Concept Name Similarity Calculation Based on Wordnet and Ontology

Yun Ma and Jie Liu*
College of Information and Engineering, Capital Normal University, Beijing, P.R.China
Email: 195307728@qq.com, liujxxxy@126.com

Zhengtao Yu
School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming Yunnan Province, China
Email: ztyu@hotmail.com

***Abstract*—Concept name similarity calculation between ontologies is the basis of ontology mapping. Many concept name words are polysemous, but in traditional word similarity algorithms, the similarity between the most similar senses of two words was regarded as the word similarity. It were ignored that the true meanings of concept names in the context, which resulted in falling of the precision of similarity calculation and ontology mapping. A new algorithm was presented in this paper: The true meaning of hypernym concept of an ontology concept and the comment of this concept were seen as two conceptual features. The conceptual WordNet sense the most similar to the features in all senses was determined as the true sense of this concept in ontology context. Each of the two features can generate a corresponding concept sense determination method and a kind of concept name similarity. At last, the weighted sum of the two kinds of similarities was seen as the final concept name similarity. The experimental result showed that our algorithm outperforms other existing algorithms. The ambiguity of the concept name words can be eliminated, and the precision of both of concept name similarity calculation and ontology mapping can be increased dramatically.**

***Index Terms*—concept name similarity, Word Sense Disambiguation, ontology, WordNet**

I. Introduction

Ontology mapping is applied widely in the fields of knowledge management and knowledge processing in Semantic Web. The concept name similarity calculation between ontologies is the basis of achieving the ontology mapping, and depends on the correct expression of the meanings of words in concept names, but many concept names, namely words, are polysemous. If there is no conceptual context, the sense of concept name will not be able to be determined. In an ontology, all features associated with a concept can be used as conceptual context to serve for concept sense disambiguation.

In the majority of traditional word similarity algorithms, the similarity between the most similar senses of two words, namely the maximum similarity between two words, is regarded as the word similarity. In these algorithms, it are ignored that the inherent meanings of the words in the ontology context, which results in the falling of the accuracy rate of the concept name similarity calculation and ontology mapping.

A new algorithm is presented in this paper: The true meaning of hypernym concept of an ontology concept and the comment of this concept are seen as two important conceptual features. The sense the most similar to the features in all senses is determined as the correct WordNet sense of this concept in ontology context. Each of the two features can generate a corresponding concept sense determination method and a kind of concept name similarity. At last, the weighted sum of the two kinds of similarities is seen as the final concept name similarity.

In this paper, the benchmark tests in OAEI ontology matching campaign 2011 were used as the experimental data sets. For each of 109 ontology pairs, the concept name similarity between this ontology pair was computed in different concept name similarity algorithms. Lastly, the *precision*, *recall* and *f-meature* of the results in the different algorithms were got. The experimental results showed our algorithm outperformed other existing algorithms.

The rest of this paper is as follows: the related research work is presented in section II; the general process of concept name similarity calculation algorithm in section III; our algorithm is specified in section IV, V and VI; experiment and result evaluation in section VII; the conclusion is in section VIII.

II. Related Research Work

In the existing algorithms for calculating the concept name similarity: In [1], [2], Jaccard coefficient and edit

Manuscript received May 9, 2012; revised July 9, 2012; accepted August 11, 2012.

This paper is supported by Beijing Natural Science Foundation (4123094). The Science and Technology Project of Beijing Municipal commission of Education (No.KM201110028020, KM201010028019). National Nature Science Foundation (No.60863011, 61175068, 61163022). Beijing Key Construction Discipline "Computer Application Technology".

Corresponding author is Jie Liu. Email: liujxxxy@126.com.

doi:10.4304/jsw.8.3.746-753

distance were used to calculate syntactic similarities between concepts, but it was ignored that different words in syntax may be synonymous; In [3], semantic similarities between concepts were computed by WordNet, and the similarity between the most similar senses of two words, namely the maximum similarity between two words, was regarded as the word similarity. But it were ignored that the inherent meanings of the words in the ontology context; The viewpoint in [4], [5] is that if two ontology concepts are related to each other with hypernym-hyponym relation, the correct WornNet sense of this two concepts are also related to each other with the same relation. So, seeking the correct WordNet sense of a concept can be transformed into looking for two WordNet nodes related by hypernym-hyponym relation. This algorithm is so strict that it was ignored that the hierarchies of many ontologies are inconsistent with that of WordNet; The algorithm of the references [4], [5] was improved in [6]. The viewpoint in [6] is that when the true meanings of two ontology concepts are related by hypernym-hyponym relation, there is a close succession relation and a high similarity between the two meanings, and this high similarity also exists between two WordNet senses representing respectively the two meanings. The strictness of this algorithm was reduced, but the most similar senses of two concepts are likely to be not related by hypernym-hyponym relation. In addition, this algorithm will not be able to be used if there are very few concepts related by hypernym-hyponym relations; In [7], by Latent Semantic Analysis (LSA), it was computed that the sentence similarity between the comment of an ontology concept and the gloss of each of all WordNet senses of this concept. And the sense the most similar to the comment was seen as the correct WordNet sense of this concept. The accuracy of similarity calculation was increased, but the contribution of syntactic structure to similarity calculation was not taken into account.

The proposed algorithm in this paper involves the sentence similarity computation. In the existing algorithms about sentence similarity calculation: In [8], [9], the sentence similarity was calculated by vector space model (VSM), and in [10], the number of shared words in two sentences was used to compute the sentence similarity. In this two algorithms, it was ignored that the contribution of syntactic structure to similarity calculation. And it was considered that the syntactic similarity rather than the semantic similarity; In [11], LSA was used to calculate the sentence similarity. It was considered that latent semantic similarity of sentences, but LSA is not suitable for computing the similarity about short text or sentence; In [12], according to the part of speech, the semantic space of LSA was divided into multiple small semantic spaces. Syntactic structure was considered, but each of these small semantic spaces has fewer words than before and LSA is not fit for processing short text; In [13], [14], [15], the sentence was parsed by the syntactic parser into the syntactic tree including subject, predicate and object. It were calculated that the word similarities between the same grammatical constituents of two sentences. Finally, the weighted sum of the similarities of the three different grammatical constituents is seen as the final sentence similarity. The syntactic structure was taken into account, but words were not disambiguated when word similarities were calculated.

III. General Process of Concept Name Similarity Calculation Algorithm

The general process of the new algorithm presented in this paper is showed in Fig. 1: Firstly, On the condition that the correct WordNet sense of an top-level ontological concept is determined manually by experts, the correct WordNet sense of the hyponym concept of this top-level concept can be defined as that sense the most similar to the correct sense of this top-level concept in all the senses of this hyponym concept; Secondly, by LSA based on syntactic analysis, it is computed that the sentence similarity between the comment of the above hyponym concept in the ontology and the gloss of each of all WordNet senses of this hyponym concept. And the sense the most similar to the comment is seen as the correct WordNet sense of this hyponym concept; Lastly, concept name similarity equals the similarity between the correct senses of two concepts. So, the above two steps, namely two methods of determining the correct sense, can generate two kinds of concept name similarities. The weighted sum of the two kinds of similarities is seen as the final concept name similarity.

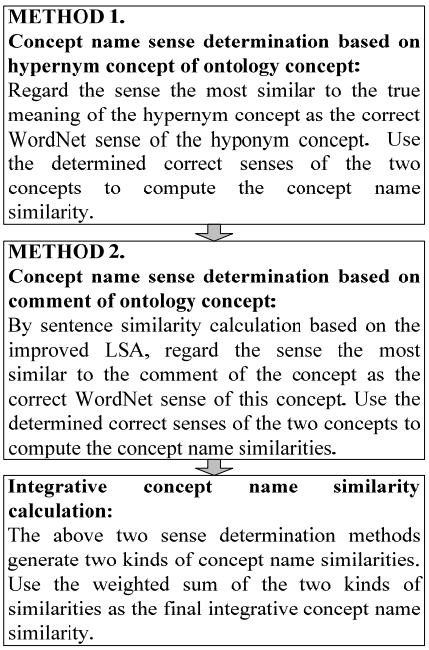

Figure 1. General process of concept name similarity calculation algorithm.

IV. Concept Name Sense Determination Based on Hypernym Concept of Ontology Concept

When the true meanings of two ontology concepts are related by hypernym-hyponym relation, there is a high similarity between the two meanings, and this high similarity also exists between two WordNet senses

representing respectively the two meanings [6]. In this section, on the condition that the correct WordNet sense of an ontological top-level concept is determined manually by experts, the correct WordNet sense of the hyponym concept of this top-level concept can be defined as that sense the most similar to the correct sense of this top-level concept in all the senses of this hyponym concept.

The specific steps of this operation are as follows:

(1) Extract a top-level concept $C\_hyper \in O_1$ and its hyponym concept $C\_hypo \in O_1$ in the ontology O_1. Determine manually the correct WordNet sense of C_hyper. And find out all the WordNet senses of this hyponym concept $Sense\_hypo_j, j = 1,2,...,n$, in which n means the number of the senses of C_hypo [4], [5].

(2) Using the algorithm "WuAndPalme" in the Java API "Java WordNet Similarity", find the sense the most similar to the correct sense of this top-level concept C_hyper in $Sense\_hypo_j, j = 1,2,...,n$. This found sense is regarded as the correct WordNet sense of C_hypo.

(3) According to the above two steps, exact all the top-level ontology concepts having the hyponym concepts. Determine manually the correct WordNet sense of each of these top-level concepts. For a certain top-level concept, regard its correct sense as the reference sense, and the correct sense of any of its hyponym concepts is defined as the sense the most similar to the reference sense in all the senses of this hyponym concept.

In summary, in the sense determination method of this section, the true meaning of hypernym concept of an ontology concept is seen as the conceptual context, for which the ambiguity of the concept name can be eliminated to some extent. This sense determination method is faster and more accurate than the sense determination method referring to sentence similarity calculation in section V. However, firstly, the most similar WordNet senses of two concepts are likely to be not related by hypernym-hyponym relation, which will produce the wrong results of sense determination; secondly, too many top-level concepts having the hyponym concepts will burden manual determination of senses; thirdly, this sense determination method will not be able to be used if there are very few concepts related by hypernym-hyponym relation. According to the above advantages and disadvantages, the sense determination method of this section is combined with that of section V to determine the correct sense of concept.

V. Concept Name Sense Determination Based on Comment of Ontology Concept

In this section, it is computed that the sentence similarity between the comment of an ontology concept and the gloss of each of all WordNet senses of this concept. And the WordNet sense the most similar to the comment is seen as the correct WordNet sense of this concept. In many existing sentence similarity algorithms, sentence similarity calculation derives from word similarities calculation between the same grammatical constituents of two sentences. But these algorithms still refer to a problem that the ambiguity of words can not be eliminated. In this paper, to avoid the drawback of word similarity calculation, LSA based on syntactic analysis is used to calculate the sentence similarity.

A. Syntactic Analysis of Sentence

Each of words in a sentence can get a syntactic weight by syntactic analysis, and the weight is used in LSA. Here, the premise of getting the syntactic weight of a word is gaining firstly the syntactic importance degree of this word.

The specific steps of gaining the syntactic importance degree are as follows:

(1) Exact the comment of an ontology concept $S\_comment$ and the glosses of all WordNet senses of this concept $S\_sense_j, j = 1,2,...,n$, in which n means the number of the senses of this concept. Both comment and gloss are sentence.

(2) Using the syntactic parser "Stanford Parser", parse these exacted sentences into the syntactic trees including three grammatical constituents, namely subject, predicate and object. For each of words in these sentences, note the part of speech and the grammatical constituent the word belongs to. For example, Fig. 2 is the syntactic tree of the sentence "The teacher sings a good song". In Fig. 2, NP is the noun phrase, and VP is the verb phrase, and NN is the noun, and VBZ is the third person singular of the verb, and JJ is the adjective, and DT is the qualifier. The subject, the predicate and the object of this sentence can be extracted from the syntactic tree, shown in Fig. 3.

(3) Preprocess the words and the sentences. Remove the stopwords of these sentences, such as qualifiers and prepositions. And recover the rest words to the form of the word root.

(4) Use all the remaining preprocessed words in a sentence to compose the keyword set of this sentence. Here, the comment $S\_comment$ can produce a keyword set $KEYWORDSET\_comment$, and the n glosses $S\_sense_j, j = 1,2,...,n$ can produce n keyword sets $KEYWORDSET\_sense_j, j = 1,2,...,n$.

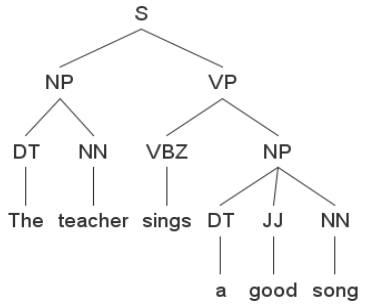

Figure 2. The syntactic tree of a sentence

(S (NP The teacher)
(VP sings
(NP a good song)))

Figure 3. Three grammatical constituents of a sentence

(5) The syntactic importance degree is showed in Table 1, and its equation is:

$$w\_sentenceSyntax = a_{pos} \cdot b_{gc},$$
$$\text{thereinto, } pos = \text{N,V,A,AD}; gc = \text{S,P,O}; \quad (1)$$
$$0 \le a_{pos} \le 1; 0 \le b_{gc} \le 1.$$

Thereinto, the syntactic importance degree equals the importance degree of part of speech multiplied by the importance degree of grammatical constituent; *pos* means the part of speech, and *gc* means the grammatical constituent; b_S , b_P and b_O are respectively the importance degree of subject, predicate and object, in which $b_S = b_P > b_O$; a_N , a_V , a_A and a_{AD} are respectively the importance degree of noun, verb, adjective and adverb, in which $a_N = a_V > a_A = a_{AD}$. If there exist no some grammatical constituents or some part of speech in a sentence, the importance degree of other constituents or part of speech will be increased to some extent.

TABLE I.
SYNTACTIC IMPORTANCE DEGREE BASED ON PART OF SPEECH AND GRAMMATICAL CONSTITUENT

Part of Speech	Grammatical Constituent		
	Subject (b_S)	Predicate(b_P)	Object(b_O)
Noun(a_N)	$a_N \cdot b_S$	--	$a_N \cdot b_O$
Verb(a_V)	$a_V \cdot b_S$	$a_V \cdot b_P$	$a_V \cdot b_O$
Adjective(a_A)	$a_A \cdot b_S$	--	$a_A \cdot b_O$
Adverb(a_{AD})	$a_{AD} \cdot b_S$	$a_{AD} \cdot b_P$	$a_{AD} \cdot b_O$

B. Latent Semantic Analysis Based on Syntactic Analysis

The improved LSA based on syntactic analysis can achieve sentence similarity calculation. The basic idea of LSA is: There exist latent semantic structures among words in document or sentence. There are the same semantic structures among synonyms and there are different semantic structures among polysemes. In the semantic space, through removing the less significant semantic structures, namely unimportant dimensions, LSA can eliminate the noise arising from synonyms and polysemes. In the semantic space whose dimensionality reduced, if the semantic structures of the words are similar to each other, there exist the high semantic similarities among these words. It is beneficial for calculating the sentence similarity. Moreover, the semantic structures among words are related to the frequency of occurrence of words in the document or sentence [7]. So, the latent semantic structure can be quantified through the statistical approach. LSA can avoid the defect that vector space model can not identify semantic similarity.

The specific steps of LSA based on syntactic analysis are as follows:

(1) Expanding the keyword set

In LSA, if the number of the keywords of a sentence is too few, the meaning of this sentence can not be expressed correctly. It results in that the accuracy rate of sentence similarity calculation is reduced. The comment of an ontology concept or the gloss of the WordNet sense of this concept has commonly about seven keywords, which is not good for LSA. It is necessary to expand the keyword set of sentence in order to increase the precision of LSA.

In this paper, for one of all WordNet senses of an ontology concept, preprocess the synonym set and the hypernym set of this WordNet sense. And generate several new keywords from the preprocessed synonym set and hypernym set. And then, merge these new keywords into the original keyword set of this WordNet sense. It makes the number of the keywords representing a sense be increased. Finally, expand *n* original keyword sets produced from *n* WordNet senses of this ontology concept into *n* new keyword sets $KEYWORDSET\_sense_j', j = 1,2,...,n$. However, note that the comment of this ontology concept $S\_comment$ has no the synonym set and the hypernym set in WordNet. So, the original keyword set produced from this comment remains unchanged and can not be expanded.

The keywords produced from the synonym set and the hypernym set of a WordNet sense do not come from the sentence, but these keywords still have their respective syntactic importance degrees. The syntactic importance degrees of the keywords produced from the synonym set and the hypernym set are denoted respectively by $w\_synsetSyntax$ and $w\_hypersetSyntax$. Their equations are:

$$w\_synsetSyntax = a_{pos} \cdot b_S, \; pos = \text{N,V,A,AD}. \quad (2)$$

$$w\_hypersetSyntax = c(a_{pos} \cdot b_S),$$
$$\text{thereinto, } pos = \text{N,V,A,AD}; 0<c<1. \quad (3)$$

Thereinto, the importance degrees of grammatical constituent of both $w\_synsetSyntax$ and $w\_hypersetSyntax$ are b_S , namely the importance degree of subject; the coefficient *c* in (3), makes the inequation $w\_synsetSyntax > w\_hypersetSyntax$ always correct, because the synonym is more important than the hypernym.

(2) Determining the integrative syntactic weight of a keyword

The equation of the integrative syntactic weight of a keyword in a sense is:

$$w\_syntax_{ij} = \frac{w\_h_i}{\sum_{q=1}^{N\_keyword_j} w\_l_q}, \quad \text{thereinto, } h,l = sentenceSyntax, synsetSyntax, hyperSyntax. \tag{4}$$

Thereinto, w_syntax_{ij} means that the integrative syntactic weight of the *i*-th keywords in the *j*-th WordNet senses; $N\_keyword_j$ means that the number of keywords in the *j*-th WordNet senses. The integrative syntactic weight of the *i*-th keywords equals the syntactic importance degree of this keyword divided by the sum of the syntactic importance degrees of all keywords in the *j*-th WordNet senses.

The equation of the integrative syntactic weight of a keyword in the comment is:

$$w\_syntax_i = \frac{w\_sentenceSyntax_i}{\sum_{q=1}^{N\_keyword} w\_sentenceSyntax_q}. \tag{5}$$

Thereinto, w_syntax_i means that the integrative syntactic weight of the *i*-th keywords in the comment; $N\_keyword$ means that the number of keywords in the comment. The integrative syntactic weight of the *i*-th keywords equals the syntactic importance degree of this keyword divided by the sum of the syntactic importance degrees of all keywords in the comment.

(3) Determining the keyword-sense matrix and the original comment vector

According to *n* new keyword sets produced from *n* WordNet senses of this ontology concept $KEYWORDSET\_sense'_j$, $j = 1,2,...,n$, construct a $m \times n$ keyword-sense matrix:

$$\boldsymbol{SENSE} = (w\_sense_{ij})_{m\times n}, \quad \text{thereinto, } w\_sense_{ij} = w\_syntax_{ij} \times \log_2(tf_{ij}+1). \tag{6}$$

Thereinto, *m* means that the number of keywords in the union of all keywords produced from the comment and *n* WordNet senses; *n* means the number of the senses of this concept; w_sense_{ij} means the weight of the *i*-th keywords in the *j*-th WordNet senses; tf_{ij} means the frequency of occurrence of the *i*-th keywords in the *j*-th WordNet senses. w_sense_{ij} equals the integrative syntactic weight of the *i*-th keywords multiplied by the logarithm of the sum of the keyword frequency and 1. The weights of high frequency keywords may be excessively prominent, which is not beneficial for LSA. So, the logarithm of the keyword frequency is adopted to reflect the contribution of keyword frequency to keyword weight, and it can reduce the effect of high frequency on LSA [16], [17].

According to the keyword set produced from the comment of this ontology concept $KEYWORDSET\_comment$, construct an original *m*-dimensional comment vector $\boldsymbol{comment}$:

$$\boldsymbol{comment} = (w\_comment_i)_{m\times 1}, \quad \text{thereinto, } w\_comment_i = w\_syntax_i \times \log_2(tf_i+1). \tag{7}$$

Thereinto, *m* means that the number of keywords in the union of all keywords produced from the comment and *n* WordNet senses; $w\_comment_i$ means the weight of the *i*-th keywords in the comment; tf_i means the frequency of occurrence of the *i*-th keywords in the comment. $w\_comment_i$ equals the integrative syntactic weight of the *i*-th keywords multiplied by the logarithm of the sum of the keyword frequency and 1.

(4) Singular Value Decomposition (SVD) and changing the original comment vector into the coordinate vector in the *k*-dimensional semantic space

Divide the matrix $\boldsymbol{SENSE}$ into three matrices by SVD:

$$\boldsymbol{SENSE} = \boldsymbol{TSD}^T \tag{8}$$

Thereinto, $\boldsymbol{T}$ means the left singular value matrix of $\boldsymbol{SENSE}$, and is a $m \times m$ orthogonal matrix, and represents *m*-dimensional semantic space of *m* keywords; $\boldsymbol{D}$ means the right singular value matrix of $\boldsymbol{SENSE}$, and is a $n \times n$ orthogonal matrix, and represents the coordinate positions of *n* senses in this semantic space; $\boldsymbol{S}$ means a diagonal matrix, and its diagonal elements are the singular values of $\boldsymbol{SENSE}$.

To remove the noise and simplify the matrix computation, only extract the top *k* maximum of singular values of $\boldsymbol{SENSE}$, namely the top *k* most important semantic structures of the keywords or the senses. And then, find the corresponding *k* order diagonal matrix in $\boldsymbol{S}$, and the corresponding *k* columns in $\boldsymbol{T}$, and the corresponding *k* rows in $\boldsymbol{D}^T$. And they are denoted respectively by $\boldsymbol{S}_k$, $\boldsymbol{T}_k$ and $\boldsymbol{D}_k^T$. After dimensionality reduced, the matrix $\boldsymbol{SENSE}$ is changed into its approximate matrix $\boldsymbol{SENSE}_k$. The equation of $\boldsymbol{SENSE}_k$ is:

$$\boldsymbol{SENSE}_k = \boldsymbol{T}_k \boldsymbol{S}_k \boldsymbol{D}_k^T. \tag{9}$$

Thereinto, in the metrix $\boldsymbol{SENSE}_k$, the original semantic relations between the keywords and the senses are retained and the noise arising from synonyms and polysemes are also removed.

Changing the original comment vector $\boldsymbol{comment}$ into the coordinate vector in the *k*-dimensional semantic space $\boldsymbol{d\_comment}$. The equation of $\boldsymbol{d\_comment}$ is:

$$\boldsymbol{d\_comment} = (\boldsymbol{comment}^T \boldsymbol{T}_k \boldsymbol{S}_k^{-1})^T = (dc_1, dc_2, ..., dc_k)^T \tag{10}$$

Thereinto, the elements in ***d_comment*** are the top *k* most important semantic dimensions of the vector ***comment*** .

(5) Calculating the sentence similarity between the comment of this ontology concept and the gloss of any of all WordNet senses of this concept

Exact the coordinate vector of the *j*-th senses of this concept in the *k*-dimensional semantic space $\boldsymbol{d}_j = (ds_1, ds_2, ..., ds_k)^T$. Calculate the cosine of the angle between ***d_comment*** and $\boldsymbol{d}_j$ as the similarity between them. The equation of the cosine of the angle is:

$$\cos(\boldsymbol{d\_comment}, \boldsymbol{d}_j) = \frac{\sum_{i=1}^{k}(dc_i \cdot ds_i)}{\sqrt{\sum_{i=1}^{k} dc_i^2} \cdot \sqrt{\sum_{i=1}^{k} ds_i^2}}, \quad (11)$$

thereinto, $j = 1, 2, ..., n$.

Thereinto, $\boldsymbol{D}_k^T = \boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_n$. If the cosine value between ***d_comment*** and a certain $\boldsymbol{d}_j$ is the largest in all *n* cosine values, the WordNet sense represented by this $\boldsymbol{d}_j$ is the sense the most similar to the comment and is the correct WordNet sense of this ontology concept.

In summary, in the sense determination method of this section, the comment of an ontology concept is seen as the conceptual context, for which the ambiguity of the concept name can be eliminated to some extent. In this paper, the syntactic analysis is added into LSA to reflect the contribution of the syntactic structure to the sentence similarity; And the keyword sets are expanded so that LSA becomes suitable for computing the similarity about short text or sentence. This sense determination method is more universal than the sense determination method referring to the true meaning of hypernym concept of an ontology concept in section IV. And there is no strict concept hierarchy demand in this sense determination method. However, the comments of the ontology concepts are created subjectively by domain experts, and have no synonym sets and hypernym sets in WordNet. It may result in a big difference between the keywords in the comment of a certain concept and the keywords in the WordNet sense of this concept, and it may reduce the precision of LSA. According to the above advantages and disadvantages, the sense determination method of this section is combined with that of section IV to determine the correct sense of concept.

VI. Concept Name Similarity Calculation

In this paper, there are two kinds of the concept name sense determination methods. One is based on the hypernym concept of this ontology concept, introduced in the section IV, and another is based on the comment of this concept, introduced in the section V.

It is taken for granted that concept name similarity equals the similarity between the correct senses of two concepts. So, the above two sense determination methods can generate two kinds of concept name similarities. This two kinds of concept name similarities are denoted respectively by $sim\_hyperHypo(C_1, C_2)$ and $sim\_commentSense(C_1, C_2)$; Lastly, the weighted sum of the two kinds of similarities is seen as the final concept name similarity. The equation of the final integrative concept name similarity is:

$$sim(C_1, C_2) = w\_hyperHypo \cdot sim\_hyperHypo(C_1, C_2) + w\_commentSense \cdot sim\_commentSense(C_1, C_2). \quad (12)$$

Thereinto, $w\_hyperHypo$ and $w\_commentSense$ are respectively the weights of this two kinds of similarities, in which $w\_hyperHypo + w\_commentSense = 1$.

In this paper, both of two kinds of the concept name sense determination methods are used, which not only combine the advantages of both and but also complement the disadvantages of both.

VII. Experiment and Result Evaluation

A. Experimental Data Sets and Criterion

In this paper, the benchmark tests from OAEI ontology matching campaign 2011 were used as the experimental data sets, because this benchmark tests is authoritative in the fields of ontology matching all over the world. The benchmark tests contain 110 ontologies, and there are about 36 concepts in every ontologies. Among them, the ontology #101 is the reference ontology in bibliographic domains. And the other ontologies are those whose linguistics or structure are changed from this reference ontology, named as variants. The concept name similarity calculation between these variants and this reference ontology were implemented. At last, the experimental results were evaluated according to OAEI criteria (*precision*, *recall* and *f-meature*). Thereinto, *precision* means the ratio of the number of the similar concept pairs discovered correctly to the number of all discovered similar concept pairs; *recall* means the ratio of the number of the similar concept pairs discovered correctly to the number of all truly similar concept pairs; $f\text{-}meature = 2pr/(p+r)$.

B. Experiment Design

In the existing algorithms for calculating the concept name similarity, there are four the most main and the most universal algorithms. The processes of the four algorithms are respectively:

(1) “MAX”: Semantic similarities between concepts are computed by WordNet, and the similarity between the most similar senses of two words, namely the maximum similarity between two words, is regarded as the word similarity [3].

(2) “HYPER_HYPO”: The correct WordNet sense of an ontology concept can be defined as that sense the most similar to the correct sense of the hypernym concept of this concept in all the senses of this concept [6]. Through this sense determination method, the correct WordNet

senses of two concepts are used in concept name similarity calculation.

(3) "LSA": By LSA, it is computed that the sentence similarity between the comment of an ontology concept and the gloss of each of all WordNet senses of this concept. The WordNet sense the most similar to the comment is seen as the correct WordNet sense of this concept [7]. Through this sense determination method, the correct WordNet senses of two concepts are used in concept name similarity calculation.

(4) "SYNTAX": The sentence is parsed by the syntactic parser into the syntactic tree including subject, predicate and object. It are calculated that the word similarities between the same grammatical constituents of two sentences. Finally, the weighted sum of the similarities of the three different grammatical constituents is seen as the final sentence similarity [13], [14], [15]. By the above the sentence similarity calculation method, it is computed that the sentence similarity between the comment of an ontology concept and the gloss of each of all WordNet senses of this concept. The WordNet sense the most similar to the comment is seen as the correct WordNet sense of this concept. Through this sense determination method, the correct WordNet senses of two concepts are used in concept name similarity calculation.

In order to evaluate our algorithms, the above four algorithms and our algorithm need to be compared mutually.

C. Experimental Result and Evaluation

The comparison result of the five algorithms is showed in Fig. 4:

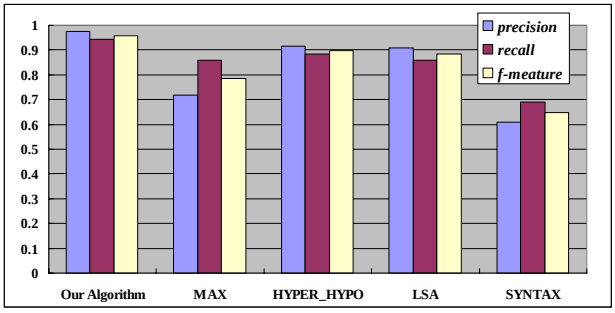

Figure 4. Comparison of the algorithms of concept name similarity calculation

(1) The *precision*, *recall* and *f-meature* of "SYNTAX" are all lower than 70%, and are the lowest in all algorithms.

(2) The result of "MAX" is better than "SYNTAX", but the *precision* of "MAX" is lower than 80%, and is far less than the *recall*.

(3) The *recall* of "HYPER_HYPO" and "LSA" are both a little larger than that of "MAX". But the *precision* of "HYPER_HYPO" and "LSA" are both higher than 90%, and are far higher than that of "MAX". It indicates that after the correct senses of concept names are determined, the *precision* of concept name similarity calculation can be increased dramatically on the basis of that the *recall* is not reduced. In addition, the *precision* and *recall* of "HYPER_HYPO" are both a little higher than that of "LSA", and the results of "HYPER_HYPO" and "LSA" are very similar to each other.

(4) The *precision*, *recall* and *f-meature* of our algorithm are respectively 97.41%, 94.32% and 95.84%, and are all far higher than that of "HYPER_HYPO" and "LSA", and are the highest in all algorithms. It indicates that the result of the algorithm referring to both of "HYPER_HYPO" and the improved "LSA" is more precise than the result of the algorithm only referring to "HYPER_HYPO" or "LSA". In other words, combining two kinds of conceptual context to determine the conceptual senses is better than only using one of the two kinds of conceptual context. Our algorithm is more universal than "HYPER_HYPO". The syntactic analysis is added into LSA in our algorithm, so our algorithm is more reasonable and more accurate than "LSA". The experimental results showed our algorithm outperformed other existing algorithms.

VIII. Conclusion

A new algorithm is presented in this paper: Firstly, the correct WordNet sense of an ontology concept can be defined as that sense the most similar to the correct sense of the hypernym concept of this concept in all the senses of this concept; Secondly, by LSA based on syntactic analysis, it is computed that the sentence similarity between the comment of this concept and the gloss of each of all WordNet senses of this concept. The WordNet sense the most similar to the comment is seen as the correct WordNet sense of this concept. The syntactic analysis is added into LSA to reflect the contribution of the syntactic structure to the sentence similarity. And the keyword sets are expanded so that LSA becomes suitable for computing the similarity about short text or sentence; Lastly, the above two sense determination methods can generate two kinds of concept name similarities. The weighted sum of the two kinds of similarities is seen as the final concept name similarity. The two kinds of the concept name sense determination methods are both used, which not only combine the advantages of both and but also complement the disadvantages of both. The experimental results showed that our algorithm is superior to other existing algorithms on the *precision* and *recall*. Ultimately, in our algorithm, the ambiguity of the concept name can be eliminated, and the accuracy rate of the concept name similarity calculation and ontology mapping can be increased dramatically.

Acknowledgment

This paper is supported by Beijing Natural Science Foundation (4123094). The Science and Technology Project of Beijing Municipal commission of Education (No.KM201110028020, KM201010028019). National Nature Science Foundation (No.60863011, 61175068, 61163022). This work is supported by Beijing Key Construction Discipline "Computer Application Technology".

REFERENCES

[1] Wei Gao, and Li Liang, "Improved ontology mapping method based on semantic similarity", *Journal of Gansu Lianhe University* (*Natural Sciences*), Vol. 23, No. 5, pp. 59-63, 2009.

[2] Yan Li, "Study of ontology mapping based on BP neural networks", Qingdao, Shandong: Shandong University of Science and Technology, 2008.

[3] Feng Yang, and Lei Liu, "A cooperative framework for effective ontology matching", *Information Engineering and Computer Science, ICIECS 2009 International Conference, Wuhan, Hubei*, pp. 1-4, 2009, doi: 10.1109/ICIECS.2009.5363494.

[4] Hiep Phuc Luong, Susan Gauch, and Mirco Speretta, "Enriching concept descriptions in an amphibian ontology with vocabulary extracted from wordnet", *Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium, Albuquerque, NM*, pp. 1-6, September 2009, doi: 10.1109/CBMS.2009.5255389.

[5] Chantal Reynaud, and Brigitte Safar, "Exploiting WordNet as background knowledge", *International Semantic Web Conference (ISWC), Busan, Korean*, pp. 271-275, September 2007.

[6] Xi Li, and Dezhi Xu, "The research of name strategy based on WordNet in ontology matching", *Computing Technology and Automation*, Vol. 128, No. 12, pp. 54-58, June 2009.

[7] Guangjun Huang, Jianguo Sun, and Junli Luo, "Concept name similarity algorithm based on Latent Semantic Analysis", *Computer Engineering*, Vol. 35, No. 14, pp. 69-74, July 2009.

[8] Xu Liang, and Dongjiao Wang , and Ming Huang, "Improved sentence similarity algorithm based on VSM and its application in Question Answering System", *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference, Xiamen, Fujian*, pp. 368-371, December 2010, doi: 10.1109/ICICISYS.2010.5658525.

[9] Shady Shehata. "A WordNet-based semantic model for enhancing text clustering", *Data Mining Workshops, 2009. ICDMW '09. IEEE International Conference, Miami, Florida*, pp. 477–482, December 2009, doi: 10.1109/ICDMW.2009.86.

[10] C.T. Meadow, B.R. Boyce, and D.H. Kraft, *Text Information Retrieval Systems*, second ed., London: Academic Press, 2000.

[11] P.W. Foltz, W. Kintsch, and T.K. Landauer, "The measurement of textual coherence with Latent Semantic Analysis", *Discourse Processes*, Vol. 25, Nos. 2-3, pp. 285-307, 1998.

[12] Mingche Lee, Jiawei Zhang, Wenxiang Lee, and Hengyu Ye, "Sentence similarity computation based on POS and semantic nets", *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference, Seoul, Korean*, pp. 907–912, November 2009, doi: 10.1109/NCM.2009.379.

[13] Hongsheng Wang, Lu Jing, and Hong Shao, "Research on method of sentence similarity based on ontology", *Intelligent Systems, 2009. GCIS '09. WRI Global Congress, Xiamen, Fujian*, pp. 465–469, August 2009, doi: 10.1109/GCIS.2009.86.

[14] Jianfang Shan, Zongtian Liu, and Wen Zhou. "Sentence similarity measure based on events and content words", *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference, Tianjin*, pp. 623–627, December 2009, doi: 10.1109/FSKD.2009.926.

[15] Wen Zhou, "Research and implementation of the English-Chinese translation memory system based on syntax and semantic", Changsha, Hunan: Hunan University, 2007.

[16] Xiang'en Hu, and Zhiqiang Cai, "A modified weight function in Latent Semantic Analysis", *Journal of Chinese Information Processing*, Vol. 119, No. 16, pp. 64-69, 2005.

[17] Yongping Chen, Sichun Yang, Xin Su, and Wansheng Mao, "Answer extraction based on weighted Latent Semantic Analysis", *Computer Systems & Applications*, Vol. 21, No. 1, pp. 40-44, 2012.

Yun Ma is a Master degree candidate whose specialty is Application of Computer at College of Information and Engineering, Capital Normal University, Beijing, P. R. China. She received the degree of Bachelor of Management Studies in Information Management and System (E-commerce) at Capital Normal University. Her main research interests are ontology application and natural language processing.

She has two published articles. One is "Chinese fuzzy ontology mapping based on support vector machine" published in China Communications, and can be retrieved in SCI. Another is "The study of semi-supervised learning algorithm for web information classification" published in Journal of Information and Computational Science, and can be retrieved in EI. She has one accepted paper "Reuse of Chinese Domain Ontology for the restricted domain Question Answering System", and this paper will be published in Journal of Computers. Her current research interests are the uncertain ontology mapping.

Jie Liu is an associate professor at College of Information and Engineering, Capital Normal University, Beijing, P. R. China. He received PH.D degree in Computer Application Technology at Beijing Institute of Technology. His main research interests are the restricted domain Question Answering System and ontology application. *The corresponding author. Email: liujxxxy@126.com.

Zhengtao Yu received his Ph.D. degree in computer application technology from Beijing Institute of Technology, Beijing, China, in 2005. He is currently a professor in the School of Information Engineering and Automation, Kunming University of Science and Technology, China. His main research interests include natural language process, Chinese question answering system and machine learning.