2009 Ninth IEEE International Conference on Data Mining

GRAPE: A Graph-Based Framework for Disambiguating People Appearances in Web Search

Lili Jiang†, Jianyong Wang§, Ning An†, Shengyuan Wang§, Jian Zhan§, Lian Li†
†Eng. Res. Ctr. of Open Source Software and Real-time Syst., Ministry of Education
School of Information Science and Engineering, Lanzhou University, China
jianglili06@lzu.cn; {nan,lil}@lzu.edu.cn
§Department of Computer Science and Technology, Tsinghua University, Beijing, China
{jianyong,wwssyy,zhanjian}@tsinghua.edu.cn

***Abstract*—Finding information about people using search engines is one of the most common activities on the Web. However, search engines usually return a long list of Web pages, which may be relevant to many namesakes, especially given the explosive growth of Web data. To address the challenge caused by name ambiguity in Web people search, this paper proposes a novel graph-based framework, GRAPE (abbr. a Graph-based fRamework for disAmbiguating People appEarances in Web search). In GRAPE, people tag information (e.g., people name, organization, and email address) surrounding the queried people name is extracted from the search results, a graph-based unsupervised algorithm is then developed to cluster the extracted tags, where a new method, $Cohesion$, is introduced to measure the importance of a tag for clustering, and each final cluster of tags represents a unique people entity. Experimental results show that our proposed framework outperforms the state-of-the-art Web people name disambiguation approaches.**

***Keywords*-People Name Disambiguation, Named Entity, Tag Extraction, Clustering**

I. Introduction

Statistics show that people search makes up 30% of all searches on Google or Yahoo [1]. However, due to the increasing availability of Web data, finding the relevant information about a particular person becomes more and more difficult. People information is often scattered across many pages, and names are heavily ambiguous in Web people search. Unfortunately, regular search engines do not provide effective solutions for people search. For example, when Google is queried with a person name of "John Smith", the top 100 returned Web pages refer to at least 10 different namesakes, including famous English soldier, singer, and common people. In order to find the information about the target "John Smith", users are forced to either add keywords (which could lead to degradation of recall) or browse every Web page (which is time-consuming). Thus, solving the problem of name ambiguity is a major concern in Web people search.

Although several research efforts have been made to address Web people search and related challenges [2][3][4][5], more efficient and effective approach in Web people name disambiguation has yet to be developed.

In this paper, a people entity denotes "a unique person", and a people name may correspond to several people entities. It is observed that people tag information (e.g., organization, location, and occupation) is more informative for users than the set of Web pages retrieved by search engines. Taking a people name query "John Smith" for example, if tags "politician" and "Glasgow University Union" co-occur in several retrieved pages, it may be suggested that these pages describe the same people entity of "John Smith". The co-occurrence of these tags can help achieve high disambiguation quality. However, not any tag set can represent a unique people entity, for example, a combination of tags "politician" and "USA" are more likely to refer to multiple people entities. This suggests that different types of tags make different contributions in name disambiguation, and we will make full use of this observation in this work.

Given a people name as a query, let document corpus $D = \{d_1, d_2, \ldots, d_n\}$ be the set of the top n returned results from a search engine. Differently from the previous methods [6][5][7][8], this paper proposes a weighted graph-based framework, GRAPE, to both disambiguate and tag people appearances in Web search. First, eight types of tags are extracted from each document $d \in D$, including people name, organization, location, email address, phone number, birth date, occupation, and URL domain. Then the proposed framework views the unique tag corpus $A = \{a_1, a_2, \ldots, a_m\}$ extracted from D as a graph with m nodes, where a node is created for each unique tag $a \in A$, and an edge is added between two tags when they co-occur in the same document. Secondly, the importance of each tag in A is measured, and a clustering algorithm is performed on the graph to group these tags into clusters, while the tags in each cluster are used to represent a certain people entity.

The contributions of this work are summarized as follows.

- A novel weighted graph model is introduced to represent the relationships among the tags.
- An effective method is proposed to measure the importance of a tag, and a graph clustering algorithm is devised.

1550-4786/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDM.2009.25

- Experiments on four real data sets have been conducted and the empirical results indicate that the GRAPE framework is very effective in disambiguating Web people names and outperforms the previous approaches.

The rest of this paper is organized as follows. In Section II, we introduce the related work. Section III provides some details of the graph-based framework. Section IV presents the experimental results. We conclude our paper and discuss some future work in Section V.

II. Related Work

Numerous approaches have been mainly proposed for author disambiguation in publication citations and Web people name disambiguation [9][10][2][11], among which we focus on the later. In [2], Bagga and Baldwin applied agglomerative clustering technique based on context similarity. However, they tested in their experiment a single person name of "John Smith", which is insufficient for generalizing the real world challenge. Except for using the context information in name disambiguation, the approach taken by [8] applies clustering techniques over the rich extracted biographic features. This study further shows the importance of categorical information (e.g., occupation and birth year), which can improve the performance of namesake distinction. Additionally, [4] combines two unsupervised frameworks to leverage an existing social network of people to aid disambiguation, among which one is based on link structure of the Web pages while another adopts agglomerative/conglomerative double clustering. Experimental results show that the hybrid model is effective in name disambiguation but requires background knowledge about the target person.

Since most of the given corpus include limited information about the ambiguous people name, online resources are often used to mine more information about the targeted people name. [6] creates extended queries to a Web search engine to disambiguate different people with the same name. [5] tries to utilize the online encyclopedia for people information detection and disambiguation. Moreover, Kalashnikov et al. [12] presented a new Web people search approach based on collecting co-occurrence information from the Web. Their method performs well on the name disambiguation task, however, it is more suitable on severside since much time is consumed on the task of Web co-occurrence counting.

Some graph-based approaches have also been proposed to solve similar problems. [13] proposes a method to represent a corpus of email messages as a graph and effectively solves the problem of name ambiguity in email documents. Likewise, [14] builds a graph for Web people search results, each node represents one text type, such as token, webpage, metadata, title, or body. The authors proposed a random walk model to determine whether two Web pages in the graph refer to the same person. However, their experimental study shows that the performance of this approach is not very good.

One of the major tasks of workshops SemEval-2007 [15] and WePS-2009 [16] is to disambiguate people names in a Web searching scenario. Besides, many meta-search engines including Vivisimo (http://ww-w.vivisimo.com) and Carrot2 (http://www.carrot2.org) are working on clustering to resolve the name ambiguity problem. Recently, some commercial systems for people search have also emerged, such as Wink (www.wink.com) and Spock (www.spock.com).

This paper aims to solve the problem of name ambiguity in Web people search. Differently from others, our work uses a graph-based framework to enable a wider range of information (e.g., people tag types and tag relationships). Additionally, a new method is proposed to measure the importance of people tags in name disambiguation. Note that a very preliminary version of the paper was published as a poster in WWW'09 conference [17]. In this study, we make further enhancements, and especially provide large amounts of experimental validation for the proposed framework.

III. The GRAPE Framework

A people name query in this paper consists of a first name and a last name, and a collection of documents obtained by inputting the query to a search engine is denoted by $D = \{d_1, d_2, \ldots, d_n\}$. Assume D contains a set of people entities, and each people entity is relevant to a few tags. Our approach for Web people appearance disambiguation is to group the extracted tags into different clusters, such that each resulting tag cluster corresponds to a people entity.

Figure 1 depicts the GRAPE framework. After preprocessing and extracting people tags from the set of documents D, a graph is modeled on the extracted tags, and then a clustering algorithm is performed on the graph with the purpose of obtaining the final tag clusters. In this section, we describe the framework in detail.

A. Preprocessing

Preprocessing is a critical initial step to clean and select data for further development. Given the document corpus D, some noisy documents which are irrelevant to the given people name should be removed. We first fix three legal mentions of the given queried people name, namely, <$firstname$> <$lastname$>, <$firstname$> <$middlename$> <$lastname$>, and <$lastname$> <$firstname$>. Meanwhile, a constraint of word boundary is employed to detect illegal people name mentions. For example, with respect to the name query of "Edward Fox", the mention of "Edward Foxe" will be ignored. Then documents without any legal mention of the given people name are discarded. Next, the remaining documents are cleaned off their HTML markups and Javascript coding. Finally, we transform both the html escape characters and ASCII code to the source string, for

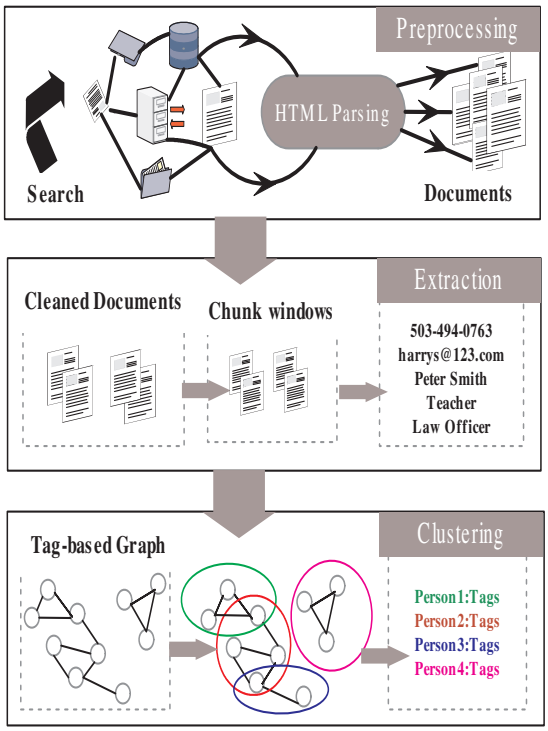

Figure 1. The GRAPE Framework

example, " " is converted to blank, and "A" is converted to letter "A".

As the preprocessing proceeds, junk documents and format contents in the documents are filtered out. However, a Web document may not totally focus on the queried people name, but also contain other irrelevant tag information. To compensate for this, we apply a chunk window around each occurrence of the given people name. We have tested different window sizes on training data and compared the disambiguation performance versus window sizes as shown in Figure 2. In the figure, "Full" denotes the entire document, F_{PI} and F_{EB} are the performance evaluation measures, which will be discussed in detail in Section IV-B. The experimental results show that chunk window of 2500 makes the best performance. Although the difference between using the window size of 2500 and using the full document is not large, several unusually long documents always happen, which dramatically increases the burden of running time. Therefore, the chunk window of 2500 characters in length before and after the name is recommended. To avoid information loss, the chunk window starts (ends) at the beginning (end) of a complete sentence and an overlap between the windows results in a merged window. These chunk windows can serve two purposes. On the one hand, it avoids extracting some noisy tags, on the other hand, it can balance the length of documents in D to some extent.

It must be noted that accurate tag extraction often depends

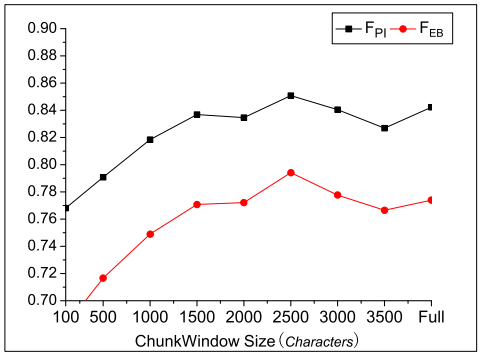

Figure 2. Chunk Window Validation

on the rich context information surrounding the queried people name. In order to preserve complete context information, stopwords (i.e., high frequency common words) removal and word stemming (i.e., reducing derived words to their stem form) are not performed at this step. For instance, in the context of "John Smith was appointed as Foundation Professor at the University of Sydney", if all the stopwords "as", "at", "the", and "of" are removed beforehand, the tag "University of Sydney" is probably missed in extraction.

B. Tag Extraction

As shown in Figure 1, after preprocessing, tags are extracted from the chunk windows in each document. Many different types of tags can be used to describe a person. In this paper, we select eight common types of people tags including people name, organization, location, email address, phone number, birth date, occupation, and URL domain. Intuitively, some combinations of these tags can uniquely characterize a specific people entity. In this subsection, we introduce the tag extraction patterns and filtering rules.

1) Extraction Patterns: Both model-based and rule-based methods are employed in tag extraction. People name, organization, and location are detected using both the character language model and hidden Markov model from a natural language processing toolkit Lingpipe [1], other types of tags are extracted based on a set of extraction rules.

Email address is a string of characters separated into two parts by an "@", that is, "username@domain-name". Phone number is a sequence of numbers, whose format is standardized by ITU-T (http://www.i-tu.int/ITU-T/). The phone number pattern accepts tags in both local format (e.g., 6258657) and international format (e.g., 1-626-780-7552). In addition, a number of variants of the phone number are considered. For example, "(503) 494-6862" are often represented as "503-494-6862" or "5034946862". According

[1]http://alias-i.com/lingpipe/

Table I
EXTRACTION RULES FOR BIRTH DATE, OCCUPATION, AND URL DOMAIN

Tag Type	Extraction Pattern
Birth Date	<*name*> was born on <*birthdate*>
	<*name*> was born in <*birthdate*>
	<*name*> (<*birthdate*> - ####)
	<*name*> b. <*birthdate*>
	<*name*> (<*birthdate*>)
	<*name*> (#### <*birthdate*>)
	<*name*> #### *birth* <*birthdate*>
Occupation	<*name*>(<*occupation*>, ####)
	<*name*>, <*occupation*>
	<*name*>*was/is a/an*<*occupation*>
URL domain	^https?://<*domain*>(/|$)

to the regular standards discussed above, rules are generated for email address and phone number respectively.

In order to improve the extraction accuracy, we combine the extraction rules and a predefined common occupation dictionary for occupation detection, that is, $occupation$ must be one mention in the dictionary, which is collected from Wikipedia in advance. Additionally, as the full URLs of the hyperlinks seem to be too specific, we define a rule to output the root of each URL. For example, given the extracted URL "http://www.macombbar.org/associations/", the rule will return "www.macombbar.org" as the URL domain tag.

Table 1 shows the extraction rules for birth date, occupation and URL domain, where $name$ denotes one of the legal mentions of the queried people name, $birthdate$ is in the form of date, $occupation$ is one mention in the common occupation dictionary, and $domain$ represents the extracted URL domain name.

2) Filtering Rules: The extracted tags are used to identify each people entity for a given people name. And yet, preprocessing is not perfect, and some extracted tags make little sense for this purpose, which will be filtered out from the tag corpus as follows.

- The stopwords, which are commonly used, often appear in the context of different people entities. To detect stopwords, 15000 common terms are selected from Wikipedia (http://www.wikipedia.org) to form a stopword list S. If a tag is in S, it will be filtered out.
- Generally speaking, except some legal characters like '–', '&' in organization and location, '@','_', '–' in email address or '–' ,'(', ')', '.' in phone number, a tag containing some special characters like '!', '=', '/' is likely to be an invalid tag. The filter algorithm will delete any tags containing these illegal characters.
- Some extracted emails, such as "webmaster@domain-name, support@domain-name, feedback@domain-name, info@domain-name" are not relevant with the query name but often occur as the contact of a Web site, therefore, they are removed from the tag set.
- Suppose the query name is "John Smith", tags such as "John K. Smith", "John Duin Smith", "John", "Smith", "John Smith" or "Smith John" are often extracted as related people names, "John K. Smith" and "John Duin Smith" may be the full names of certain "John Smith" and should be retained for disambiguation. But the others mentioned above do not have any effect in disambiguating different "John Smith" in the following clustering and will be discarded.

The quality of tags is extremely important for clustering. These filtering rules, which can improve the accuracy of tag extraction, lead to better clustering results and help represent people entity characteristically.

C. Graph Modeling

Through Web document preprocessing and tag extraction, eight types of tags $T = \{$ people name, organization, location, email address, phone number, birth date, occupation, URL domain$\}$ are extracted, and a union of eight tag sets $A = \bigcup_{i=1}^{8} T_i$ is generated, where T_i contains the non-repetitive (i.e., unique) tags with the type $t_i \in T$.

Regarding the tag corpus A for a people name, we build an undirected labeled graph $G = (V, E)$, where the node set $V = \{v_1, v_2, \ldots, v_m\}$ is a set of unique tags (where m is the total number of unique tags), and each edge $(v_i, v_j) \in E$ represents the co-occurrence of the tags corresponding to v_i and v_j in the same document in D. The graph G for any given query name is not always a connected graph. In the following, we first give some definitions, then introduce how to compute a weight for each edge and each node in G.

1) Definition: Given a tag graph $G{=}(V, E)$ and one of its subgraphs $G'{\subset}G$, if the subgraph G' is a maximal clique subgraph containing all the tags in a single document, it is called a **micro-cluster**. In fact, micro-clusters denote the initial co-occurrence relationships among the tags. Figure 3 gives an example of a tag graph and there are in total four micro-clusters. In the following we introduce several other definitions.

Definition 1. (**Bridge-tag**) Node v is defined as a bridge-tag, if v links at least two micro-clusters. In Figure 3, nodes 3 and 6 each corresponds to a bridge-tag.

Definition 2. (**Reachable**) Any node v is reachable from w, if there is a chain of nodes $u_1, u_2, \ldots, u_m$, $u_1{=}v$, $u_m{=}w$, such that there is an edge between u_i and u_{i+1}, $\forall i \in \{1, \ldots, m-1\}$, which can be equivalently formalized as:

$$Reachable(v, w) \iff$$
$$\forall v, w \in V, \text{if } \exists u_1, \ldots, u_n \in V : u_1 = v \wedge\ u_n = w \wedge (u_i, u_{i+1}) \in E$$

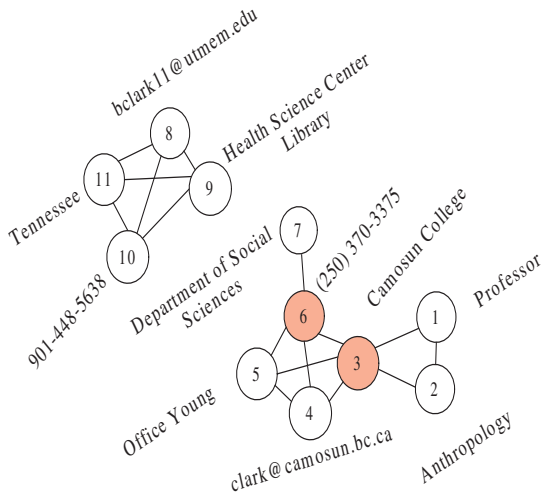

Figure 3. An Example of the Tag Graph

Definition 3. (**Macro-cluster**) A subgraph C is called a macro-cluster when it consists the maximal micro-clusters, which interconnect with strong connectivity strength. Details on the measure of connectivity strength are postponed to be discussed in Section III-D, and we note that each bridge-tag belongs to at least one macro-cluster. According to the definition of macro-cluster, we can get the following property.

Property 1. If C is a macro-cluster, all nodes in C are $reachable$ from each other and C has the maximal reachability (i.e., if a non-bridge tag v is in C, any other non-bridge tags connected with v are also in C), formally:

$Macro\text{-}cluster(C) \Rightarrow$

(1) $\forall v, w \in C : Reachable(v, w)$
(2) $\forall v, w \in V \wedge v, w$ *are not bridge-tags*: $v \in C \wedge (v, w) \in E \rightarrow w \in C$

Proof:

Let tags v, w belong to micro-clusters M_v and M_w respectively. The proof is outlined as follows.

(1) We first prove the reachability of a macro-cluster C. Assume that $\exists v, w \in C$ are not reachable, so any tag in M_v is also not reachable from tags in M_w. Then the connectivity strength between M_v and M_v is zero, which draws the conclusion that v and w are not in the same macro-cluster. Therefore, we conclude that $\forall v, w \in C$, Reachable(v, w) must be true.

(2) Suppose $v \in C, w \in V$ and v, w are not bridge tags, we have $M_v \subseteq C$. Since $(v, w) \in$ E, we can get $w \in M_v$ and $M_w = M_v$. Thus, we conclude that $w \in C$.

2) Edge Weighting: Roughly speaking, if two distinct tags co-occur frequently in multiple documents, they may be strongly relevant. Accordingly, in the proposed graph model, an edge indicates the relevance of the connected nodes corresponding to tags v and w, and the edge weight, $edgeweight(v, w)$, is defined as the number of documents where v and w co-occur.

3) Node Weighting: Although the eight tag types are all used to describe people on the Web, different types of tags have their own characteristics. Generally, there is a low probability that an email address is shared by two namesakes, while two namesakes could be relevant with the same organization or location. This observation leads us to assign different type weights to nodes with different tag types.

Intuitively, an ideal tag type should have the following property that each of its tags is uniquely associated with one people entity, while the more the number of distinct tags of one type for each people entity w.r.t. a query name, the less this tag type contributes in name disambiguation. For instance, if a person has cooperated with two companies, two different company names may occur around this person name in two documents d_1 and d_2 respectively. Obviously, the company names (with the tag type of "organization") misguide name disambiguation regarding whether d_1 and d_2 mention the same people entity.

In this work, we assign a type weight to each node according to the following heuristic: the higher the number of unique tags of a certain type, the smaller the weight of each node with this tag type. Thus, the tag weighting function is defined as Formula 1, where v denotes any tag with the type $t_i \in T$. $|A|$ is the number of non-repetitive tags in D, and $|T_i|$ is the number of unique tags of the type "t_i".

$$typeweight(v) = \begin{cases} \dfrac{\log |A|}{\log |T_i|} & \text{if } |T_i| > 1 \\ 1 & \text{if } |T_i| = 1 \end{cases} \quad (1)$$

$|T_i|$=1 means that there is only one tag typed t_i in D. Although this situation is very rare given a large document corpus, we assign constant 1 to avoid zero-valued divisor.

D. Clustering

It is found that each of the majority of the documents in D usually refers to a single people entity. However, the influence that a document mentions more than one people entity should be considered, therefore, we have performed experiments, in which documents overlapping in different clusters was allowed when the same document mentions several people entities (i.e., namesakes). The experimental results show a higher precision but much lower recall, which results in the degradation of the overall performance (i.e., the harmonic mean of precision and recall). Hence, our work proposes a clustering algorithm for name disambiguation

under the assumption that a document exactly concerns one people entity w.r.t. the given query name.

Accordingly, this algorithm first assigns tags in a micro-cluster to a separate initial cluster. Then the algorithm groups micro-clusters into several macro-clusters, each of which represents a unique people entity. In order to complete this task, the connectivity strength between micro-clusters has to be considered to judge whether tags from different micro-clusters are actually referred to the same people entity.

1) Connectivity Strength (CS): Vector Space Model(VSM) is traditionally used to measure the similarity between documents [2][18][19], but documents with many tags are poorly represented in VSM because they have a small scalar product and a large dimensionality. To overcome this shortcoming, we measure the similarity between any two micro-clusters M_i and M_j, denoted by *Connectivity Strength*, using Formula 2. If the *Connectivity Strength* is above a predefined threshold λ, the tags in these clusters are considered to refer to the same people entity and should be merged into one macro-cluster. [11] has defined a function of "Connection Strength" in a entity-relationship graph, and used a topic-based correlation clustering through training the skylines. Differently, our "Connectivity Strength" is defined in a tag graph to measure the similarity between micro-clusters based on bridge-tags.

$$CS(M_i, M_j) = \sum_{k=1}^{n} \frac{Cohesion(b_k, M_i) + Cohesion(b_k, M_j)}{2} \tag{2}$$

Here, b_k is the bridge-tag linking micro-clusters M_i and M_j (k=1, . . . , n), and $Cohesion(b_k, M_i)$ is the internal cohesion of tag b_k in M_i, which will be defined later. Although different documents might miss different types of tags about the queried people, they usually contain some types of tags, which connect different documents. Thus, the micro-clusters corresponding to the same people entity are usually connected by the bridge tags. This formula means that the more the bridge-tags connecting these two micro-clusters and the higher the cohesion weight of each bridge-tag, the larger the connectivity strength between them. Based on the cohesion value of a bridge tag, the proposed graph-based approach effectively clusters the tags referred to the same people entity scattering in different cliques together.

Most of the existing work weight the words by TF-IDF [2][5]. That is, suppose f(w) is the frequency of word w in document d, N is the total number of documents, and $df(w)$ is the document frequency of word w, then the weight of word w in the TF-IDF representation of d is defined as follows.

$$d_w = f(w) \ln \frac{N}{df(w)}$$

The measure of TF-IDF regards words with higher frequency and lower document frequency more important. In this work, the worthless tags with high document frequency have been filtered out as stop words in Section III-B2. On the other hand, some infrequent but relevant tags which have high type weights are retained as they are effective in name disambiguation. For this reason, we develop a more effective method, $Cohesion$, by applying a similar framework with TF-IDF. The proposed new method is employed to compute the importance of a tag, taking both tag frequency and type weight into account. This method emphasizes on the internal cohesion of a tag in a micro-cluster. The higher the cohesion value, the more important the tag v to micro-cluster M.

Let $p(v, M)$ be the probability of tag v in micro-cluster M obtained from the number of occurrences of tag v divided by the total number of all tag occurrences in the micro-cluster M. Assuming the independence between any two distinct tags in the same micro-cluster, we have the co-occurrence probability for any two connected nodes v and w in the micro-cluster M, $p(vw, M) = p(v, M)p(w, M)$. We compute the internal cohesion of tag v in M as follows.

$$Cohesion(v, M) = \alpha(typeweight(v) \times \sum_{j=1}^{m} edgeweight(v, w_j)p(vw_j, M)) \tag{3}$$

As defined in Formula 3, a high $Cohesion$ weight is reached by a high type weight and a high joint occurrence probability of v with other nodes w_j in M (m is the number of nodes connected with v in M). In addition, the $Cohesion$ formula is factorized by a co-occurring weighting parameter α, which is the minimum number of v involved bridge-tags linking M and any other micro-clusters. This factor is introduced to measure the relevancy of tag v with other tags in M. Since common tags, which occur too frequently have been filtered out as stop words, if tag v often occurs with many other tags linking M and other micro-clusters, there is a high relevancy of tag v with M, and accordingly the cohesion weight of v in M is high. Compared with TF-IDF, the $Cohesion$ measure is more suitable for the tag evaluation in people name disambiguation on the Web.

2) Clustering Algorithm: The clustering is performed in two steps. In the first step, all the micro-clusters (i.e., maximal clique subgraphs) are first identified. In the second step, a single link clustering algorithm is adopted to merge two connected micro-clusters with the connectivity strength value above a predefined threshold λ to form a larger micro-cluster until all the micro-clusters are visited. In this way, we get a final set of macro-clusters.

The optimal threshold λ for clustering is acquired through training data. The training data set is divided into 10 parts randomly, and then 10-fold cross validation [20] is adopted to estimate the disambiguation performance. Finally, the threshold, having the best average performance, is determined as the clustering parameter λ on test data sets. In this

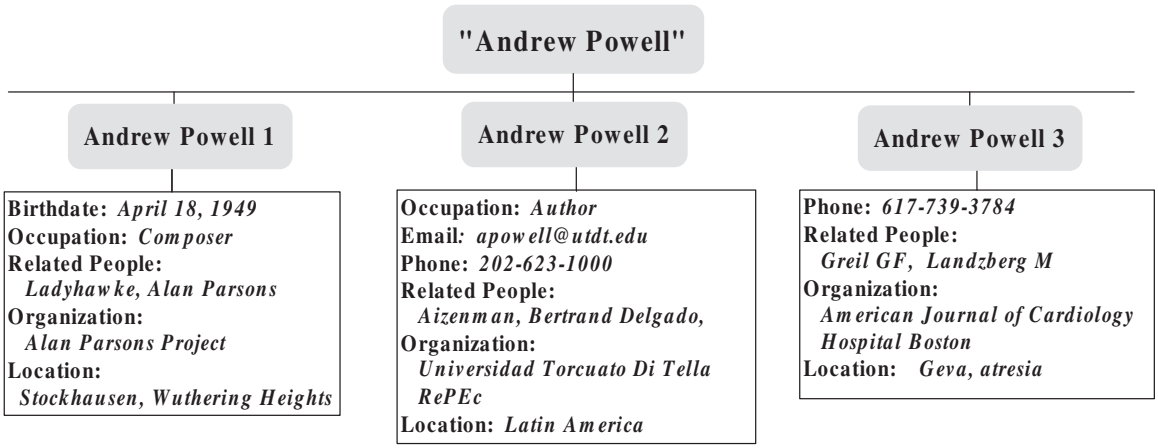

Figure 4. An Example of Tag Clusters

Table II
SUMMARY OF GRAPH MODELS W.R.T. EACH DATA SET

Data Set	average No. of tags	average No. of bridge-tags	average No. of micro-clusters
SIGIR'05	2,923	297	90
WWW'05	2,273	295	68
WePS'07	3,157	528	86
WePS'09	4,417	764	109

work, the threshold is trained to be 0.45 for the GRAPE model.

3) Tagging: The output of the above clustering algorithm is a set of tag clusters, each of which represents a unique people entity. According to the framework in Figure 1, we applied our GRAPE approach in a real Web application. Given a query name, after preprocessing and tag extraction from the returned top 100 results of a Web search engine, we get people tag clusters by using the clustering algorithm. Finally, we select tags with high frequency in each type of tag set to represent each people entity effectively.

As illustrated in Figure 4, we demonstrate the tag clusters generated by our approach for three people entities regarding the query name of "Andrew Powell". The final set of tags except URL domain in each cluster are listed in a table under the corresponding people entity.

IV. EXPERIMENTAL STUDY

A. Data Sets

We evaluated our GRAPE approach using four different public data sets. The first one is SIGIR'05 [1], which contains 9 common people names and 110 Web pages for each name. The second data set, WWW'05 [4], is composed of 12 person names. The third one is the test data set of task WePS'07 in SemEval-2007 [15], containing 30 people names. Both WWW'05 and WePS'07 consist of collections of Web pages obtained from the top 100 results for a person name query to an Internet search engine. The forth data set is WePS'09 [16], which contains 30 people names, and 150 returned Web pages for each people name.

We used the training data of the WePS competition [2] to learn the clustering threshold. This data set contains 49 different people names and has also been designed as the training data of both WePS'07 and WePS'09 tasks.

For each people name in the data set, a tag-based graph is modeled and the proposed clustering algorithm groups the micro-clusters in the graph into several macro-clusters, each of which characterizes a unique people entity (i.e., namesake). The average number of tags, bridge-tags and micro-clusters for each query name in the test data sets are summarized in Table II.

B. Evaluation Measures

Two measures, F_{EB} [21] and F_{PI} [15], were used for evaluation. Herein, F_{EB} is the extended B-cubed measure based on Precision and Recall, while F_{PI} is the harmonic mean of the standard Purity and Inverse-Purity clustering measures. As stated in [21], the measure of F_{PI} focuses on the match between clusters and category, and F_{EB} can handle the problem of overlapping clustering effectively. Moreover, F_{EB} measure satisfies most of the clustering constraints and is recommended for generic clustering problems. In the following evaluation, Precision, Recall, Purity, and Inverse-Purity are abbreviated as EB-P, EB-R, P, and IP respectively.

C. Empirical Results

Four different sets of experiments were conducted to validate the effectiveness of the proposed framework in disambiguating and tagging people names.

[2]http://nlp.uned.es/weps/

Table III
EXPERIMENTAL RESULTS ON SIGIR'05

	TF-IDF	EQW	GRAPE
People Name	F_{EB}/F_{PI}	F_{EB}/F_{PI}	F_{EB}/F_{PI}
Ann Hill	0.83/0.83	0.87/0.87	**0.90/0.90**
Brenda Clark	0.93/0.93	**0.94/0.94**	**0.94/0.94**
Christine King	0.68/0.72	0.86/0.87	**0.91/0.90**
Helen Miller	**0.93/0.94**	0.91/0.92	0.92/0.94
Lisa Harris	0.65/0.62	0.68/0.72	**0.75/0.80**
Mary Johnson	0.74/0.72	**0.81/0.79**	**0.81/0.79**
Nancy Thompson	0.87/0.90	**0.96/0.96**	**0.96/0.96**
Samuel Baker	**0.71/0.64**	0.61/0.56	0.70/0.63
Sarah Wilson	0.77/0.80	0.69/0.78	**0.77/0.82**
Mean	0.78/0.79	0.81/0.82	**0.85/0.85**

Table IV
EXPERIMENTAL RESULTS ON WWW'05

	TF-IDF	EQW	GRAPE
People Name	F_{EB}/F_{PI}	F_{EB}/F_{PI}	F_{EB}/F_{PI}
Adam Cheyer	**0.87/0.93**	0.84/0.92	0.81/0.90
William Cohen	**0.90/0.94**	0.86/0.92	0.85/0.91
Steve Hardt	0.45/0.66	**0.56/0.75**	0.45/0.67
David Israel	0.65/0.75	**0.68/0.77**	**0.68/0.77**
Leslie Kaelbling	0.94/0.88	**0.95/0.98**	**0.95/0.98**
Bill Mark	0.72/0.79	**0.74/0.82**	**0.74/0.82**
Andrew McCallum	0.72/0.83	0.75/0.84	**0.87/0.93**
Tom Mitchell	0.75/0.81	0.81/0.85	**0.86/0.89**
David Mulford	0.72/0.83	**0.74/0.83**	0.73/0.83
Andrew Ng	0.64/0.76	0.78/0.84	**0.84/0.88**
Fernando Pereira	0.65/0.69	**0.79/0.86**	**0.79/0.86**
Lynn Voss	0.59/0.65	0.59/0.65	**0.60/0.66**
Mean	0.72/0.80	0.76/0.84	**0.77/0.84**

We used TF-IDF and Equal-Weight (denoted by EQW) as the baseline methods on SIGIR'05 and WWW'05. TF-IDF employs the vector space model in clustering based on the tag frequency and inverse document frequency. On the other hand, EQW assumes each tag $typeweight$ in Cohesion measure is equal, which aims to verify the effectiveness of the $typeweight$ measure. The names from WWW'05 are primarily of researchers in related research area and more diverse than SIGIR'05, we tested them separately to indicate the effectiveness of our proposed method for different real world challenges.

The experimental results on SIGIR'05 and WWW'05 are presented in Table III and Table IV respectively, where the best scores are in boldface. As the results indicate, the proposed method makes a big improvement over the method TF-IDF in name disambiguation. The idea of assigning different types of tags with different weights makes significant improvement on SIGIR'05 and a little improvement on WWW'05. In addition, as demonstrated in Section III-D3, our approach generates a set of meaningful tags to describe each cluster (corresponding to a namesake).

Table V
EXPERIMENTAL RESULTS ON WEPS'07

System	EB-P	EB-R	P	IP	F_{EB}/F_{PI}
CU-COMSEM	0.67	**0.81**	0.72	**0.88**	0.71/0.79
IRST-BP	0.68	0.73	0.75	0.80	0.68/0.77
PSNUS	0.68	0.71	0.73	0.82	0.67/0.77
GRAPE	**0.80**	0.79	**0.84**	0.87	**0.78/0.85**

Table VI
EXPERIMENTAL RESULTS ON WEPS'09

System	EB-P	EB-R	P	IP	F_{EB}/F_{PI}
PolyUHK	0.87	0.79	0.91	0.86	0.82/0.88
UVA-1	0.85	0.80	0.89	0.87	0.81/0.87
ITC-UT-1	**0.93**	0.73	**0.95**	0.81	0.81/0.87
GRAPE	0.85	**0.83**	0.88	**0.90**	**0.83/0.89**

Some experiments were also performed on a Web people search task using data sets from WePS'07 [15] and WePS'09 [16]. Table V and Table VI compare our approach with the top 3 best systems in WePS'07 and WePS'09, respectively. We used the same training data and test data as the systems in both WePS'07 and WePS'09 tasks to make sure that the performance comparison is unbiased.

From the results in Table V and Table VI we see, the proposed GRAPE approach achieves better overall performance (measured by F_{EB} and F_{PI}) than the state-of-the-art systems. Due to some uncertain information in documents, even annotator in person cannot decide whether some document belongs to a cluster or a new people entity. According to [16], the F_{EB}/F_{PI} performance of Oracle systems which know what is the best clustering threshold for each test instance on WePS'09 is 0.85/0.90, which indicates that our GRAPE approach works well and almost achieves the optimal performance.

Furthermore, we conducted a set of experiments for different tag sets. Table VII shows the results. NE denotes the combination of people name, organization, and location, while BioTag denotes the tag set including email address, phone number, birth date, and occupation. [12] represents the social network of a namesake using only people names and organizations, as they speculate that locations cannot be reliable for people disambiguation as many people might be linked to one location. Thus, we also report the disambiguation effectiveness using people name and organization only (denoted by "NE − Location" in Table VII). The results in Table VII show that the tag type of location is useful in most of cases, thus we retain location in name disambiguation and tagging. Although different tag combinations play different roles over different data sets, the experimental results demonstrate the effectiveness of the combination of NE, BioTag, and URL domain in name disambiguation.

Table VII
EXPERIMENTAL RESULTS BASED ON DIFFERENT TAG SETS

	SIGIR'05		WWW'05		WePS'07		WePS'09	
Tag Set	F_{EB}	F_{PI}	F_{EB}	F_{PI}	F_{EB}	F_{PI}	F_{EB}	F_{PI}
NE − Location	0.79	0.81	0.72	0.80	0.73	0.79	0.80	0.86
NE	0.84	0.84	0.71	0.80	0.73	0.80	0.82	0.88
NE + BioTag	0.79	0.81	0.72	0.81	0.75	0.82	0.82	0.88
NE + URL domain	0.84	0.84	0.75	0.83	0.74	0.81	0.82	0.88
NE+BioTag+URL domain	**0.85**	**0.85**	**0.77**	**0.84**	**0.78**	**0.85**	**0.83**	**0.89**

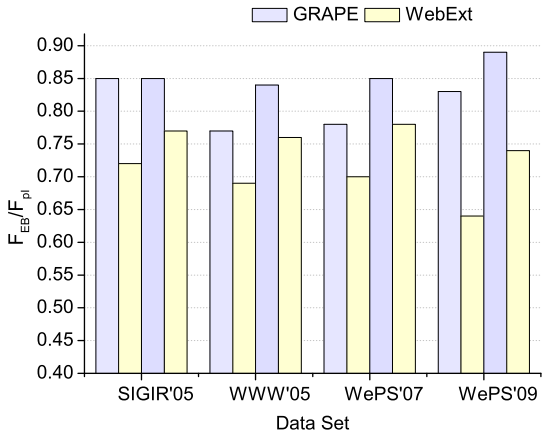

Figure 5. Comparison Results of two Clustering Algorithms

There have been many attempts to utilize the extended Web resources for disambiguating people names. We implemented another clustering algorithm (denoted by WebExt), which uses additional Web resources in Web people name disambiguation as presented in [6]. Given a Web documents corpus, WebExt extracts terms from each document, and then compares these terms by inputting each term as a query to Web search engines and collecting the snippets in the top 100 returned results. Based on the term similarities, WebExt computes the document similarity by using group-average agglomerative clustering algorithm (GAAC). Figure 5 shows the comparison results, where F_{EB} and F_{PI} measure results for the same algorithm are presented separately with the same color in each data set.

According to the results in Figure 5, our GRAPE approach performs better than the WebExt algorithm. WebExt considers the entire document, which adds much noise. Moreover, WebExt is over an order of magnitude slower than GRAPE. Therefore, it is a big challenge to exploit the external Web resources to improve the performance of people name disambiguation. Bad external resources usually bring in much noisy information, which costs much time and leads to inaccurate clustering results.

V. CONCLUSION AND FUTURE WORK

This paper proposes a novel weighted-graph based framework, GRAPE, to both disambiguate and tag the people names in Web search. A graph structure is devised to model the relationships among people tags, and an unsupervised clustering algorithm is developed to cluster all the tags for each people entity, which effectively solves the problem of name disambiguation in Web people search. Specially, the proposed method takes type weight into account to measure the importance of tags. Furthermore, an extensive performance study was performed using several public data sets and the proposed approach outperforms all the existing solutions.

A number of approaches exist to improve the capability of Web search for people information searching. However, it still remains a challenging research issue. For future work, we have three main research interests. One is people tag extraction. Most natural language processing tools were previously used to train a model for a news corpus, such as Lingpipe used in this paper, yet the Web is noisier. Thus, some specific extraction models are needed for more accurate tag extraction in Web pages. The second direction is to explore the social networks. It is observed that people name is the most frequent type of tag, people name indices or family trees are often presented in the returned pages. These pages are too ambiguous for name disambiguation, but provide a good source for social network construction. Hence, an extension of this work is to mine the people relationships on the Web. Furthermore, we plan to apply the proposed framework in a real Web people search system, and provide the analysis of efficiency for further validation.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grant No. 60833003 and No. 90924025, National Basic Research Program of China under Grant No. 2006CB303103, an HP Labs Innovation Research Program award, a research award from Google, Inc., and the Program for New Century Excellent Talents in University under Grant No. NCET-07-0491, State Education Ministry of China, the Guangdong Provincial Government, State Education Ministry of China under Grant No. 0712226-100097.

REFERENCES

[1] J. Artiles, J. Gonzalo, and F. Verdejo, "A testbed for people searching strategies in the www," in *Proceedings of the 28th annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2005)*, 2005, pp. 569–570.

[2] A. Bagga and B. Baldwin, "Entity-based cross-document coreferencing using the vector space model," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL)*, 1998, pp. 79–85.

[3] M. B. Fleischman and E. Hovy, "Multi-document person name resolution," in *Proceedingsof the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, 2004.

[4] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *Proceedings of the 14th International World Wide Web Conference (WWW 2005)*, 2005, pp. 463–470.

[5] R. Bunescu and M. Pasca, "Using encyclopedic knowledge for named entity disambiguation," in *Proceedings of EACL-06*, 2006.

[6] D. Bollegara, Y. Matsuo, and M. Ishizuka, "Disambiguating personal names on the web using automatically extracted key phrases," *In Proceedings of the biennial European Conference on Artificial Intelligence (ECAI 2006)*, 2006.

[7] Y. Ravin and Z. Kazi, "Is hillary rodham clinton the president? disambiguating names across documents," in *Proceedings of the ACL 1999 Workshop on Conference and its Applications*, 1999.

[8] G. S. Mann and D. Yarowsky, "Unsupervised personal name disambiguation," in *Proceedings of the 7th Conference on Computational Natural Language Learning. Edmonton, Canada*, 2003.

[9] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, "Two supervised learning approaches for name disambiguation in author citations," in *Proceedings of JCDL 2004*, 2004.

[10] X. Fan, J. Wang, B. Lv, L. Zhou, and W. Hu, "Ghost: An effective graph-based framework for name distinction," in *Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM 2008)*, 2008.

[11] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray-Turan, "Web people search via connection analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1550–1565, 2008.

[12] 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 the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*. New York, NY, USA: ACM, 2008, pp. 27–34.

[13] E. Minkov, W. W. Cohen, and A. Y. Ng, "Contextual search and name disambiguation in email using graphs," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, 2006.

[14] J. Iria, L. Xia, and Z. Zhang, "Wit: Web people search disambiguation using random walks," in *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-2007)*, 2007, pp. 480–483.

[15] J. Artiles, J. Gonzalo, and S. Sekine, "The semeval-2007 weps evaluation: Establishing a benchmark for web people search task," in *Proceedings of Semeval 2007, Association for Computational Linguistics*, 2007, pp. 9–16.

[16] A. Javier, J. Gonzalo, and S. Sekine, "Weps 2 evaluation campaign: overview of the web people search clustering task," in *Proceedings of In 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, 2009.

[17] L. Jiang, J. Wang, N. An, S. Wang, J. Zhan, and L. Li, "Two birds with one stone: A graph-based framework for disambiguating and tagging people names in web search," in *Proceedings of the 18th International World Wide Web Conference (WWW 2009)*, 2009.

[18] C. Niu, W. Li, Srihari., and R. K., "Weakly supervised learning for cross-document person name disambiguation supported by information extraction," *ACL Results Rand and Evaluation*, 2004.

[19] A. Huang, D. N. Milne, E. Frank, and I. H. Witten, "Clustering documents with active learning using wikipedia," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, 2008, pp. 839–844.

[20] P. A. Devijver and J. Kittler, "Pattern recognition: A statistical approach," *Prentice-Hall, London*, 1982.

[21] E. Amigo, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval Journal, Springer, Heidelberg*, 2008.