Structural Semantic Relatedness: A Knowledge-Based Method to Named Entity Disambiguation

Xianpei Han **Jun Zhao***

National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing 100190, China
{xphan,jzhao}@nlpr.ia.ac.cn

Abstract

Name ambiguity problem has raised urgent demands for efficient, high-quality named entity disambiguation methods. In recent years, the increasing availability of large-scale, rich semantic knowledge sources (such as *Wikipedia* and *WordNet*) creates new opportunities to enhance the named entity disambiguation by developing algorithms which can exploit these knowledge sources at best. The problem is that these knowledge sources are heterogeneous and most of the semantic knowledge within them is embedded in complex structures, such as graphs and networks. This paper proposes a knowledge-based method, called *Structural Semantic Relatedness* (*SSR*), which can enhance the named entity disambiguation by capturing and leveraging the structural semantic knowledge in multiple knowledge sources. Empirical results show that, in comparison with the classical BOW based methods and social network based methods, our method can significantly improve the disambiguation performance by respectively 8.7% and 14.7%.

1 Introduction

Name ambiguity problem is common on the Web. For example, the name "Michael Jordan" represents more than ten persons in the Google search results. Some of them are shown below:

Michael (Jeffrey) Jordan, Basketball Player
Michael (I.) Jordan, Professor of Berkeley
Michael (B.) Jordan, American Actor

The name ambiguity has raised serious problems in many relevant areas, such as web person search, data integration, link analysis and knowledge base population. For example, in response to a person query, search engine returns a long, flat list of results containing web pages about several namesakes. The users are then forced either to refine their query by adding terms, or to browse through the search results to find the person they are seeking. Besides, an ever-increasing number of question answering and information extraction systems are coming to rely on data from multi-sources, where name ambiguity will lead to wrong answers and poor results. For example, in order to extract the birth date of the Berkeley professor *Michael Jordan*, a system may return the birth date of his popular namesakes, e.g., the basketball player *Michael Jordan*.

So there is an urgent demand for efficient, high-quality named entity disambiguation methods. Currently, the common methods for named entity disambiguation include name observation clustering (Bagga and Baldwin, 1998) and entity linking with knowledge base (McNamee and Dang, 2009). In this paper, we focus on the method of name observation clustering. Given a set of observations $O = \{o_1, o_2, \ldots, o_n\}$ of the target name to be disambiguated, a named entity disambiguation system should group them into a set of clusters $C = \{c_1, c_2, \ldots, c_m\}$, with each resulting cluster corresponding to one specific entity. For example, consider the following four observations of *Michael Jordan*:

1) ***Michael Jordan*** *is a researcher in Computer Science.*
2) ***Michael Jordan*** *plays basketball in Chicago Bulls.*
3) ***Michael Jordan*** *wins NBA MVP.*
4) *Learning in Graphical Models:* ***Michael Jordan****.*

A named entity disambiguation system should group the 1st and 4th *Michael Jordan* observations into one cluster for they both refer to the Berke-

* Corresponding author

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 50–59,
Uppsala, Sweden, 11-16 July 2010.

ley professor *Michael Jordan*, meanwhile group the other two *Michael Jordan* into another cluster as they refer to another person, the Basketball Player *Michael Jordan*.

To a human, named entity disambiguation is usually not a difficult task as he can make decisions depending on not only contextual clues, but also the prior background knowledge. For example, as shown in Figure 1, with the background knowledge that both *Learning* and *Graphical models* are the topics related to *Machine learning*, while *Machine learning* is the sub domain of *Computer science*, a human can easily determine that the two *Michael Jordan* in the 1st and 4th observations represent the same person. In the same way, a human can also easily identify that the two *Michael Jordan* in the 2nd and 3rd observations represent the same person.

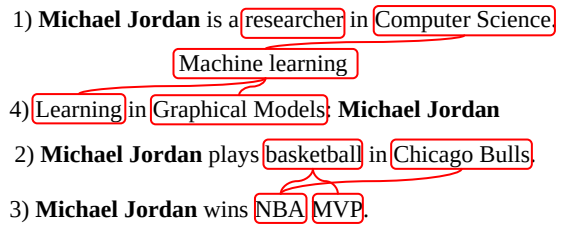

Figure 1. The exploitation of knowledge in human named entity disambiguation

The development of systems which could replicate the human disambiguation ability, however, is not a trivial task because it is difficult to capture and leverage the semantic knowledge as humankind. Conventionally, the named entity disambiguation methods measure the similarity between name observations using the *bag of words* (BOW) model (Bagga and Baldwin (1998); Mann and Yarowsky (2006); Fleischman and Hovy (2004); Pedersen et al. (2005)), where a name observation is represented as a feature vector consisting of the contextual terms. This model measures similarity based on only the co-occurrence statistics of terms, without considering all the semantic relations like social relatedness between named entities, associative relatedness between concepts, and lexical relatedness (e.g., acronyms, synonyms) between key terms.

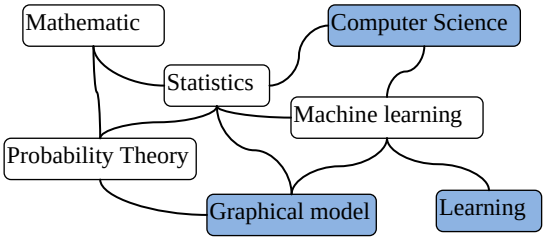

Figure 2. Part of the link structure of Wikipedia

Fortunately, in recent years, due to the evolution of Web (e.g., the *Web 2.0* and the *Semantic Web*) and many research efforts for the construction of knowledge bases, there is an increasing availability of large-scale knowledge sources, such as *Wikipedia* and *WordNet*. These large-scale knowledge sources create new opportunities for knowledge-based named entity disambiguation methods as they contain rich semantic knowledge. For example, as shown in Figure 2, the link structure of Wikipedia contains rich semantic relations between concepts. And we believe that the disambiguation performance can be greatly improved by designing algorithms which can exploit these knowledge sources at best.

The problem of these knowledge sources is that they are heterogeneous (e.g., they contain different types of semantic relations and different types of concepts) and most of the semantic knowledge within them is embedded in complex structures, such as graphs and networks. For example, as shown in Figure 2, the semantic relation between *Graphical Model* and *Computer Science* is embedded in the link structure of the Wikipedia. In recent years, some research has investigated to exploit some specific semantic knowledge, such as the social connection between named entities in the Web (Kalashnikov et al. (2008), Wan et al. (2005) and Lu et al. (2007)), the ontology connection in DBLP (Hassell et al., 2006) and the semantic relations in Wikipedia (Cucerzan (2007), Han and Zhao (2009)). These knowledge-based methods, however, usually are specialized to the knowledge sources they used, so they often have the knowledge coverage problem. Furthermore, these methods can only exploit the semantic knowledge to a limited extent because they cannot take the structural semantic knowledge into consideration.

To overcome the deficiencies of previous methods, this paper proposes a knowledge-based method, called *Structural Semantic Relatedness* (*SSR*), which can enhance the named entity disambiguation by capturing and leveraging the structural semantic knowledge from multiple knowledge sources. The key point of our method is a reliable semantic relatedness measure between concepts (including WordNet concepts, NEs and Wikipedia concepts), called *Structural Semantic Relatedness*, which can capture both the explicit semantic relations between concepts and the implicit semantic knowledge embedded in graphs and networks. In particular, we first extract the semantic relations between two concepts from a variety of knowledge sources and

represent them using a graph-based model, *semantic-graph*. Then based on the principle that "two concepts are semantic related if they are both semantic related to the neighbor concepts of each other", we construct our *Structural Semantic Relatedness* measure. In the end, we leverage the structural semantic relatedness measure for named entity disambiguation and evaluate the performance on the standard WePS data sets. The experimental results show that our *SSR* method can significantly outperform the traditional methods.

This paper is organized as follows. Section 2 describes how to construct the structural semantic relatedness measure. Next in Section 3 we describe how to leverage the captured knowledge for named entity disambiguation. Experimental results are demonstrated in Sections 4. Section 5 briefly reviews the related work. Section 6 concludes this paper and discusses the future work.

2 The Structural Semantic Relatedness Measure

In this section, we demonstrate the structural semantic relatedness measure, which can capture the structural semantic knowledge in multiple knowledge sources. Totally, there are two problems we need to address:

1) How to extract and represent the semantic relations between concepts, since there are many types of semantic relations and they may exist as different patterns (the semantic knowledge may exist as explicit semantic relations or be embedded in complex structures).

2) How to capture all the extracted semantic relations between concepts in our semantic relatedness measure.

To address the above two problems, in following we first introduce how to extract the semantic relations from multiple knowledge sources; then we represent the extracted semantic relations using the semantic-graph model; finally we build our structural semantic relatedness measure.

2.1 Knowledge Sources

We extract three types of semantic relations (semantic relatedness between Wikipedia concepts, lexical relatedness between WordNet concepts and social relatedness between NEs) correspondingly from three knowledge sources: Wikipedia, WordNet and NE Co-occurrence Corpus.

1. **Wikipedia**[1], a large-scale online encyclopedia, its English version includes more than 3,000,000 concepts and new articles are added quickly and up-to-date. Wikipedia contains rich semantic knowledge in the form of hyperlinks between Wikipedia articles, such as *Polysemy* (disambiguation pages), *Synonym* (redirect pages) and *Associative relation* (hyperlinks between Wikipedia articles). In this paper, we extract the semantic relatedness *sr* between Wikipedia concepts using the method described in Milne and Witten(2008):

$$sr(a,b) = 1 - \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))}$$

where *a* and *b* are the two concepts of interest, *A* and *B* are the sets of all the concepts that are respectively linked to *a* and *b*, and *W* is the entire Wikipedia. For demonstration, we show the semantic relatedness between four selected concepts in Table 1.

	Statistics	**Basketball**
Machine learning	0.58	0.00
MVP	0.00	0.45

Table 1. The semantic relatedness table of four selected Wikipedia concepts

2. **WordNet 3.0**[2] (Fellbaum et al., 1998), a lexical knowledge source includes over 110,000 WordNet concepts (word senses about English words). Various lexical relations are recorded between WordNet concepts, such as *hyponyms*, *holonym* and *synonym*. The lexical relatedness *lr* between two WordNet concepts are measured using the Lin (1998)'s WordNet semantic similarity measure. Table 2 shows some examples of the lexical relatedness.

	school	**science**
university	0.67	0.10
research	0.54	0.39

Table 2. The lexical relatedness table of four selected WordNet concepts

3. **NE Co-occurrence Corpus**, a corpus of documents for capturing the social relatedness between named entities. According to the fuzzy set theory (Baeza-Yates et al., 1999), the degree of named entities co-occurrence in a corpus is a measure of the relatedness between them. For example, in Google search results, the "Chicago Bulls" co-occurs with "NBA" in more than

[1] http://www.wikipedia.org/
[2] http:// wordnet.princeton.edu/

7,900,000 web pages, while only co-occurs with "EMNLP" in less than *1,000* web pages. So the co-occurrence statistics can be used to measure the social relatedness between named entities. In this paper, given a NE Co-occurrence Corpus *D*, the social relatedness *scr* between two named entities ne_1 and ne_2 is measured using the Google Similarity Distance (Cilibrasi and Vitanyi, 2007):

$$scr(ne_1, ne_2) = 1 - \frac{\log(\max(|D_1|, |D_2|)) - \log(|D_1 \cap D_2|)}{\log(|D|) - \log(\min(|D_1|, |D_2|))}$$

where D_1 and D_2 are the document sets correspondingly containing ne_1 and ne_2. An example of social relatedness is shown in Table 3, which is computed using the Web corpus through Google.

	ACL	NBA
EMNLP	0.61	0.00
Chicago Bulls	0.19	0.55

Table 3. The social relatedness table of four selected named entities

2.2 The Semantic-Graph Model

In this section we present a graph-based representation, called *semantic-graph*, to model the extracted semantic relations as a graph within which the semantic relations are interconnected and transitive. Concretely, the semantic-graph is defined as follows:

*A **semantic-graph** is a weighted graph **G = (V, E)**, where each node represents a distinct concept; and each edge between a pair of nodes represents the semantic relation between the two concepts corresponding to these nodes, with the edge weight indicating the strength of the semantic relation.*

For demonstration, Figure 3 shows a semantic-graph which models the semantic knowledge extracted from Wikipedia for the *Michael Jordan* observations in Section 1.

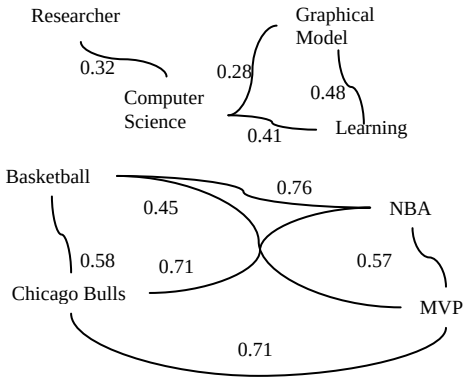

Figure 3. An example of semantic-graph

Given a set of name observations, the construction of semantic-graph takes two steps: concept extraction and concept connection. In the following we respectively describe each step.

1) Concept Extraction. In this step we extract all the concepts in the contexts of name observations and represent them as the nodes in the semantic-graph. We first gather all the N-grams (up to 8 words) and identify whether they correspond to semantically meaningful concepts: if a N-gram is contained in the WordNet, we identify it as a WordNet concept, and use its primary word sense as its semantic meaning; to find whether a N-gram is a named entity, we match it to the named entity list extracted using the open-Calais API3, which contains more than 30 types of named entities, such as Person, Organization and Award; to find whether a N-gram is a Wikipedia concept, we match it to the Wikipedia anchor dictionary, then find its corresponding Wikipedia concept using the method described in (Medelyan et al, 2008). After concept identification, we filter out all the N-grams which do not correspond to the semantic meaningful concepts, such as the N-grams "*learning in*" and "*wins NBA MVP*". The retained N-grams are identified as concepts, corresponding with their semantic meanings (a concept may have multiple semantic meaning explanation, e.g., the "*MVP*" has three semantic meaning, as "*most valuable player, MVP*" in WordNet, as the "*Most Valuable Player*" in Wikipedia and as a named entity of Award type).

2) Concept Connection. In this step we represent the semantic relations as the edges between nodes. That is, for each pair of extracted concepts, we identify whether there are semantic relations between them: 1) If there is only one semantic relation between them, we connect these two concepts with an edge, where the edge weight is the strength of the semantic relation; 2) If there is more than one semantic relations between them, we choose the most reliable semantic relation, i.e., we choose the semantic relation in the knowledge sources according to the order of WordNet, Wikipedia and NE Co-concurrence corpus (Suchanek et al., 2007). For example, if both Wikipedia and WordNet provide the semantic relation between *MVP* and *NBA*, we choose the semantic relation provided by WordNet.

[3] http://www.opencalais.com/

2.3 The Structural Semantic Relatedness Measure

In this section, we describe how to capture the semantic relations between the concepts in semantic-graph using a semantic relatedness measure. Totally, the semantic knowledge between concepts is modeled in two forms:

1) The edges of semantic-graph. The edges model the direct semantic relations between concepts. We call this form of semantic knowledge as *explicit semantic knowledge*.

2) The structure of semantic-graph. Except for the edges, the structure of the semantic-graph also models the semantic knowledge of concepts. For example, the neighbors of a concept represent all the concepts which are explicitly semantic-related to this concept; and the paths between two concepts represent all the explicit and implicit semantic relations between them. We call this form of semantic knowledge as *structural semantic knowledge*, or *implicit semantic knowledge*.

Therefore, in order to deduce a reliable semantic relatedness measure, we must take both the edges and the structure of semantic-graph into consideration. Under the semantic-graph model, the measurement of semantic relatedness between concepts equals to quantifying the similarity between nodes in a weighted graph. To simplify the description, we assign each node in semantic-graph an integer index from 1 to $|\boldsymbol{V}|$ and use this index to represent the node, then we can write the adjacency matrix of the semantic-graph $\boldsymbol{G}$ as $\boldsymbol{A}$, where $\boldsymbol{A[i,j]}$ or $\boldsymbol{A_{ij}}$ is the edge weight between node *i* and node *j*.

The problem of quantifying the relatedness between nodes in a graph is not a new problem, e.g., the *structural equivalence* and *structural similarity* (the SimRank in Jeh and Widom (2002) and the similarity measure in Leicht et al. (2006)). However, these similarity measures are not suitable for our task, because all of them assume that the edges are uniform so that they cannot take edge weight into consideration.

In order to take both the graph structure and the edge weight into account, we design the structural semantic relatedness measure by extending the measure introduced in Leicht et al. (2006). The fundamental principle behind our measure is "a node *u* is semantically related to another node *v* if its immediate neighbors are semantically related to *v*". This definition is natural, for example, as shown in Figure 3, the concept *Basketball* and its neighbors *NBA* and *Chicago Bulls* are all semantically related to *MVP*. This definition is recursive, and the starting point we choose is the semantic relatedness in the edge. Thus our structural semantic relatedness has two components: the neighbor term of the previous recursive phase which captures the graph structure and the semantic relatedness which captures the edge information. Thus, the recursive form of the structural semantic relatedness S_{ij} between the node *i* and the node *j* can be written as:

$$S_{ij} = \lambda \sum_{l \in N_i} \frac{A_{il}}{d_i} S_{lj} + \mu A_{ij}$$

where λ and μ control the relative importance of the two components and

$N_i=\{j \mid A_{ij} > 0\}$ is the set of the immediate neighbors of node *i*;

$d_i = \sum_{j \in N_i} A_{ij}$ is the degree of node *i*.

In order to solve this formula, we introduce the following two notations:

$\boldsymbol{T}$: The relatedness transition matrix, where $T[i,j]=A_{ij}/d_i$, indicating the transition rate of relatedness from node *j* to its neighbor *i*.

$\boldsymbol{S}$: The structural semantic relatedness matrix, where $S[i,j]=S_{ij}$.

Now we can turn our first form of structural semantic relatedness into the matrix form:

$$S = \lambda TS + \mu A$$

By solving this equation, we can get:

$$S = \mu(I - \lambda T)^{-1} A$$

where $\boldsymbol{I}$ is the identity matrix. Since μ is a parameter which only contributes an overall scale factor to the relatedness value, we can ignore it and get the final form of the structural semantic relatedness as:

$$S = (I - \lambda T)^{-1} A$$

Because the *S* is asymmetric, the finally relatedness between node *i* and node *j* is the average of S_{ij} and S_{ji}.

The meaning of λ : The last question of our structural semantic relatedness measure is how to set the free parameter λ . To understand the meaning of λ**,** let us expand the similarity as a power series thus:

$$S = (I + \lambda T + \lambda^2 T^2 + ... + \lambda^k T^k + ...)A$$

Noting that the $[T^k]_{ij}$ element is the relatedness transition rate from node *i* to node *j* with path length *k*, we can view the λ as a penalty factor for the transition path length: by setting the λ with a value within (0, 1), a longer graph path will contribute less to the final relatedness value. The optimal value of λ is 0.6 through a learning

process shown in Section 4. For demonstration, Table 4 shows some structural semantic relatedness values of the Semantic-graph in Figure 3 (*CS* represents *computer science* and *GM* represents *Graphical model*). From Table 4, we can see that the structural semantic relatedness can successfully capture the semantic knowledge embedded in the structure of semantic-graph, such as the implicit semantic relation between *Researcher* and *Learning*.

	Researcher	CS	GM	Learning
Researcher	---	0.50	0.27	0.31
CS	0.50	---	0.62	0.73
GM	0.27	0.62	---	0.80
Learning	0.31	0.73	0.80	---

Table 4. The structural semantic relatedness of the semantic-graph shown in Figure 3

3 Named Entity Disambiguation by Leveraging Semantic Knowledge

In this section we describe how to leverage the semantic knowledge captured in the structural semantic relatedness measure for named entity disambiguation. Because the key problem of named entity disambiguation is to measure the similarity between name observations, we integrate the structural semantic relatedness in the similarity measure, so that it can better reflect the actual similarity between name observations.

Concretely, our named entity disambiguation system works as follows: 1) Measuring the similarity between name observations; 2) Grouping name observations using the clustering algorithm. In the following we describe each step in detail.

3.1 Measuring the Similarity between Name Observations

Intuitively, if two observations of the target name represent the same entity, it is highly possible that the concepts in their contexts are closely related, i.e., the named entities in their contexts are socially related and the Wikipedia concepts in their contexts are semantically related. In contrast, if two name observations represent different entities, the concepts within their contexts will not be closely related. Therefore we can measure the similarity between two name observations by summarizing all the semantic relatedness between the concepts in their contexts.

To measure the similarity between name observations, we represent each name observation as a weighted vector of concepts (including named entities, Wikipedia concepts and WordNet concepts), where the concepts are extracted using the same method described in Section 2.2, so they are just the same concepts within the semantic-graph. Using the same concept index as the semantic-graph, a name observation o_i is then represented as $o_i = \{w_{i1}, w_{i2}, ..., w_{in}\}$, where w_{ik} is the k^{th} concept's weight in observation o_i, computed using the standard TFIDF weight model, where the DF is computed using the Google Web1T 5-gram corpus[4]. Given the concept vector representation of two name observations o_i and o_j, their similarity is computed as:

$$SIM(o_i, o_j) = \sum_l \sum_k w_{il} w_{jk} S_{lk} \Big/ \sum_l \sum_k w_{il} w_{jk}$$

which is the weighted average of all the structural semantic relatedness between the concepts in the contexts of the two name observations.

3.2 Grouping Name Observations through Hierarchical Agglomerative Clustering

Given the computed similarities, name observations are disambiguated by grouping them according to their represented entities. In this paper, we group name observations using the hierarchical agglomerative clustering(HAC) algorithm, which is widely used in prior disambiguation research and evaluation task (WePS1 and WePS2). The HAC produce clusters in a bottom-up way as follows: Initially, each name observation is an individual cluster; then we iteratively merge the two clusters with the largest similarity value to form a new cluster until this similarity value is smaller than a preset merging threshold or all the observations reside in one common cluster. The merging threshold can be determined through cross-validation. We employ the single-link method to compute the similarity between two clusters, which has been applied widely in prior research (Bagga and Baldwin (1998); Mann and Yarowsky (2003)).

4 Experiments

To assess the performance of our method and compare it with traditional methods, we conduct a series of experiments. In the experiments, we evaluate the proposed *SSR* method on the task of personal name disambiguation, which is the most common type of named entity disambiguation. In the following, we first explain the general experimental settings in Section 4.1, 4.2 and 4.3; then evaluate and discuss the performance of our method in Section 4.4.

[4] www.ldc.upenn.edu/Catalog/docs/LDC2006T13/

4.1 Disambiguation Data Sets

We adopted the standard data sets used in the First Web People Search Clustering Task (**WePS1**) (Artiles et al., 2007) and the Second Web People Search Clustering Task (**WePS2**) (Artiles et al., 2009). The three data sets we used are **WePS1_training** data set, **WePS1_test** data set, and **WePS2_test** data set. Each of the three data sets consists of a set of ambiguous personal names (totally 109 personal names); and for each name, we need to disambiguate its observations in the web pages of the top *N* (100 for **WePS1** and 150 for **WePS2**) Yahoo! search results.

The experiment made the standard "one person per document" assumption, which is widely used in the participated systems in WePS1 and WePS2, i.e., all the observations of the same name in a document are assumed to represent the same entity. Based on this assumption, the features within the entire web page are used to disambiguate personal names.

4.2 Knowledge Sources

There were three knowledge sources we used for our experiments: the WordNet 3.0; the Sep. 9, 2007 English version of Wikipedia; and the Web pages of each ambiguous name in WePS datasets as the NE Co-occurrence Corpus.

4.3 Evaluation Criteria

We adopted the measures used in WePS1 to evaluate the performance of name disambiguation. These measures are:

Purity (**Pur**): *measures the homogeneity of name observations in the same cluster*;

Inverse purity (**Inv_Pur**): *measures the completeness of a cluster*;

F-Measure (**F**): *the harmonic mean of purity and inverse purity*.

The detailed definitions of these measures can be found in Amigo, et al. (2008). We use F-measure as the primary measure just liking WePS1 and WePS2.

4.4 Experimental Results

We compared our method with four baselines: (1) ***BOW:*** The first one is the traditional *Bag of Words* model (*BOW*) based methods: hierarchical agglomerative clustering (HAC) over term vector similarity, where the features including single words and NEs, and all the features are weighted using TFIDF. This baseline is also the state-of-art method in WePS1 and WePS2. (2) ***SocialNetwork:*** The second one is the social network based methods, which is the same as the method described in Malin et al. (2005): HAC over the similarity obtained through random walk over the social network built from the web pages of the top *N* search results. (3)***SSR-NoKnowledge:*** The third one is used as a baseline for evaluating the efficiency of semantic knowledge: HAC over the similarity computed on semantic-graph with no knowledge integrated, i.e., the similarity is computed as:

$$SIM(o_i, o_j) = \sum_l w_{il} w_{jl} \Big/ \sum_l \sum_k w_{il} w_{jk}$$

(4) ***SSR-NoStructure***: The fourth one is used as a baseline for evaluating the efficiency of the semantic knowledge embedded in complex structures: HAC over the similarity computed by only integrating the explicit semantic relations, i.e., the similarity is computed as:

$$SIM(o_i, o_j) = \sum_l \sum_k w_{il} w_{jk} A_k \Big/ \sum_l \sum_k w_{il} w_{jk}$$

4.4.1 Overall Performance

We conducted several experiments on all the three WePS data sets: the four baselines, the proposed ***SSR*** method and the proposed ***SSR*** method with only one special type knowledge added, respectively ***SSR-NE***, ***SSR-WordNet*** and ***SSR-Wikipedia***. All the optimal merging thresholds used in HAC were selected by applying leave-one-out cross validation. The overall performance is shown in Table 5.

Method	WePS1_training		
	Pur	**Inv_Pur**	**F**
BOW	0.71	0.88	**0.78**
SocialNetwork	0.66	0.98	**0.76**
SSR-NoKnowledge	0.79	0.89	**0.81**
SSR-NoStructure	0.87	0.83	**0.83**
SSR-NE	0.80	0.86	**0.82**
SSR-WordNet	0.80	0.91	**0.83**
SSR-Wikipedia	0.82	0.90	**0.84**
SSR	0.82	0.92	**0.85**
	WePS1_test		
	Pur	**Inv_Pur**	**F**
BOW	0.74	0.87	**0.74**
SocialNetwork	0.83	0.63	**0.65**
SSR-NoKnowledge	0.80	0.74	**0.75**
SSR-NoStructure	0.80	0.78	**0.78**
SSR-NE	0.73	0.80	**0.74**
SSR-WordNet	0.81	0.77	**0.77**
SSR-Wikipedia	0.88	0.77	**0.81**
SSR	0.85	0.83	**0.84**
	WePS2_test		
	Pur	**Inv_Pur**	**F**
BOW	0.80	0.80	**0.77**
SocialNetwork	0.62	0.93	**0.70**
SSR-NoKnowledge	0.84	0.80	**0.80**
SSR-NoStructure	0.84	0.83	**0.81**
SSR-NE	0.78	0.88	**0.80**
SSR-WordNet	0.85	0.82	**0.83**
SSR-Wikipedia	0.84	0.81	**0.82**
SSR	0.89	0.84	**0.86**

Table 5. Performance results of baselines and SSR methods

From the performance results in Table 5, we can see that:

1) The semantic knowledge can greatly improve the disambiguation performance: compared with the BOW and the SocialNetwork baselines, SSR respectively gets 8.7% and 14.7% improvement on average on the three data sets.

2) By leveraging the semantic knowledge from multiple knowledge sources, we can obtain a better named entity disambiguation performance: compared with the *SSR-NE*'s 0% improvement, the *SSR-WordNet*'s 2.3% improvement and the *SSR-Wikipedia*'s 3.7% improvement, the *SSR* gets 6.3% improvement over the *SSR-NoKnowledge* baseline, which is larger than all the *SSR* methods with only one type of semantic knowledge integrated.

3) The exploitation of the structural semantic knowledge can further improve the disambiguation performance: compared with *SSR-NoStructure*, our *SSR* method achieves 4.3% improvement.

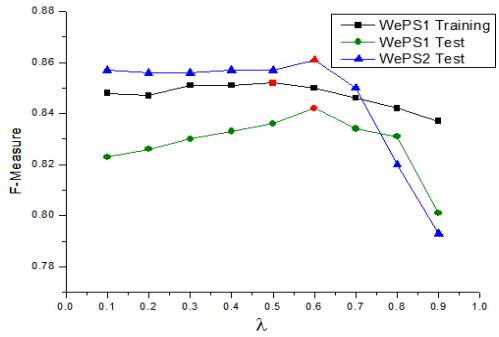

Figure 4. The F-Measure vs. λ on three data sets

4.4.2 *Optimizing Parameters*

There is only one parameter λ needed to be configured, which is the penalty factor for the relatedness transition path length in the structural semantic relatedness measure. Usually a smaller λ will make the structural semantic knowledge contribute less in the resulting relatedness value. Figure 4 plots the performance of our method corresponding to the special λ settings. As shown in Figure 4, the SSR method is not very sensitive to the λ and can achieve its best average performance when the value of λ is *0.6*.

4.4.3 *Detailed Analysis*

To better understand the reasons why our *SSR* method works well and how the exploitation of structural semantic knowledge can improve performance, we analyze the results in detail.

The Exploitation of Semantic Knowledge. The primary advantage of our method is the exploitation of semantic knowledge. Our method exploits the semantic knowledge in two directions:

1) *The Integration of Multiple Semantic Knowledge Sources.* Using the semantic-graph model, our method can integrate the semantic knowledge extracted from multiple knowledge sources, while most traditional knowledge-based methods are usually specialized to one type of knowledge. By integrating multiple semantic knowledge sources, our method can improve the semantic knowledge coverage.

2) *The exploitation of Semantic Knowledge embedded in complex structures.* Using the structural semantic relatedness measure, our method can exploit the implicit semantic knowledge embedded in complex structures; while traditional knowledge-based methods usually lack this ability.

The Rich Meaningful Features. One another advantage of our method is the rich meaningful features, which is brought by the multiple semantic knowledge sources. With more meaningful features, our method can better describe the name observations with less information loss. Furthermore, unlike the traditional N-gram features, the features enriched by semantic knowledge sources are all semantically meaningful units themselves, so little noisy features will be added. The effect of rich meaningful features can also be shown in Table 5: by adding these features, the *SSR-NoKnowledge* respectively achieves 2.3% and 9.7% improvement over the *BOW* and the *SocialNetwork* baseline.

5 Related Work

In this section, we briefly review the related work. Totally, the traditional named entity disambiguation methods can be classified into two categories: the shallow methods and the knowledge-based methods.

Most of previous named entity disambiguation researches adopt the shallow methods, which are mostly the natural extension of the *bag of words* (*BOW*) model. Bagga and Baldwin (1998) represented a name as a vector of its contextual words, then two names were predicted to be the same entity if their cosine similarity is above a threshold. Mann and Yarowsky (2003) and Niu et al. (2004) extended the vector representation with extracted biographic facts. Pedersen et al. (2005) employed significant bigrams to represent

a name observation. Chen and Martin (2007) explored a range of syntactic and semantic features.

In recent years some research has investigated employing knowledge sources to enhance the named entity disambiguation. Bunescu and Pasca (2006) disambiguated the names using the category information in Wikipedia. Cucerzan (2007) disambiguated the names by combining the *BOW* model with the Wikipedia category information. Han and Zhao (2009) leveraged the Wikipedia semantic knowledge for computing the similarity between name observations. Bekkerman and McCallum (2005) disambiguated names based on the link structure of the Web pages between a set of socially related persons. Kalashnikov et al. (2008) and Lu et al. (2007) used the co-occurrence statistics between named entities in the Web. The social network was also exploited for named entity disambiguation, where similarity is computed through random walking, such as the work introduced in Malin (2005), Malin and Airoldi (2005), Yang et al.(2006) and Minkov et al. (2006). Hassell et al. (2006) used the relationships from DBLP to disambiguate names in research domain.

6 Conclusions and Future Works

In this paper we demonstrate how to enhance the named entity disambiguation by capturing and exploiting the semantic knowledge existed in multiple knowledge sources. In particular, we propose a semantic relatedness measure, *Structural Semantic Relatedness*, which can capture both the explicit semantic relations and the implicit structural semantic knowledge. The experimental results on the WePS data sets demonstrate the efficiency of the proposed method. For future work, we want to develop a framework which can uniformly model the semantic knowledge and the contextual clues for named entity disambiguation.

Acknowledgments

The work is supported by the National Natural Science Foundation of China under Grants no. 60875041 and 60673042, and the National High Technology Development 863 Program of China under Grants no. 2006AA01Z144.

References

Amigo, E., Gonzalo, J., Artiles, J. and Verdejo, F. 2008. A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval.

Artiles, J., Gonzalo, J. & Sekine, S. 2007. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In SemEval.

Artiles, J., Gonzalo, J. and Sekine, S. 2009. WePS2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In WePS2, WWW 2009.

Baeza-Yates, R., Ribeiro-Neto, B., et al. 1999. Modern information retrieval. Addison-Wesley Reading, MA.

Bagga, A. & Baldwin, B. 1998. Entity-based cross-document coreferencing using the vector space model. Proceedings of the 17th international conference on Computational linguistics-Volume 1, pp. 79-85.

Bekkerman, R. & McCallum, A. 2005. Disambiguating web appearances of people in a social network. Proceedings of the 14th international conference on World Wide Web, pp. 463-470.

Bunescu, R. & Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. Proceedings of EACL, vol. 6.

Chen, Y. & Martin, J. 2007. Towards robust unsupervised personal name disambiguation. Proceedings of EMNLP and CoNLL, pp. 190-198.

Cilibrasi, R. L., Vitanyi, P. M. & CWI, A. 2007. The google similarity distance, IEEE Transactions on knowledge and data engineering, vol. 19, no. 3, pp. 370-383.

Cucerzan, S. 2007, Large-scale named entity disambiguation based on Wikipedia data. Proceedings of EMNLP-CoNLL, pp. 708-716.

Fellbaum, C., et al. 1998. WordNet: An electronic lexical database. MIT press Cambridge, MA.

Fleischman, M. B. & Hovy, E. 2004. Multi-document person name resolution. Proceedings of ACL, Reference Resolution Workshop.

Han, X. & Zhao, J. 2009. Named entity disambiguation by leveraging Wikipedia semantic knowledge. Proceeding of the 18th ACM conference on Information and knowledge management, pp. 215-224.

Hassell, J., Aleman-Meza, B. & Arpinar, I. 2006. Ontology-driven automatic entity disambiguation in unstructured text. Proceedings of The 2006 ISWC, pp. 44-57.

Jeh, G. & Widom, J. 2002. SimRank: A measure of structural-context similarity, Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, p. 543.

Kalashnikov, D. V., Nuray-Turan, R. & Mehrotra, S. 2008. Towards Breaking the Quality Curse. A Web-Querying Approach to Web People Search. In Proc. of SIGIR.

Leicht, E. A., Petter Holme, & M. E. J. Newman. 2006. Vertex similarity in networks. Physical Review E , vol. 73, no. 2, p. 26120.

Lin., D. 1998. An information-theoretic definition of similarity. In Proc. of ICML.

Lu, Y. & Nie , Z. et al. 2007. Name Disambiguation Using Web Connection. In Proc. of AAAI.

Malin, B. 2005. Unsupervised name disambiguation via social network similarity. SIAM SDM Workshop on Link Analysis, Counterterrorism and Security.

Malin, B., Airoldi, E. & Carley, K. M. 2005. A network analysis model for disambiguation of names in lists. Computational & Mathematical Organization Theory, vol. 11, no. 2, pp. 119-139.

Mann, G. S. & Yarowsky, D. 2003. Unsupervised personal name disambiguation, Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, p. 40.

McNamee, P. & Dang, H. Overview of the TAC 2009 Knowledge Base Population Track. In Proceedings of Text Analysis Conference (TAC-2009), 2009.

Medelyan, O., Witten, I. H. & Milne, D. 2008. Topic indexing with Wikipedia. Proceedings of the AAAI WikiAI workshop.

Milne, D., Medelyan, O. & Witten, I. H. 2006. Mining domain-specific thesauri from wikipedia: A case study. IEEE/WIC/ACM International Conference on Web Intelligence, pp. 442-448.

Minkov, E., Cohen, W. W. & Ng, A. Y. 2006. Contextual search and name disambiguation in email using graphs, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 27-34.

Niu C., Li W. and Srihari, R. K. 2004. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. Proceedings of ACL, pp. 598-605.

Pedersen, T., Purandare, A. & Kulkarni, A. 2005. Name discrimination by clustering similar contexts. Computational Linguistics and Intelligent Text Processing, pp. 226-237.

Strube, M. & Ponzetto, S. P. 2006. WikiRelate! Computing semantic relatedness using Wikipedia, Proceedings of the National Conference on Artificial Intelligence, vol. 21, no. 2, p. 1419.

Suchanek, F. M., Kasneci, G. & Weikum, G. 2007. Yago: a core of semantic knowledge, Proceedings of the 16th international conference on World Wide Web, p. 706.

Wan, X., Gao, J., Li, M. & Ding, B. 2005. Person resolution in person search results: Webhawk. Proceedings of the 14th ACM international conference on Information and knowledge management, p. 170.

Witten, D. M. & Milne, D. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA, pp. 25-30.

Yang, K. H., Chiou, K. Y., Lee, H. M. & Ho, J. M. 2006. Web appearance disambiguation of personal names based on network motif. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 386-389.