Computing Semantic Relatedness Using a Word-Text Mutual Guidance Model

Bingquan Liu[1], Jian Feng[1], Ming Liu[1], Feng Liu[1], Xiaolong Wang[1], and Peng Li[2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, 150001
[2] School of Software, Harbin University of Science and Technology, Harbin, China, 150080

Abstract. The computation of relatedness between two fragments of text or two words is a challenging task in many fields. In this study, we propose a novel method for measuring semantic relatedness between word units and between text units using an iterative process, which we refer to as the word-text mutual guidance (WTMG) method. WTMG combines the surface and contextual information when computing word or text relatedness. The iterative process can start in two different ways: calculating relatedness between texts using the initial relatedness of the words, or computing the relatedness between words using the initial relatedness of the texts. This method obtains the final relatedness result after the iterative process reaches convergence. We compared WTMG with previous relatedness computation methods, which showed that obvious improvements were obtained in terms of the correlation with human judgments.

Keywords: Semantic Relatedness, Mutual Guidance, Iterative Process, Initialization.

1 Introduction

The computation of semantic relatedness requires the estimation of the degree of association between two text fragments, which can be words, sentences, or documents (texts). For example, we may want to determine how two words are semantically related, such as *dog* and *cat*, or two pieces of text, such as *preparing a manuscript* and *writing an article*. Semantic relatedness measures have been applied in many natural language processing tasks such as information retrieval [4] and question answering [10].

Making judgments about the relatedness of different units is a common but complex task, which requires the surface meaning of the units and contextual knowledge about where they appear. Thus, we need to consider statistical information and the semantic information related to the words or texts. In this study, we introduce a new semantic relatedness computation model called the word-text mutual guidance model (WTMG). The mutual guidance between concept

C. Zong et al. (Eds.): NLPCC 2014, CCIS 496, pp. 67–78, 2014.

A and concept B is defined as a process where A can be derived from B and B can be derived from A. In our model, we compute the relatedness among words using the text relatedness and the relatedness among texts can be calculated based on the word relatedness. We propose an iterative process that computes the relatedness among words and texts. Our model considers the semantic information obtained from a hierarchical lexical database such as WordNet and the statistical information contained in the corpus involved. The proposed method comprises two main steps. First, we establish the initial word relatedness or text relatedness and we construct a relatedness matrix. Second, the word relatedness and text relatedness are calculated iteratively.

The main contributions of this study are as follows. First, to exploit the information associated with words and texts, we propose the WTMG model to make full use of the internal relationships between words and texts. Second, we performed comparisons of many word and text semantic relatedness initialization methods. The remainder of this paper is organized as follows. We provide an overview of related work on word and text relatedness in Section 2. The WTMG model is described in detail in Section 3. Section 4 presents the experimental results and our conclusions are given in Section 5.

2 Related Work

Many methods have been proposed for semantic relatedness computation but they can generally be grouped into two categories: knowledge-based and corpus-based methods.

Knowledge-based methods, such as those of L&C [7], Wu&Palmer [13], Resnik [11], J&C [5], and Lin [8], employ information extracted from manually constructed lexical taxonomies, e.g., WordNet [1]. Previous studies have focused on developing appropriate measures while using WordNet as the primary knowledge source and they obtain relatively good results compared with corpus-based methods.

Corpus-based measures, such as LSA [6], ESA [2], SSA[3], employ probabilistic approaches to compute the semantic relatedness among words and texts. Most of these corpus-based measures map words or texts to the corresponding article in Wikipedia, which has emerged as a promising conceptual network for semantic relatedness computing in recent years.

However, these knowledge-based or corpus-based methods only consider semantic or statistical information, thus they ignore the fact that there must be relationships between a text and its component words. We propose the WTMG model to mine deeper relationship between words and texts. A similar approach was reported by [12], who aimed to calculate the short text similarity, but Wenyin's model has two drawbacks. First, it computes the initial word similarity by reconstructing WordNet, which is time consuming and unstable. Second, the model only selects the word relatedness to begin the iteration process and it ignores the initial text relatedness, which is also an important factor. Our model uses the most typical word relatedness computation method, which is available

directly with WordNet, to initialize the word relatedness matrix. Next, the initial text relatedness matrix can be calculated based on the initial word relatedness matrix, and the word and text relatedness are then calculated using an iterative algorithm.

3 WTMG Model

The proposed method comprises the following steps. Given a set of raw texts, the model first computes the initial relatedness between words using a knowledge-based method and the initial word relatedness matrix is then constructed. The text relatedness is computed based on the word relatedness matrix, and the word relatedness and text relatedness are then calculated iteratively until convergence. There is an alternative method for starting the iteration process, where we can calculate the initial text relatedness first and the word relatedness can then be calculated based on the initial text relatedness, but the two alternative methods both rely on an iterative process. Sections 3.1 and 3.2 introduce the typical initial word and text relatedness computation methods, respectively.

3.1 Word Relatedness Initialization

The measures used for computing the semantic relatedness belong to two categories.

Path-Based Measures. These measures compute the word relatedness as a function of the number of edges in the taxonomy along the path between two conceptual nodes c_1 and c_2 onto which the words w_1 and w_2 are mapped. The simplest path-based measure is the basic edge counting method, which defines the semantic distance as the number of nodes in the taxonomy along the shortest path between two concepts. The semantic relatedness is defined based on the semantic distance.

Information Content-Based Measures. [11] defined a criterion of similarity between two concepts as the extent to which they share common information. The information content is defined as $IC(c) = -logP(c)$, where $P(c)$ is the probability that a randomly selected word in a corpus is an instance of concept c. Semantic relatedness between concepts are then calculated based on the information content.

In our study, we calculated the initial word relatedness with both path-based measures and information content-based measures to compare the performance of these approaches. Meanwhile, we need to ensure that the relatedness is computed between words with the same parts of speech. This is because most word-to-word knowledge-based measures cannot be applied across parts of speech, thus we added this restriction to all of the word-to-word relatedness measures.

3.2 Text Relatedness Initialization

An alternative method for initializing our WTMG model is calculating the text relatedness first. The typical approach finds the relatedness between two text

segments using the vector space model or latent semantic analysis. Although these methods are successful to some degree, these corpus-based relatedness methods cannot always identify the semantic relatedness of texts. For example, there is an obvious relatedness between the two text segments *I own a dog* and *I have an animal*, but most current text relatedness metrics fail with relatively short texts.

[9] proposed a method for measuring the semantic relatedness of texts by exploiting the information that can be extracted from the relatedness of their component words. The relatedness of two texts t_1 and t_2 is defined in Eq. (1).

$$sim(t_1, t_2) = \frac{1}{2}\left[\frac{\sum_{w\in\{t_1\}}(max(w, t_2) * idf(w))}{\sum_{w\in\{t_1\}} idf(w)} + \frac{\sum_{w\in\{t_2\}}(max(w, t_1) * idf(w))}{\sum_{w\in\{t_2\}} idf(w)}\right] \quad (1)$$

where $max(w, t)$ represents the maximum relatedness between w and component words of t. This method focuses on measuring the semantic relatedness of short texts by exploiting the deep relationships between words and texts, and combining the word relatedness to obtain the text relatedness, *COMB* in Table 2 shows the text relatedness results obtained using the word relatedness.

3.3 Iterative Procedure

We calculate the relatedness between words and texts in Sections 3.1 and 3.2, respectively, where both the word relatedness and text relatedness were calculated independently. In most cases, however, there are relationship among texts and words. Thus, if we want to compute the relatedness between *text1* and *text2*, the words in *text1* and *text2* can affect the relatedness between them, which is similar to Eq. (1). Normally, two texts may be similar if they share more co-occurring words. In addition, each word has synonyms, thus if two texts include synonymous information, they should be similar. Similarly, two words may share a common or similar concept if they co-occur in many texts or they appear in similar texts.

In general, the most straightforward method for calculating relatedness between two texts (e.g., t_p and t_q) is to use the text vector derived from the word-text matrix, which is defined by Eq. (2):

$$R(t_p, t_q) = sim(V(t_p), V(t_q)) \quad (2)$$

where $V(t_p)$ and $V(t_q)$ denote two text vectors formed by words.

Equation (2) is based on vector representation and many measurements are required to perform this calculation. Thus, we use *Cosine* as an example and Eq. (2) changes into Eq. (3):

$$R(t_p, t_q) = \sum_{k=1}^{N}(tf_{pk}) * (tf_{qk}) \quad (3)$$

where tf_{ij} is the frequency of word w_j in the ith text and N is the dimensionality of the feature space, or the total number of words that appear in the corpus.

In fact, texts are composed of words, thus if two texts share more topics with similar words, these two texts are relevant. This idea is useful when calculating the text relatedness based on the word relatedness. Thus, we expand the text relatedness to Eq. (4).

$$R(t_p, t_q) = \sum_{k=1}^{N} (tf'_{pk}) * (tf'_{qk}) \tag{4}$$

tf'_{pk} and tf'_{qk} can then be calculated using Eq.(5):

$$tf'_{pk} = \sum_{j=1}^{N} (tf_{pj} P_{jk}) \quad tf'_{qk} = \sum_{j=1}^{N} (tf_{qj} P_{jk}) \tag{5}$$

where P_{jk} indicates the normalized word relatedness, which can be calculated using Eq. (6).

$$P_{jk} = \frac{sim(w_j, w_k)}{\sqrt{\sum_{l=1}^{N} sim(w_j, w_l)^2}} \tag{6}$$

By incorporating guidance based on the word relatedness, Eq. (4) changes into Eq. (7).

$$R(t_p, t_q) = \sum_{k=1}^{N} \left[\left(\sum_{j=1}^{N} tf_{pj} P_{jk} \right) \left(\sum_{j=1}^{N} tf_{qj} P_{jk} \right) \right] \tag{7}$$

Similarly, the word relatedness can also be calculated by the vector represented by texts, which is defined in Eq. (8):

$$sim(w_p, w_q) = sim(V(w_p), V(w_q)) \tag{8}$$

where w_p and w_q denote two word vectors formed by texts.

We use *Cosine* as an example to compute the relatedness and Eq. (8) changes into Eq. (9):

$$sim(w_p, w_q) = \sum_{k=1}^{M} (tf_{kp}) * (tf_{kq}) \tag{9}$$

where M is the number of texts in the corpus. Two words may be more similar if they co-occur in many texts or they appear in similar texts. Based on this fact, Eq. (9) changes into Eq. (10):

$$sim(w_p, w_q) = \sum_{k=1}^{M} (tf'_{kp}) * (tf'_{kq}) \tag{10}$$

where tf'_{kp} and tf'_{kq} are defined by Eq. (11):

$$tf'_{kp} = \sum_{i=1}^{M} (tf_{ip} Q_{ik}) \quad tf'_{kq} = \sum_{i=1}^{M} (tf_{iq} Q_{ik}) \tag{11}$$

where Q_{ik} indicates the normalized text relatedness, which can be calculated using Eq. (12).

$$Q_{ik} = \frac{sim(t_i, t_k)}{\sqrt{\sum_{l=1}^{M} sim(t_i, t_l)^2}} \tag{12}$$

By incorporating guidance based on the text relatedness, the word relatedness can be computed using Eq. (13).

$$sim(w_p, w_q) = \sum_{k=1}^{M} \left[\left(\sum_{i=1}^{M} tf_{ip} Q_{ik} \right) \left(\sum_{i=1}^{M} tf_{iq} Q_{ik} \right) \right] \tag{13}$$

It is obvious that P_{jk} is derived from the relatedness between words and that Q_{ik} is derived from the relatedness between texts. We can see that the definitions of the relatedness between words and the relatedness between texts are cyclic. Thus, the relatedness between words and the relatedness between texts can be calculated using an iterative algorithm. Figure 1 shows the iterative process employed with WTMG. Two operations are repeated alternately, where one operation uses the word similarity to guide the text relatedness calculation and the other is the opposite. It is clear that the process can operate in two possible ways, which start from two initial points. The dotted line starts from $Sim^{(0)}(w_p, w_q)$ and the real line starts from $R^{(0)}(t_p, t_q)$. Thus, the process only requires the setting of one parameter: $Sim^{(0)}(w_p, w_q)$ or $R^{(0)}(t_p, t_q)$.

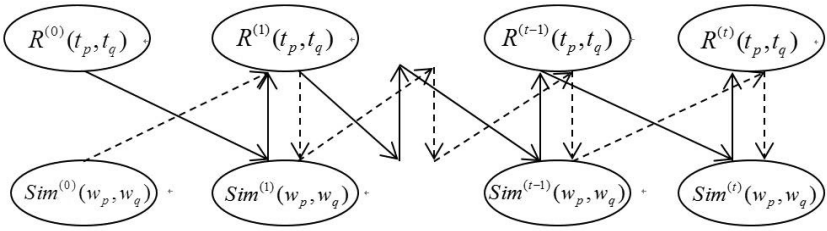

Fig. 1. Iterative Procedure of WTMG

The iterative process can be defined using Eqs. (14) and (15).

$$sim^{(t+1)}(w_p, w_q) = \sum_{k=1}^{M} \left[\left(\sum_{i=1}^{M} tf_{ip} Q_{ik}^{(t)} \right) \left(\sum_{i=1}^{M} tf_{iq} Q_{ik}^{(t)} \right) \right] \tag{14}$$

$$R^{(t+1)}(t_p, t_q) = \sum_{k=1}^{N} \left[\left(\sum_{j=1}^{N} tf_{pj} P_{jk}^{(t+1)} \right) \left(\sum_{j=1}^{N} tf_{qj} P_{jk}^{(t+1)} \right) \right] \tag{15}$$

Equations (14) and (15) show that the process begins with $R^{(0)}(t_p, t_q)$, and the process can also start in another way.

To verify the effectiveness of WTMG, we selected five similar text pairs (designated as S1 to S5) and five dissimilar text pairs (designated as U1 to U5). We applied our WTMG model to this small corpus example and the results are shown in Figures 1 and 2. Figure 1 shows the text relatedness results obtained after starting from $R^{(0)}(t_p, t_q)$ with WTMG, whereas Figure 2 shows the text relatedness results obtained after starting from $Sim^{(0)}(w_p, w_q)$ with WTMG.

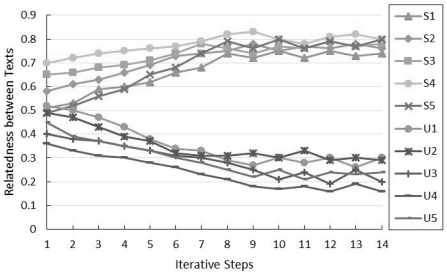

Fig. 2. Initialize Text Relatedness

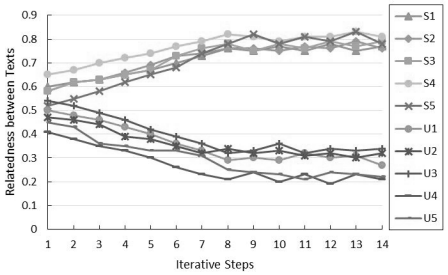

Fig. 3. Initialize Word Relatedness

Figures 2 and 3 demonstrate that our model obtained reasonable results, where it allowed the relatedness between similar texts to be greater, and the relatedness between dissimilar texts to be smaller. The results calculated by starting from the initial word relatedness or initial text relatedness were also similar. However, this "convergence" does not mean that the process converge to specific values and it simply refers to a balanced condition, where the values fluctuated in a small range repeatedly. To make the iterative process converge, we add a damping factor λ and λ is set by a kernel function that regresses as time passes. By minimizing λ, Eqs. (14) and (15) can finally converge. After adding the damping factor λ, the iterative equations change into Eqs. (17) and (16).

$$sim^{(t+1)}(w_p, w_q) = (1-\lambda) sim^{(t)}(w_p, w_q) + \lambda \sum_{k=1}^{M} \left[\left(\sum_{i=1}^{M} tf_{ip} Q_{ik}^{(t)} \right) \left(\sum_{j=1}^{M} tf_{iq} Q_{ik}^{(t)} \right) \right] \tag{16}$$

$$R^{(t+1)}(t_p, t_q) = (1-\lambda) R^{(t)}(t_p, t_q) + \lambda \sum_{k=1}^{N} \left[\left(\sum_{j=1}^{N} tf_{pj} P_{jk}^{(t+1)} \right) \left(\sum_{j=1}^{N} tf_{qj} P_{jk}^{(t+1)} \right) \right] \tag{17}$$

λ can be set between 0 and 1, and t represents the tth iteration. Theoretically, λ can be different in Eqs. (16) and (17), but we used the same value of λ in our experiments for simplicity. Theoretically, the convergence of relatedness cannot be guaranteed, thus we decrease λ by 20% during each iteration in practice.

3.4 Time Complexity Analysis

After matrix Q and P have been calculated, we can calculate the relatedness. the process that Eq. (7). and Eq. (13) show can be written as:

$$R_t = TQQ'T' \tag{18}$$

$$S_w = T'PP'T \tag{19}$$

where T is the matrix of terms in text, Q and P are regularizations of S_w and R_t. T multiply Q whose time complexity is $O(kn^2)$, so the time complexity of Eq.(18) is $O(kn^2 + k^2n)$, similarly Eq.(19) is $O(k^2n + kn^2)$. Suppose after t step, the model has converged, so the time complexity of calculating P is $O(tkn^2 + tk^2n)$, similarly Q is $O(tk^2n + tkn^2)$. Matrix T, Q, P are sparse matrix, and the size of nonzero number were written as L_t, L_q, L_p. As the complexity of multiplying between matrix and vector is linear, the complexity of TQ can be declined as $O(kL_q)$ and the complexity of Eq.(18) can be written as $O(kL_q + kL_t)$, similarly Eq.(19) as $O(nL_p + nL_t)$. So the time complexity for calculating matrix P and Q are $O(tkL_q + tkL_t)$ and $O(tnL_p + tnL_t)$. Thus the time complexity of the algorithm is $O(tkL_q + tkL_t + tnL_p + tnL_t)$

4 Experiments

In our experiments, parameter λ was set to 0.5 and our model was applied using two alternative methods. First, we started our model based on the initial word relatedness, and the text relatedness and word relatedness were calculated iteratively until convergence, which we denote by WTMGW. Second, we initialized the text relatedness first, and the word relatedness and text relatedness were computed iteratively until convergence, which we denote by WTMGT.

The word relatedness measures use WordNet as a resource and the results shown in Table 1 indicate that the *Lin* measure performed the best among all the path-based and information content-based measures. Thus, we used the *Lin* method to calculate the word-to-word relatedness in *COMB*. In our WTMG model, we used the *Lin* measure to initialize the word relatedness and the text relatedness was initialized with the *COMB* measure, i.e., *WTMGW* and *WTMGT* respectively.

We used several standard word-to-word and text-to-text datasets to evaluate the representation strength of our mutual guidance semantic relatedness model. Correct correlations are typically used to evaluate the semantic relatedness, thus we used *Pearson's correlation coefficient* γ and *Spearman's rank correlation coefficient* ρ in our study, both of which are important for semantic relatedness evaluations.

4.1 Word Relatedness

To evaluate the effectiveness of the WTMG model in determining the word-to-word relatedness, we employed three standard datasets that have been used widely in previous studies.

Rubenstein and Goodenough(**RG65**) comprises 65 word pairs that range from synonymy pairs (e.g.,*car-automobile*) to completely unrelated terms (e.g., *noon-string*). The 65 noun pairs were annotated by 51 human subjects. All of the noun pairs are non-technical words and they are scored using a scale from 0 (not related) to 4 (perfect synonymy).

Miller-Charles(**MC30**) is a subset of the Rubenstein and Goodenough dataset that comprises 30 word pairs. The relatedness of each word pair was rated by 38 human subjects using a scale from 0 to 4.

WordSimilarity-353(**WS353**) is known as Finkelstein-353 and it comprises 353 word pairs annotated by 13 human experts using a scale from 0 (unrelated) to 10 (very closely related or identical). The Miller-Charles set is a subset of the WordSimilarity-353 dataset. Unlike the Miller-Charles dataset, which only contains single generic words, the WordSimilarity-353 set also includes phrases (e.g., "Wednesday news"), proper names, and technical terms, thus it presents an additional degree of difficulty for any relatedness metric.

Mturk-771(**MT771**) comprises 771 English word pairs and with their mean relatedness scores. The scores were collected using the Amazon Mechanical Turk and at least 20 ratings were collected for each word pair, where each judgment task comprised a batch of 50 word pairs. The ratings were collected on a scale of 1 to 5, where 5 denotes highly related and 1 denotes not related. The relatedness value of each word pair was the mean score given by the users.

We used the *Reuters News*[1] dataset when calculating the word relatedness, which is available on the Web. It contains *10,788* documents and approximately *130* million words. We used this large dataset to calculate the word relatedness and text relatedness iteratively because we needed a corpus that would include all the word pairs found in the standard datasets described above.

Table 1. Pearson and Spearman results for the word relatedness datasets

Method		Pearson(γ)				Spearman(ρ)			
		MC30	RG65	WS353	MT771	MC30	RG65	WS353	MT771
Knowledge-based	Wup	0.778	0.784	0.282	0.477	0.750	0.755	0.339	0.398
	J&C	0.695	0.731	0.354	0.498	0.820	0.804	0.318	0.402
	L&C	0.779	0.839	0.313	0.503	0.768	0.797	0.302	0.410
	Lin	0.835	0.858	0.329	0.513	0.750	0.788	0.348	0.424
	Resnik	0.813	0.836	0.362	0.431	0.693	0.731	0.353	0.404
Corpus-based	LSA	0.725	0.644	0.563	–	0.662	0.609	0.581	–
	ESA	0.588	–	0.503	–	0.727	–	0.629	–
	SSA	0.778	0.850	0.590	–	0.843	0.800	0.537	–
Ours	WTMGW	**0.879**	**0.861**	**0.622**	**0.572**	**0.846**	0.826	**0.750**	**0.480**
	WTMGT	0.871	0.847	0.602	0.539	0.820	**0.830**	0.748	0.477

Table 1 shows the results obtained using our mutual guidance model compared with the state-of-the-art methods (knowledge-based and corpus-based). The left

[1] `http://about.reuters.com/researchandstandards/corpus/`

column in Table 1 shows the measurement methods. The results in bold indicate the best results for a dataset and underlining denotes the second best results. Table 1 shows that the knowledge-based methods obtained very good results with the *MC30* and *RG65* datasets, which can be explained by the deliberate inclusion of familiar and frequently used dictionary words in these sets. As expected, our model performed better with large datasets such as *WS353* (γ from 0.282 to 0.622 and ρ from 0.302 to 0.750) and *MT771* (γ from 0.431 to 0.572 and ρ from 0.398 to 0.480), probably because the large datasets contained more technical and culturally biased terms, which cannot be covered by knowledge-based measures.

We can also conclude from Table 1 that our proposed model *WTMGW* performed best with most of the datasets, followed by *WTMGT*. Clear, the results obtained with *WTMGW* and *WTMGT* are similar because they only differed in terms of their beginning points. They shared the same iterative process, thus the results were similar after several iterations. It is also interesting to note that the performance of *WTMGW* was superior to that of the *SSA* method, although *SSA* uses Wikipedia as its knowledge resource. This may be because Wikipedia is very complicated and it contains a high level of noisy information, whereas our model only utilizes the context information, which is reliable and the semantics are abundant.

4.2 Text Relatedness

To evaluate the effectiveness of the WTMG model in determining the text-to-text relatedness, we used two datasets that have been employed in previous studies.

Lee50 comprises 50 documents collected from the Australian Broadcasting Corporation's news mail service. Each document was scored by ten annotators based on their semantic relatedness to all the other documents. The user annotations were then averaged per document pair, thereby yielding 2,500 documents pairs and their similarity score annotations. We found that there were no significant differences between the annotations when order of the documents in a pair differed, thus the evaluations used only 1225 document pairs after ignoring duplicates.

Li30 is a sentence pair similarity dataset, which was obtained by replacing each of the Rubenstein and Goodenough word-pairs with their respective definitions in the Collins Cobuild dictionary. Each sentence pair was scored by 32 native English speakers and the scores were averaged to generate a single relatedness score per sentence pair. The scores were skewed toward low similarity sentence-pairs, so a subset of 30 sentences was selected manually from the 65 sentence pairs to maintain an even distribution across the similarity rage.

AG400 is a domain-specific dataset related to computer science, which is used to evaluate the semantic relatedness of real-world applications such as short answer grading. The original dataset comprises 630 student answers and their corresponding questions. Each answer was graded by two judges on a scale from 0 to 5 and the Pearson's correlation coefficient between human judges was found to be 0.64. We noted a large skew in the grade distribution toward the high

Table 2. Pearson and Spearman results for the text relatedness datasets

Method		Pearson(γ)			Spearman(ρ)		
		Li30	Lee50	AG400	Li30	Lee50	AG400
Knowledge-based	COMB	0.810	0.702	0.480	0.832	0.356	0.365
Corpus-based	VSM	0.759	0.639	0.386	0.773	0.289	0.304
	LSA	0.810	0.635	0.425	0.812	0.437	0.389
	ESA	0.838	0.696	0.365	0.863	0.463	0.318
	SSA	0.848	0.684	0.567	0.832	0.480	0.495
Ours	WTMGW	**0.886**	**0.724**	**0.602**	**0.878**	**0.488**	0.486
	WTMGT	0.872	0.673	0.584	0.870	0.452	**0.512**

end of the grading scale, thus we randomly eliminated 230 of the highest grade answers to obtain more normally distributed scores.

Table 2 shows the semantic relatedness results obtained with the text datasets, where WTMG was compared with the knowledge-based and corpus-based methods. The results show that *WTMGW* obtained very good results with *Li*30 (γ=0.886 and ρ=0.878) and *Lee*50 (γ=0.724 and ρ=0.488). It is also interesting to note that large improvements were obtained using *WTMGW* (γ=0.602, ρ=0.486) and *WTMGT* (γ=0.584, ρ=0.512) compared with *LSA*, *ESA*, and *SSA* based on evaluations with the AG400 dataset.

As shown in Tabel 2, *WTMGW* and *WTMGT* clearly delivered the best performance, and they provided great improvements with the *AG400* dataset, but they were only slightly better than other methods with relatively small datasets.

5 Conclusions

The existing methods used to measure semantic relatedness consider the knowledge and the corpus independently, and they ignore the internal relationships between words and texts. In this study, we developed a word-text mutual guidance model to mine the deep relationships between words and texts, which combines semantic and statistical information using an iterative process. The initial word relatedness is calculated based on WordNet, which is semantically richer than corpus-based approaches. The text relatedness is then calculated based on the word relatedness, where this process utilizes the relationship between a text and its component words in an effective manner. The experimental results demonstrated that our proposed model is more effective than the state-of-the-art methods for semantic computing. The evaluations using standard word-to-word and text-to-text relatedness benchmarks confirmed the superiority and consistency of our model.

However, the model remains time consuming and there is still room for improvement, e.g., it may be possible to optimize the algorithm using dimensionality reduction. In future work, we will apply this semantic relatedness model to other NLP tasks such as text clustering or relationship classification.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant No.61100094, 61272383, 61300114 and 61103149). We thank the anonymous reviewers for their insightful comments.

References

1. Fellbaum, C.: WordNet. Wiley Online Library (1999)
2. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI 7, 1606–1611 (2007)
3. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: AAAI (2011)
4. Jain, V., Singh, M.: Ontology based information retrieval in semantic web: A survey. International Journal of Information Technology & Computer Science 5(10) (2013)
5. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
6. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes 25(2-3), 259–284 (1998)
7. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. WordNet: An Electronic Lexical Database 49(2), 265–283 (1998)
8. Lin, D.: An information-theoretic definition of similarity. In: ICML, vol. 98, pp. 296–304 (1998)
9. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, vol. 6, pp. 775–780 (2006)
10. Moreda, P., Llorens, H., Saquete, E., Palomar, M.: Combining semantic information in question answering systems. Information Processing & Management 47(6), 870–885 (2011)
11. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
12. Wenyin, L., Quan, X., Feng, M., Qiu, B.: A short text modeling method combining semantic and statistical information. Information Sciences 180(20), 4031–4041 (2010)
13. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138. Association for Computational Linguistics (1994)