Assessing Text Semantic Similarity Using Ontology

Hongzhe Liu
Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China
Email: liuhongzhe@buu.edu.cn

Pengfei Wang
Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China
Email: feipengwang767@163.com

***Abstract*—Sentence and document similarity assessment is key to most NLP applications. This paper presents a novel measure of calculating the similarity between sentences or between documents using ontology. The similarity is assessed using sentence or document concept vector forming from finding the linkage between ontology terms and sentence or document content, the linage can be used to generate semantic indexes of sentences or document and apply them to implement highly efficient searching algorithms to compute sentence or document similarity, and the difference between the sentence and document similarity measurement is articulated. Results were verified through experiments. Experiments show that this technique is efficient and compares favorably to other similarity measures, and it is flexible enough to allow the user to make comparisons without any additional dictionary or corpus information. We believe that this method can be applied in a variety of text knowledge representation and discovery applications.**

***Index Terms*—Sentence similarity, document similarity, WordNet, word similarity, ontology**

I. INTRODUCTION

Sentence in this paper is usually a short text while document is a long text. Many natural language processing applications require that the similarity between sentences or between documents be calculated quickly and reliably. The samples, usually pairs of sentences or documents, are considered similar if they are judged to have the same meaning or discuss the same subject. A method that can automatically calculate semantic similarity scores is much more valuable than simple lexical matching for applications such as question answering (QA), information extraction (IE), multi-document summarization, and evaluations of machine translation (MT). Most existing measures rely not only on ontology knowledge base like WordNet, but also on large text corpora which serve as additional knowledge resources. However, in many applications, especially in domain-based applications, large text corpus cannot be expected to be readily available. Many applications store ontology relations in a relational database, which do not fully represent the rich relations imbedded in the original text collections. In these cases, the similarities between these sentences or documents have to be extracted from the limited representations in the database only. In this paper, we focus on the challenge of assessing sentence similarities or document similarity by using the structural information inherent in a given ontology structure, we propose an unsupervised efficient approach to automatically calculate sentence or document level text similarities based on ontology structure, without using any external knowledge from other training corpora. The main contributions of our work to the field are as follows:

1. A means of deriving direct connecting nodes and relevancy nodes of a text (sentence or document) from its domain ontology is defined.
2. A means of forming text concept vector based on its domain ontology is proposed
3. A means of computing text similarity based on its concept vector is proposed.

The remainder of this paper is organized as follows: section II describes the related work of this paper including related work about sentence similarity and document similarity, section III describes our proposed methods, section IV is the experiment evaluation of the method, and section V gives a conclusion.

II. RELATED WORK

A. Sentence Similarity

There are mainly three classes of measures that can be used for identifying the similarity between sentences. These are Word Overlap Measures, TF-IDF Measures and Linguistic Measures according to [10][12]. We listed them in the following table Ⅰ.

Analyzing the above three class of sentence similarity measures, word overlap measures is relatively simple. However, in short texts or sentences, word co-occurrence may be rare or even null. This is mainly due to the inherent flexibility of natural language enabling people to express similar meanings using quite different sentences in terms of structure and word content, so deciding sentence similarity by their surface similarity is not reliable [10]; TF-IDF Measures need a large text corpus for statistics computation, but such a text corpus is hard

doi:10.4304/jsw.9.2.490-497

to obtain in my domain specific application; The linguistic measures outperform other classes of measures in accuracy, precision and recall metrics[10], but they have relatively low performance. Given two 50 words sentences, linguistic measures need to access WordNet structure 50*50 times for the word similarity computation, so they have to spend at least 2500 times of corresponding word similarity computation time for one pair of sentences.

B. Document Similarity

There are mainly four classes of measures that can be used for identifying the similarity between documents. These are binary similarity models, count similarity models, LSA similarity models, ontology based similarity models. We Listed them in the following table Ⅱ.

Analyzing the above four class of document similarity

TABLE I.
MEASURES OF RELATED METHODS

Measure classes	Class description	Main measures	Method description
Word Overlap Measures	A family of combinatorial similarity measure that compute similarity score based on a number of words shared by two sentences	Jaccard similarity coefficient[1]	The size of the intersection of the words in the two sentences compared to the size of the union of the words in the two sentences
		Simple word overlap fraction [2]	The proportion of words that appear in both sentences normalized by the sentence's length
		IDF overlap [2]	The proportion of words that appear in both sentences weighted by their inverse document frequency
		Zipfian overlap [3]	The Zipfian relationship between the length of phrases and their frequencies in a text collection
Corpus based Measures	Computing a cosine similarity between the corpus based vector representations of the two sentences as similarity, a set of words that appear in the sentence pair is used as a feature set	LSA[14]	Analysis a large corpus of natural language that generate representation to capture semantics
		HAL[15]	Statistics lexical concurrence between words to produce a high dimensional space to compute short text similarity
		TF-IDF measure[4]	Based on the sum of the product of term frequency and inverse document frequency of words that appear in both sentences
		Identity measure [5]	The sum of inverse document frequency of the words that appear in both sentences normalized by the overall lengths of the sentences and the relative frequency of a word between the two sentences
Linguistic Measures	Utilize semantic relations between words and their syntactic composition, to determine the similarity of sentences	Li et al. [9]	A linear combination of semantic vector similarity and word order similarity.
		Mihalcea et al. [6]	Six knowledge based word semantic similarity measure[16]-[21] with word specificity scores to form sentence vector to compute sentence similarity.
		Malik et al.[7]	The sum of maximum word similarity scores of words in the same part-of-speech class normalized by the sum of sentence's lengths
		Islam and Inkpen [11][13]	Semantic word similarity with corpus-based word specification and a normalized and modified version of the Longest Common Subsequence (LCS) string matching algorithm.
		WSD based measure[8]	A method based on a comparison of word sense disambiguation and string similarity.

TABLE II.
MEASURES OF RELATED METHODS

Measure classes	Class description	Main measures	Method description
Binary similarity models	word-based, keywords-based and n-gram measure to determine similarity	Tversky's Contrast Model[22]	Measures similarity as the ratio of common to common and distinctive features.
		Common Features Model[23]	Assumes simply that similarity is measured by the proportion of common features
		the Distinctive Features based contrast Model[24]	Assumes that two stimuli become more dissimilar to the extent that one stimulus has a feature that the other does not
Count similarity models	Similarity models mainly based on the corpus representations using counts[25]	the Correlation model	Correlation measure
		the Jaccard model	Jaccard measure
		the Cosine model	Cosine-vector
		the Overlap model	Overlap measures
LSA similarity models	Latent Semantic Analysis models	the local weighting function[26]	Measures the importance of a word within a document
		The global weighting function [27]	Measures the importance of a word across the entire corpus of documents, normalized each word using the local weighting function; an inverse document frequency measure, an entropy measure.
		Local and global weighting functions[26]	Local and global weighting functions are used to generate a weighted corpus representation and it is subjected to singular value decomposition
Ontology based similarity models	Based on ontology like WorldNet or Wikipedia to compute similarity	WorldNet based[28][30][31]	Identifying similar documents based on a conceptual tree-similarity measure
		Wikipedia based ESA[29]	Represent the meaning of any text as a weighted vector of Wikipedia-based concepts

measures, similar to the word overlap measures for sentence similarity, binary similarity models is relatively simple, however, natural language enabling people to express similar meanings using quite different words, so deciding text similarity by their surface similarity of their words or phrases is not reliable. Count similarity models and LSA similarity models need a large text corpus for statistics computation, but such a text corpus is hard to obtain in my domain specific application. The existing ontology based similarity methods [31] are usually based on direct word mappings from text to concept, so their similarity computation is also based on literal similarity to some extent.

C. Summary of the Related Work

In summary, a number of methods for text similarity measurement have emerged, and some of them are mature and perform well in specific application. At the same time, each kind of methods has its own shortcomings. The common problems of the sentence similarity and the document similarity computing is that simple measures are not reliable, the statistics measures relay on a additional text corpus, complex measures may have performance problem, and quite a few.of the methods are based on surface similarity instead of semantic similarity.

III. Our Proposed Methods

In this work we consider ontology as knowledge structures that specify terms, their properties and relations among them to enable share and reuse of knowledge. Ontology collects and organizes terms of references. We represent ontology using a tree based model that reflects structural and semantic relationships between terms. Finding the linkage between ontology terms and sentence content can be used to generate semantic indexes of sentences and apply them to implement highly efficient searching algorithms to compute sentence similarity and document similarity.

A. Sentence Similarity

A.i The Definitions

First we define a tree based ontology which is basic of our method.

Definition 1 (hierarchical concept tree). HCT is denoted as T (N, E), a rooted tree where N is the set of concept nodes in the tree and E is the set of edges between the parent/child pairs in T. The semantic coverage of the child concept nodes is the partition of the semantic coverage of their parent concept node. HCT is a kind of tree structure based ontology.

The HCT is the basis of our method, as we represent ontology using HCT that reflects structural and semantic relationships between concepts. Our similarity computation is derived from cosine similarity, which is based on the orthogonality of its components, so the semantic coverage of the concept nodes should be independent. The limits of the semantic coverage of the child concept nodes are the partition (instead of covering) of the semantic coverage of their parent concept nodes. That is, the concepts subsumed by sibling concept nodes are usually non-overlapping; the relationship between two siblings is captured only through their ancestor concept nodes.

How to extract nodes that relevant to a target node is important to our problem, especially it is the key to converting the surface sentence similarity computing to the semantics similarity computing. The nature of the HCT is tree structure, and from tree structure we know that the ancestor concept nodes of any given concept node in the hierarchy subsume its attributes, and its descendent concept nodes inherit them. So we define that the ancestor and descendent concept nodes are relevant to that concept node in definition 2.

Definition 2 (concept node's relevancy nodes). The relevancy nodes of a given concept node in the HCT are its ancestor and descendent concept nodes.

A sentence may be composed of many words, in which some are articles, prepositions, conjunctions and particle, etc. These words are less important to the meaning of sentence, so we discard them when conducting similarity computation. The rest words are key words of the sentence. We map the key words of the sentence to the domain ontology. The sentence direct connecting nodes and relevancy nodes are extracted from ontology in definition 3 and definition 4.

Definition 3 (sentence direct connecting nodes). If a key word in sentence is equivalent to concept node in HCT, then the concept node is named the word's direct connecting node. Direct connecting nodes of all the key words composing the sentence are named sentence direct connecting nodes.

Definition 4 (sentence relevancy nodes). The relevancy nodes of a given sentence in the HCT are the union of all the relevancy nodes of each direct connecting node of the sentence.

Lastly, a sentence concept vector is formed based on direct connecting nodes and relevancy nodes in definition 5.

Definition 5 (direct connecting nodes and relevancy nodes based sentence concept vectors). Given an HCT with n concept nodes, the concept vector of a sentence is denoted as S= (v_1, v_2,…,v_n) and v_i (i =1, 2, … n) is the dimension value corresponding to all concept nodes of the ontology relative to the particular sentence pair, defined as follows using (1):

$$v_i = \begin{cases} 1\,(if\ C_i\ is\ the\ document\ \ direct\ \ connecting\ \ nodes) \\ w_1\,(if\ C_i\ is\ the\ document\ \ relevancy\ nodes\ )(0 < w_1 < 1) \\ 0\ \ (Otherwise) \end{cases} \quad (1)$$

As the document relevancy nodes in ontology are not as important as document direct connecting nodes, so we give them less weight w_1.

A.ii Working Steps of the Measure

Given an ontology is depicted in the following figure 1, two sentence contain key words set T_1 and T_2.

T_1 = {key words in S_1} = {$word_1$, $word_2$}
T_2 = {key words in S_2} = {$word_a$, $word_b$}

We have four steps to compute the sentence similarity:

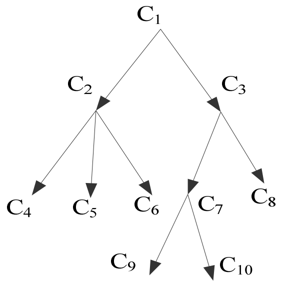


Figure 1. Tree based ontology example

Step 1: Map the key words in sentences to concepts in ontology to find the sentence direct connecting nodes.

Given $word_1$, $word_2$ in sentence 1 mapped to concept nodes C_3, C_4 of the ontology, according to **definition 3**, C_3, C_4 are the direct connecting nodes of sentences 1; Given $word_a$, $word_b$ in sentence 2 mapped to concept nodes C_2, C_9 of the ontology, C_2, C_9 are the direct connecting nodes of the sentence 2, then T_1 and T_2 can be represented as figure 2:

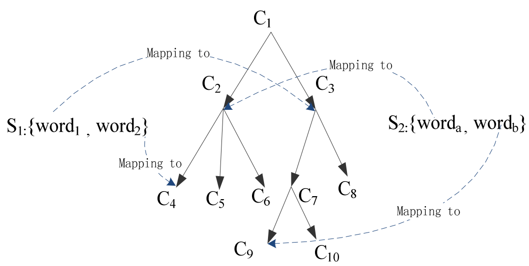


Figure 2. Direct connecting nodes mapping

T_1= {key words in S_1} = {$word_1$, $word_2$} $\xrightarrow{\text{mapped to}}$ {C_3, C_4}

T_2= {key words in S_2} = {$word_a$, $word_b$} $\xrightarrow{\text{mapped to}}$ {C_2, C_9}

Step 2: Expand the direct connecting nodes of the sentence according to their relevancy nodes in ontology.

In figure 1, according to **definition 2**, relevancy nodes of C_3 (Rel(C_3)) are C_1, C_7, C_8, C_9, C_{10}, relevancy nodes of C_4(Rel(C_4)) are C_1, C_2 , according to **definition 4**, the sentence relevancy nodes of S_1 are Rel(C_3) ∪ (Rel(C_4).

Relevancy nodes of C_2 (Rel(C_2)) are C_1, C_4, C_5, C_6, relevancy nodes of C_9 (Rel(C_9)) are C_1, C_3, C_7, according to **definition 4**, the sentence relevancy nodes of S_2 are Rel(C_2) ∪ (Rel(C_9). The direct connecting node of S_1 and S_2 can be expanded to direct connecting node plus the relevancy nodes:

T_1= {words in S_1} = {$word_1$, $word_2$} $\xrightarrow{\text{mapped to}}$ {C_3, C_4} $\xrightarrow{\text{expand to}}$ {C_3, C_4, C_1, C_2, C_7, C_8, C_9, C_{10}}
T_2= {words in S_2} = {$word_a$, $word_b$} $\xrightarrow{\text{mapped to}}$ {C_2, C_9} $\xrightarrow{\text{expand to}}$ {C_2, C_9, C_1, C_3, C_4, C_5, C_6, C_7}

From the above example, we can see that there are no surface overlap between S_1 and S_2 originally, but after expanding with domain ontology, there are overlaps between them.

Step 3: Forming concept vector V_1, V_2 for sentence S_1, S_2.

The connections by the relevancy nodes are some "shallow" connections compared to the connections by the direct connecting nodes. So give the relevancy nodes relatively lower weight w_1 ($0 < w_1 < 1$). The dimension corresponds to all concept nodes of ontology in figure 1. According to definition 5, form the direct connecting nodes and relevancy nodes based sentence concept vector. If we were to list all concept nodes in sequential order of concept vectors according to the tree's breadth-first traversal sequence, we would have the concept vectors:

$V_1 = (w_1 , w_1 , 1, 1, 0, 0, w_1 , w_1 , w_1 , w_1)$
$V_2 = (w_1, 1, w_1, w_1 , w_1 , w_1 , w_1 , 0, 1, 0)$

Step 4: Compute the sentence similarity using vector cosine similarity.

$$sim(V_1,V_2) = \frac{V_1 \bullet V_2}{\|V_1\|\|V_2\|} \qquad (2)$$

B. Document Similarity

The weighting of words in document

Sentence similarity and document similarity computing are similar, and we have the similar definition for document like the definition of the document direct connecting nodes and the document relevancy nodes. The main difference between sentence and document is that the document contains more words; the importance of the words differs largely in a document. So unlike the sentence direct connecting nodes were given equal importance to similarity computing, the document direct connecting nodes should have different importance to similarity computing. Recall the previous solution to most of this problem is the tf–idf weighting scheme. The tf–idf weighting scheme is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Two intuitions are at play in the tf–idf weighting scheme:

- the more frequently a word occurs in a text document, the more important for the document is it (the *term frequency* intuition);
- the more documents a word occurs in, the less discriminating is it, i.e. the smaller its contribution is in characterizing the semantics of a document in which it occurs (the *inverse document frequency* intuition).

The idf measure is also known as statistical specificity. Its value is usually extracted from large text corpus. If in some domain application, a large text corpus is not available, but domain taxonomy is usually there. Next we will analysis how to extract the 'idf weight' from a taxonomy structure.

In a tree structure, the upper the concept nodes located, the more abstract of its meaning and the less discriminating is it. The deeper the concept nodes located in a tree, the more concrete of its meaning and the more discriminating is it. Based on the above intuition, the 'idf *weight'* is generated as follows:

For each taxonomy in the concept vector, the algorithm recursively propagates weights to the parent node until the root node is reached. Weights are assigned to parents according to the following (3):

$$W_{Parent} = \alpha_2 * W_{Child} \quad (3)$$

- Where W_{Parent} is the weight of the parent.
- W_{Child} is the weight of the child and α is the weight propagation factor.

The weight propagation factor α_2 is used to determine how much of a child's weight is propagated to its parent. When $\alpha_2 = 0$, the parents will not be assigned any part of the child's weight. Given taxonomy in figure 1, $w_{c9}=w_{c10}=1$, and the following figure 3 illustrates the weight propagation of other nodes.

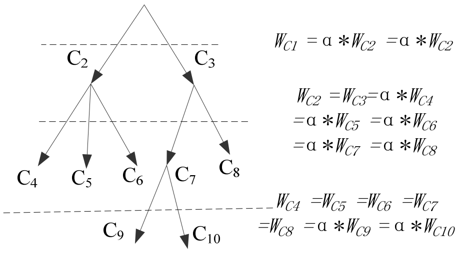


Figure 3. Weight propagation

IV. EXPERIMENTAL EVALUATION

A. *Sentence Similarity Measurement*

In order to fully evaluate our method, we incorporate all 14 sentences similarity measures of the three classes in reference [10] to compare with our method.

A.i Data Sets

The Microsoft Research paraphrase corpus (MSRP) data set is a known dataset for ground truth of sentence similarity calculation. The dataset includes 1,725 test pairs taken from Internet news articles [32]. Each sentence pair is judged by two human assessors whether they are semantically equivalent or not. Overall, 67% of the total sentence pairs are judged to be the positive examples. Semantically equivalent sentences may contain either identical information or the same information with minor differences in detail according to the principal agents and the associated actions in the sentences. Sentence that describes the same event but is a superset of the other is considered to be a dissimilar pair. Note that this rule is similar to the one used in text entailment task.

A.ii Ontology

WordNet is the product of a Princeton University research project that has attempted to model the lexical knowledge of a native speaker of English [34]. The system uses both online thesauri and online dictionaries to organize each part of speech (such as nouns and verbs) into taxonomies that render each node into a set of synonyms (synset). These synsets are represented as one sense. Words with more than one sense appear in multiple synsets. WordNet also defines the semantic and lexical relations between synsets and word senses

Nouns and verbs are organized into hierarchies based on the hypernymy/hyponymy or hyponymy/troponymy relationships between synsets. We use only WordNet nouns and verbs in this work to map words for the sentences in the MSRP data set. Hyponym/hypernym and hyponymy/troponymy relations take up about 80% of all WordNet relations. Strictly speaking, HCT based on WordNet nouns and verbs is not a tree, as there are a few nodes have more than one father. We choose one of its fathers randomly to form our concept vector (experiment show that choosing different father has very small influence on result). Figure 4 is the tree structure in our experiment.

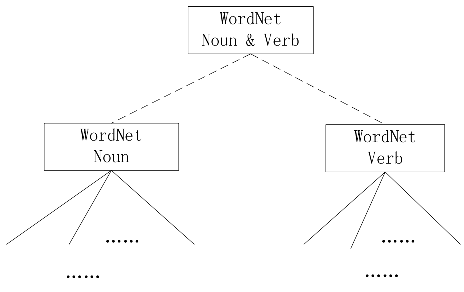


Figure 4. Tree structure of WordNet nouns and verbs

A.iii Evaluation Criteria

We look into four different evaluation measures, accuracy, precision, recall, and F-measure which capture different aspects of semantic similarity. The meaning of them is defined as following:

- Accuracy=(TP + TN)/(TP + TN+ FP + FN)
- Precision= TP/(TP + FP)
- Recall= TP/(TP + FN)
- F-measure= (1 + β)PR/(βP + R)= 2PR / (P + R) (when β=1, precision and recall have the same weight)

TP: Number of sentences predicted to be similar sentences that actually are similar.

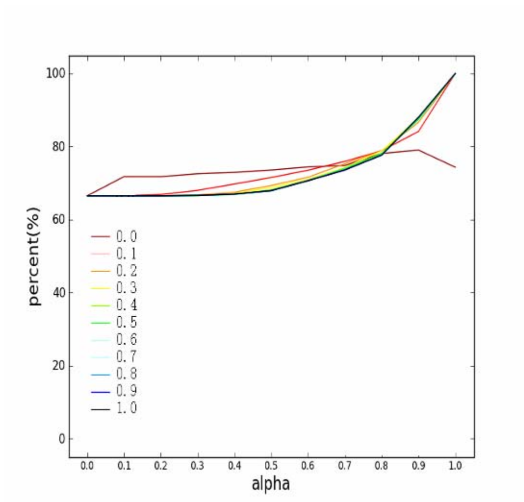
TN*:* Number of sentences predicted to be dissimilar sentences that actually are dissimilar

FP: Number of sentences predicted to be similar that are actually dissimilar

FN: Number of sentences predicted to be dissimilar that are actually similar

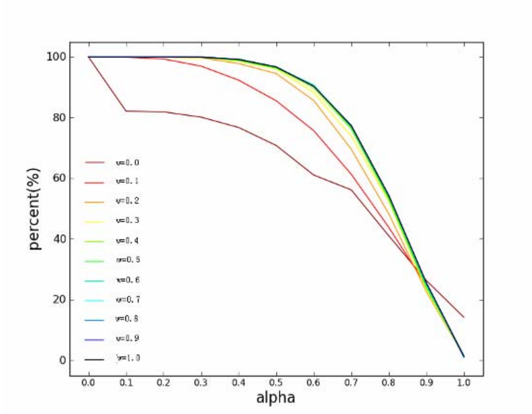
A.iv Result Analysis

We evaluated the results in terms of accuracy, precision, recall, and F-measure. The following figure 5 and figure 6 illustrated that the precision and recall values with different similarity threshold *(a_1)* and different relevancy nodes weight *(w_1)*. We can see from the cures that the measure obtain similar performance with precision and recall when relevancy nodes weight w_1 >= 0.4.

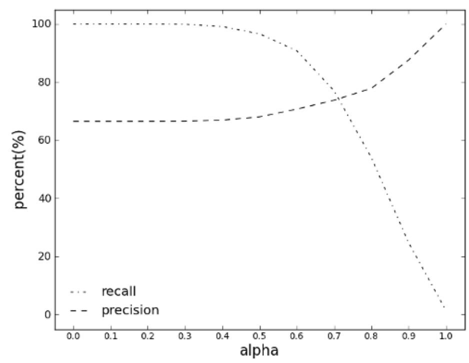


x-axis: Similarity threshold (a_1)
y-axis: Precision values (percent(%))
Figure 5. Precision with different similarity threshold (a_1) and different relevancy nodes weight w_1

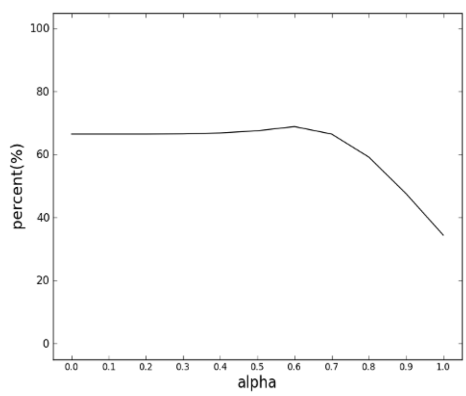
We choose $w_1 = 0.6$ as our relevancy nodes weight, the next figure 7 is precision and recall with different similarity threshold value. Figure 8 is accuracy values with different similarity threshold value when $w_1=0.6$; we can see that maximum accuracy value is 0.69 when similarity threshold $a_1=0.6$.



x-axis: Similarity threshold (a_1)
y-axis: Recall values (percent(%))
Figure 6. Recall with different similarity threshold (a1) and different relevancy nodes weight (w1)



x-axis: Similarity threshold (a_1)
y-axis: Precision and recall values (percent(%))
Figure 7. Precision and recall with different similarity threshold value (weight w_1=0.6)



x-axis: Similarity threshold (a_1)
y-axis: Accuracy values (percent(%))
Figure 8. Accuracy with different similarity threshold value （weight w_1=0.6）

We evaluated these results in terms of accuracy, precision, recall, and f-measure. *Recall* is a proportion of correctly predicted similar sentences compared to all similar sentences. *Precision* is a proportion of correctly predicted similar sentences compared to all predicted similar sentences. Most methods yield good results for precision or recall, but very few do so for both. *Accuracy* is a proportion of all correctly predicted sentences compared to all sentences; the higher of the value marks the more accurate of the result. In table III, the average precision, recall, accuracy and F-measure of the three kinds compared with our methods are from reference [10]. The average accuracy of word overlap methods is 0.62, the average accuracy of tf-idf methods is 0.63, and the average accuracy of linguistic methods is 0.65, though ours climbed as high as 69% when we use the similarity threshold score 0.6 and relevancy node weight 0.6. F_measure is a uniform harmonic mean of precision and recall. We also obtain the highest F-measure value (0.80) when accuracy equal 69%.

TABLE III. :
AVERAGE PRECISION, RECALL, ACCURACY, AND F-MEASURE OF RELATED METHODS

method	Precision	Recall	Accuracy	F_measure
word Overlap measures	0.78	0.62	0.62	0.66
TF-IDF measures	0.74	0.75	0.63	0.69
Linguistic measures.	0.68	0.89	0.65	0.76
Our method (w_1=0.6, a_1 =0.6)	0.71	0.91	0.69	0.80

It is noteworthy that the average processing time for one pair of sentence of our measure takes 0.025 second, while on the same computer platform, it take at least 1 minutes averagely for computing the similarity of one pair of sentences for word to word similarity based measures.

B. Document Similarity Measurement

B.i Data sets and ontology

For document similarity, we used the Michael D. Lee document dataset [33], a collection of 50 documents from the Australian Broadcasting Corporation’s news mail service. These documents were paired in all possible ways, and each of the 1,225 pairs has 8–12 human judgments. When human judgments have been averaged for each pair, the collection of 1,225 relatedness scores has only 67 distinct values. Spearman correlation is not appropriate in this case, and therefore we used Pearson’s linear correlation coefficient. We used WordNet as base Ontology as in sentence similarity measurement.

B.ii Result analysis

WordNet based document similarity calculation, setting an indirect connection concept node w_2=0.4 weight, layer decreasing factor α_2 =0.85, get Pearson linear correlation coefficient of 0.55. It can be seen from the literature, based on word overlap similarity calculation method and most corpus-based statistical document similarity calculation method, the best results of linear correlations for Pearson 0.4-0.5; in the method based on the statistics, the LSA similarity model gets the best pearson linear correlation coefficient between 0.5-0.6; some documents based on ontology similarity calculation method based on Wikipedia, for example document similarity calculation by ESA method have higher Pearson linear correlation coefficient, but it is very difficult to apply to domain data. Our document similarity methods can be built based on domain ontology, which do not rely on additional information, compared with the people's judgment, our method has obtained good Pearson linear correlation value, and it is highly efficient.

V. CONCLUSION

Experiments prove that our method compares favorably to related measures, our method perform efficient and does not need any external knowledge from other training corpora. This method allows the user make efficient comparisons between sentences and document based solely on ontological structure without requiring on any additional dictionary or corpus of information. In this way, our approach can be applied directly to any application with domain ontology. We believe this is a very attractive feature in building new applications.

ACKNOWLEDGEMENTS

This project was supported by the Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges under Beijing Municipality (CIT&TCD20130513), Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality (Grant No. PHR201108419), and the National Natural Science Foundation of China (Grant No. 60972045 and grant No.61271369).

REFERENCES

[1] Jacob B, Benjamin C(2008) Calculating the Jaccard Similarity Coefficient with Map Reduce for Entity Pairs in Wikipedia,http://www.infosci.cornell.edu/weblab/papers/ Bank2008.pdf .
[2] Metzler, D., Bernstein, Y., Croft, W., Moffat, A., and Zobel, J. (2005) Similarity measures for tracking information flow. Proceedings of CIKM, 517–524.
[3] Banerjee, S. and Pedersen, T. (2003). Extended gloss overlap as a measure of semantic relatedness. In Proceedings of IJCAI’03, Acapulco, Mexico, 805-810.
[4] Allan, J., Bolivar, A., and Wade, C. (2003) Retrieval and novelty detection at the sentence level. In Proceedings of SIGIR’03, 314–321.
[5] Hoad, T. and Zobel, J. (2003) Methods for identifying versioned and plagiarized documents. Journal of the American Society of Information Science and Technology,54(3), 203–215.
[6] Mihalcea, R., Corley, C., and Strapparava, C. (2006) Corpus-based and knowledge-based measures of text semantic similarity, in Proceedings of AAAI 2006.vol.1.
[7] Malik, R., Subramaniam, V., and Kaushik, S. (2007) Automatically Selecting Answer Templates to Respond to Customer Emails. In Proceedings of IJCAI’07, Hyderabad, India, 1659-1664.
[8] Chukfong H.,Masrah A. Azmi M.,Rabiah A. K.(2003), Shyamala C. Doraisamy. Word sense disambiguation-based sentence similarity, Proceedings of the 23rd International Conference on Computational Linguistics, 418-426

[9] Li, Y., Bandar, Z., and McLean, D.(2003). An approach for measuring semantic similarity using multiple information sources. IEEE Transaction Knowledge Data Eng. 15(4).871–882.
[10] Palakorn A., Xiaohua H., Xiajiong S. (2008). The Evaluation of Sentence Similarity Measures, Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery.305-316
[11] Aminul I., Diana I.(2008). Semantic text similarity using corpus-based word similarity and string similarity, ACM Transactions on Knowledge Discovery from Data (TKDD), 2(2)
[12] Junsheng Z., Yunchuan S., Huilin W.(2011). Calculating Statistical Similarity between Sentences, Journal of Convergence Information Technology, 6(2)
[13] A. Islam and D. Inkpen(2009). "Semantic similarity of short texts". Recent Advances in Natural Language Processing V: Selected Papers from Ranlp 2007, 227-231.
[14] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.
[15] Burgess, C., Livesay, K., & Lund, K.(1998) Explorations in Context Space: Words, Sentences, Discourse. Discourse Processes, 211 - 257.
[16] A.Leacock and M. Chodorow(1998). Combining local context and WordNet sense similarity for word sense identification. In WordNet, an Electronic Lexical Database, 265–283.
[17] P.Resnik(1999) Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, Journal of Articial Intelligence Research, 95–130.
[18] S. Banerjee and T. Pedersen (2003). Extended gloss overlaps as a measure of semantic relatedness. In Proceedings of IJCAI, 805–810.
[19] Z. Wu and M. Palmer.(1994) Verb semantics and lexical selection. In Proceedings of ACL.133–138
[20] J.JiangandD.Conrath(1997).Semantic similarity based on corpus statistics and lexical taxonomy.In Proceedings of COLING.19(1)17-30
[21] D. Lin(2008). An information-theoretic definition of similarity. In Proceedings of ICML. 296–304.
[22] Tversky, A. (1977). Features of similarity. Psychological Review, 84(4):327–352
[23] Shepard, R. N. and Arabie, P. (1979). Additive clustering representations of similarities as combinations of discrete overlapping properties. Psychological Review, 86(2):87–123.
[24] Rohde, D. L. T. (2002). Methods for binary multidimensional scaling. Neural Computation, 14(5):1195– 1232.
[25] Rorvig, M. E. (1999). Images of similarity: A visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets. Journal of the American Society for Information Science, 50(8):639–651.
[26] Lee, M. D. and Navarro, D. J. (2002). An Empirical Evaluation of Models of Text Document Similarity. The annual conference of Cognitive Science Society 1254-1259.
[27] Pincombe, B. M. (2004). Comparison of human and latent semantic analysis (LSA) judgments of pairwise document similarities for a news corpus. Defence Science and Technology Organisation Research Report DSTO–RR–0278.
[28] Susan G., Mirco S. (2008).Document similarity based on concept tree distance. Proceedings of the 19th ACM Conference on Hypertext and Hypermedia, 127-132.
[29] Evgeniy G., Shaul M.(2007).Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, International Joint Conference on Artificial Intelligence ,1606-1611.
[30] Yun M., Jie L.， Zhengtao Y.(2013). Concept Name Similarity Calculation Based on Wordnet and Ontology，journal of software, Vol8(3), 746-753.
[31] Rushdi S.,Adel E.(2010). A Corpus-based Evaluation of a Domain-specific Text to Knowledge Mapping Prototype，Journal of computers, Vol 5(1), 69-80.
[32] Dolan, W., Quirk, C., and Brockett, C.(2004) Unsupervised construction of large paraphrase corpora: Exploiting massively parallel new sources. In Proceedings of the 20thInternational Conference on Computational Linguistics.350-356
[33] Lee, M.D., Pincombe, B.M., & Welsh, M.B. (2005). An empirical evaluation of models of text document similarity. Proceedings of the 27th Annual Conference of the Cognitive Science Society, 1254-1259.
[34] Wordnet website, available on: http://wordnet.princeton.edu/.2011.5.1

Hong-zhe Liu, Ph.D of school of computer and information technology from Beijing Jiao tong University, Beijing, China, she receive her M.S. degree in computer science from California State University, USA, in 1999. She is now an Assistant Processor of Computer Science Department, Beijing union University, vice director of Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China. Her research interests include semantic computing, artificial intelligence and distributed systems

Peng-fei Wang, master of software engineering of Beijing Key Laboratory of Information Service Engineering from Beijing Union University, Beijing, China, he received his bachelor degree in electronic information engineering from Beijing Union University, Beijing, China, in 2012.