p(w_m \mid t_i) \geq \lambda$ and $p(w_n \mid t_i) \geq \lambda$
5.          Modify $V^{(1)}$ and $V^{(2)}$ according to Eq.(10)
     **End If**
   **End For**
6. $Sim$ := $cosine(V^{(1)}, V^{(2)})$
7. **Return** $Sim$
```

Fig. 1 The similarity algorithm

2. In the second experiment, we test our method for the question categorization task. A question is a special format of short text snippet, and the categorization of questions is a technique that automatically assigns one of the predefined categories to a newly posted question according to the topic or content of the question and the categories. There has been some research work [22,23] on question type classification, which aim at classifying questions into some predefined categories according to certain constraints on corresponding potential answers, e.g., "person", "location" and so on. However, among many online services, such as on-line interactive question answering systems [1], there emerges another need of categorizing questions according their topics. In other words, the question topic categorization task assigns a category (such as "computer", "education", "sports" and so on) to an input question if and only if the content of the question is in accordance with the topic of the category. To classify a question, many classifiers (such as K-Nearest Neighbor classifier, or KNN in short) need to firstly calculate the similarity between two questions. Since we propose a new method to compute the similarity between two short text snippets, it is intuitive and convenient to apply it in the state-of-the-art machine learning techniques, such as KNN, to classify questions.

Table 1 The result of our and other methods on the Microsoft paraphrase corpus

Metric	Accuracy	Precision	Recall	F1
TBS	69.9	1.0	69.9	82.3
The methods presented by Mihalcea et al.				
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
J&C	69.3	72.2	87.1	79.0
L&C	69.5	72.4	87.0	79.0
Lesk	69.3	72.4	86.6	78.9
Lin	69.3	71.6	88.7	79.2
W&P	69.0	70.2	92.1	80.0
Resnik	69.0	69.0	96.4	80.4

Moreover, the effectiveness of the proposed similarity measure can be verified through the categorization performance. However, since the vector for each short text snippet in the proposed method is not fixed but is determined by other text (see Eq. (10)), some of the popular classifiers (such as Support Vector Machine) are inappropriate to verify our method.

5.1 Paraphrasing

Paraphrasing, which expresses one thing in other words, plays an important role in showing the variety and complexity of natural language document [24]. Mihalcea et al. in their work [16] use the Microsoft paraphrase corpus [24] to test their method on identifying whether each pair of sentences is a paraphrase. The corpus consists of 4,076 training and 1,725 testing pairs, and each pair is labeled by two human annotators to determine whether two sentences in this pair are close enough to be a paraphrase. The agreement of the two human annotators on all the pairs in this corpus is approximately 83%, and this value is considered as an upperbound for the task of automatically identifying the paraphrases in this corpus. Note that Mihalcea et al. take only the testing data into consideration in their work. However, for our method, the training data is used for extracting probabilistic topics and the testing data is for testing. Similar to Mihalcea et al. [16], the text similarity for each candidate paraphrase pair in the test set is calculated with our method, and the candidate paraphrase pair will be considered as a paraphrase if the similarity score exceeds 0.5. Accordingly, the number of correctly identified paraphrase pairs in the test set can be derived.

We evaluate the results in terms of *accuracy*, *precision*, *recall* and $F1$ [25,26], calculated with respect to the true values in the corpus. The number of iterations in Gibbs sampling used in this paper is set as 300, as we find that Gibbs sampling usually can approach the target distribution after 300 rounds of iterations. The parameters α and β are set as $\alpha = 50/Z$ and $\beta = 0.01$, respectively, as stated above. The number of topics Z is set as 100 and λ is 0.05. The result is shown in Table 1, and the proposed topic based similarity measure is denoted as TBS. For comparison, we also list the results of corpus-based methods (PMI-IR, LSA) and knowledge-based measures (J&C, L&C, Lesk, Lin, W&P and Resnik) presented by Mihalcea et al [16] as the baselines.

As reported by Mihalcea et al. [16], one of the corpus-based method, PMI-IR, performs the best among all the involved measures in the paraphrasing task. Another corpus-based measure, LSA, also shows promising F1 performance. According to the above description about the proposed method, TBS, we can find that TBS is also a kind of corpus-based measure. This is because TBS focuses on mining the implicit relationship between two short text snippets based on probabilistic topics extracted from background data (corpus). The performance of TBS is also significant according to the results in Table 1, where the new method TBS achieves better *accuracy* and $F1$ than the other 8 methods introduced by Mihalcea et al. [16], except for the *accuracy* on PMI-IR which is equal to our method. The table also shows our method achieves much better *precision* than the other 8 methods though its *recall* is rather lower. The reason behind this phenomenon is probably due to that the other 8 methods are more likely to assign a larger similarity score to each candidate paraphrase pair and their recall values are therefore higher but their precision values are lower. In contrast, TBS has a trend of assigning a relatively lower similarity score to each candidate paraphrase. However, since $F1$is the combination of precision and recall, the overall performance of the presented methods can be examined from $F1$.

From the evaluation in the paraphrase task with *accuracy* and $F1$ as the performance metric, we can find the proposed similarity method shows improvement in evaluating whether two sentences are a paraphrase. This is one way to demonstrate the effectiveness of the new similarity method. Next, another experiment is conducted to evaluate TBS on question corpus for the question categorization task.

5.2 Question categorization

In this section we first present several question samples to show the effectiveness of the proposed method on calculating similarity. A series of experiments for question categorization are also conducted to evaluate the proposed method. As we know, the key tasks in KNN are the similarity calculation and selection of a proper number of neighbors. If we leave out the selection of the proper number of neighbors, the performance of KNN will heavily rely on the precision of similarity calculation. If our method can calculate the similarity among short text snippets more precisely than TF-IDF, the KNN will select more appropriate neighbors for a test case and the classification performance of KNN will be higher. Vice versa, if we obtain a better result of KNN with our similarity measure than with TF-IDF, we think the improvement is due to our similarity measure. Therefore, if the performance of the KNN based on the proposed short text similarity method outperforms that of using TF-IDF as the similarity metric, we claim that it is rational to prove that our method is more effective on calculating short text similarity than the TF-IDF measure.

5.2.1 Data collection

We use two question collections[1] for question categorization in our experiments. One is obtained from BuyAns [1], which contains 1,120 questions derived from 32 boards (categories) of the system. The number of questions in each category varies from 14 to 108. The other question dataset is crawled from Yahoo! Answers [20], which is composed of 2,400 questions spreading into 11 categories. Each category contains at least 100 questions and at most 400 questions. After the removal of stop words, each word is stemmed into its root form. Each question of the two collections is composed of several English words (usually less than 10 words). No feature selection is performed in our experiments.

[1] The two question collections can be found at: http://www.cs.cityu.edu.hk/~liuwy/questionset/.

Table 2 Some sample questions from BuyAns

Category	Question
Category 1	Q1: How to operate MS Excel software?
	Q2: What is the newest version of Microsoft word?
	Q3: Can I use Microsoft Excel to draw pictures or any figures?
	Q4: Where can I buy the newest Microsoft Office?
Category 2	Q5: What is the price of Casio's DC?
	Q6: How to use a Digital Camera?
Category 3	Q7: What are the major tasks of mobile phones?
	Q8: What is color of N70?

5.2.2 Evaluation methodology

For the question categorization task, 70% of questions from each category are randomly selected for training, and the rest is for testing. The categorization process is executed for 10 iterations and the average performance is recorded. In our experiments, we compare the performance of three classifiers implemented as follows: (1) KNN, (2) Support Vector Machine (SVM), and (3) the KNN method combining our proposed topic based similarity measure (KNN_TBS). The KNN and SVM are both constructed using the TF-IDF method to weight each vector component of the question, and are used as baseline to compare with our proposed method. The number of neighbors in KNN and KNN_TBS are both fixed as 30. The parameters of Gibbs sampling in KNN_TBS are set as the same in Sect. 5.1. The SVM algorithm is implemented based on the libSVM tool [27]. In our experiments the linear kernel of SVM is used due to its competitive performance in the context of text categorization. The parameters in the linear kernel are set to their default values. The standard measures *precision*, *recall* and $F1$ [25,26] are employed to measure the performance of these methods. We also calculate the *Micro-Averages* and *Macro-Averages* [28] for the three metrics of *precision*, *recall* and $F1$.

5.2.3 Sample questions for evaluation

We select some questions from the BuyAns system as samples to test the proposed similarity measure, as shown in Table 2. Since it is difficult to determine the actual similarity between two questions, we simply assume that questions in the same category should be more similar than those in different categories. Hence, if the proposed topics based similarity method can increase the similarities between questions in the same category without augmenting the similarity of questions from different categories, it is rational to claim that the proposed method is more effective. We use the bag-of-words model (BOW) [29] combined with the cosine measure as the baseline of this test, and the results are exhibited in Table 3.

As shown in Table 3 the topics based similarity measure can indeed increase the similarity of questions from the same category. However, as each of the questions contains only few distinguishing terms after removing stop words, the improvement is limited. Moreover, the proposed method correctly detects the relationship between questions from different categories (we assume the questions from different categories have little relationship) as shown in Table 3. Note that there are two exceptions in the example, which are the results in the last two

Table 3 The similarities calculated by the proposed topic based method and the TF-IDF method

Question pair	TBS	BOW + Cosine
Q1: How to operate MS Excel software?	0.0758	0
Q2: What is the newest version of Microsoft word?		
Q2: What is the newest version of Microsoft word?	0.1998	0.1384
Q3: Can I use Microsoft Excel to draw pictures or any figures?		
Q2: What is the newest version of Microsoft word?	0.6953	0.6438
Q4: Where can I buy the newest Microsoft Office?		
Q2: What is the newest version of Microsoft word?	0	0
Q5: What is the price of Casio's DC?		
Q4: Where can I buy the newest Microsoft Office?	0.2079	0.1612
Q3: Can I use Microsoft Excel to draw pictures or any figures?		
Q4: Where can I buy the newest Microsoft Office?	0	0
Q6: How to use a Digital Camera?		
Q4: Where can I buy the newest Microsoft Office?	0	0
Q7: What are the major tasks of mobile phones?		
Q6: How to use a Digital Camera?	0.1072	0
Q7: What are the major tasks of mobile phones?		
Q6: How to use a Digital Camera?	0.1762	0
Q8: What is color of N70?		

rows of Table 3. The two pairs of questions are respectively derived from different categories but each of the pairs is assigned a nonzero similarity score. The reason for this phenomenon is that even though Q6 and Q7, as well as Q6 and Q8, belong to different categories, they are related in the extracted probabilistic topics. This is rational since the terms "Digital Camera" and N70 (mobile phone) are related to some extent in practice.

5.2.4 Question categorization

The number of topics in the proposed method is one of the most important parameters. As illustrated above, the optimal topics number is decided by the categorization performance. We test the *MicroF1* (*Micro-Averages* of $F1$) and *MacroF1* (*Macro-Averages* of $F1$) performance when different numbers of topics are selected. As shown in Figs. 2 and 3, the categorization achieves its best performance when the number of topics is 160 on the BuyAns dataset and 120 when on Yahoo! dataset.

The parameter λ in the proposed method is an important threshold deciding whether two distinguishing term sets are related to each other through the probabilistic topics. A larger value of λ may lead to neglect of the real relationship between the distinguishing term sets, while a smaller value may misjudge the relationship between the two sets. This parameter is determined according to the categorization performance empirically. As shown in Figs. 4 and 5, the optimal λ is 0.05 for BuyAns collection and 0.15 for Yahoo! collection. Note that the optimal value of λ relies on the training dataset for extracting probabilistic topics, since the probabilities of words in each topic may vary greatly when different training datasets are used.

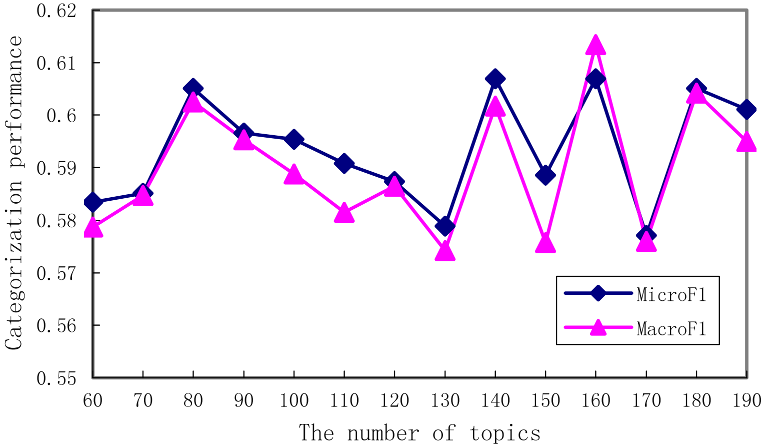

Fig. 2 *MicroF1* and *MacroF1* on BuyAns questions collection with different numbers of topics

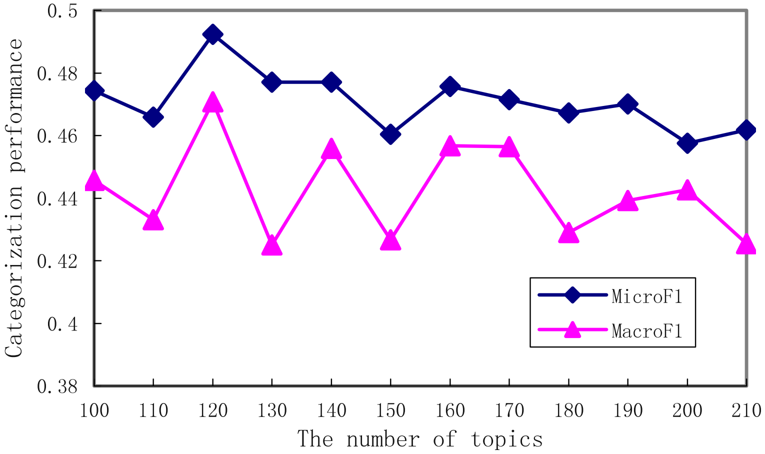

Fig. 3 *MicroF1* and *MacroF1* on Yahoo! Answer questions collection with different numbers of topics

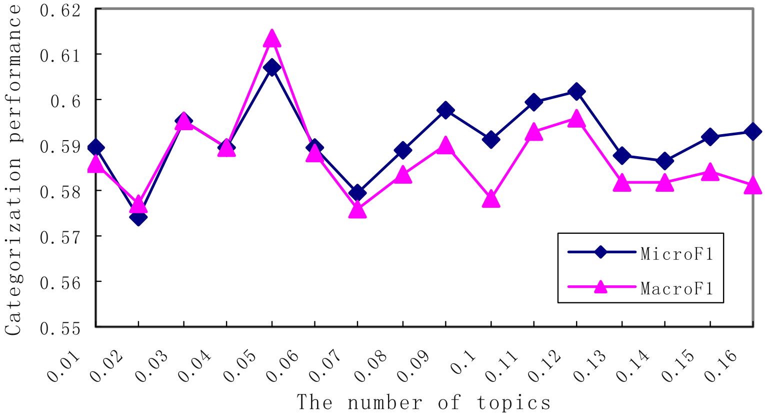

Fig. 4 Question categorization performance on BuyAns collection with different parameter λ

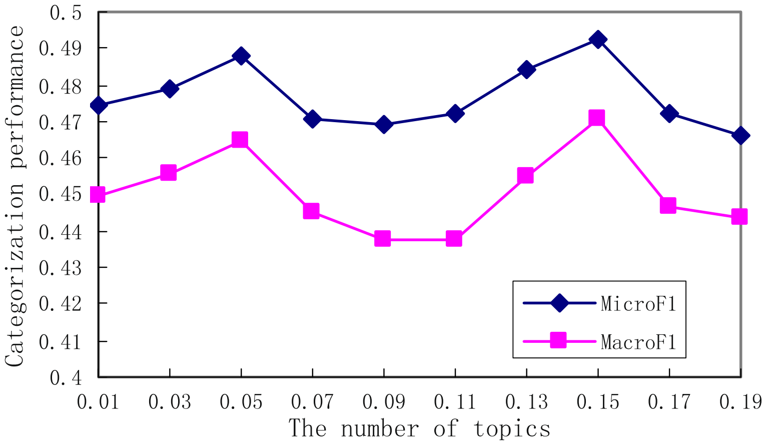

Fig. 5 Question categorization performance on Yahoo! collection with different parameter λ

Table 4 Question categorization results on the BuyAns dataset

Performance	KNN	KNN_TBS	SVM
Precision			
Micro-	0.5003	0.6069	0.5786
Macro-	0.5304	0.6582	0.7178
Recall			
Micro-	0.5003	0.6069	0.5786
Macro-	0.4577	0.5746	0.5369
F1			
Micro-	0.5003	0.6069	0.5786
Macro-	0.4911	0.6135	0.6142

Table 5 Question categorization results on the Yahoo! Dataset

Performance	KNN	KNN_TBS	SVM
Precision			
Micro-	0.3703	0.4924	0.3940
Macro-	0.3253	0.5316	0.4197
Recall			
Micro-	0.3703	0.4924	0.3940
Macro-	0.2881	0.4225	0.3236
F1			
Micro-	0.3703	0.4924	0.3940
Macro-	0.3054	0.4708	0.3652

The overall results are presented in Tables 4 and 5. For the BuyAns dataset, the number of topics in the proposed method is set as 160 and λ is 0.05, while the number of topics is 120 and λ is 0.15 for the Yahoo! collection. From these tables, we can see that the performance of KNN_TBS outperforms KNN by 0.10 on the BuyAns dataset, and it is also higher than the SVM based method in most of the performance metrics. On the Yahoo dataset, the performance increase of KNN_TBS is as high as 0.12–0.21 over the KNN method, and is also significant over the SVM based method in most of the performance metrics.

6 Conclusion

In this paper, we propose a novel method for calculating the similarity between two short text snippets. The proposed method computes the similarity between two short text snippets from two aspects. One is the common words between them and the other component is their distinguishing terms. In our method, the vector representation of each short text snippet can be different when it is compared with two different short text snippets. For comparison of each pair of short text snippets, we first find the distinguishing terms of them and then mine the relationship between the distinguishing terms. Our method mines the implicit relationship between the distinguishing terms by comparing them with several third-party topics. Specifically, the relationship between the distinguishing terms is discovered by examining their probabilities in a series of probabilistic topics, extracted by the Gibbs sampling algorithm. Finally, the similarity between two short text snippets can be calculated based on their common words and the relationship of their distinguishing terms. Two experiments on paraphrasing and question categorization are conducted to test the effectiveness of the proposed method. From the experimental results we can obtain the following conclusions:

1. The *accuracy* and $F1$ performance of our method for paraphrasing on the Microsoft paraphrase corpus are both higher than most of the methods introduced by Mihalcea et al. [16]
2. The performance of the KNN classifier for question categorization is greatly improved by using the proposed similarity measure. In addition, the proposed method has been applied in the question categorization module in the BuyAns system [1]. When a user submits a question, the system automatically suggests three most related boards (categories) for the user to confirm to host the question. It yields quite accurate categorization suggestions and user satisfaction.

In further work, we will explore the applications of the proposed method to more tasks, such as clustering and question recommendation. Furthermore, as the questions only contain several words and there are limited distinguishing terms between them, it is necessary to test the proposed method on other short text snippets that contain more words than questions but fewer words than documents. In addition, how to select the optimal parameters, especially λ and the number of topics, is still an important issue that worth further research.

Acknowledgments The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 7002488) and the China Semantic Grid Research Plan (National Grand Fundamental Research 973 Program, Project no. 2003CB317000).

References

1. Wenyin L, Hao TY, Chen W, Feng M (2009) A web-based platform for user-interactive question-answering. World Wide Web: Internet Web Inform Syst 12(2):107–124
2. Park EK, Ra DY, Jang MG (2005) Techniques for improving web retrieval effectiveness. Inform Process Manag 41:1207–1223

3. Atkinson-Abutridy J, Mellish C, Aitken S (2004) Combining information extraction with genetic algorithms for text mining. IEEE Intell Syst 19:22–30
4. Metzler D, Dumais S, Meek C (2007) Similarity measures for short segments of text. In: Proceedings of the 29th European conference on information retrieval (ECIR 2007). Lecture notes in computer science, vol 4425, Springer, Berlin (2007) pp 16–27
5. Phan XH, Nguyen ML, Horiguchi S (2008) Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th international conference on World Wide Web. ACM Press, New York, pp 91–100
6. Salon G (1989) Automatic text processing: the transformation, analysis and retrieval of information by computer. Addison-Wesley, Reading
7. Bollegala D, Matsuo Y, Ishizuka M (2007) Measuring semantic similarity between words using Web search engines. In: Proceedings of the 16th international conference on World Wide Web (WWW2007). ACM Press, New York, pp 757–766
8. Sahami M, Heilman T (2006) A Web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th international conference on World Wide Web (WWW2006). ACM Press, New York, pp 377–386
9. Yih W, Meek C (2007) Improving similarity measures for short segments of text. In: Proceedings of twenty-second conference on artificial intelligence (AAAI-07), Vancouver, July 22–26, pp 1489–1494
10. Fellbaum C (1998) WordNet: an electronic lexical database. MIT Press, Cambridge
11. Li YH, McLean D, Bandar ZA et al (2006) Sentence similarity based on semantic nets and corpus statistics. IEEE Trans Knowl Data Eng 18:1138–1150
12. Griffiths T, Steyvers M (2004) Finding scientific topics. Natl Acad Sci 101:5228–5235
13. Salon G, Yang CS (1973) On the specification of term values in automatic indexing. J Documentation 29(4):351–372
14. Hatzivassiloglou V, Klavans J, Eskin E (1999) Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning. In: Proceedings of joint SIGDAT conference on empirical methods in NLP and very large corpora., College Park, MD, USA, June 21–22
15. Okazaki N, Matsuo Y, Matsumura N et al (2003) Sentence extraction by spreading activation through sentence similarity. IEICE Trans Inform Syst E86D(9):1686–1694
16. Mihalcea R, Corley C, Strapparava C (2006) Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the American association for artificial intelligence (AAAI 2006), Boston, July 2006, pp 775–780
17. Lavrenko V, Croft WB (2001) Relevance based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, New Orleans, Louisiana, September 9–13. ACM Press, New York, pp 120–127
18. Zhai C, Lafferty J (2001) Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of the tenth international conference on Information and knowledge management, Atlanta, Georgia, October 5–10. ACM Press, New York, pp 403–410
19. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
20. http://answers.yahoo.com
21. Hofmann T (1999) Probabilistic latent semantic analysis. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence, Stockholm, Sweden, July 30–August 1, pp 289–296
22. Li X, Roth D (2002) Learning question classifiers. In: Proceedings of the 19th international conference on computational linguistics, Taipei, Taiwan, August 24–September 01, pp 1–7
23. Zhang D, Lee WS (2003) Question classification using support vector machine. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval, Toronto, Canada, July 28–August 01. ACM Press, New York, pp 26–32
24. Dolan WB, Quirk C, Brockett C (2004) Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In: Proceedings of the 20th international conference on computational linguistics, Geneva, Switzerland, August 23–27, No 350
25. Cesario E, Folino F, Locane A et al (2008) Boosting text segmentation via progressive classification. Knowl Inform Syst 15(3):285–320
26. Peng T, Zuo W, He F (2008) SVM based adaptive learning method for text classification from positive and unlabeled documents. Knowl Inform Syst 16(3):281–301
27. Chang C, Lin C (2001) LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/cjlin/libsvm/
28. Fragoudis D, Meretakis D, Likothanassis SD (2005) Best terms: an efficient feature-selection algorithm for text categorization. Knowl Inform Syst 8(1):16–33
29. Salton G, McGill M (1983) Introduction to modern information retrieval. McGraw-Hill, New York

Author Biographies

Xiaojun Quan received the B.E. degree in computer science from the Chang'an University in 2005 and the M.E. degree in computer science from University of Science and Technology of China in 2008. He is currently a research assistant in department of computer science, City University of Hong Kong. His research interests include data mining, information retrieval, question answering and anti-phishing.

Gang Liu received the B.E. degree in computer science from Tsinghua University. He is currently pursuing the Ph.D. degree in the department of computer science, City University of Hong Kong. His research interests include artificial intelligence approaches to computer security and privacy, web document analysis, information retrieval, and natural language processing.

Zhi Lu received his B.E. in computer science from Nanjing University of Science and Technology in 2007 and MSc degree in computer science from City University of Hong Kong in 2008. Currently, he is an MPhil student in department of computer science, City University of Hong Kong. His research interests include data mining, information retrieval, and question answering.

Xingliang Ni is a Ph.D. student in the department of computer science at University of Science and Technology of China. He also joints in the collaborated Ph.D. education schema of the City University of Hong Kong. He received his B.S. degree from the Hefei University of Technology. His research interests include information retrieval, machine learning and natural language processing.

Liu Wenyin is an assistant professor in the computer science department at the City University of Hong Kong. Before that, he was a full time researcher at Microsoft Research China/Asia. His research interests include question answering, anti-phishing, graphics recognition, and performance evaluation. He has a BEng and MEng in computer science from Tsinghua University, Beijing and a DSc from the Technion, Israel Institute of Technology, Haifa. In 2003, he was awarded the International Conference on Document Analysis and Recognition Outstanding Young Researcher Award by the International Association for Pattern Recognition (IAPR). He is also TC10 chair of IAPR and a guest professor of University of Science and Technology of China (USTC). He is a senior member of IEEE and a member of the editorial board of the International Journal of Document Analysis and Recognition (IJDAR).