Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge

Xianpei Han Jun Zhao
Institute of Automation, Chinese Academy of Sciences
HaiDian District, Beijing, China
+86 10 8261 4468
{xphan, jzhao}@nlpr.ia.ac.cn

ABSTRACT

Name ambiguity problem has raised an urgent demand for efficient, high-quality named entity disambiguation methods. The key problem of named entity disambiguation is to measure the similarity between occurrences of names. The traditional methods measure the similarity using the bag of words (*BOW*) model. The *BOW*, however, ignores all the semantic relations such as social relatedness between named entities, associative relatedness between concepts, polysemy and synonymy between key terms. So the *BOW* cannot reflect the actual similarity. Some research has investigated social networks as background knowledge for disambiguation. Social networks, however, can only capture the social relatedness between named entities, and often suffer the limited coverage problem.

To overcome the previous methods' deficiencies, this paper proposes to use Wikipedia as the background knowledge for disambiguation, which surpasses other knowledge bases by the coverage of concepts, rich semantic information and up-to-date content. By leveraging Wikipedia's semantic knowledge like social relatedness between named entities and associative relatedness between concepts, we can measure the similarity between occurrences of names more accurately. In particular, we construct a large-scale semantic network from Wikipedia, in order that the semantic knowledge can be used efficiently and effectively. Based on the constructed semantic network, a novel similarity measure is proposed to leverage Wikipedia semantic knowledge for disambiguation. The proposed method has been tested on the standard WePS data sets. Empirical results show that the disambiguation performance of our method gets 10.7% improvement over the traditional *BOW* based methods and 16.7% improvement over the traditional social network based methods.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information storage and retrieval– *Information Search and Retrieval.*

General Terms

Algorithms, Experimentation

Keywords

Named Entity Disambiguation, Name ambiguity, Coreference Resolution, Record Linkage, Semantic Knowledge

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 Jordan, Footballer
Michael (B.) Jordan, American Actor

The name ambiguity has raised serious problems in many different 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 looking for. Besides, an ever-increasing number of question answering, information extraction systems are coming to rely on data from multi-sources, name ambiguity will lead to wrong answers and poor results. For example, in order to extract the birth date of the Berkeley's professor *Michael Jordan*, an information extraction system may return the birth date of his popular namesake basketball player *Michael Jordan*. Furthermore, ambiguous names are not unique identifiers for specific entities and, as a result, there are many confounders in the construction of knowledge base or social network about named entities. So there is an urgent demand for efficient, high-quality named entity disambiguation methods, which can disambiguate occurrences of names by grouping them according to their represented named entities.

Named entity disambiguation, however, is by no means a trivial task. In order to group occurrences of names, the disambiguation system must decide whether the occurrences of a specific name represent the same entity. The manner by which a human makes a decision is often contingent on contextual clues as well as prior background knowledge. For example, when a reader encounters the following four occurrences of the name "Michael Jordan":

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

the reader must decide whether these occurrences represent the same person. With the background knowledge that the *machine learning* in the context of the first *Michael Jordan* occurrence is semantic related to the *Graphical Models* in the context of the fourth *Michael Jordan* occurrence via associative relation, it is obvious that the first *Michael Jordan* represents the same person as the fourth *Michael Jordan*. And with the background knowledge that the entity *Chicago Bulls* is semantic related to the entity *NBA* via social relation, it is clear that the second and the third occurrence of *Michael Jordan* represent the same person.

Conventionally, named entity disambiguation methods determine whether two occurrences of a specific name represent the same entity by measuring the similarity between them. The traditional methods measure the similarity using the bag of words (*BOW*) model (Bagga and Baldwin[1]; Mann and Yarowsky[13]; Fleischman[21]; Pedersen et al.[26]), where an occurrence of name is represented as a term vector consisting of the terms that appear in the context and their associated weights. By "terms" we mean words, phrases or extracted named entities, but in most cases they are single words. In this model, similarity is measured by the co-occurrence statistics of terms. Hence the disambiguation algorithm can only group the occurrences of names containing the identical contextual terms, while all semantic relations (Hjørland, Birger[29]) like social relatedness between named entities, associative relatedness between concepts, and acronyms, synonyms, spelling variations between key terms are ignored. Thus, the *BOW* based similarity cannot reflect the actual similarity between name occurrences. Background knowledge is needed to capture the various semantic relations.

Recent research has investigated social networks as background knowledge for disambiguation (Malin and Airoldi[3]; Minkov et al.[10]; Bekkerman and McCallum[23]). Social networks can capture the social relatedness between named entities, so the similarity can be bridged by the socially related named entities. For example, although they share no identical contextual terms, the following two occurrences of *Michael Jordan*: "*Michael Jordan plays basketball in Chicago Bulls*" and "*Michael Jordan wins NBA MVP*" will still be identified as the same person if a social network can provide the information that *NBA* is socially related to *Chicago Bulls*. By leveraging social relatedness among entities, the social network based methods are more reliable than the *BOW* based methods in some situations. However, the social network based methods has a number of limitations: First, social networks can only capture a special type of semantic relations - the social relatedness between named entities, while all other semantic relations such as associative relation, hierarchical relation and equivalence relation between concepts are still ignored (e.g., the associative relatedness between *basketball* and *MVP* in above example); Second, social networks usually have limited coverage: most recent research uses social networks built from specific corpora, or some existing social networks of special domain, such as the IMDB for movie domain(Malin and Airoldi[3]) and the DBLP for research domain(Joseph et al.[17]).

To overcome the deficiencies of previous methods, in this paper we propose to use Wikipedia as the background knowledge for disambiguation, which surpasses other knowledge bases by the coverage of concepts, rich semantic information and up-to-date content (D. Milne, et al. [7]). By leveraging the semantic knowledge in Wikipedia like social relatedness between named entities, associative relatedness between concepts, acronyms and spelling variations between key terms, we can obtain a more accurate similarity measure between occurrences of names for disambiguation. In particular, we construct a large-scale semantic network from Wikipedia, in order that the semantic knowledge can be used efficiently and effectively: Wikipedia concepts in documents can be recognized, semantic relations between concepts can be identified and semantic relatedness between concepts can be measured. Based on the constructed semantic network, we first represent every occurrence of names as a Wikipedia concept vector; then the similarity between concept vectors are computed using a novel similarity measure which can leverage various types of semantic relations; finally a hierarchical agglomerative clustering algorithm is applied to grouping occurrences of names based on the similarity. To evaluate the performance of the proposed method, we have performed an empirical evaluation on the standard WePS data sets. The experimental results show that, with the help of Wikipedia semantic knowledge, the disambiguation performance of our proposed method is greatly improved over the previous methods.

This paper is organized as follows. In the next section we state the named entity disambiguation problem and briefly review the related work. Next in Section 3 we describe how to construct a semantic network from Wikipedia. In Section 4 we describe our proposed method in detail. Experimental results are discussed in Sections 5. Section 6 concludes this paper and discusses the future work.

2. PROBLEM STATEMENT AND RELATED WORK

Conventionally, a named entity disambiguation system is defined as a six-tuple $M = \{N, E, D, O, K, \delta\}$, where:

$N=\{n_1,n_2,\ldots,n_l\}$ is a set of ambiguous names which need to be disambiguated, e.g., {"Michael Jordan", …… };

$E=\{e_1,e_2,\ldots,e_k\}$ is a reference entity table containing the entities which the names in N may represent, e.g., {"Michael Jordan (Basketball player)", "Michael Jordan (Professor)", ……};

$D=\{d_1,d_2,\ldots,d_n\}$ is a set of documents containing the names in N;

$O=\{o_1,o_2,\ldots,o_m\}$ is all name observations in D which need to be disambiguated. In this paper, we use the term *observation* to denote the basic unit to be disambiguated: an occurrence of a particular name combined with its context. For example, "*NBA.com: Michael Jordan Bio*" is an observation of "Michael Jordan". The name occurrence's context can be various forms, such as the contextual words within a fixed window size or sometimes the entire document;

K is the background knowledge used in named entity disambiguation. The background knowledge has been exploited along a continuum, from the *BOW* model which includes no background knowledge, to the social network based methods which employ the social relatedness between named entities;

$\delta : O \times K \rightarrow E$ is the disambiguation function, the key component of named entity disambiguation, which groups the observations according to their represented entities.

Obviously, the perfect reference entity table E is in most cases unavailable, so disambiguation must be conducted on the condition that the reference entity table is incomplete. Therefore, in most cases the disambiguation problem is regarded as a

clustering task, where $\delta : O \times K \rightarrow E$ is a clustering algorithm, which clusters all the observations of a particular name, with each resulting cluster corresponding to one specific entity.

A lot of research has focused on named entity disambiguation. The traditional methods disambiguate names based on the bag of words (*BOW*) model. Bagga and Baldwin [1] represented a name as a vector of its contextual words, then the similarity between two names was determined by the co-occurring words, and finally two names were predicted to be the same entity if their similarity is above a threshold. Cucerzan [24] disambiguated names through linking them to Wikipedia entities by comparing their term vector representations. Mann and Yarowsky [13] extended the name's vector representation by extracting biographic facts. Pedersen et al. [26] employed significant bigrams to represent the context of a name. Fleischman [21] trained a Maximum Entropy model to give the probability that two names represent the same entity, then used a modified agglomerative clustering algorithm to cluster names using the probability as the similarity. Bunescu and Pasca [22] disambiguated the names in Wikipedia by linking them to the most similar Wikipedia entities using the similarity computed using a disambiguation SVM kernel.

All the similarity measures used in the above *BOW* based methods are only determined by the co-occurrences of terms, while all semantic relations like social relatedness between named entities and associative relatedness between concepts are all ignored. So background knowledge is needed to capture the various semantic relations. Recent research has investigated social networks as background knowledge for disambiguation. Bekkerman and McCallum [23] disambiguated names based on the link structure of the Web pages between a set of socially related persons, their model leveraged hyperlinks and the content similarity between web pages. Malin [2] and Malin and Airoldi [3] measured the similarity based on the probability of walking from one ambiguous name to another in the social network constructed from corpora. Minkov et al. [10] disambiguated names in email documents by building a social network from email data, then employed a random walk algorithm to compute the similarity. Joseph et al. [17] used the relationships from DBLP to pinpoint names in research domain to the persons in DBLP. Kalashnikov et al. [8] enhanced similarity measure by collecting named entity co-occurrence information via web search.

Social networks can enhance similarity measure by leveraging social relatedness between named entities. However, as mentioned in Section 1, social networks can only capture a special type of semantic relations - the social relatedness between named entities, and often suffer the limited coverage problem. To overcome these deficiencies, we propose to use Wikipedia as the background knowledge. In the following sections, we will show how to leverage semantic knowledge in Wikipedia for disambiguation.

3. WIKIPEDIA AS A SEMANTIC NETWORK

Wikipedia is the largest encyclopedia in the world and surpasses other knowledge bases in its coverage of concepts, rich semantic information and up-to-date content. Its English version contains more than 2,800,000 articles and new articles are added quickly and up-to-date[1]. Each article in Wikipedia describes a single concept; its title is a succinct, well-formed phrase that resembles a term in a conventional thesaurus (Milne, et al.[7]). Wikipedia contains concepts in a wide range[2], such as people, organizations, occupations and publications. Wikipedia contains rich semantic structures, such as disambiguation pages (polysemy), redirect pages (synonym), and hyperlinks between Wikipedia articles (associative relatedness and social relatedness, etc.). Moreover, Wikipedia has high coverage on both concepts and semantic relations. For example, in Food and Agriculture domain, the June 3, 2006 Version of English Wikipedia covers 72% useful concepts, 95% synonymy relations, 69% hierarchical relations and 56% associate relations(Milne, et al.[7]). And, with the growth of Wikipedia, these coverage rates will be further improved.

However, Wikipedia is an open data resource built for human use, so it includes much noise and the semantic knowledge within it is not suitable for direct use in named entity disambiguation. To make it clean and easy to use, we construct a semantic network from Wikipedia, in order that the semantic knowledge can be used efficiently and effectively for disambiguation: Wikipedia concepts within documents can be recognized, semantic relations between concepts can be identified and semantic relatedness between concepts can be measured efficiently and accurately.

3.1 Wikipedia Concepts

As shown above, each article in Wikipedia describes a single concept and its title can be used to represent the concept it describes, e.g., the title "IBM" and "Professor". However, some articles are meaningless – it is only used for Wikipedia management and administration, such as "1980s", "Wikipedia:Statistics", etc. Hence, we filter the noisy Wikipedia concepts using some rules from Hu, et al.[16], which is described below(titles satisfy one of the below will be filtered):

- *The article belongs to categories related to chronology, i.e. "Years", "Decades" and "Centuries".*
- *The first letter is not a capital one.*
- *The title is a single stop word.*

3.2 Surface Forms of Wikipedia Concepts

In many tasks, we need to recognize Wikipedia concepts in documents (plain texts, web pages, etc.). Usually the recognition is affected by two factors: First, a Wikipedia concept may appear in various surface forms. For example, the *IBM* can appear in more than 40 forms, such as *IBM*, *Big Blue* and *International Business Machine*. Second, a surface form may represent several Wikipedia concepts. For example, as shown in Table 1, the surface form *AI* can represent more than 6 Wikipedia concepts, such as *Artificial intelligence* and *Ai (singer)*.

Figure 1. Three anchor texts of *IBM*

Taking into account the above two factors, we collect a table of the surface forms (full name, acronyms, alternative names, and

[1] http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

[2] http://en.wikipedia.org/wiki/Portal:Contents/Categorical_index

spelling variations) of Wikipedia concepts for Wikipedia concept recognition. The surface forms of Wikipedia concepts can be collected from anchor texts in Wikipedia: each link in Wikipedia is associated with an anchor text, and the anchor text can be regarded as the surface form of its target concept. For example, the three anchor texts of *IBM* in Figure 1 are respectively its full name "International Business Machines", acronyms "IBM" and alternative name "Big Blue". Using the anchor text collection in Wikipedia, we can collect all surface forms and, for each of the surface forms, we summarize its target concepts together with the count information it's used as the anchor text of a specific Wikipedia concept. Part of the surface form table is shown in Table 1. Using the collected surface form table, we are able to recognize Wikipedia concepts in documents and the detailed description of recognition method is shown in Section 4.1.

Surface Form	Target Concept	Count
IBM	IBM	3685
	IBM mainframe	2
	IBM DB2	2
	…	…
International Business Machine	IBM	1
AI	Artificial intelligence	581
	Game artificial intelligence	48
	Ai (singer)	10
	Angel Investigations	9
	Strong AI	3
	Characters in the Halo series	2
	…	…

Table 1. Part of the surface form table of Wikipedia concepts

3.3 Semantic Relations between Wikipedia Concepts

Wikipedia contains rich relation structures, such as synonymy (Redirect page), Polysemy (disambiguation page), social relatedness and associative relatedness (internal page link). All these semantic relations express in the form of hyperlinks between Wikipedia articles, and as Milne et al. [6] mentioned that, links between articles are only tenuously related. Therefore in the constructed semantic network, two Wikipedia concepts are considered to be semantic related if there are hyperlinks between them. In this way, the constructed semantic network can incorporate all the semantic relations expressed by the hyperlinks between Wikipedia articles. For example, Figure 2 shows a part of the constructed semantic network, which contains all the semantic related concepts of the Berkeley's professor *Michael Jordan*.

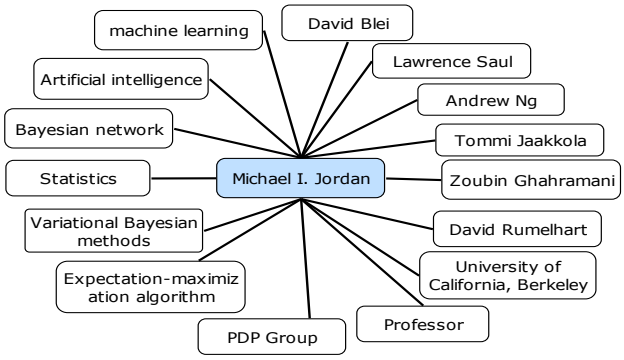

Figure 2. The semantic related concepts of Berkeley professor *Michael I. Jordan* in the constructed semantic network

3.4 Semantic Relatedness between Wikipedia Concepts

Semantic relations can provide the information about whether two concepts are related, but it doesn't explicitly provide the value of the semantic relation's strength. In order to incorporate Wikipedia semantic into similarity measure, we must measure the semantic relation's strength (semantic relatedness) between concepts. There has been several research which focus on computing the semantic relatedness between Wikipedia concepts (Strube and Ponzetto [25]; Gabrilovich and Markovich[12]; Milne and Witten[6]). In this paper, we adopt the method described in Milne and Witten [6] to compute the semantic relatedness between Wikipedia concepts. Based on the idea that the higher semantic related Wikipedia concepts will share more semantic related concepts, this method measures the semantic relatedness as:

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

where a and b are the two concepts of interest, A and B are the sets of all concepts that link to a and b respectively, and W is the entire Wikipedia. We show an example of semantic relatedness between four selected concepts in Table 2, where the semantic relatedness can reveals the associate relatedness between *Bayesian network* and *Machine learning*, and the social relatedness between *Chicago Bulls* and *NBA*.

	Bayesian network	Chicago Bulls
Machine learning	0.74	0.00
NBA	0.00	0.71

Table 2. The semantic relatedness table of four selected concepts

4. NAMED ENTITY DISAMBIGUATION BY LEVERAGING WIKIPEDIA SEMANTIC KNOWLEDGW

In this section, we describe our proposed method in detail and show how to leverage Wikipedia semantic knowledge for disambiguation. There are three steps in total: (1) representing name observations as Wikipedia concept vectors; (2) computing the similarity between name observations; (3) grouping name observations using a hierarchical agglomerative clustering algorithm. The critical innovation of the proposed method is a novel similarity measure which can accurately measure the similarity between name observations by incorporating the various semantic relations in Wikipedia.

4.1 Representing Name Observations as Wikipedia Concept Vectors

Intuitively, if two name observations represent the same entity, it is highly possible that the Wikipedia concepts in their contexts are highly related. In contrast, if two name observations represent different entities, the Wikipedia concepts in their contexts will not be closely related. Thus, a name observation o can be represented by the Wikipedia concepts in its context, i.e., a Wikipedia concept vector $o = \{(c_1, w(c_1, o)), (c_2, w(c_2, o)), ..., (c_m, w(c_m, o))\}$, where each concept c_i is assigned a weight $w(c_i, o)$ indicating the relatedness between c_i and o.

In order to represent a name observation as a Wikipedia concept vector, we recognize the Wikipedia concepts in its context. In this paper, we use the collected table of surface forms and take the same route as Milne and Witten [5] to recognize Wikipedia concepts. The recognition takes three steps: (1) identifying surface forms; (2) mapping them to Wikipedia concepts; (3) concepts weighting and pruning for a better similarity measure. The detail description is as follows.

Surface form identification. In order to recognize Wikipedia concepts, we first identify all occurrences of surface forms. Given a name observation's context as input, we gather all N-grams (up to 8 words) and match them to the surface forms in the collected surface form table (described in Section 3.2). Not all matches are considered, because even stop words such as "is" and "a" may represent a concept. We use Mihalcea and Csomai [19]'s keyphraseness feature to select helpful surface forms. In detail, for each surface form *s*, we first calculate its probability of representing a concept as $f_a(s)/(f_a(s)+f_t(s))$, where $f_a(s)$ is the number of Wikipedia articles in which the surface form represents a concept, and $f_t(s)$is the number of articles in which the surface form appears in any form. Then surface forms with low probabilities are discarded.

Mapping surface forms to concepts. As mentioned earlier, surface forms may be ambiguous for they can represent more than one concept, such as the *IBM* in Table 1, the concept candidates it may represent include *IBM*, *IBM mainframe* and *IBM DB2*, etc. So a mapping step is needed to identify which concept a surface form actually represents. In this paper, we adopt the mapping method described in Medelyan et al. [19]: First, the method detect the "context concepts" *T* in name observations, i.e., the concepts which the unambiguous surface forms (which has only one target concept, e.g., the *International Business Machine* in Table 1) mapped to. Then, the method scores the final mapping between a surface form *s* and a candidate concept *c* by combining the average similarity of a candidate concept with the commonness of this mapping:

$$Score(s,c) = \frac{\sum_{t \in T} sr(t,c)}{|T|} \times Commonness_{s,c}, \text{ where}$$

$$Commonness_{s,c} = \frac{Count(s,c)}{Count(s)}$$

Finally the candidate concept with highest score will be taken as the target concept of a surface form. Using this method, the mapping accuracy can be up to 93.3%. More details about this method can be found in Medelyan et al.[19].

Concepts weighting and pruning. After the first two steps, a name observation is represented as a Wikipedia concept vector $o=\{c_1,c_2, \ldots, c_m\}$. However, not all concepts in representation are equally helpful for named entity disambiguation: documents may contain noisy concepts (this is very common in web pages) and some concepts are only loosely related to the observed name. So here we expect to preserve the concepts that are highly related to the observed name, and discard the outliers that are only loosely related to the observed name. This paper select the helpful concepts by assign each concept with a weight indicating its relatedness to the observed name. In detail, for each concept *c* in a name observation *o*, we assign it a weight by averaging the semantic relatedness of *c* to all other concepts in *o*, i.e.:

$$w(c,o) = |o|^{-1} \left(\sum_{c_i \in o, c_i \neq c} sr(c,c_i) \right)$$

Based on the computed weights, we are able to prune concepts to improve both efficiency and accuracy for disambiguation using a weight threshold which can be learned in a learning process.

4.2 Measuring the Similarity between Name Observations by Leveraging Wikipedia Semantic Knowledge

Through the method described in Section 4.1, a name observation is represented as a Wikipedia concept vector:

$$o = \{(c_1, w(c_1,o)), (c_2, w(c_2,o)), \ldots, (c_m, w(c_m,o))\}$$

where each concept c_i is assigned with a weight $w(c_i,o)$. For example, given the following three observations *MJ1*, *MJ2* and *MJ3* of "Michael Jordan", their concept vector representations are shown in Figure 3.

MJ1: ***Michael Jordan*** *is a leading researcher in machine learning and artificial intelligence.*
MJ2: ***Michael Jordan*** *has published over 300 research articles on topics in computer science, statistics and cognitive science.*
MJ3: ***Michael Jordan*** *wins NBA MVP.*

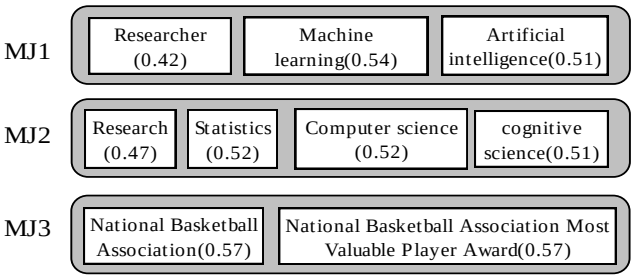

Figure 3. The concept representations of *MJ1*, *MJ2* and *MJ3*

After obtaining the concept vector representations of name observations, previous methods' similarity measures can be applied to compute the similarity of two name observations. However, previous methods' similarity measures cannot take into consideration the semantic relations: the *BOW* based methods typically measure the similarity between name observations using the cosine of their term vectors, so that matches of terms indicate relatedness and mismatches indicate otherwise; the social network based methods measure the similarity using only the social relatedness between named entities.

So in this paper, we propose a novel similarity measure which allows us to take into account the full semantic relations indicated by hyperlinks within Wikipedia, rather than just term overlap or social relatedness between named entities. Given two name observations o_l and o_k, the proposed similarity measure is computed as follows:

Step 1. Concept alignment between two concept vector representations. In order to measure the similarity between two concept vector representations, firstly we must define the correspondence between the concepts from one vector to those from another. A simple alignment strategy is to assign a concept to the target concept which is exactly the same match, e.g., assign "Research" to "Research", "Machine learning" to "Machine

learning". This alignment strategy, however, cannot take semantic relations between concepts into account. Therefore, we use the following strategy to align concepts: for each concept c in an observation o_l, we assign it a target concept $Align(c, o_k)$ in another observation o_k, which will maximize the semantic relatedness between the concept pair, i.e.,

$$Align(c, o_k) = \underset{c_i \in o_k}{\operatorname{argmax}}\, sr(c, c_i)$$

We use two examples shown in Figure 4 and Figure 5 to demonstrate the proposed concept alignment strategy based on the semantic relatedness table shown in Table 3.

	Researcher	**Machine Learning**	**Artificial intelligence**
Research	0.54	0.38	0.40
Statistics	0.32	0.58	0.46
Computer science	0.44	0.50	0.60
Cognitive science	0.44	0.66	0.65

Table 3. The semantic relatedness table of between the concepts in *MJ1* and *MJ2*

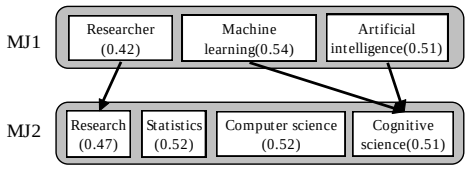

Figure 4. The concept alignment from *MJ1* to *MJ2*

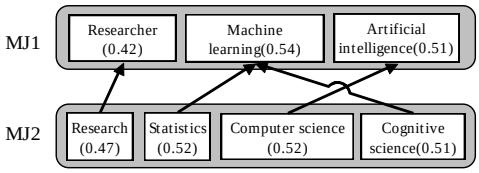

Figure 5. The concept alignment from *MJ2* to *MJ1*

Step 2. Compute the semantic relatedness from one concept vector representation to another. We define the semantic relatedness from a source concept vector representation o_k to target representation o_l as the weighted average of all the semantic relatedness between the source concepts in o_k and their aligned target concepts in o_l :

$$SR(o_k \rightarrow o_l) = \frac{\sum_{c \in o_k} w(c, o_k) \times w(Align(c, o_l), o_l) \times sr(c, Align(c, o_l))}{\sum_{c \in o_k} w(c, o_k) \times w(Align(c, o_l), o_l)}$$

Using the alignments shown in Figure 4 and Figure 5, *SR(MJ1→ MJ2)* is computed as $(0.42 \times 0.47 \times 0.54 + 0.54 \times 0.51 \times 0.66 + 0.51 \times 0.51 \times 0.65)/(0.42 \times 0.47 + 0.54 \times 0.51 + 0.51 \times 0.51) = 0.62$, and *SR(MJ2→MJ1)* is computed as $(0.47 \times 0.42 \times 0.54 + 0.52 \times 0.54 \times 0.58 + 0.52 \times 0.51 \times 0.60 + 0.51 \times 0.54 \times 0.66)/(0.47 \times 0.42 + 0.52 \times 0.54 + 0.52 \times 0.51 + 0.51 \times 0.54) = 0.60$.

Step 3: Compute similarity between two concept vector representations. We compute the similarity between o_l and o_k as the average of the semantic relatedness from o_l to o_k and that from o_k to o_l:

$$SIM(o_k, o_l) = \frac{1}{2} \times (SR(o_k \rightarrow o_l) + SR(o_l \rightarrow o_k))$$

Because the semantic relatedness $sr(c,c_i)$ is always in [0,1], $SR(o_l \rightarrow o_k)$ will also bounded within [0,1], thus the $SIM(o_k,o_l)$ between two name observations will also bounded within [0,1]: while 0 indicates the named entities represented by the two name observations are completely unrelated and 1 indicates the named entities represented by the two name observations are mostly related.

Using the proposed similarity measure, the semantic similarity *SIM(MJ1, MJ2)* is computed as (0.60 + 0.62)/2 = 0.61, *SIM(MJ2, MJ3)* is computed as *0.10* and *SIM(MJ1, MJ3)* is computed as *0.0.* These similarities indicate that, although (*MJ1, MJ2),(MJ1, MJ3)* and (*MJ2,MJ3)* all have no concept overlap, the similarity values measured by leveraging Wikipedia semantic knowledge can still successfully reveal the fact that (*MJ1, MJ2)* is highly possible represents the same entity, while (*MJ1, MJ3)* and *(MJ2, MJ3)* are unlikely represent the same entity.

4.3 Grouping Name Observations Using Hierarchical Agglomerative Clustering

Given the computed similarities, name observations are disambiguated by grouping them according to their represented entities. In this paper, we grouping 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 average-link method to compute the similarity between two clusters which has been applied in prior disambiguation research (Bagga and Baldwin[1]; Mann and Yarowsky[13]), where similarity between different clusters, denoted $CSIM(u_i, u_j)$, is calculated as follows:

$$CSIM(u_i, u_j) = (\|u_i\| \|u_j\|)^{-1} \sum_{s \in u_i, t \in u_j} SIM(s, t)$$

where s, t are name observations in cluster u_i and cluster u_j.

5. 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 our proposed method on the disambiguation of personal names, which is the most common type of named entity disambiguation. The experiments are conducted on a standard disambiguation data set, the WePS data set [14,15]. In the following, we first explain the general experimental settings in Section 5.1, 5.2 and 5.3, then evaluate and discuss the performance of our method.

5.1 Wikipedia Data

Wikipedia data can be obtained easily from http://download.wikipedia.org for free research use. It is available in the form of database dumps that are released periodically. The version we used in our experiments was released on Sep. 9, 2007.

We identified over 4,600,000 distinct concepts for the construction of semantic network. The concepts are highly inter-linked: averagely each concept links to 10 other concepts. This indicates the rich semantic relations between Wikipedia concepts.

5.2 Disambiguation Data Sets

We adopted the standard data sets used in the First Web People Search Clustering Task (**WePS1**) ([14]) and the Second Web People Search Clustering Task (**WePS2**) ([15]). All the three data sets were used: **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, its observations in the web pages of the top N (100 for **WePS1** and 150 for **WePS2**) Yahoo! search results are needed to be disambiguated.

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

5.3 Evaluation Criteria

We adopted the measures used in WePS1 ([14]) to evaluate the performance of name disambiguation. These measures are:

Purity (**Pur**): measures the homogeneity of the observations of names 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. [11]. Because purity and inverse purity are often positively correlated, they do not always get their peaks at the same point. In this case, we used F-measure as the most important measure just like WePS1 and WePS2.

5.4 Experimental Results

We compared our method with three baselines: (1) The first one is the traditional *BOW* based methods: hierarchical agglomerative clustering (HAC) over term vector similarity, where a web page is represented as the features including single words and NEs, and all the features are weighted using TFIDF- we denote this baseline as ***BOW***, which is also the state-of-art method in WePS1 and WePS2; (2) The second one is social network based methods, which is the same as the method described in Malin and Airoldi [3]: HAC over the similarity obtained through random walk over the social network built from the web pages of top N search results - we denote this baseline as ***SocialNetwork***; (3) The third one evaluates the efficiency of Wikipedia concept representation: HAC over the cosine similarity between the Wikipedia concept representations of the name observations-we denoted it as ***WikipediaConcept***.

5.4.1 Overall Performance

We conducted several experiments on all the three WePS data sets: the baseline ***BOW***, the baseline ***SocialNetwork***, the baseline ***WikipediaConcept***, the proposed method with all Wikipedia concepts assigned the same weight 1.0(***WS-SameWeight***), and the proposed method with concept weighting and pruning (***WS***). All the optimal merging thresholds used in HAC were selected by applying leave-one-out cross validation. The concept pruning threshold for ***WS*** was set to 0.04 through a learning process which will be introduced detailedly in the next section. The overall performance is shown in Table 4.

Method	**WePS1_training**		
	Pur	**Inv_Pur**	**F**
BOW	0.71	0.88	**0.78**
SocialNetwork	0.66	0.98	**0.76**
WikipediaConcept	0.80	0.88	**0.82**
WS-SameWeight	0.84	0.89	**0.85**
WS	0.88	0.89	**0.87**
	WePS1_test		
	Pur	**Inv_Pur**	**F**
BOW	0.74	0.87	**0.74**
SocialNetwork	0.83	0.63	**0.65**
WikipediaConcept	0.73	0.72	**0.71**
WS-SameWeight	0.83	0.87	**0.84**
WS	0.88	0.90	**0.88**
	WePS2_test		
	Pur	**Inv_Pur**	**F**
BOW	0.80	0.80	**0.77**
SocialNetwork	0.62	0.93	**0.70**
WikipediaConcept	0.71	0.84	**0.75**
WS-SameWeight	0.84	0.82	**0.83**
WS	0.85	0.89	**0.86**

Table 4. Performance results of baselines, *WS-SameWeight* and *WS*

From the performance results in Table 4, we can see that within the three baselines:

1) *BOW* and *WikipediaConcept* perform better than the *SocialNetwork*: In comparison with *SocialNetwork*, *BOW* gets 6% improvement and *WikipediaConcept* gets 5.7% improvement. We believe this is because *SocialNetwork* only used the named entities within context, which is usually insufficient for named entity disambiguation: compared with *BOW*, it ignores helpful contextual words; compared with *WikipediaConcept*, it ignores helpful concepts of other types.

2) There is no clear winner between *BOW* and *WikipediaConcept*: the winner is different on different data sets. This may indicate that Wikipedia concept representations contain considerable information as the *BOW*'s representations do.

By comparing the proposed method with the three baselines, we found that by leveraging Wikipedia semantic knowledge, our method can greatly improve the disambiguation performance: compared with *BOW*, *WS-SameWeight* gets 7.7% improvement and *WS* gets 10.7% improvement on average on the three data sets; compared with *SocialNetwork*, *WS-SameWeight* gets 13.7% improvement and *WS* gets 16.7% improvement; Compared with *WikipediaConcept*, *WS-SameWeight* gets 8% improvement and *WS* gets 11% improvement on average on the three data sets.

Comparing the performances of the two proposed methods, *WS-SameWeight* and *WS*, we can find that the concept weighting and pruning can improve the proposed method by 3% on average.

Representation	Features
Terms	machine**(5)**, learning**(5)**, networks**(2)**, statistics**(2)**, David**(2)**, cognitive**(2)**, Department**(2)**, students**(2)**, postdocs**(2)**, field**(2)**
Named Entities	Andrew Ng, David Blei, David E. Rumelhart, Lawrence Saul, Tommi Jaakkola, Zoubin Ghahramani, Berkeley, California, Department of EECS, Department of Statistics, PDP Group
Wikipedia Concepts	Statistics **(0.273)**, Machine learning**(0.269)**, Artificial intelligence**(0.267)**, University of California, Berkeley **(0.225)**, David Rumelhart **(0.218)**, Inference**(0.215)**, Professor **(0.210)**, Bayesian network**(0.210)**, Expectation-maximization algorithm**(0.201)**, Doctor of Philosophy**(0.194)**, Variational Bayesian methods**(0.193)**, Postgraduate education**(0.169)**, Zoubin Ghahramani**(0.167)**, Student**(0.158)**, Researcher**(0.157)**, Postdoctoral researcher**(0.144)**, Cognitive model**(0.124)**, Perspective (cognitive)**(0.108)**, Recurrent neural network**(0.103),** Formal system**(0.082)**

Table 5. The representations of Professor *Michael Jordan*'s Wikipedia Page

5.4.2 Optimizing Parameters

Our proposed method selects helpful concepts for disambiguation by assigning them with weights and pruning them. A weight threshold needs to be set for pruning the outliner concepts. Usually a larger threshold will filter out more outliner concepts but meanwhile it will also filter out more helpful concepts. Figure 6 plots the tradeoff. For WePS1_training and WePS1_testing data sets, a threshold 0.04 will result in the best performance. But the pruning of concepts will lead to a decline in performance on WePS2_testing data set. Overall, the best threshold can only enhance the performance in a limited extend (0.1% on average). We believe this is because the concept weighting is good enough for disambiguation, so a pruning step cannot make significant improvements.

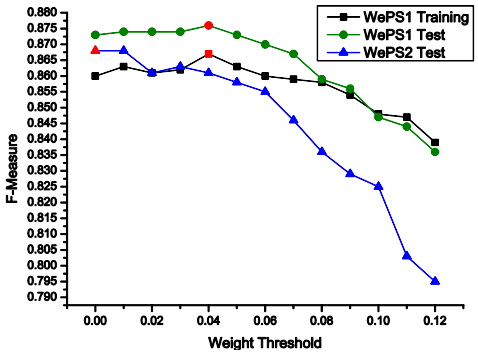

Figure 6. The F-Measure vs. Concept Weight Threshold on three data sets

Representation	Features
Terms	vol**(9)**, pp**(9)**, Research**(6)**, Learning**(6)**, Machine**(5)**, Bayesian**(4)**, Science**(4)**, Fellow**(4)**, 2006**(4)**, Electrical**(3)**, Engineering**(3)**, Berkeley**(3)**, Statistical**(3)**
Named Entities	A. Y. Ng, B. Taskar, D. M. Blei, Z. Ghahramani, P. Xing, W. Teh, D. Wolpert, AAAI, AAAS, IEEE, IMS, American Statistical Association, Arizona State University, Berkeley, Department of Electrical Engineering and Computer Science, Department of Statistics, University of California,
Wikipedia Concepts	Computer science**(0.257)**, Statistics**(0.253)**, Neural network**(0.244)**, Artificial intelligence**(0.242)**, Cognitive science**(0.238)**, Research**(0.237)**, Bioinformatics**(0.234)**, Massachusetts Institute of Technology**(0.225)**, Machine learning**(0.223)**, Inference**(0.215)**, Robotics**(0.211)**, Institute of Electrical and Electronics Engineers**(0.200)**, Bayesian inference**(0.192)**, Molecular biology**(0.190)**, Bioengineering**(0.185)**, University of California, Berkeley**(0.183),** Distributed computing**(0.177)**, Prediction**(0.169)**, Doctor of Philosophy**(0.168)**, Computational biology**(0.160)**

Table 6. The representations of Professor *Michael Jordan*'s Home Page

5.4.3 Detailed Analysis

To better understand the reasons why our proposed method works better than the BOW based methods and the social network based methods, we analyze the features of name observations generated by different methods and show how they affect the similarity measures.

For demonstration, Table 5 and 6 respectively show the top weighted features of two web pages which talks about the Berkeley professor *Michael Jordan*: one is his Wikipedia page[3] and the other is his Berkeley homepage[4]. The occurrence counts of terms and the weights of Wikipedia concepts are shown within the brackets after them.

Comparison of Representations. As shown in Table 5 and 6, the feature representations generated by different methods are different: the *BOW* based methods represent a name observation as a vector of terms; the social network based methods represent a name observation as a set of named entities; our method represents a name observation as a Wikipedia concept vector. Compared with the term vector representation and the named entities representation, the Wikipedia concept vector representation has the following advantages:

[3] http://en.wikipedia.org/wiki/Michael_I._Jordan

[4] http://www.eecs.berkeley.edu/Faculty/Homepages/jordan.html

1) Compared with the term vector representation, the Wikipedia concept vector representation is more meaningful. All the features in Wikipedia concept representation are concepts which themselves are semantic units, while some terms in term vector representation cannot explain their actual meaning on behalf of themselves. For example, the proposed method extracts a feature *Machine learning* from the phrase "Machine learning" while the term vector representation extract two separate terms *machine* and *learning*.

2) Compared with the social network based methods, our method can generate features in a larger scope. Except for the named entities, our method can also extract the concepts of other types contained in Wikipedia such as *occupation*, *subject* and *degree* , which is also very useful for disambiguation. For example, the concepts *Statistics*, *Professor*, *Computer science* and *Machine learning* in Table 5 and 6.

3) All the features in Wikipedia concept vector representation are corresponding to Wikipedia articles, rather than their surface forms. So our method can handle the acronyms and spelling variations by mapping them into the same concept, while the other two representations usually lack this ability. For example, *Andrew Ng* in Table 5 and *A. Y. Ng* in Table 6 are actually the same person, *AAAS* and *American Statistical Association* in Table 6 are actually the same organization, but the social network based methods cannot recognize them as the same one. On the other hand, it is obvious that the semantic knowledge incorporation will be more effectively and efficiently using Wikipedia concept representation: for every feature there is an article in Wikipedia which can provide the detailed knowledge about it.

Comparison of Similarity Measures. When measuring the similarity between name observations, the three methods (*BOW*, social network and the proposed method) use different measures: The term vector similarity used in the *BOW* based methods is determined by the term co-occurrence statistics; the social network similarity is determined by the social relatedness between contextual named entities; and our proposed similarity is determined by the semantic relatedness between Wikipedia concepts. Compared with other two similarity measures, the proposed similarity measure shows the following advantages:

1) Compared with the other two similarity measures, our proposed similarity measure can incorporate more semantic relations between features. The term vector similarity ignores all the semantic relations between terms, such as associate relatedness between *statistics* and *Bayesian*, and social relatedness between *Berkeley* and *David*. The social network based methods can only capture social relatedness between named entities, such as that between *University of California* and *Department of EECS*, *Andrew Ng* and *Z. Ghahramani* in Table 5 and 6. But it cannot incorporate semantic relations of other types, such as associate relatedness between *machine learning* and *statistics*, *Bayesian network* and *Cognitive science*. Compared with the above two similarity measures, our proposed similarity measure can incorporate all these semantic relations.

2) The relatedness measure between features (terms, named entities and concepts) used in the proposed similarity measure is more reliable and accurate. The term vector similarity measures the relatedness between terms as either *0* or *1*, this usually conflicts reality. For example, the two terms *statistics* and *statistical* is obvious more related than *statistics* and *pp*, but the term vector similarity gives them the same relatedness *0*. Currently the social relatedness between named entities are usually set by manually defined heuristic rules (Malin and Airoldi[3], Minkov et al.[10]). While based on the large-scale and semantic information rich data in Wikipedia, the semantic relatedness measure between concepts has shown their efficiency in Milne and Witten [6].

5.4.4 Comparison with State-of-art Performance

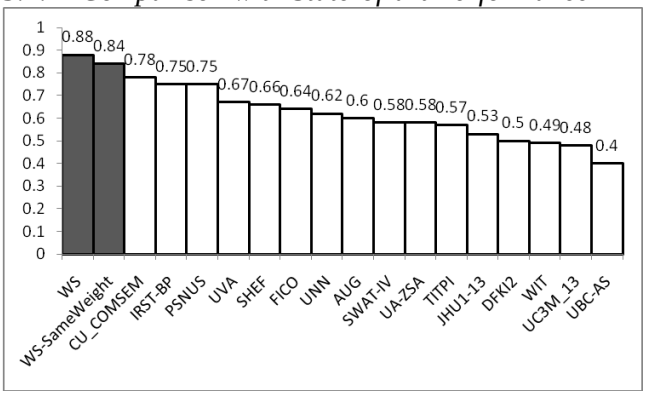

Figure 7. A comparision with WePS1 systems

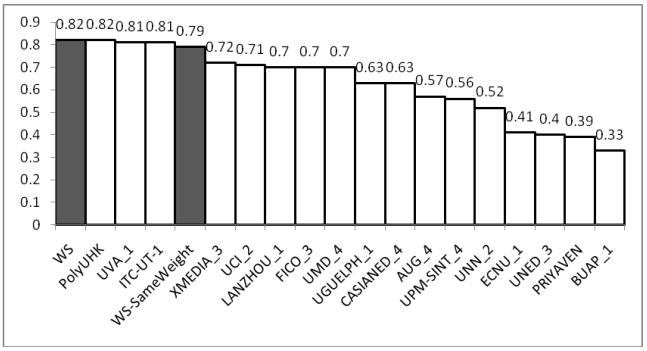

Figure 8. A comparision with WePS2 systems using B-Cubed F-measure

We also compared our method with the state-of-art Performance in WePS1 (Artiles, et al.[14]) and WePS2 (Artiles, et al.[15]). Because WePS2 evaluated the participating systems using the B-Cubed measures, we compared our method with the systems participating in WePS2 by optimizing our method on the B-Cubed F-measure. The comparison results are shown in Figure 7 and 8. As shown in Figure 7, our method gets 10% improvement over the best system of WePS1. As shown in Figure 8, in comparison with the systems participating in WePS2, our method can obtain the same performance as the best solution. We believe our method is competitive: the best solution in WePS2 extracted additional features such as the title words of the root page of the given web page and used some large additional resources such as the Web 1T 5-gram corpus of Google, while these features and knowledge are not used in our proposed method. And we believe our method can be further improved by collecting additional disambiguation evidence from the web.

6. CONCLUSIONS AND FUTURE WORKS

In this paper we demonstrate how to leverage the semantic knowledge in Wikipedia, so the performance of named entity

disambiguation can be enhanced by obtaining a more accurate similarity measure between name observations. Concretely, we construct a large-scale semantic network from Wikipedia, in order that the semantic knowledge can be used efficiently and effectively. Based on the constructed semantic network, a novel similarity measure is proposed to leverage Wikipedia semantic knowledge for disambiguation. On the standard WePS data sets, our method can achieve appealing results: it gets 10.7% improvement over the traditional *BOW* based method and 16.7% improvement over the traditional social network based methods.

For future work, because Wikipedia also provides other semantic knowledge like category hierarchy and structural description of entities (e.g. the infobox), so Wikipedia semantic knowledge can also be used to tag and generate a concise structural summary of disambiguation results. Furthermore, Wikipedia semantic knowledge is also very useful in many other different tasks, such as knowledge base population, link analysis, document clustering and classification.

7. ACKNOWLEDGMENTS

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

8. REFERENCES

[1] Bagga and Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model, In Proc. of HLT/ACL, 1998.

[2] B. Malin. Unsupervised Name Disambiguation via Social Network Similarity, In Proc. of SIAM, 2005.

[3] B. Malin and E. Airoldi. A Network Analysis Model for Disambiguation of Names in Lists. In Proc. of CMOT, 2005.

[4] Cheng Niu, Wei Li and Srihari. Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. In Proc. of ACL, 2004.

[5] D. Milne and Ian H. Witten. Learning to Link with Wikipedia. In Proc. of CIKM, 2008.

[6] D. Milne and Ian H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proc. of AAAI, 2008.

[7] D. Milne, O. Medelyan and Ian H. Witten. Mining Domain-Specific Thesauri from Wikipedia: A case study. In Proc. of IEEE/WIC/ACM WI, 2006.

[8] D. V. Kalashnikov, R. Nuray-Turan and S. Mehrotra. Towards Breaking the Quality Curse. A Web-Querying Approach to Web People Search. In Proceedings of SIGIR. 2008.

[9] E. Gabrilovich and S. Markovitch. Feature Generation for Text Categorization Using World Knowledge. In Proc. of IJCAI, 2005.

[10] Einat Minkov, William W. Cohen and Andrew Y. Ng. Contextual Search and Name Disambiguation in Email Using Graphs. In Proc. of SIGIR, 2006.

[11] Enrique Amigo, Julio Gonzalo, Javier Artiles and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval, 2008.

[12] E. Gabrilovich, and S. Markovich. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In Proc. of the IJCAI, 2007.

[13] Gideon S. Mann and David Yarowsky. Unsupervised Personal Name Disambiguation. In Proc. of CONIL, 2003.

[14] Javier Artiles, Julio Gonzalo and Satoshi Sekine. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In SemEval, 2007.

[15] Javier Artiles, Julio Gonzalo and Satoshi Sekine. WePS2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In WePS2, WWW 2009, 2009.

[16] Jian Hu, Lujun Fang, Yang Cao, et al. Enhancing Text Clustering by Leveraging Wikipedia Semantics. In Proc. Of SIGIR, 2008.

[17] J. Hassell, B. Aleman-Meza and IB Arpinar. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In Proc. of ISWC, 2006

[18] Kai-Hsiang Yang, Kun-Yan Chiou, Hahn-Ming Lee and Jan-Ming Ho. Web Appearance Disambiguation of Personal Names Based on Network Motif. In Proc. of WI, 2006.

[19] O. Medelyan, Ian H. Witten and D. Milne. Topic Indexing with Wikipedia. In WIKIAI, AAAI 2008. 2008.

[20] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In Proc. of CIKM. 2007.

[21] Michael Ben Fleischman. Multi-Document Person Name Resolution, In Proc. of ACL, 2004.

[22] Razvan Bunescu and Marius Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In Proc. of EACL, 2006.

[23] Ron Bekkerman and Andrew McCallum. Disambiguating Web Appearances of People in a Social Network. In Proc. of WWW, 2005

[24] Silviu Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In Proc. of EMNLP, 2007.

[25] Strube, M. and Ponzetto, S. P. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In Proc. of AAAI, 2006.

[26] Ted Pedersen, Amruta Purandare and Anagha Kulkarni. Name Discrimination by Clustering Similar Contexts. In Proc. of CICLing, 2005.

[27] Xiaojun Wan, Jianfeng Gao, Mu Li and Binggong Ding. Person Resolution in Person Search Results: WebHawk. In Proc. of CIKM, 2005.

[28] Ying Chen, James Martin. Towards Robust Unsupervised Personal Name Disambiguation. In Proc. of EMNLP, 2007.

[29] Hjørland, Birger. Semantics and Knowledge Organization. Annual Review of Information Science and Technology 41:367 -40, 2007.