International Journal of Innovative
Computing, Information and Control
Volume **13**, Number **5**, October **2017**

 ISSN 1349-4198
pp. **1693–1708**

TEXT SIMILARITY MEASUREMENT WITH SEMANTIC ANALYSIS

Xu Li, Na Liu, Chunlong Yao and Fenglong Fan

School of Information Science and Engineering
Dalian Polytechnic University
No. 1, Qinggongyuan, Ganjingzi District, Dalian 116304, P. R. China
lixu102@aliyun.com

Received February 2017; revised June 2017

Abstract. *The traditional text similarity measurement methods based on word frequency vector ignore the senses of words and the semantic relationships between words, which have become obstacles to text similarity calculation, together with the high-dimensionality and sparsity of document vector. To address the problems, word sense disambiguation based on unsupervised learning is used to identify the senses of words and the key senses are selected to construct the feature vector for document representation. The definition of concept similarity and the similarity weighting factor between vectors are proposed to calculate similarity between two documents on the semantic level. The experimental results on benchmark corpus demonstrate that the proposed approach promotes the evaluation metrics of F-measure.*

Keywords: Text similarity measurement, Word sense disambiguation, Unsupervised learning, Concept similarity computing

1. **Introduction.** Document clustering automatically partitions the whole document collection into groups of clusters. In a good cluster, all the documents within a cluster are very similar, while the documents in other clusters are different. How to measure the similarity between documents is the core issue of document clustering [1]. Text similarity measurement is also widely used in many other fields. Text similarity measurement is considered to be one of the best ways to improve the efficiency of information retrieval. In image retrieval, a better precision can be obtained by the textual information around image [2]. In automatic evaluation of machine translation and text summarization, an accurate similarity measurement between the reference sample and the machine generated one may help to improve the performance of the machine translation and automatic abstract system [3]. Moreover, text similarity measurement also plays an important role in text classification, automatic discovery of similar papers and document copy detection [4].

Traditional text similarity measurements based on VSM (Vector Space Model) commonly use the bag-of-words model. Each document is represented as a vector and each member of the vector denotes the feature value of key word in the document. The feature value could be word frequency, relative word frequency or TF-IDF (Term Frequency-Inverse Document Frequency). Cosine similarity, Jaccard similarity and Manhattan distance are usually used to measure similarity between two vectors [5]. TF-IDF calculation is relatively simple, together with the high accuracy and recall. Therefore, TF-IDF becomes the most widely used weight calculation method, which is a combined measure of the importance of the word and how much information the word provides. Typically, the words which have the greater frequency in a document and occur very rarely in the document set are more related to the topic of the document. However, TF-IDF contains

only statistical information and cannot express the senses of words and semantic relation between words. In practical applications, it is not enough to consider only TF-IDF. For example, *pen* can have two senses: *a writing instrument* or *an enclosure where small children can play*. There are two documents regarding pen, which refer to writing instrument and enclosure respectively. The traditional text similarity measurements based on word frequency vector are likely to regard them as the similar documents. On the other hand, two documents concerning fruits may be considered the dissimilar ones because the word *apple* or *orange* occurs in two documents respectively. Furthermore, the dimensionality of a document vector is huge and the feature values of document vectors are usually sparse, i.e., a lot of feature values are zeros. Such high-dimensionality and sparsity have become serious obstacles to text similarity measurement [6].

Many research works on the similarity measurements between words have been carried out. The knowledge-based similarity measurements use the knowledge in specific fields and recognize the synonyms, semantic redundancies and textual entailments in the documents to calculate the text similarity [7]. The establishment of a knowledge base is a complex and ambitious project, and therefore a knowledge base is generally replaced by the comprehensive word dictionary [8], such as WordNet or HowNet, in the existing research works. Elias et al. [9, 10] organized all the words to form a semantic network and examined the edges of the network, node densities, node depths and link types between them to calculate the similarity between words. The similarity measurement between words can be expanded into paragraph similarity calculation, and then the paragraph similarity calculation can be further expanded into article similarity calculation. Piepaolo et al. [11] used the Lesk-based word sense disambiguation approach to improve the traditional text similarity measurements based on word frequency vector. F-measure of this approach had an improvement over the traditional one; however, the high dimensional feature space not only consumes the time of the clustering algorithm but also reduces the accuracy. Based on the clustering technology, Kamal et al. [12] proposed the similarity measurement between sentences and applied it to automatic text summarization. Zhang et al. [13] introduced ontology to recalculate and reorder the relevance between documents for the results returned by the search engine; however, this approach required the interactions with users in order to obtain more accurate results. Ma et al. [14] used WordNet to analyze the concepts of words, synonyms and hyponymy words and replaced the word frequency vector of vector space model with the new one where each word in the document was extended to synonyms or hyponymy word. However, there is a lack of similarity measurement definition between two documents in the approach. Bellegarda [15] proposed LSI (Latent Semantic Index) in statistical language modeling and employed SVD (Singular Value Decomposition) to reduce the dimensionality of the word-document matrix. However, this approach did not analyze the effect of the decomposed space dimension on text clustering. If the decomposed space dimension is too small, the word-document matrix is compressed too much to represent the original semantics. If the decomposed space dimension is too large, the effect of the dimensionality reduction is not ideal and a lot of noises retained. Besides, for too sparse corpus, LSI cannot provide a good reflection of its potential semantics. In summary, the main problems with the existing text similarity measurements are as follows. a) The polysemous problem and synonym problem are the main obstacles to the semantic analysis of text similarity measurements. The polysemous words have multiple possible senses, each of which is only appropriate in certain contexts. Ambiguity may result in the misunderstanding of a text. b) Although the LSI can be used to compress the feature space to reduce dimensionality of text representation model, there is no mature and effective selection method of singular values.

To address these problems, a text similarity measurement with semantic analysis is proposed in this paper. The contributions of this paper are as follows. a) The proposed word sense disambiguation based on unsupervised learning tags the most appropriate sense of an ambiguous word in the given contexts and the concept similarity computing based on WordNet measures the semantic similarity between words. b) The proposed approach selects the important senses that have greater TF-IDF values to construct the feature vector and determines the optimal proportion of the selected key senses, which can effectively reduce the dimensionality of text representation model and remove noises. c) Weight calculation takes account of the degree of semantic relevance between two feature vectors, TF-IDF value, the occurrence position of sense, and the semantic similarity between concepts, which is more complete and scientific. The experimental results on benchmark corpus demonstrate that the proposed approach promotes the evaluation metrics of F-measure and has robustness for synonyms replacement.

The rest of this paper is organized as follows. Related work is discussed in Section 2. The flow chart of our approach is shown in Section 3. The proposed word sense disambiguation based on unsupervised learning and text similarity measurement are described in Section 4 and Section 5. Experimental results and discussions are presented in Section 6. Finally, concluding remarks are given in Section 7.

2. **Related Work.** Word Sense Disambiguation (WSD) is defined as the selection of the intended sense of an ambiguous word from a known and finite set of possible meanings. This choice is based upon a probabilistic model that tells which member of the set of possible meanings is the most likely given context in which the ambiguous word occurs [16]. Corpus-based approaches are employed which make disambiguation decisions based on probabilistic models learned from large quantities of naturally occurring text. The approaches take advantage of the abundance of text available online and do not require deep understanding of the linguistic structure and the availability of rich sources of real-world knowledge.

These probabilistic models are learned via supervised and unsupervised approaches. Supervised approaches need to annotate manually the most appropriate sense of the examples to serve as training data. A generalized model from the set of examples is built and it is used to disambiguate instances of the ambiguous word found in test data. Unfortunately, sense-tagged text only exists in small quantities and is expensive to create. The bottleneck is addressed by developing unsupervised approaches that learn probabilistic models from raw untagged text. In unsupervised framework, the sense is treated as a latent or missing feature. The EM (Expectation Maximization) algorithm is usually used to estimate the parameters of a probabilistic model [17].

At the heart of the EM algorithm lies the Q-function. This is the expected value of the log of the likelihood fun for the complete data sample, $D = (Y, S)$, where Y is the observed data and S is the missing sense value:

$$Q\left(\Theta^{new}|\Theta^{old}\right) = E\left[\ln p(Y, S|\Theta^{new})|\Theta^{old}, Y\right] \tag{1}$$

Here, Θ^{old} is the previous value of the maximum likelihood estimates of the parameters and Θ^{new} is the improved estimate; $p(Y, S|\Theta^{new})$ is the likelihood of observing the complete data given the improved estimate of the model parameters. When approximating the maximum of the likelihood function, the EM algorithm starts from a randomly generated initial estimate of the model parameters and then replaces Θ^{old} by the Θ^{new} which maximizes $Q(\Theta^{new}|\Theta^{old})$. This is a two-step process, where the first step is known as the expectation step, i.e., the E-step, and the second is the maximization step, i.e., the M-step [18]. The E-step finds the expected values of the sufficient statistics of the

complete model using the current estimates of the model parameters. For decomposable models these sufficient statistics are the frequency counts of events defined by the marginal distributions of the model. The M-step makes maximum likelihood estimates of the model parameters using the sufficient statistics from the E-step. These steps iterate until the parameter estimates Θ^{old} and Θ^{new} converge.

The word sense disambiguation approach based on EM algorithm utilizes an iterative estimation procedure to classify an ambiguous word into one of several predetermined senses. However, the approach usually computes expensively and converges slowly because of the large number and random initialization of model parameters. In this paper, an improved approach is proposed, which makes use of mutual information theory based on Z-test to select features and uses a statistical learning algorithm to estimate initial parameter values.

3. **The Processing Flow of the Proposed Approach.** The flow chart of the proposed approach is shown in Figure 1.

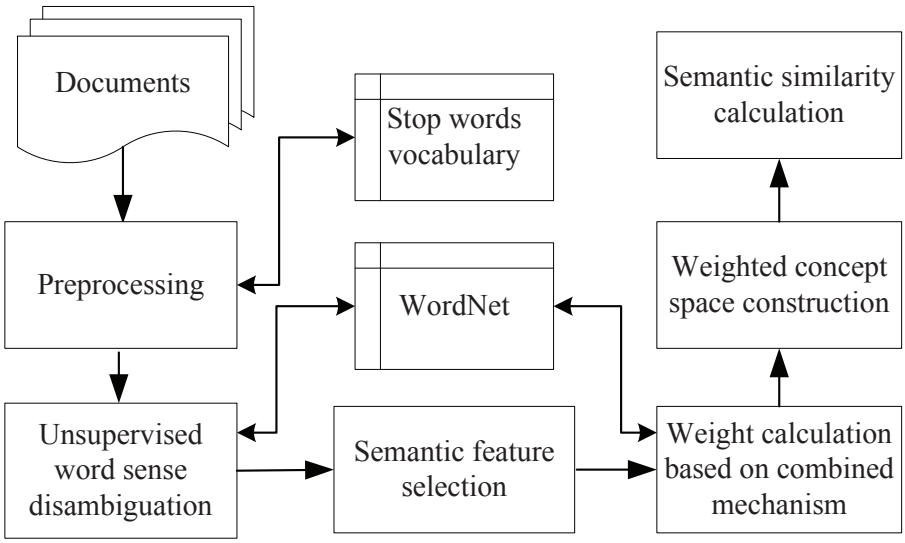

FIGURE 1. The flow chart of the proposed approach

Although original documents contain the most complete text information, natural language processing technology has not fully dealt with them. Preprocessing is necessary before converting document into vector. Preprocessing includes deleting stop words and function words. The semantics of a document is directly related to the senses of its contained words, and one can expect to capture more precise content of a document when the senses are used. In the proposed approach, the words in a preprocessed document are firstly mapped to senses by the improved word sense disambiguation based on unsupervised learning. Secondly the key senses are selected as features if the TF-IDF values of them are greater than the given threshold. The weight based on combined mechanism is calculated according to the importance of each feature to the topic of the document. The document can be represented as a vector with the weight of the feature as a member of the vector. Finally, the semantic similarity between the two documents is calculated according to the proposed similarity measurement definition.

4. Word Sense Disambiguation Based on Unsupervised Learning.

4.1. **The probabilistic model.** The probabilistic model can indicate which sense of an ambiguous word is most probable given the context in which it occurs. The probabilistic models consist of a parametric form and parameter estimates. The parametric form shows which contextual features affect the values of other contextual features as well as which contextual features affect the sense of the ambiguous word. The parameter estimates tell how likely certain combinations of values for the contextual features are to occur with a particular sense of an ambiguous word.

It is assumed the parametric form is Naive Bayes. Unsupervised learning requires that the variable value associated with the sense of an ambiguous word be treated as missing or unobserved data in the sample. The sense of an ambiguous word is represented by a feature variable S. The observed contextual features are represented by F_1, F_2, ..., F_i, ..., F_n, where F_i is the ith feature variable. Here is the assumption that all the features of an event are dependent. Bayesian estimate is the product of the prior probability, $p(S)$, and the conditional probability, $p(F_i|S)$. This product defines the posterior probability function, $p(S|F_1, F_2, \ldots, F_n)$, defined by Bayes Rule as:

$$p(S|F_1, F_2, \ldots, F_n) = \frac{p(F_1, F_2, \ldots, F_n, S)}{p(F_1, F_2, \ldots, F_n)} = \frac{p(S) \times \prod_{i=1}^{n} p(F_i|S)}{\sum_s p(F_1, F_2, \ldots, F_n, S)} \tag{2}$$

The parameters of the model are $p(S)$ and $p(F_i|S)$.

4.2. **Feature selection.** WSD acquires linguistic knowledge from the given context in which the ambiguous word occurs. However, not all the contexts contribute to the sense classifier [19]. In this paper, we make use of mutual information theory based on Z-test to select the senses that have contributions to the value of a classification variable as contextual features. In this way, the proposed model not only reduces the noises brought in the disambiguation model, but also decreases the amounts of computation. There is an assumption that contextual features are only defined within the boundaries of the sentence in which an ambiguous word occurs. In other words, only information that occurs in the same sentence is used to resolve the meaning of an ambiguous word.

Mutual information describes the relation between variables in the information theory. In the proposed model, an ambiguous word is represented by w and a contextual word is represented by w_j, where w_j is the jth contextual word. Their mutual information, $I(w, w_j)$, is defined to be

$$I(w, w_j) = \log_2 \frac{p(w, w_j)}{p(w)p(w_j)} = \log_2 \frac{Mf(w, w_j)}{f(w)f(w_j)} \tag{3}$$

Mutual information compares the joint probability of observing w and w_j together with the probabilities of observing w and w_j independently. If there is a genuine relation between w and w_j, the joint probability $p(w, w_j)$ will be much greater than the chance of $p(w)p(w_j)$, and consequently $I(w, w_j) \gg 0$. If there is no interesting relation between w and w_j, then $p(w, w_j) \approx p(w)p(w_j)$, and thus, $I(w, w_j) \approx 0$. If w and w_j are in complementary distribution, then $p(w, w_j)$ will be much less than $p(w)p(w_j)$, and forcing $I(w, w_j) \ll 0$. Word probabilities $p(w)$ and $p(w_j)$ are estimated by counting the number of observations of w and w_j in a corpus, $f(w)$ and $f(w_j)$, and normalizing by M, the size of the corpus. The joint probability, $p(w, w_j)$, is estimated by counting the number of times that w and w_j co-occur, $f(w, w_j)$, and normalizing by M.

The threshold should be different when the mutual information has a different distribution. In practice, it is difficult to achieve. To address the problem, Z-test is used to transform the distribution of mutual information into the standard normal distribution in

proposed model. In this way, we can use a uniform threshold to select features regardless of the distribution of mutual information.

The standard normal distribution is a normal distribution with $\mu = 0$ and $\sigma^2 = 1$. An arbitrary normal distribution can be converted into the standard normal distribution by taking $\mu = 0$ and expressing deviations from μ in standard deviation units. If $X \sim N(\mu, \sigma^2)$, then $Z = (X - \mu)/\sigma \sim N(0, 1)$. The mean, the variance and the value of Z are calculated as follows.

$$E = \frac{1}{n} \sum_{i=1}^{n} I(w, w_j) \tag{4}$$

$$D = \frac{1}{n} \sum_{i=1}^{n} (I(w, w_j) - E)^2 \tag{5}$$

$$Z = \frac{I(w, w_j) - E}{\sqrt{D}} \tag{6}$$

If the calculated value of Z is greater than the given threshold, the contextual word w_j will be selected as a feature. According to the 3σ theorem in the probability theory, we can know that the area included under the standard normal curve between $Z = -3$ and $Z = 3$ is 99.74%. Consequently, the threshold should be in the range of $(-3, +3)$. The larger the threshold, the more information the selected feature will contribute to ambiguous word sense classifier.

4.3. **Initial parameter estimate.** The EM algorithm cannot guarantee to find the global optimal solution. There is a closed relationship between the final solution and the initial estimate of the model parameters. The irrelevant initial estimate will lead to a poor solution. In this paper, a statistical learning algorithm is proposed to estimate the initial parameters of EM iteration.

Unsupervised learning is based upon raw or untagged text. No external knowledge sources are employed. It seems that the parameters of the model cannot be calculated from raw text. However, it is not truth. Maximum entropy principle in information theory tells us that an undetermined distribution should obey the uniform distribution. It is the assumption that the probability and the frequency of each sense of an ambiguous word appearing in the given context are all uniform. However, some words in the corpus have the same senses. These senses occur repeatedly and the occurring frequencies are different. Consequently, the status of the uniform distribution will be broken easily in the machine learning. The different statistical information will come forth. The sense distributions of an ambiguous word are not uniform in the final statistical results. The imbalance of distribution is a natural reflection of the real text. Based on the above opinion, we can estimate the distribution of a sense in raw corpus. The proposed statistical learning algorithm is as follows:

Input: The corpus.

Output: Estimated initial value of $p(S)$ and $p(F_i|S)$.

Step 1: The possible senses of an ambiguous word are defined by WordNet.

Step 2: A $(m + 1) \times (n + 2)$ table of word frequency is constructed, where m is the number of sense items and n is the number of contextual words. The number of the occurrence of sense item, the number of the co-occurrence of each item and its context occurring in the corpus are both calculated. The number of the occurrence of sense item s_i is represented by $f(s_i)$ and the number of times that s_k and its contextual word w_j co-occur is represented by $f(s_k, w_j)$. The data in the table is initialized to zero. The statistic table of word frequency is shown in Table 1.

TABLE 1. Statistic table

sense	$f(s_i)$	$f(s_i, w_1)$	...	$f(s_i, w_n)$
sense1	$f(s_1)$	$f(s_1, w_1)$	...	$f(s_1, w_n)$
sense2	$f(s_2)$	$f(s_2, w_1)$	...	$f(s_2, w_n)$
...	...	...	...	...
sense$k$	$f(s_k)$	$f(s_k, w_1)$	...	$f(s_k, w_n)$
...	...	...	...	...
sense$m$	$f(s_m)$	$f(s_m, w_1)$	...	$f(s_m, w_n)$

Step 3: If an ambiguous word has i sense items, the number of the occurrence of each sense item and the number of co-occurrence of it and its context will be added the number $1/i$.

Step 4: The initial values of the model are calculated by the following formulas.

$$p(s_i) = \frac{f(s_i)}{\sum_{i=1}^{m} f(s_i)} \tag{7}$$

$$p(w_j|s_i) = \frac{f(s_i, w_j)}{f(s_i)} \tag{8}$$

According to law of large numbers in probability theory, we can know that the estimated distributions obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.

The sense of a word is represented in sense no. formats. In WordNet, each synset has a unique offset in the database. This offset can be used as the unique ID for this sense. All the synonyms in this synset share the same offset. For example, the offset of the first sense of noun *course* is 00831838. The offset representation format is helpful for both the synonym problem and the polysemous problem.

4.4. **Parameter estimates of the probabilistic model.** EM algorithm is used to estimate the parameters of the probabilistic model in proposed approach, in which E-step and M-step processes iterate until the parameter estimates converge. In E-step process, the expected values of the sufficient statistics of the Naive Bayes model are computed. These are the frequency counts of marginal events of the form (F_i, S) and are notated $freq(F_i, S)$. Since S is unobserved, the values for it must be imputed before the marginal events can be counted. During the first iteration of the EM algorithm, values for S are imputed by the special parameter initialization following the proposed initial parameter estimates algorithm above. Thereafter, S is imputed with values that maximize the probability of observing a particular sense for an ambiguous word in a given context:

$$S = \overset{\text{argmax}}{S_x} \; p(S|f_1, f_2, \ldots, f_{n-1}, f_n) \tag{9}$$

From $p(a|b) = p(a, b)/p(b)$ and $p(a, b) = \Sigma_c p(a, b, c)$ it follows that:

$$S = \overset{\text{argmax}}{S_x} \frac{p(S) \times \prod_{i=1}^{n} p(f_i|S)}{\sum_s p(f_1, f_2, \ldots, f_n, S)} \tag{10}$$

This calculation determines the value of S to impute for combination of selected feature values. Given imputed values for S, the expected values of the marginal event counts, $freq(F_i, S)$, are determined directly from the data sample. These counts are the sufficient statistics for the Naive Bayes model.

In M-step process, the sufficient statistics from the E-step are used to re-estimate the model parameters. This new set of estimates is designated Θ^{new} while the previous set of parameter estimates is called Θ^{old}. The model parameters $p(S)$ and $p(F_i|S)$ are estimated as follows:

$$p(S) = \frac{freq(S)}{N} \tag{11}$$

$$p(F_i|S) = \frac{freq(F_i, S)}{freq(S)} \tag{12}$$

If the difference between the parameter estimates obtained in the previous and current iteration is less than some pre-specified threshold ε, i.e.,

$$\left\|\Theta^{old} - \Theta^{new}\right\| < \varepsilon \tag{13}$$

then the parameter estimates have converged and the EM algorithm stops. If the difference is greater than the threshold, Θ^{new} is renamed Θ^{old} and the EM algorithm continues.

The pre-specified threshold ε is usually a smaller positive number, i.e., 0.01 or 0.001. The smaller the threshold is, the more iterations the EM algorithm will run and the slower the algorithm converges.

5. Text Similarity Measurement.

5.1. **Key senses selection.** TF-IDF method is used to select key senses in a document in this paper, which is a combination of sense frequency and inverse document frequency here. Sense frequency is based on the assumption: the weight of a sense that occurs in a document is simply proportional to the sense frequency. Typically, the senses which have the greater frequency are more related to the topic of the document. Inverse document frequency is a measure of how much information the sense provides, which is incorporated which diminishes the weight of senses that occur very frequently in the document set and increases the weight of senses that occur rarely [20]. The formula for TF-IDF is given below:

$$tfidf(sno_{ik}) = tf(sno_{ik}) \times idf(sno_{ik}) = \frac{t_{ik}}{\sum_{k=1}^{m} t_{ik}} \times \log \frac{n}{|\{i : sno_k \in d_i\}|} \tag{14}$$

where

- t_{ik}: the number of times the sense sno_k occurs in document d_i;
- $\sum_{k=1}^{m} t_{ik}$: sum of the number of times each sense occurs in document d_i;
- n: total number of documents in the corpus;
- $|\{i : sno_k \in d_i\}|$: number of documents where the sense sno_k appears.

According to information theory, IDF actually is a cross-entropy of sense probability distribution in the certain conditions. If we select the key senses which have the greater TF-IDF values to construct the feature vector to represent a document, not only the content of a document can be characterized precisely but also the dimensions of text representation model can be reduced effectively. In proposed approach, the TF-IDF values of the senses in a document are sorted and the senses with high TF-IDF values are selected to construct feature vector.

5.2. **Concept similarity computing.** WordNet is a large lexical database of English, which was created in the cognitive science laboratory of Princeton Unversity. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synsets, each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The most frequently encoded relation among synsets is the hyponymy relation. Hyponymy relation is transitive: if an armchair is a kind of chair, and if a chair is a kind

of furniture, then an armchair is a kind of furniture [21]. Since the distance and depth between concepts are easily available and the two properties are better able to distinguish conceptual-semantic, we take advantage of the distance and depth between concepts to measure semantic similarity. Related definitions are as follows.

Definition 5.1. *Concept Hyponymy Graph (CHG). The number of concept hyponymy relations accounts for nearly eighty percent of the total number of all relations in WordNet. A graph is formed by the hyponymy relations between concepts, which is called CHG.* $CHG = (V, E, r)$*, where* V *denotes the set of concept nodes,* E *is a binary relation on the set* V*,* r *is the root node.*

Definition 5.2. *Conceptual path. A sequence of concept nodes* $P = (v_1, v_2, \ldots, v_n)$ *is given, where* P *is the conceptual path from node* v_1 *to* v_n *if and only if* $E(v_i, v_{i+1})$ $(0 < i < n)$ *exists.*

Definition 5.3. *Length of conceptual path. The conceptual path* $P = (v_1, v_2, \ldots, v_n)$ *from* v_1 *to* v_n *is given, then the length of* P *is* $n - 1$*. It is referred to as* $L_p = n - 1$*.*

Definition 5.4. *Depth of concept node. The concept node* v *and the shortest conceptual path* $P = (r, v_1, v_2, \ldots, v_n, v)$ *from node* v *to* r *are given, then the depth of* v *is defined as the length of* P*, denoted by* $D_v = L_p$*.*

Definition 5.5. *Distance between concept nodes. The concept nodes* v_1*,* v_2 *and the common ancestor node set* V *are given, and* P_{i1} *and* P_{i2} *are the shortest conceptual path of the concept node* v_i $(v_i \in V)$ *to* v_1 *and* v_2 *respectively and the length* $L_{p_{i1}}$ *and* $L_{p_{i2}}$ *are known, then the distance between concept nodes* v_1 *and* v_2 *is defined as* $L(v_1, v_2) = \min\limits_{0<i\leq|V|}(L_{p_{i1}} + L_{p_{i2}})$ *and the distance from the concept node to itself is defined as zero.* $L(v_1, v_2) = \min\limits_{0<i\leq|V|}(L_{pi1} + L_{pi2})$ *and* $L(v_2, v_1) = \min\limits_{0<i\leq|V|}(L_{pi1} + L_{pi2})$*, therefore* $L(v_1, v_2) = L(v_2, v_1)$*, that is the distance between concept nodes is symmetrical.*

Definition 5.6. *Depth of concept nodes. The concept nodes* v_1*,* v_2 *and the common ancestor node set* V *are given, and* P_{i1} *and* P_{i2} *are the shortest conceptual paths of the concept node* v_i $(v_i \in V)$ *to* v_1 *and* v_2 *respectively and the lengths* $L_{p_{i1}}$ *and* $L_{p_{i2}}$ *are known, then the depth of concept nodes is defined as* $D(v_1, v_2) = D_i|_{\min\limits_{0<i\leq|V|}}\left(L_{p_{i1}} + L_{p_{i2}}\right)$*, where* D_i *is the depth of the concept node* v_i*.* $D(v_1, v_2) = D_i|_{\min\limits_{0<i\leq|V|}}(L_{p_{i1}} + L_{p_{i2}})$ *and* $D(v_2, v_1) = D_i|_{\min\limits_{0<i\leq|V|}}(L_{p_{i1}} + L_{p_{i2}})$*, therefore* $D(v_1, v_2) = D(v_2, v_1)$*, that is the depth of concept nodes is symmetrical.*

The shorter the distance between concept nodes is, the greater semantic consistency between them exists. The bigger the depth between concept nodes is, the greater semantic consistency between them will be obtained. The semantic similarity between concepts is calculated by the hierarchical structure composed of super-subordinate relations in this paper. The concept similarity definition is as follows.

$$sim(v_1, v_2) = \frac{2D(v_1, v_2)}{2D(v_1, v_2) + L(v_1, v_2)} \tag{15}$$

where $D(v_1, v_2)$ is the depth of concept nodes and $L(v_1, v_2)$ is the distance between concept nodes.

The recognition ability of human being is limited. Yang and Powers [22] found that the upper limit of the distance between concept nodes which can effectively distinguish semantic similarity is 12. The maximum depth of concept node in WordNet is 16. We

define the value of $L(v_1, v_2)$ to be 12 if the value is greater than 12, and limit the range of $L(v_1, v_2)$ to be $LV = \{i|0 \leq i \leq 12, i \in Z\}$ and the range of $D(v_1, v_2)$ to be $DV = \{i|0 \leq i \leq 16, i \in Z\}$.

5.3. **Semantic similarity calculation.** The key senses represent the most important information of a document. Therefore, the text similarity can be described by the similarity between the vectors of key senses. Text similarity measurement is converted to vector similarity measurement. In the vector space model, the cosine of the angle between two feature vectors is usually used to measure the similarity between two documents. Let $\bar{v}_i = (a_{i1}, a_{i2}, \ldots, a_{ik}, \ldots, a_{in})$ and $\bar{v}_j = (a_{j1}, a_{j2}, \ldots, a_{jl}, \ldots, a_{jn})$ be the feature vectors of the ith document and the jth document respectively, where a_{ik} $(1 \leq k \leq n)$ denotes the feature value of sno_k in the ith document. In this paper, the formula for the feature value is as follows.

$$a_{ik} = tfidf_i(sno_{ik}) \times pwf(sno_{ik}) \times \max_{1 \leq l \leq n} sim(sno_{ik}, sno_{jl}) \tag{16}$$

where $tfidf(sno_{ik})$ denotes the TF-IDF of sno_{ik} in the ith document, $pwf(sno_{ik})$ denotes the position weighting factor of sno_{ik}, $\max_{1 \leq l \leq n} sim(sno_{ik}, sno_{jl})$ denotes the maximum semantic similarity with other concepts in the vector $\bar{v}_j$.

The position of a sense in the document should be considered in the text similarity measurement. The senses in the title are more important than the ones in the main body of document. Even in the main body, the senses at the beginning and end of documents are more important than the ones in the middle of documents. Therefore, the senses appearing in the title and important positions should be weighted to improve the accuracy of clustering. The position weighting factor of a sense is calculated as follows.

$$pwf(sno_{ik}) = 1 + \log_2(1 + n(sno_{ik})) \tag{17}$$

where $n(sno_{ik})$ denotes the total number of sno_{ik} occurring in the title, abstract, keywords, conclusions. The senses which occur more times in the above positions are more important for text clustering, and their position weighting factors are greater.

The semantic similarity between two vectors is calculated as follows.

$$VectSim\,(\bar{v}_i, \bar{v}_j) = \frac{\bar{v}_i \bar{v}_j}{\|\bar{v}_i\| \|\bar{v}_j\|} \tag{18}$$

If the senses which have greater semantic relevance with other ones are numerous and the TF-IDF values of these items are greater in the respective documents, it is implied that these senses contribute more to text clustering. In this paper, the vectors are weighted according to the proportion of the TF-IDF values of senses, which satisfy a given semantic relevance demand, in the sum of TF-IDF values throughout the text. The relevance weighting factor between vectors is defined as follows.

$$\begin{aligned} wf = 1 + \frac{1}{2} \times & \left(\frac{\sum\limits_{k \in \Lambda_i} tfidf(sno_{ik})}{\sum\limits_{k=1}^{m} tfidf(sno_{ik})} + \frac{\sum\limits_{l \in \Lambda_j} tfidf(sno_{jl})}{\sum\limits_{l=1}^{n} tfidf(sno_{jl})} \right) \\ & \times \left(\sqrt{VectSim\,(\bar{v}_i, \bar{v}_j)} - VectSim\,(\bar{v}_i, \bar{v}_j) \right) \end{aligned} \tag{19}$$

where $tfidf(sno_{ik})$ denotes the TF-IDF value of the sense sno_{ik}. $\sum_{k \in \Lambda_i} tfidf(sno_{ik}) / \sum_{k=1}^{m} tfidf(sno_{ik})$ shows that the percentage of the TF-IDF values of all the key senses, of which the semantic similarities between them and other concepts in the vector $\bar{v}_j$ exceed

the given threshold, in the sum of TF-IDF values of all the senses in the vector $\bar{v}_i$. The set Λ_i and Λ_j of the above expression are defined as follows.

$$\Lambda_i = \left\{ k : 1 \le k \le m, \max_{1 \le l \le n} \{sim(sno_{ik}, sno_{jl})\} \ge \sigma \right\} \tag{20}$$

$$\Lambda_j = \left\{ l : 1 \le l \le n, \max_{1 \le k \le m} \{sim(sno_{jl}, sno_{ik})\} \ge \sigma \right\} \tag{21}$$

If the semantic similarity between the sense sno_{ik} in the vector $\bar{v}_i$ and the sense sno_{jl} in the vector $\bar{v}_j$ is larger than the given threshold, the sense sno_{ik} is put into the set Λ_i. Similarity, each element contained in the set Λ_j is selected respectively.

The text similarity measurement between two documents is given as follows.

$$TextSim\left(\bar{v}_i, \bar{v}_j\right) = wf \times VectSim\left(\bar{v}_i, \bar{v}_j\right) \tag{22}$$

6. **Experiments.** Semcor 3.0, which is constructed by Princeton University according to WordNet, is used as the corpus of word sense disambiguation. One hundred and fifty tagfiles are selected as the training data and thirty-six tagfiles are selected as the test data from brown1 and brown2. Three approaches are compared their disambiguation accuracies in the experiment. The first one uses the traditional EM algorithm to estimate the values of the senses of ambiguous words. The second one adds the proposed feature selection before EM algorithm. The third one adopts the proposed model in this paper, which not only adds the feature selection, but also uses the statistical learning algorithm to estimate the initial parameters of the model before EM algorithm is employed. In order to discuss the relation between the accuracy of the classification and the size of the training corpus, the training corpora of different sizes are employed. The training corpora are A (50tagfiles), B (100tagfiles) and C (150tagfiles) respectively. The experimental results of the three approaches in the open test are shown in Table 2.

Table 2. The experimental accuracies

Approach	A-open	B-open	C-open
The first approach	.689	.701	.734
The second approach	.747	.763	.797
The third approach	.751	.769	.804

From Table 2 we can know that the third approach is the best in WSD, the second one is secondary, and the first one is the worst. The comparison between the second approach and the first one verifies the active effect of feature selection. By selecting features, the dimensionality of the problem and the jamming of useless features are reduced. The accuracy of WSD is obviously improved. The comparison between the third approach and the second one shows that the initial parameter estimation of the model has also positive effect on improving the accuracy of WSD. From the table we can also know that the accuracy of WSD will tend to become higher as the more training data is employed. According to the statistical results, the average number of feature variables is decreased from 9 to 5 because of feature selection. EM algorithm estimates the parameters of the model $p(S)$ and $p(F_i|S)$. Thus the computation of each iteration in proposed approach is reduced about 44.4%.

Reuters-21578 and 20Newsgroups corpora are used in the text similarity measurement experiment. There are significant differences in the size of text, the number of clustering and text distribution in these data sets. The format of the text in the Reuters-21578 is SGM rather than plain text, and thus the documents need be preprocessed. Three text

TABLE 3. Characteristics of the experimental data

Set	Total number of documents	Number of clusters	Minimum number of documents in a cluster	Maximum number of documents in a cluster	Average number of documents in a cluster
R1	100	8	9	16	13
R2	300	8	30	57	38
R3	500	8	51	78	63
N1	200	10	15	25	20
N2	500	10	40	60	50
N3	1000	10	80	120	100

subsets are selected in each data set, namely R1, R2, R3 from the Reuters-21578 and N1, N2, N3 from the 20Newsgroups. Each document in the data set is divided into one or more specific classes in advance. Table 3 shows the characteristics of each data subset.

K-Means (KM) and Bisecting K-Means (BKM) clustering algorithms provided in the CLUTO software package [23] are employed. F-measure is evaluated, which considers both the precision and the recall of the test to compute the score. Given a set of labeled documents belonging to i classes, we assume the clustering algorithm to partition them into j clusters. The number of the cases in class i is represented by n_i and the number of cases in cluster j is represented by n_j. The number of the cases that are both in cluster j and class i is represented by n_{ij}. The precision, recall, and F-measure are defined as follows.

$$P(i,j) = \frac{n_{ij}}{n_j} \tag{23}$$

$$R(i,j) = \frac{n_{ij}}{n_i} \tag{24}$$

$$F(i,j) = \frac{2 \times P(i,j) \times R(i,j)}{P(i,j) + R(i,j)} \tag{25}$$

where P is the precision of cluster j and R is the recall of cluster j.

The F-measure of all of the clusters is the sum of the F-measures of each class weighted by its size:

$$F = \sum_i \frac{n_i}{n} \max_j F(i,j) \tag{26}$$

An F-measure reaches its best value at 1 and worst at 0.

The optimal proportion of the selected key senses need be determined in the experiment. BKM algorithm is used for clustering and the concept similarity threshold σ is set to zero, that is the concept similarity measurement is not used to weight and all the senses are equally important. Figure 2 shows the F-measures of the selected key senses of different proportions. Experimental results demonstrate that the clustering is best when the top 30 percentage of senses are selected as the key ones. When the percentage is less than 30, the number of the selected key senses is too small to the lack of extracted text feature information. When the percentage is more than 30, the extracted text feature information is sufficient; however, many of senses have very little contribution to the text clustering. The more selected key senses are, the more noises are kept. Consequently, the clustering result is not satisfactory when the number of selected key senses is continuously increased.

The clustering results obtained by the different concept similarity thresholds are shown in Figure 3. F-measures are gradually improved except in the N1 data set when the

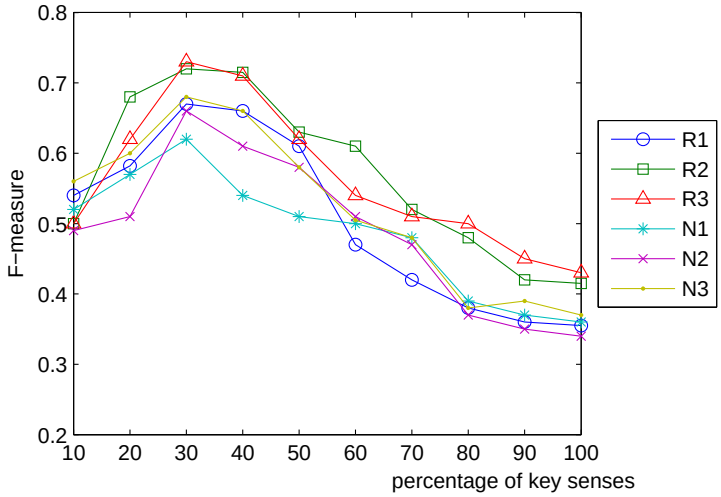

FIGURE 2. The clustering results of key senses of different percentages

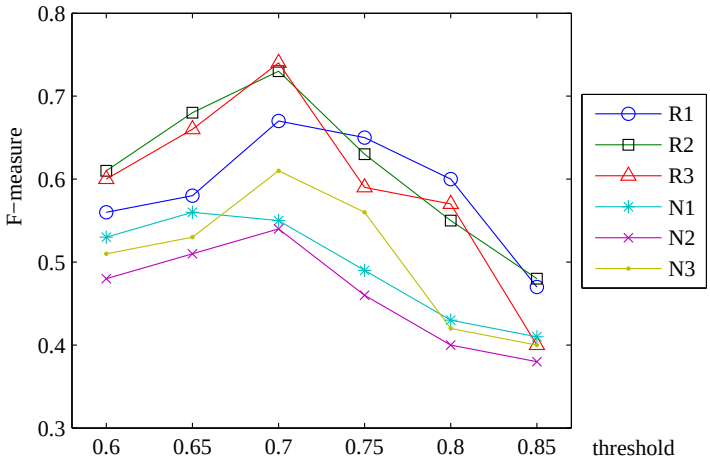

FIGURE 3. The clustering results of the different concept similarity thresholds

concept similarity threshold σ is less than or equal to 0.7 and it is increased. The reason is that the semantic discrimination between documents becomes larger when the concept similarity threshold is increased. The clustering effect is getting better. F-measure of the clustering reaches a peak when the concept similarity threshold is equal to 0.7. However, when the similarity threshold is more than 0.7 and further improved, F-measure declines rapidly. The reason is that the proportion of senses, of which the maximum semantic similarity with other concepts exceeds 0.7 in the data set, is small. The guidance effect of the relevance weighting factor between vectors is weakened, and the overall F-measure is reduced.

The proposed approach is compared with the traditional TF-IDF method and the WordSim method presented in paper [13]. The top 30 percentage of senses are selected as the key ones and the concept similarity threshold σ is set to 0.7 in the experiments. The WordSim method uses WordNet to integrate background knowledge into the text representation. The clustering results of the three approaches using KM and BKM algorithms are shown in Figure 4 and Figure 5. As shown, our method performs better than others. The results demonstrate that word sense disambiguation based on unsupervised learning

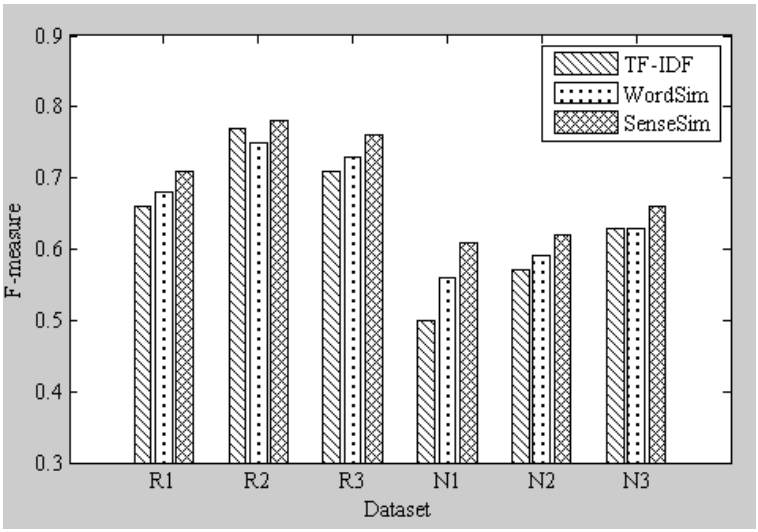

FIGURE 4. The clustering results of the different approaches using KM algorithm

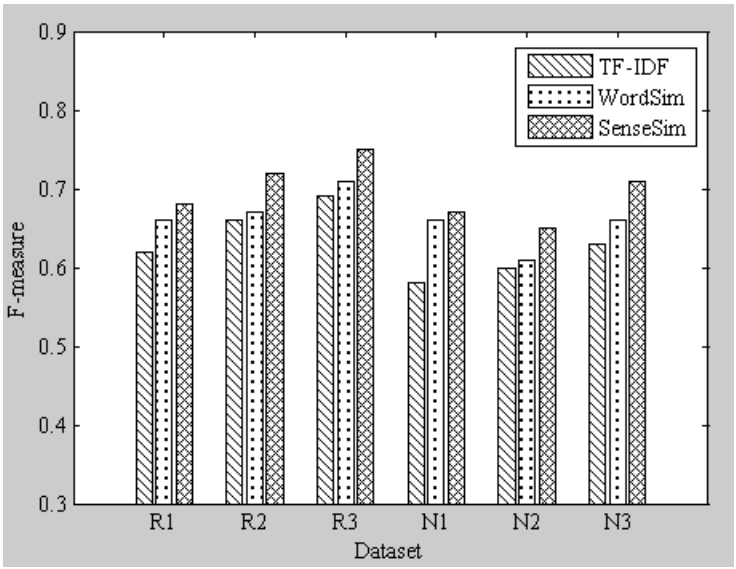

FIGURE 5. The clustering results of the different approaches using BKM algorithm

and the proposed semantic similarity definitions effectively promote F-measure of text similarity measurement.

7. **Conclusions.** We have presented a text similarity measurement with semantic analysis. The proposed approach has the following characteristics.

- Word sense disambiguation distinguishes effectively the senses of words in different contexts. In addition, the proposed approach obtains automatically the knowledge of word from raw text, which avoids the time-consuming sense annotation and data sparsity of supervised learning.
- The selection of key senses reduces the dimensionality of text representation model and removes noises, which improves the efficiency of text clustering.
- By computing the concept similarity and weighting the vector of key senses, the proposed approach improves the importance of those key senses that have both

larger semantic relevances with other concepts and greater TF-IDF values in text similarity calculation, which has positive effect on text clustering.

Experimental results on benchmark corpus have shown that the proposed approach promotes the evaluation metrics of F-measure. The proposed approach can be widely applied in the fields of machine translation, document summarization, document classification and document copy detection.

In future work, we will introduce deep neural network to extract the latent features which are difficult to artificially constructed. Deep learning uses multi-layer neural networks and requires complex computation. With the rapid development of machine learning theory and computer hardware, we will use the advanced hardware architecture such as GPU (Graphic Processing Unit) or TPU (Tensor Processing Unit) computing to reduce computational complexity.

Acknowledgment. This work is partially supported by National Natural Science Foundation of China (No. 61402069) and Scientific Research Fund of Liaoning Provincial Education Department (No. L2015047). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Y.-C. Tseng, C.-W. Yang and B.-C. Kuo, Using SVM to combine Bayesian networks for educational test data classification, *International Journal of Innovative Computing, Information and Control*, vol.12, no.5, pp.1679-1690, 2016.

[2] A. Noha and I. Ali, An efficient fast-response content-based image retrieval framework for big data, *Computers & Electrical Engineering*, vol.54, no.1, pp.522-538, 2016.

[3] B. Mehrez and J. Mohamed, Learning sign language machine translation based on elastic net regularization and latent semantic analysis, *Artificial Intelligence Review*, vol.46, no.2, pp.145-166, 2016.

[4] X. Peng, S. Liu, Z. Liu, W. Gan and J. Sun, Mining learners' topic interests in course reviews based on like-LDA model, *International Journal of Innovative Computing, Information and Control*, vol.12, no.6, pp.2099-2110, 2016.

[5] C. Hiram and M. Oscar, Integrated concept blending with vector space models, *Computer Speech & Language*, vol.40, pp.79-96, 2016.

[6] M. Andri, Clustering and latent semantic indexing aspects of the singular value decomposition, *International Journal of Information and Decision Sciences*, vol.8, no.1, pp.53-72, 2016.

[7] S. Fraihat, Ontology-concepts weighting for enhanced semantic classification of documents, *International Journal of Innovative Computing, Information and Control*, vol.12, no.2, pp.519-531, 2016.

[8] J. Sergio, Effectively combining paraphrase database, string matching, wordnet, and word embedding for semantic textual similarity, *Proc. of the 10th International Workshop on Semantic Evaluation*, San Diego, USA, pp.749-757, 2016.

[9] L. Elias and P. Alexandros, Similarity computation using semantic networks created from web-harvested data, *Natural Language Engineering*, vol.21, no.1, pp.49-79, 2015.

[10] P. Li, H. Wang and K. Zhu, A large probabilistic semantic network based approach to compute term similarity, *IEEE Trans. Knowledge and Data Engineering*, vol.27, no.10, pp.2604-2617, 2015.

[11] B. Piepaolo, C. Annalina and S. Giovanni, An enhanced lesk word sense disambiguation algorithm through a distributional semantic model, *Proc. of the International Conference on Computational Linguistics*, Dublin, Ireland, pp.1591-1600, 2014.

[12] S. Kamal, S. Khushbu and G. Avishikta, Improving graph based multidocument text summarization using an enhanced sentences similarity measure, *Proc. of the 2nd IEEE International Conference on Recent Trends in Information Systems*, Kolkata, India, pp.359-365, 2015.

[13] R. Zhang, S. Xiong and Z. Chen, An ontology-based approach for measuring semantic similarity between words, *Proc. of the 11th International Conference on Intelligent Computing*, Fuzhou, China, pp.510-516, 2015.

[14] Y. Ma, J. Liu and Z. Yu, Concept name similarity calculation based on wordnet and ontology, *Journal of Software*, vol.8, no.3, pp.746-753, 2013.

[15] R. Bellegarda, Exploiting latent semantic information in statistical language modeling, *Proc. of the IEEE*, vol.8, pp.1279-1296, 2000.

[16] W. Zhu, Semi-supervised word sense disambiguation using von Neumann kernel, *International Journal of Innovative Computing, Information and Control*, vol.13, no.2, pp.695-701, 2017.
[17] M. Gonzalez, C. Minuesa and D. Puerto, Maximum likelihood estimation and expectation-maximization algorithm for controlled branching processes, *Computational Statistics & Data Analysis*, vol.93, pp.209-227, 2016.
[18] M. Mitesh, J. Salil and B. Pushpak, A bilingual unsupervised approach for estimating sense distributions using expectation maximization, *Proc. of the 15th Internaitonal Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, pp.695-704, 2011.
[19] K. Daisuke and P. Martha, Single classifier approach for verb sense disambiguation based on generalized features, *Proc. of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland, pp.4210-4213, 2014.
[20] A. Shereen, F. Sebastien and E. Bernard, An effective TF/IDF-based text-to-text semantic similarity measure for text classification, *Proc. of the 15th International Conference on Web Information Systems Engineering*, Thessaloniki, Greece, pp.105-114, 2014.
[21] J. Liu, L. Qin and J. Gao, A similarity-based concepts mapping method between ontologies, *IEICE Trans. Information & Systems*, vol.98, no.5, pp.1062-1072, 2015.
[22] D. Yang and D. Powers, Measuring semantic similarity in the taxonomy of wordnet, *Proc. of the 28th Australasian Computer Science Conference*, Newcastle, NSW, Australia, pp.315-322, 2005.
[23] G. Karypis, *Cluto – A Clustering Tookit*, Technical Report, vol.4, no.2, pp.163-165, 2003.