# Explore Person Specific Evidence in Web Person Name Disambiguation

**Liwei Chen, Yansong Feng, Lei Zou, Dongyan Zhao**
Institute of Computer Science and Technology
Peking University
Beijing
{clwclw88,fengyansong,zoulei,zhaodongyan}@pku.edu.cn

## Abstract

In this paper, we investigate different usages of feature representations in the web person name disambiguation task which has been suffering from the mismatch of vocabulary and lack of clues in web environments. In literature, the latter receives less attention and remains more challenging. We explore the feature space in this task and argue that collecting person specific evidences from a corpus level can provide a more reasonable and robust estimation for evaluating a feature's importance in a given web page. This can alleviate the *lack of clues* where discriminative features can be reasonably weighted by taking their corpus level importance into account, not just relying on the current local context. We therefore propose a topic-based model to exploit the person specific global importance and embed it into the person name similarity. The experimental results show that the corpus level topic information provides more stable evidences for discriminative features and our method outperforms the state-of-the-art systems on three WePS datasets.

## 1 Introduction

Resolving ambiguity associated with person names found on the Web is a key challenge in many Internet applications, such as information retrieval, question answering, open information extraction, automatic knowledge acquisition(Wu and Weld, 2008) and so on. For example, if you want to know more about a guy named *George Foster* and feed Yahoo! with his name, the results are not satisfactory where you get more than 40 different persons named *George Foster* scattering in the top 100 returned pages. None of the dominant search engines currently helps users group those returned pages into clusters according to whether they refer to the same person. Users thus have to either read those pages carefully or adjust their queries by adding extra modifiers. This motivates an intensive study in automatically resolving person name ambiguity in various web applications.

However, resolving web person name ambiguity is not a trivial task. Due to the difficulties in figuring out or predicting the number of namesakes in the returned pages, the task has been investigated in an unsupervised learning fashion in the literature, which is apparently different from the traditional word sense disambiguation or entity linking/disambiguation tasks, where the inventories of candidate word senses or entities are usually known given the target word or entity mention.

A general framework for this task can be formulated as first extracting various features from the web pages, and then grouping these pages into several clusters each of which is assumed to represent one specific person. Despite of the inevitably noisy nature of web data, a key challenge is how to handle the data sparsity problem which we mean as: **mismatch of vocabulary** and **lack of clues**. The former refers to the case that two web pages may describe the same person but use different words thus the word overlap between them are small. Various features, including entities, biographical information, URL, etc., have been introduced to bridge the gap(Mann and Yarowsky, 2003; Kalashnikov et al., 2008a; Ikeda et al., 2009; Jiang et al., 2009),



*Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 832–842, Jeju Island, Korea, 12–14 July 2012. 

and external knowledge resources are also employed to capture the semantic relationship between entities(Han and Zhao, 2009, 2010). However, a more challenging scenario is that there are few clues available in the web pages. For example, there is a page mentioning a nutritionist *Emily Bender* in WePS2 dataset(Javier et al., 2009). Throughout the whole page we can find only one word, *nutrition*, related to her identification, while other pages about the nutritionist in the dataset contain substantial materials about her profession and job. In this case, current efforts, focusing on either feature engineering or background knowledge, are incapable to exploiting these limited clues from the current page to the whole *Emily Bender* document set, where *nutrition*, as an important feature for recognizing a nutritionist, should be paid more attention.

As far as we know, there is less work focusing on exploring person specific information to relieve the *lack of clues* problem. Traditional vector space model (VSM) is most widely used to accommodate various features, but it ignores any relations between them(Mann and Yarowsky, 2003; Ikeda et al., 2009). Beyond bag-of-features, two kinds of features are explored, co-occurrences of entities and Wikipedia based semantic relationship between entities, both of which provide a reasonable relatedness for entity pairs. More recent works adopt one of these relationships(Jiang et al., 2009; Kalashnikov et al., 2008a; Han and Zhao, 2009). Han and Zhao try to model both aspects, but their co-occurrence estimation, estimated from held-out resources, fails to capture the person specific importance for a feature, which is crucial to enhance limited clues in a corpus level, e.g., the significance of *nutrition* for *Emily Bender* in WePS1 dataset.

In this paper, we explore different usages of features and propose an approach which mines cross document information to capture the person specific importance for a feature. Specifically, we construct a semantic graph from Wikipedia concepts appearing in all documents that contain the target name (which we refer to *name observation set*), then group them into several topics and further weight each feature by considering both the relatedness of the feature to its corresponding topic and the importance of this topic in the current name observation set. By incorporating both the Wikipedia and topic information into our person name similarity, our model exploits both Wikipedia based background knowledge and person specific importance. We argue that the corpus level importance provides more stable evidences for discriminative features in various scenarios, especially the tough case. We compared our model with the state of the arts on three WePS datasets (from the First and Second Web People Search Clustering Task), and our experiments show that our model consistently outperforms other competitive models on all three datasets.

In the rest of this paper, we first review related work, and in Section 3, show how we exploit the person specific importance in our disambiguation model. Experiment results are discussed in Section 4. We conclude this paper in Section 5.

## 2 Related Work

Web person name ambiguity resolution can be formally defined as follows: Given a set of web pages $\{d_1, d_2, ..., d_n\}$, where each page $d_i$ $(i = 1, ..., n)$ contains an ambiguous name $N$ which may correspond to several persons holding this name among these pages. The disambiguation system should group these name observations into $j$ cluster $\{c_1, c_2, ..., c_j\}$ each of which is expected to contain web pages about the same person.

As mentioned before, the task is usually formulated in a unsupervised fashion, including two steps: feature extraction and person clustering. Most research efforts so far have been made to the former, exploring various features according to specific applications, while the second step is currently dominated by hierarchical agglomerative clustering (HAC). According to the reliance of extra knowledge resources, existing works can be categorized into non-resource methods and resource-based methods. Non-resource methods extract various local features from the context of ambiguous names, and compute the similarity between feature vectors. These features include plain words(Bagga and Baldwin, 1998), biographical information(Mann and Yarowsky, 2003; Niu et al., 2004), named entities, compound key phrases, hyperlinks(Ikeda et al., 2009), etc. The similarity between namesakes are usually measured by the cosine similarity(Bagga and Baldwin, 1998), or other graph based met-

rics(Iria et al., 2007; Kalashnikov et al., 2008a; Jiang et al., 2009). Those methods pay more attention to extracting informative features and their co-occurrences, but they usually treat the features locally, and ignore the semantic relatedness of features beyond the current document.

Resource-based approaches, on the other hand, can leverage external resources to benefit from rich background knowledge, which is crucial to remedy the data sparsity problem. The employed resources include raw texts available on the web and online encyclopedias. Kalashnikov et al. and Yiming et al. use extra web corpora to obtain co-occurrences between named entities. Rao et al. use Google Snippets to provide more contexts. By employing Wikipedia, the largest online encyclopedia, rich background knowledge about the semantic relatedness between entities can be leveraged to improve the disambiguation performance, and relieve the coverage problem, to some extent. Bunescu and Pasca and Cucerzan utilize Wikipedia's category hierarchy to disambiguate entities, while Pilz uses Wikipedia's link information. Han and Zhao adopt Wikipedia semantic relatedness to compute the similarity between name observations. They also combine multiple knowledge sources and capture explicit semantic relatedness between concepts and implicit semantic relationship embedded in a semantic graph simultaneously(Han and Zhao, 2010).

Most approaches discussed above explore various features in the current page or rely on external knowledge resources to bridge the vocabulary gap, but pay less attention to the *lack of clues* since they ignore the person specific evidence in the current corpus level. Our model focuses on solving the data sparsity problem by utilizing other web pages in the same name observation set to provide a robust but person specific weighting for discriminative features beyond the current document alone. In terms of extra resources, the Wikipedia based model (WS) by Han and Zhao (2009) is close to our model. The WS model uses Wikipedia to capture the relationship between entities in the local context to bridge the vocabulary gap, but it is incapable to evaluate the importance of a feature with regarding to the target name, hence is unable to make use of limited clues in the current web page. Our method captures person specific evidences by generating topics from all concepts in the current name observation set and weighting a feature accordingly. In this case, discriminative features that are sparse in the current page can be globally weighted so as to provide a more accurate and stable person name similarity.

## 3 The Model

Our model consists of three steps: feature extraction, topic generation and name disambiguation. For an ambiguous name, we first extract three types of features and construct a semantic graph from all Wikipedia concepts extracted from the current name observation set. We then collect global person specific evidences by clustering these concepts on the graph into different topics, which in turn are used to weight each concept by considering the importance of its corresponding topic in the current name observation set and its highly related neighbors in both the topic and its local context. At last, we incorporate the proposed topic representation into the person name similarity functionand adopt the hierarchical agglomerative clustering (HAC) algorithm to group these web pages.

### 3.1 Feature Extraction

We extract features from the contexts of ambiguous names, including Wikipedia concepts, named entities and biographical information, such as email addresses, phone numbers and birth years.

**Wikipedia Concept Extraction** Each concept in Wikipedia is described by an article containing hyperlinks to other concepts which are supposed to related to the current one. All the linking relations in Wikipedia construct a huge semantic graph, where we can mine rich semantic relationship between concepts(David and Ian, 2008). We collect Wikipedia concepts from all web pages in the dataset by comparing all n-grams (up to 8) from the dataset to Wikipedia anchor text dictionary and checking whether it is a Wikipedia concept surface form. We further prune the extracted concepts according to their keyphraseness(Mihalcea and Csomai, 2007). Initially, each concept is weighted according to its average semantic relateness(David and Ian, 2008) with other concepts in the current page.

**Named Entity and Biographical Information Extraction** Although Wikipedia concepts can pro-

vide rich background knowledge, they suffer from the limited coverage. It is common that some discriminative features are not likely to be found in Wikipedia, such as names of infamous people or organizations, email addresses, phone numbers, etc. We therefore extract two extra kinds of features, named entities that do not appear in the Wikipedia anchor text dictionary, and biographical information. We use Stanford Named Entity Recognizer(Finkel et al., 2005) to collect named entities which are not in the Wikipedia list. We use regular expressions to extract email address, phone numbers and birth years. For convenience, we will also call concept features for Wikipedia concept features and non-concept features for the other two in the rest of this paper.

### 3.2 Topic Generation and Weighting Scheme

Now we proceed to describe the key step of our model, topic generation and weighting strategy. The purpose of introducing topics into our model is to exploit the corpus level importance of a feature for a given name so that we will not miss any discriminative features which are few in the current name observation but have shown significant importance over the whole name observation set.

**Graph Construction** In our model, we capture the topic structure through a semantic graph. Specifically, for each name observation set, we connect all Wikipedia concepts appearing in the current observation set by their pairwise semantic relatedness-David and Ian (2008)to form a semantic graph.

The constructed graph is usually very dense since any pair of unrelated concepts would be connected by a small semantic relatedness resulting in many light-weighted or even meaningless edges. We therefore propose to prune some light-weighted edges to make the graph stable and easier to harvest reasonable topics. We use the following strategies to prune the graph:

- If an edge's weight is lower than a predefined threshold, it will be pruned.
- If two vertices of an edge do not co-occur in any web page of the current observation set, then this edge will be pruned.

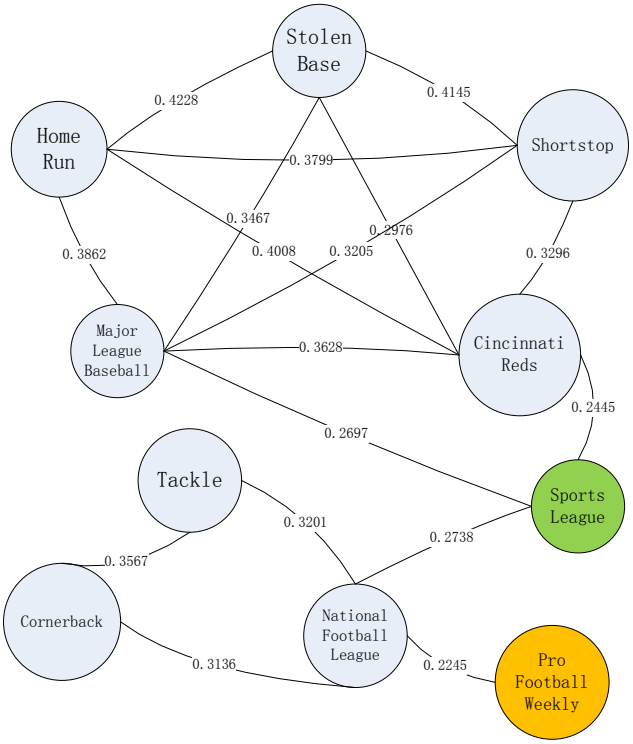


Figure 1: An abridged example of the semantic graph for *George Foster*. The green node *Sports League* is a hub node, and the yellow node *Pro Football Weekly* is an outlier.

The second rule is set to be strict and is proposed to handle the following circumstance. Some general concepts, such as swimming, football, basketball and golf, will be measured highly related with each other by Wikipedia semantic relatedness and thus are very likely to be grouped into one topic, however, they are discriminative on their own when disambiguating different persons. For example, the concept *swimming* is discriminative enough to distinguish Russian swimmer Popov from basketball player Popov. So it is not a good idea to group these concepts into one topic. The proposed co-occurrence rule is based on the above observation that it is rare that such kind of general concepts, e.g., swim and basketball, often co-occur with each other when talking about one specific person. After the pruning step, for each ambiguous name, we get a semantic graph from all Wikipedia concepts extracted in this name observation set. Figure 1 illustrates an abridged version of a semantic graph for *George Foster*.

**Graph Clustering** Considering the graph construction strategy we use, it is more suitable for us to group the concepts on the graph into several topics using a density-based clustering model.

We choose SCAN algorithm Xu et al. (2007) to perform the clustering step. The SCAN algorithm utilizes a neighborhood structure to measure the similarity between two vertices. If a vertex has a minimal of $\mu$ neighbors with a similarity larger than $\varepsilon$, it is called a *core*. The algorithm[1] starts from a random vertex in a graph, examining whether it is a core or not. If yes, the algorithm will expand a cluster from this vertex recursively, otherwise the vertex will be assigned either a hub node or an outlier depending on the number of its neighboring clusters. A hub node connects to more than one cluster, while an outlier connects to one or no cluster. Take the semantic graph in Figure 1 for example, the node *Sports League* is a hub node, while the node *Pro Football Weekly* is an outlier. Finally, all concepts in the graph are grouped into $K + 2$ parts ($K$ is the number of the clusters, and is determined automatically), including $K$ clusters, the set of hub nodes and the set of outliers.

One problem of applying SCAN in our work is that it is originally designed for unweighted graphs. We have to adapt it to our weighted graph by modifying the similarity function between two nodes as follows:

$$sim(c_1, c_2) = \alpha \times \frac{sim_{nb}(c_1, c_2)}{1+\alpha} + \frac{sr(c_1, c_2)}{1+\alpha} \quad (1)$$

and $sim_{nb}(c_1, c_2)$ is defined as:

$$sim_{nb}(c_1, c_2) = \frac{\sum_{c \in N(c_1) \cap N(c_2)} \frac{sr(c_1,c)+sr(c_2,c)}{2}}{|N(c_1) \cup N(c_2)|}$$

where $N(c)$ is the neighbor set of concept $c$. This new similarity function contains two parts: the neighborhood similarity and the semantic relatedness between two concepts. We combine them using a linear combination, where $\alpha$ is a weight tuned during training.

**Topic Generation** Now we will map the clustering results into different topics. Intuitively, each cluster will be treated as a topic. However, we found that hub nodes usually correspond to general concepts which may be related to many topics, but with a loose relatedness. We thus *distribute* each general concept into its every related topic, but with a lower weight to distinguish from ordinary concepts in this topic.

Outliers may be concepts which are far away from main themes of the corpus, or noise concepts. We calculate the average semantic relatedness of an outlier with its neighbor concepts that belong to one topic. If the result is lower than a threshold, this outlier will be discarded, otherwise it will be treated as a non-concept feature.

Now we are able to map the clustering results into different topics. Intuitively, each cluster will be treated as a topic. However, we found that hub nodes usually correspond to general concepts, e.g., *education* or *public*, which may be related to many topics, but with a loose relatedness. We thus *distribute* each general concept into its every related topic, but with a lower weight to distinguish from ordinary concepts in this topic. Outliers are found to contain concepts which are far away from main topics of the document set and look like noise concepts. We therefore calculate the average semantic relatedness of an outlier node with its neighboring concepts which belong to some topics. If the average relatedness is lower than a threshold, this node will be discarded, otherwise it will be treated as a non-concept feature.

**Weighting Topics** After generating all topics, we should weight each topic according to its importance in the current name observation set as well as the quality of the topic (cluster). Intuitively, if most concepts in the topic are considered to be discriminative in the current name set and they are closely related to each other, this topic should be weighted as important. By properly weighting the generated topics, we can capture the importance of a concept reliably in the corpus level (in the current name observation set) rather than in the current page solely.

Before we weight a topic, we first explain how we re-weight a hub concept in a topic since our initial feature weighting scheme(Han and Zhao, 2009) works on individual web page, lacks cross document information and is likely to over-estimate the impor-

[1]We omit the details of SCAN for brevity, and refer interested reader to Xu et al. (2007) for more details.

tance of a hub node (general concept) by by assigning a higher weight. Suppose a hub node $h$ connects to a topic $t$ with $n$ neighbors, namely $c_1, c_2, \cdots, c_n$. The similarity between this hub node and the topic is computed by averaging the semantic relatedness between this hub node and these $n$ neighbors:

$$sim(h,t) = \frac{1}{n}\sum_{i=1}^{n} sr(h,c_i). \quad (2)$$

We then update the weight of this hub node by considering its similarity with this topic: $w_t(h) = w(h) \times sim(h,t)$ from which we can see that the hub node receives a lower weight than before indicating that it is not as important as ordinary concepts in a topic.

Now we proceed to weight the topic $t$ by taking into account the frequencies of its concepts and the coherence between the concepts and their neighborhood in topic $t$:

$$w(t) = \frac{\sum_{i=1}^{n} f(c_i)}{n} \times \frac{\sum_{i=1}^{n} n\_coh(c_i,t)}{n} \quad (3)$$

where topic $t$ contains $n$ concepts $\{c_1, c_2, ..., c_n\}$, $f(c)$ is the frequency of concept $c$ over current name observation set, specially, when $c$ is a hub node concept, we will distribute its frequency according to equation (2), having $f_t(c) = f(c)sim(c,t)$. And $n\_coh(c,t)$ is the neighborhood coherence of concept $c$ with topic $t$, defined as:

$$n\_coh(c,t) = \frac{\sum_{q \in N(c) \cap t} sr(q,c)}{|N(c) \cap t|} \quad (4)$$

where $N(c)$ is the neighboring node set of concept $c$.

By incorporating corpus level concept frequencies into topic weighting, discriminative concepts that are sparse in one document and suppressed by conventional models can benefit from their corpus level importance as well as their coherence in related topics.

### 3.3 Clustering Person Name Observations

Now the remaining key step is to compute the similarity between two name observations. The similarity proposed in GRAPE(Jiang et al., 2009) measures two documents by bridge tags (common features) shared by two document graphs. Specifically, Jiang et al. utilize *cohesion* to weight a bridge tag in a document. The more bridge tags two documents share, the stronger the cohesion of each bridge tag is, and in turn the more similar the two documents are.

However, this similarity bears a shortcoming that the bridge tags shared by the two documents require an exact match of features, which does not take any semantic relatedness into consideration. If two web pages mentioning the same person but have few features in common, the GRAPE similarity may not work properly. We, therefore, propose a new similarity measure combining topic similarity, topic based connectivity strength and GRAPE's connectivity strength.

**Matching Topics to Person Name Observations** We first describe how to match the generated topics to different name observations. In order to avoid unreliable estimation, we only match a topic to a name observation when they share at least one concept. To measure the relatedness between a topic and a name observation, we formulate this similarity as the weighted average of semantic relatedness between each concept from one side and its closely related counterpart from the other side,defined as:

$$sim(A \to B) = \frac{\sum_{a \in A} w_A(a) \times w_B(b_a) \times sr(a,b_a)}{\sum_{a \in A} w_A(a) \times w_B(b_a)} \quad (5)$$
$$sim(A,B) = (sim(A \to B) + sim(B \to A))/2,$$

where $A$ can be a topic and $B$ a name observation or vice versa, $b_a$ is a concept in $B$ that is most related to concept $a$, $w_A(a)$ represents the weight of concept $a$ estimated by the averaged relatedness between $a$ and other concepts in $A$.

**Person Name Similarity** Now we describe the first component in our proposed measure: topic similarity, which is calculated through the common topics shared by the two name observations, $o_1$ and $o_2$:

$$TSm(o_1,o_2) = \sum_{t \in T(o_1,o_2)} sim(o_1,t) \times sim(o_2,t) \times sim(o_1 \cap t, o_2 \cap t) \times w(t) \quad (6)$$

where $T(o_1,o_2)$ contains all common topics of $o_1$ and $o_2$, $w(t)$ is the weight of topic $t$ estimated using

equation (3), both $sim(o_i, t)$ and $sim(o_1 \cap t, o_2 \cap t)$ measure the similarity between two concept sets and can be estimated using equation (5). The underlying idea of the equation is, if two name observations share more and closer common topics, and also these topics receive higher weights according to the current name observation set, then the two observations should be more related to each other.

Specifically, the factor $sim(o_1 \cap t, o_2 \cap t)$ is designed to measure the fine relatedness between $o_1$ and $o_2$ given the topic $t$. Sometimes, both $o_1$ and $o_2$ are mapped to $t$ and both close to this topic, but in fact they depict different aspects of $t$ since some of our topics are more general thus include several aspects. The comparison of their intersections will provide a more detailed view for their similarity.

Inspired by the use of bridge tags in GRAPE(Jiang et al., 2009), we propose to capture the connection strength between concept sets by the means of our topics. We consider common topics as the bridge tags and define our topic based connectivity strength between two name observations as:

$$TCS(o_1, o_2) = \frac{1}{2} \sum_{t \in T(o_1, o_2)} sim(o_1 \cap t, o_2 \cap t) \times (Cohs(o_1, t) + Cohs(o_2, t)) \quad (7)$$

Note that we still need $sim(o_1 \cap t, o_2 \cap t)$ to capture the fine differences inside a topic. $Cohs(o, t)$ is a cohesion measure to capture the relatedness between non-concept features in $o$ and concept features in $t$, defined as:

$$Cohs(o, t) = \sum_{c \in o \cap t} w(t) \times \sum_{q \in EB(o)} occ(c, q) f_o(c) f_o(q) \quad (8)$$

where $EB(o)$ contains all non-concept features in $o$ (e.g., non-Wikipedia entities and biographical information), $occ(c, q)$ is the co-occurring number of concept $c$ and feature $q$, $f_o(q)$ is the relative frequency of $q$ in observation $o$. It is easy to find that a higher cohesion can be achieved by larger overlap between $o$ and $t$, higher topic weight and more co-occurrences of concept features in $t$ and other features in $o$.

The third part is the original connectivity strength defined in GRAPE(Jiang et al., 2009): $CS(o_1, o_2)$, calculated using plain features without topics (we omit the details for brevity). Finally, we linearly combine equation (6), (7) and $CS(o_1, o_2)$ into the person name similarity function as:

$$S(o_1, o_2) = \alpha_1 \times TSm(o_1, o_2) + \alpha_2 \times TCS(o_1, o_2) + (1 - \alpha_1 - \alpha_2) \times CS(o_1, o_2) \quad (9)$$

where $\alpha_1$ and $\alpha_2$ are optimized during training.

This final similarity function will then be embedded into a normal HAC algorithm to group the web pages into different namesakes where we compute the centroid-based distance between clusters(Mann and Yarowsky, 2003).

## 4 Experiments

We compare our model with competitive baselines on three WePS datasets. In the following, we first describe the experimental setup, and then discuss the their performances.

### 4.1 Data

**Wikipedia Data** Wikipedia offers free copies of all available data to interested users in their website. We used the one released in March 6th, 2009 in our experiments. We identified over 4,000,000 highly connected concepts in this dump; each concept links to 10 other concepts in average.

**WePS Datasets** We used three datasets in our experiments, WePS1 Training and Testing (Artiles et al., 2007), WePS2 Testing (Javier et al., 2009). These datasets collected names from three different resources including Wikipedia names, program committee of a computer science conference and US census. Each name were queried in Yahoo! Search and top $N$ result pages (100 pages in WePS1 and 150 pages in WePS2) were obtained and manually labeled.

### 4.2 Baselines

We compare our model TM with four baseline methods: (1)VSM: traditional vector space model with cosine similarity. We use features extracted in Section 3.1 and weight them using TFIDF. The documents are grouped using standard HAC algorithm. (2)GRAPE(Jiang et al., 2009): we re-implement the state-of-the-art system which outperforms any models that do not use extra knowledge resources reported in WePS1 and WePS2. (3)WS: the Wikipedia

Semantic method(Han and Zhao, 2009). This system uses Wikipedia to enhance the results of name disambiguation. (4)SSR: the Structural Semantic relatedness model(Han and Zhao, 2010) creates a semantic graph to re-calculate the semantic relatedness between features, and captures both explicit semantic relations and implicit structural semantic knowledge. We also build two variants of TM: TM-nTW which removes topic weighting to examine what effect the topic weighting strategy can make and whether it can provide a person specific evidence and TM-nCP which does not use co-occurring information to prune the semantic graph to examine whether the pruning is effective.

### 4.3 Parameters

There are several parameters to be tuned in our model. In the SCAN algorithm, we use default parameters according to (Xu et al., 2007) with an exception: the weight $\alpha$ is tuned exhaustively to be 0.2. Note that the number of topics are automatically decided by SCAN. The semantic graph pruning threshold is set to 0.27 tuned on a held out set. The smoothing parameters in equation (9) are: $\alpha_1 = 0.3$, $\alpha_2 = 0.2$ which are tuned using cross validation. Optimization of some parameters will be addressed in detail in the following subsection. In HAC, all optimal merging thresholds are selected by applying leave-one-out cross validation.

### 4.4 Results and Discussion

We adopt the same evaluation process as (Han and Zhao, 2009), and evaluating these models using Purity, Inverse Purity and the F-measure (also used in WePS Task Artiles et al. (2007)). The overall performance is shown in Table 1, and the best scores are in boldface.

Let us first look at our model and its variants, TM-nTW and TM-nCP. By introducing the corpus level topic weighting scheme, our model improves in average 1.6% consistently over all datasets. Recall that our topic weightings are obtained over the whole name observation set beyond local context, this improvement indicates that this corpus level person specific evidences render the person similarity more reasonably than that of single document. On the other hand, by pruning the semantic graph, our model improves averagely 1.3% over TM-nCP. This shows that our co-occurrence based pruning strategy can help render the semantic graph with less noisy edges, thus generate more reasonable topics.

Table 1: Web person name disambiguation results on all three WePS datasets

| **WePS1 Training** | | | |
|---|---|---|---|
| **Method** | **P** | **IP** | **FMeasure** |
| *VSM* | 0.86 | 0.86 | 0.85 |
| *GRAPE* | **0.93** | 0.90 | **0.91** |
| *WS* | 0.88 | 0.89 | 0.87 |
| *SSR* | 0.82 | **0.92** | 0.85 |
| *TM-nTW* | 0.91 | 0.89 | 0.90 |
| *TM-nCP* | 0.92 | 0.90 | **0.91** |
| *TM* | **0.93** | 0.91 | **0.91** |
| **WePS1 Testing** | | | |
| **Method** | **P** | **IP** | **FMeasure** |
| *VSM* | 0.79 | 0.85 | 0.81 |
| *GRAPE* | 0.93 | 0.83 | 0.87 |
| *WS* | 0.88 | **0.90** | 0.88 |
| *SSR* | 0.85 | 0.83 | 0.84 |
| *TM-nTW* | 0.93 | 0.85 | 0.88 |
| *TM-nCP* | 0.92 | 0.86 | 0.88 |
| *TM* | **0.94** | 0.86 | **0.90** |
| **WePS2 Testing** | | | |
| **Method** | **P** | **IP** | **FMeasure** |
| *VSM* | 0.82 | 0.87 | 0.83 |
| *GRAPE* | 0.88 | **0.90** | 0.89 |
| *WS* | 0.85 | 0.89 | 0.86 |
| *SSR* | 0.89 | 0.84 | 0.86 |
| *TM-nTW* | 0.92 | 0.87 | 0.89 |
| *TM-nCP* | **0.93** | 0.88 | 0.90 |
| *TM* | **0.93** | 0.89 | **0.91** |

Generally, our proposed model works best consistently over all three datasets. Our method gains 9.3% improvement on average in three datasets compared with VSM, 1.7% improvement compared to GRAPE, 3.8% over WS and 6.7% over SSR. We also performed significance testing on F-measures: the differences between our model and other models are significant. We notice there are many noisy or short web pages which lead to inaccurate concept extraction, but this cross document evidences, to some extent, can remedy this. In the *Emily Bender* example, our system correctly groups the odd page, which contains limited clues, into the *nutritionist* cluster,

while the rest, excluding WS and SSR, failed. Surprisingly, SSR combines both kinds of relations and implicit structural knowledge, but performs in the same bulk with VSM in WePS1 training set. We think the reason may be that some name observation sets are too small to estimate non-concept relatedness via random walk. In WePS1 training set, many names in this dataset contains several namesakes, each of which corresponds to a few web pages. In this case, our corpus level weighting scheme and WS show no advantage over GRAPE which considers word co-occurrences solely. From the results, we can also find that there is no clear winner between GRAPE and WS. The former does not use Wikipedia relatedness but only includes local relationship, and performs even slightly better than WS in WePS2, which indicates that non-Wikipedia concepts are important disambiguation features as well.

### 4.5 Parameter Optimization

In this subsection, we discuss the optimization of several parameters in the proposed method. In total we need to set four parameters. The first one is the edge pruning threshold during graph construction; the second one is the weight $\alpha$ in SCAN algorithm; the third one and the forth one are the combination parameters in the final similarity function. We will address the first two in the following. The last two combination parameters are tuned by exhaustively searching the space and omitted here for brevity

First, we configure the pruning threshold. Intuitively, larger threshold can prune more unimportant edges and improve the disambiguation performance. However, if the threshold is too large, we may prune important edges and harm the results. The F-measure of our method with respect to the pruning threshold is plotted in Figure 2.

From Figure 2, we can know that in all three data sets, a pruning threshold of 0.27 will lead to the best performance. Both increasing and decreasing of this pruning threshold will cause a decline of the F-Measure, because they will either leave more noisy light-weighted edges or prune some important edges.

Secondly, we configure the neighborhood similarity weight. The larger this weight is, the more neighborhood information can influence the similarity between two nodes in the semantic graph. We plot the performance of our method regarding to the neighborhood similarity weight in Figure 3.

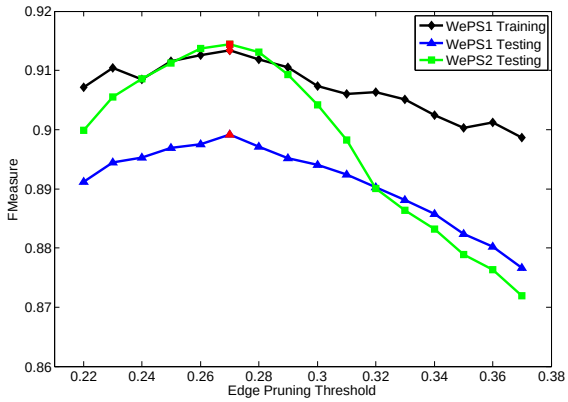


Figure 2: The F-Measure v.s. the edge pruning threshold on three data sets.

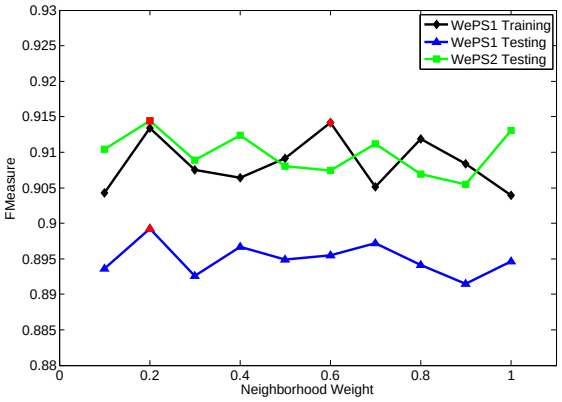


Figure 3: The F-Measure v.s. the neighborhood similarity weight on three data sets.

From Figure 3, we know that for the WePS 1 Testing and WePS2 Testing data sets, a neighborhood similarity weight of 0.2 can result in the best performance, but for WePS 1 Training set, the weight for the best performance is 0.6. In fact, when the neighborhood similarity weight varies from 0 to 1, the difference between the best and worst performance are less than 0.01, which indicates that neighborhood similarity is as considerable as semantic relatedness.

## 5 Conclusion and Future Work

In this paper, we explore the feature space in the web person name disambiguation task and propose a topic-based model which exploits corpus level person specific evidences to handle the data sparsity challenges, especially the case that limited evidences can be collected from the local context. In particular, we harvest topics from wikipedia concepts appearing in the name observation set, and weight a concept based on both the relatedness of the concept to its corresponding topic and the importance of this topic in the current name observation set, so that some discriminative but sparse features can obtain more reliable weights. Experimental results show that our weighting strategy does its job and the proposed model outperforms the-state-of-the-art systems. Our current work utilizes the topic information shared in one name observation set but is incapable to handle sparse name set, which needs more accurate relation extraction inside the name observations. Jointly modeling entity linking and person (entity) disambiguation tasks will be an interesting direction where the two tasks are closely related and usually need to be considered at the same time. Investigating the person name disambiguation task in different web applications will also be of great importance, e.g., disambiguating a name in streaming data or during knowledge base construction. In addition, graphical model, which has been studied in academic author disambiguation, may be a good choice to cope with the noises and non-standard forms in web data.

## Acknowledgments

We would like to thank Yidong Chen, Wei Wang and Tinghua Wang for their useful discussions and the anonymous reviewers for their helpful comments which greatly improved the work and the presentation. This work was supported by the National High Technology Research and Development Program of China (Grant No. 2012AA011101), National Natural Science Foundation of China (Grant No.61003009) and Research Fund for the Doctoral Program of Higher Education of China (Grant No.20100001120029).

## References

Artiles, J., Gonzalo, J., and Sekine, S. (2007). The semeval-2007 weps evaluation: establishing a benchmark for the web people search task. In *SemEval*, SemEval '07, pages 64–69, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bagga, A. and Baldwin, B. (1998). Entity-based cross-document coreferencing using the vector space model. In *ACL*, pages 79–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bunescu, R. C. and Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716. ACL.

David, M. and Ian, H. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *AAAI*, AAAI '08.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

Han, X. and Zhao, J. (2009). Named entity disambiguation by leveraging wikipedia semantic knowledge. In *CIKM*, CIKM '09, pages 215–224, New York, NY, USA. ACM.

Han, X. and Zhao, J. (2010). Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In *ACL*, ACL '10, pages 50–59, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ikeda, M., Ono, S., Sato, I., Yoshida, M., and Nakagawa, H. (2009). Person name disambiguation on the web by two-stage clustering. In *WWW*.

Iria, J., Xia, L., and Zhang, Z. (2007). Wit: web people search disambiguation using random walks. In *SemEval*, SemEval '07, pages 480–483, Stroudsburg, PA, USA. Association for Computational Linguistics.

Javier, A., Julio, G., and Satoshi, S. (2009). Weps 2 evaluation campaign: Overview of the web people search clustering task. In *WWW 2009*.

Jiang, L., Wang, J., An, N., Wang, S., Zhan, J., and Li, L. (2009). Grape: A graph-based framework for disambiguating people appearances in web search. In *ICDM*, ICDM '09, pages 199–208, Washington, DC, USA. IEEE Computer Society.

Kalashnikov, D. V., Chen, Z., Mehrotra, S., and Nuray-Turan, R. (2008a). Web people search via connection analysis. *IEEE Trans. on Knowl. and Data Eng.*, 20:1550–1565.

Kalashnikov, D. V., Nuray-Turan, R., and Mehrotra, S. (2008b). Towards breaking the quality curse.: a web-querying approach to web people search. In *SIGIR*, SIGIR '08, pages 27–34, New York, NY, USA. ACM.

Mann, G. S. and Yarowsky, D. (2003). Unsupervised personal name disambiguation. In *CONLL*, CONLL '03, pages 33–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of CIKM'07*, pages 233–242.

Niu, C., Li, W., and Srihari, R. K. (2004). Weakly supervised learning for cross-document person name disambiguation supported by information extraction. In *ACL*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pilz, A. (2010). Entity disambiguation using link based relations extracted from wikipedia. In *ICML*.

Rao, D., Garera, N., and Yarowsky, D. (2007). Jhu1: an unsupervised approach to person name disambiguation using web snippets. In *SemEval*, SemEval '07, pages 199–202, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wu, F. and Weld, D. S. (2008). Automatically refining the wikipedia infobox ontology. In *WWW*, WWW '08, pages 635–644, New York, NY, USA. ACM.

Xu, X., Yuruk, N., Feng, Z., and Schweiger, T. A. J. (2007). Scan: a structural clustering algorithm for networks. In *Proceedings of KDD*, KDD '07, pages 824–833, New York, NY, USA. ACM.

Yiming, L., Zaiqing, N., Taoyuan, C., Ying, G., and Ji-Rong, W. (2007). Name disambiguation using web connection. In *AAAI*.