Semantic disambiguation in Automatic Semantic Annotation

Xin Qi, Min Xiao
College of Computer Science and Technology
Wuhan University of Technology
Wuhan, China
qixin.whut@gmail.com

***Abstract*—In order to generate the metadata of semantic web, semantic information need be extracted from web documents. Facing the mass scale of web documents, Compared to artificial or semi-automatic semantic annotation, automatic semantic annotation is a feasible method. To recognize candidate named entities, the semantic dictionary is designed and semantic distance between entities is calculated by semantic relevance path. The most complex problem in semantic annotation is semantic disambiguation. A semantic disambiguation method based on the shortest path and n-gram is proposed. Experiments have been made on a news corpus. The result shows that the method is effective for the task of automatic semantic annotation.**

Keywords-semantic annotation; n-gram; semantic disambiguation; directed acyclic graph; knowledge base

I. INTRODUCTION

In order to realize more efficient management and access to web content, the semantic web adds the formal information and semantics to web content. The realization of the semantic web vision depends on massive metadata [1], and how to obtain the metadata is an important challenge. Billions of existing web pages are artificial semantic annotation that is obviously not a feasible method, so automatic semantic annotation gradually becomes a focused research problem.

Semantic annotation discussed in this paper is actually generating metadata layer of semantic web. Only based on the metadata layer, all the semantic web application can become a reality.

Definition 1: Semantic Annotation is mapping the knowledge base (KB) and the document repository to annotation results, recorded as $\delta ds \times kb \to \left\{ \prec_{doc_i}^{ent_m} \right\}$, and $ds = \left\{ doc_1, doc_2, .., doc_i, ..., doc_m \right\}$ means the document repository, doc_i means the i-th document in document repository, m means the quantity of documents, and $1 \le i \le m$; $kb = \left\{ ent_1, ent_2, ..., ent_m, ..., ent_k \right\}$ means entity set for annotating documents, k means the quantity of entities, and $1 \le m \le k$.

Currently, research methods of automatic semantic annotation contain: predefined rules, machine learning, classification, sequence model, model of subject-predicate-object composition, ontology, etc. Typical system that used ontology for semantic annotation is SemTag [2], which recognized candidate entity by TAP knowledge base, then generated two text vectors for context(10 words around keyword) of keyword and context of candidate entity. At last, the most matched entity is selected by calculating similarity between two vectors. Automatic semantic annotation in this paper is also based on ontology and knowledge, but in the case of matches, it not only considers the context window, but also considers the whole semantic structure of document.

Automatic semantic annotation can be regarded as the composite process of traditional Named Entity Recognition (NER) and Entity Annotation. Entity type in NER must be several common types, for example, Organization, Person, Location, Date, Money, etc. The main problem in traditional Named Entity Recognition is annotations are not encoded in an open and formal system, and entity types are not restricted, and the language resource for recognition is a private form and has no explicit semantic. It is an obstacle to reuse of language resource and annotation results in different systems.

These problems can be solved by information extracting infrastructure based on ontology and knowledge base [3]. Recognized type should be defined in ontology; recognized entity should be described in knowledge base. Due to the recognized entities are identified by URI in KB, the results of semantic annotation can be reused by semantic index and semantic retrieval.

II. SEMANTIC DICTIONARY

NER based on ontology and KB needs a semantic dictionary to realize mapping keyword to semantic object.

In the process of matching keyword to concrete ontology concept or instance, semantic dictionary plays an important role.

Logical construction of semantic dictionary is shown in Figure 1. For a term which is generated by Chinese Tokenizer, the first step is checking part-of-speech. If the term is a noun or noun phrase, find it appeared in which the attribute value of semantic objects. So the term is matched to these semantic objects, and these semantic objects are candidate entities of semantic disambiguation.

978-1-4244-9945-8/10/$26.00 ©2010 IEEE

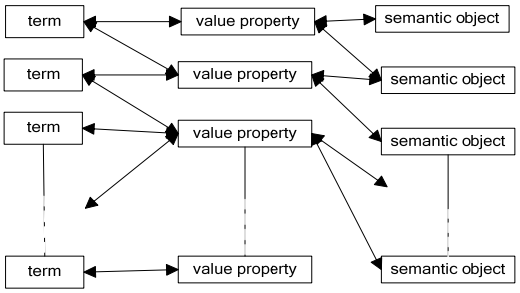

Figure 1. Mapping between terms and semantic objects

The realization in this paper is using Jena (a RDF parser) to get ub:name, rdfs:label, rdfs:comment of property value of all the entities in KB. The tuples of <term, value property, semantic object> are used to construct semantic dictionary.

III. SEMANTIC DISAMBIGUATION

According to the logic structure of semantic dictionary, it is found that some terms could refer to multiple semantic objects. So in the process of semantic annotation, a complicated problem is selecting only one semantic object from multiple candidate ones. The process is called semantic disambiguation.

When a new document need be processed, the first step is getting pure text (title and main body) by preprocessing the document. Because of the index is based on the content of the document, nouns and noun phrases should be extracted from title and main body. So get N orderly arrangement of Terms, recorded as $T_1,...,T_n$, it is known that one Term could have more than one interpretation by semantic dictionary, namely, one Term could refer to multiple semantic objects according to different contexts. As for the Term "liuwei" refers to the party secretary of WHUT, or refers to one of the teacher in College of Computer Science and Technology. By using the function $SemanticObject(T_i)=\{SO_{i1},SO_{i2},.....,SO_{ip_i}\}$ defined in semantic dictionary, it is found that Term T_i could refer to P_i semantic objects. But for the document, every Term only refers to one semantic object, namely, finding solution $[SO_1',...,SO_n']$ for vector $[T_1,...,T_n]$. The quantity of candidate solutions is $P_1\times P_2\times.....\times P_n$.

Before solving process, it can be thought the correct solution should in the same field of knowledge, or in a field of knowledge as far as possible. From the geometric perspective, the solution of $T_1,...,T_n$ should be:

1. They located in the same field of knowledge or as soon as possible.

2. In the field of knowledge of the RDF digraph, distances between they are close as possible.

Therefore, this paper proposes a semantic disambiguation method based on semantic distance of semantic objects, as to satisfy the above results.

A. Semantic Distance

Relationship between entities is a path of edges and nodes in KB. If there is a property sequence between entity E_1 and entity E_n, E_1 and E_n is path relevance. The relationship is $R=\{E_1,P_1,E_2,P_2,E_3,...,E_{n-1},P_{n-1},E_n\}$, E means entity, and P means property between two entities.

Definition 2: Semantic Distance between two entities refers to the length of semantic association path, recorded as L_R.

For $R=\{E_1,P_1,E_2,P_2,E_3,...,E_{n-1},P_{n-1},E_n\}$, the length of semantic association path between E_1 and E_n is n-1. Normally, the shorter path connection, the degree of relevance between two entities is higher.

B. Shortes Path Problem

This paper calculates semantic distance between two entities in KB based on the calculation of the shortest path algorithm in graph theory.

In order to reach the above objectives, the first step is to extract all triples in KB, then store the triples whose subject and object are entities in adjacency matrix. The adjacency matrix expresses the graph of all direct associations in KB. In order to facilitate the calculation, it is assumed that the weight of each edge in the graph is 1, and the directions of the edges are ignored.

Finding the length of shortest path of each node pair is called all-pair shortest-paths in graph theory. More specifically, for the graph G= (V, E) and u, v ∈ V, the minimum value of d(u, v) is calculated.

Dijkstra algorithm is a typical algorithm of Single-Source Shortest-Paths Problem [4]. The shortest path and the length of the shortest path between each two nodes can be calculated by Dijkstra algorithm. Using Dijkstra algorithm for $|V|$ times, the program traverses all nodes in the RDF graph to build MiniLengthMatrix which records the length of the shortest path. MinLengthMatrix will be used in semantic disambiguation.

C. Semantic Disambiguation based on shortes- path

Basic idea of semantic disambiguation based on shortest path is making candidate entities as nodes of annotation directed acyclic graph (DAG). Those semantic objects whose semantic relevance degree is high are the result of semantic annotation. Selection of the shortest path in annotation DAG as semantic annotation results that is reasonable.

According to function $SemanticObject(T_i)=\{SO_{i1},SO_{i2},.....,SO_{ip_i}\}$ defined in semantic dictionary, Term T_i could refer to P_i semantic objects. An annotation DAG is built to semantic disambiguation. The nodes of edge in the DAG are semantic object referred by T_{i-1} and semantic object referred by T_i. The weight of edge is the length of the shortest path between

T_{i-1} and T_i in RDF DAG. MinLengthMatrix which is introduced in last section provides the value of length.

The quantity of nodes in annotation DAG G is $\sum_{i=1}^{n} P_i$. If Path(i , j) means path set in which every one is the path from Node v_i to Node v_j, then the shortest path in Path(1,n) is the result of semantic disambiguation. This paper adopted Dijkstra algorithm to find the shortest path from head node to tail node.

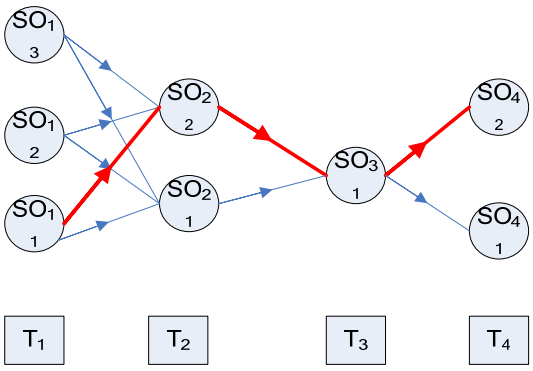

Figure 2. Schematic diagram of semantic disambiguation based on the shortest path

As Figure 2 shows, there is a vector of 4 terms $[T_1, T_2, T_3, T_4]$, T_1 has 3 candidate semantic objects, T_2 and T_4 have 2 candidate semantic object, T_3 has one candidate semantic object. All candidate semantic objects constitute the nodes of annotation DAG, the highlight path is the shortest path by Dijkstra algorithm, so semantic disambiguation result is $[SO_{11}, SO_{22}, SO_{31}, SO_{42}]$.

There is another problem to solve. Not all two semantic objects have semantic association path between them. If two candidate semantic objects have no semantic association path, the weight of the edge between them is maximum value in MinLengthMatrix add 1. So, it could guarantee that the shortest path in annotation DAG can be found.

D. Semantic Disambiguation based on n-gram

If you already have the large-scale corpus which has be semantic annotated, another semantic disambiguation method based on n-gram can get higher accuracy.

The effect of n-gram is predicting the probability of an entity sequence [5]. N-gram assumes the probability of an entity appearance is related to n-1 entities ahead in document, not related to entities earlier. In order to describe the probability distribution, an n-dimensional array is needed. The length of every dimensional of the array is the quantity of entities in KB, the quantity of elements in the array is m^n, an element $a_{i_1 i_2 \ldots i_n}$ means the probability of e_i appearing after $e_{i_1} e_{i_2} \ldots e_{i_{n-1}}$, recorded as $p(e_{i_n} \mid e_{i_1} e_{i_2} \ldots e_{i_{n-1}})$.

Semantic disambiguation is also finding out a shortest path in DAG; however the weight of every edge is not semantic distance between two entities once more, but the probability from one to another. By n-gram, the probability of every path is calculated by (1).

$$p(e_1 e_2 \ldots e_l) = p(e_1 \ldots e_{n-1}) \prod_{i=n}^{l} p(e_i \mid e_{i-1} \ldots e_{i-n+1}) \quad (1)$$

In (1), if $n \geq 3$, the current entity is related to n-1 entities ahead. So the probability of n entities appear in the same document must be get. However, On the basis of the current scale of corpus, there is a data sparseness problem, and performance could be influenced because of waste of space. So this paper adopted bi-gram model to semantic disambiguation. The model need calculate the probability of two entities appear in the same document, and assumes every entity is related to one entity ahead.

Assuming $E = e_1 e_2 \ldots e_n$ is the disambiguation result of term sequence $T = T_1 T_2 \ldots T_n$, e_i is one entity of E, e_i is only related to e_{i-1}, so:

$$P(E) = P(e_1, e_2, \ldots, e_n) = P(e_1) P(e_2 \mid e_1) \ldots P(e_n \mid e_{n-1}) \quad (2)$$

If assuming there is an e_0, a start tag, (2) can be simplified to (3):

$$P(E) = \prod_{i=1}^{n} P(e_i \mid e_{i-1}) \quad (3)$$

In (3), $P(e_i \mid e_{i-1})$ means the probability from e_{i-1} to e_i, the probability is conditional probability of two entities occurring in the same document. On the basis of large-scale corpus, according to law of large number, with the premise of large-scale sample statistic, sample frequency is close to its probability value. So the maximum likelihood estimator of $P(e_i \mid e_{i-1})$ can be calculated by $P(e_i \mid e_{i-1}) = N(e_{i-1}, e_i) / N(e_i)$, $N(e_{i-1}, e_i)$ means co-occurrence frequency of two entities, $N(e_i)$ means occurrence frequency of e_i.

If two entities (e_{i-1}, e_i) don't occur in a document, then $N(e_{i-1}, e_i) = 0$, so there is a data sparse problem. Because it couldn't enlarge unlimitedly the scale of corpus to calculate co-occurrence frequency of all entity-pair, so some appropriate strategies need to take. This paper adopted Jelinek-Mercer data smoothing strategy [6] to solve data sparse problem, the data smoothing technology mainly use linear interpolation to get high rank model from low rank model. If no enough data to estimate probability of high rank model, low rank model may provide useful information. When calculating $p(e_{i-1}, e_i)$, the value couldn't be get from training corpus, the occurrence probability of current entity e_i can take the place of $p(e_{i-1}, e_i)$, and interpolation formula is:

$$p'(e_{i-1}, e_i) = \lambda p(e_{i-1}, e_i) + (1 - \lambda) p(e_i) \quad (4)$$

In (4), $p'(e_{i-1}, e_i)$ is an estimated value of $p(e_{i-1}, e_i)$, λ is a real number between 0 and 1, the value of λ can be get by experiment or statistics. $p(e_i)$ means occurrence frequency of entity e_i, $p(e_i) \approx k_i / \sum_{j=0}^{n} k_j$, k_i is occurrence frequency

of entity e_i in training corpus. From (4), when λ is 1, this is a pure bi-gram model; when λ is 0, this degrade into one-gram model. By continually multiply of $p'(e_{i-1}, e_i)$, the P (E) value of every possible path could be calculated. At last, the path whose probability is maximum value is regard as the result of semantic disambiguation.

IV. EXPERIMENTS AND DISCUSSION

A test dataset which consists of campus news in Wuhan University of Technology (WHUT) was constructed to test the proposed method of automatic semantic annotation. Ontology in the test dataset is lightweight ontology in LUMB (Lehigh University Benchmark) benchmark, called Univ-Bench. A lot of course information, teacher information, and organization information was collected artificially for pre-population of KB. The knowledge base contains 7633 entities and 23412 properties instance, and these knowledge have been manually reviewed. 11125 web pages were collected by using web crawler in campus intranet construct a web page repository for test.

The program adopting the method is introduced in this paper semantic annotated the web page repository. 63928 entities were annotated in the repository; there are 5.75 entities per web page.

In the process of semantic annotation, the program logs disambiguation Log to test the accuracy of semantic disambiguation. By analyzing logs, there are 2753 disambiguation operations. All disambiguation were judged artificially that is impractical, so test program selected 100 disambiguation logs at random for artificial judgment. We performed three experiments to calculate and analyze the two semantic disambiguation methods.

TABLE I. DATA IN THREE EXPERIMENTS

Semantic Disambiguation Method	***Experiment 1***	***Experiment 2***	***Experiment 3***
Based on shortest-path	76	81	85
Based on n-gram	83	88	91

By the experimental data from Table 1, it can be found the two semantic disambiguation methods have satisfactory accuracy rate. When no large-scale annotation corpus, automatic semantic annotation based on the shortest path may be choose. When training corpus has high level in quantity and quality, automatic semantic annotation based on n-gram will be better.

V. CONCLUSION

Automatic semantic annotation is a complex problem for development of semantic web; nevertheless the most complex problem in automatic semantic annotation is semantic disambiguation. This paper proposed a semantic disambiguation method based on shortest-path and a semantic disambiguation method based on n-gram. The experiments show that both methods can effectively semantic annotate web document automatically.

REFERENCES

[1] Berners-Lee, T., J. Hendler, and O. Lassila, The semantic web. Scientific american, 2001. 284(5): p. 28-37.

[2] Dill, S. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. in WWW. 2003.

[3] Kiryakov, A., et al., Semantic Annotation, Indexing, and Retrieval. The SemanticWeb-ISWC 2003, 2003: p. 484-499.

[4] Shaffer, C., A Practical Introduction to Data Structures and Algorithm Analysis, Java Edition. Prentice Hall, 1998. 131: p. 132-134.

[5] ZHANG Hua-ping, LIU Qun, Model of Chinese Words Rough Segmentation Based on N-Shortest-Paths Method. Journal of Chinese Information Processing, 2002. 16(005): p. 1-7.

[6] Xu Zhiming, et al., Data Smoothing Technology in N-gram Language Model. Application Research of Computers, 1999. (7):p37-40