Social Network-based Entity Extraction for People Ontology

Tian Tian
Department of Computer Science
Manhattan College
tina.tian@manhattan.edu

Soon Ae Chun
College of Staten Island
City University of New York
soon.chun@csi.cuny.edu

***Abstract*—When users want to search people, search engines face two basic challenges. One challenge is due to the fact that there are many people (entities) with the same name, i.e. a homonym problem. The other is an entity linking issue, where several words are linked to the same person. The homonyms create the search results with a long list of hits with mingled information of the different person with the same name. The end users need to sift through the documents that fit their needs. To improve the ambiguous search arising from homonyms, we previously implemented an Ontology-Supported Web Search System (OSWS) that utilizes an ontology to disambiguate the search term and that provides search results in different possible categories that a search term may belong to. For a prototype of the OSWS system, we developed an ontology by mining person names and retrieving data from resources such as DBpedia. However, DBpedia is incomplete and often outdated. In this paper, we extend our approach to using social networks for building a People Ontology (PO). Specifically, personal profile attributes and their values of famous people are extracted from public social networks pages, cleaned and mapped to the ontology, resulting in a significant increase of the domain coverage achieved by the People Ontology to support the Ontology-Supported Web Search System.**

Keywords-social networks; ontology; mining from social networks; semantic Web search

I. INTRODUCTION

Users' information needs in the digital era can be fulfilled by keyword-based search engines. However, the major search engines do not disambiguate homonymous search terms, especially the person names that may refer to several different people. The search results thus contain information of different people of the same name. To address the homonymous names, we developed a domain-specific ontology, namely People Ontology, where people with the same name are categorized into different classes based on their properties. This category information and other properties can be used for disambiguation. The utility of the People Ontology is shown in the Ontology-Supported Web Search (OSWS) System [1] [2] we have developed. The search system uses the People Ontology to disambiguate the people search by providing users with separate search results of each homonymous person separated with different categories.

Our approach to develop a domain-specific People Ontology for the Ontology-Supported Web Search system (OSWS) involves (1) retrieving person names by Google search suggestions and (2) extracting category and attribute information from DBpedia [3]. Google search completion feature suggests a set of potential names which is used to generate a candidate list of person concepts for the People Ontology [4]. We classified these names suggested by Google search into three different famous people categories, namely, A-List, B-List and C-List. We used the working definition of the famous people as the person whose full name is suggested with a minimal substring of the name. Thus, the smaller the substring of the name is, the more famous entity it may refer to. The A-List contains more famous people than B-List since the full name is suggested with fewer substrings (e.g. first name only) than the B-List candidates where the search suggestion requires more than the first name string. Similarly, C-List contains names with the least famous people, according to our working definition. This working definition is based on the assumption that the famous (or infamous) people are more likely to be used as the search keywords, that influences the Google search suggestion.

In order to establish the unique entity for the person in these candidate lists, we used DBpedia for any additional attributes and person categories for the People Ontology. The resulting ontology in [4] contains 3,241 people instances and over 60,000 relationships emanating from them.

DBpedia is a huge public resource that can be used for developing ontology. However, it was found that many concepts in the candidate lists, especially not so famous people, did not exist in DBpedia. Furthermore, DBpedia is a slow-changing data resource. To overcome these shortcomings to improve PO, additional sources of information were needed. In this paper, we use social networks such as Facebook or Twitter profiles, to gather additional, "fast-changing" information that can further disambiguate the homonymous people concepts.

In this paper, we present how we used a social network as a secondary resource to extract knowledge in the domain of famous people. Choosing Facebook [5] as a sample social network is motivated by the fact that it has become the largest social networking site in recent years [6]. Millions of users have integrated Facebook into their daily practices [6].

One can create public pages in Facebook. Public pages are for organizations and celebrities to broadcast information about them in an official, public manner [7]. More and more famous people are joining Facebook to publicize their profiles and news.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-311-7

The rest of the paper is organized as follows. Section II briefly explains the Ontology-Supported Web Search System to illustrate the application of People Ontology. In Section III, we describe the process of mining information from a social network and report the resulting enrichment of the People Ontology (PO). Section IV presents the related work. Section V concludes the paper and proposes future directions.

II. ONTOLOGY-SUPPORTED WEB SEARCH SYSTEM

In previous research, an Ontology-Supported Web Search (OSWS) System for famous people has been developed to improve the Web search process for homonymous terms. The system visually categorizes homonyms and provides search suggestions to the users [1] [2]. The OSWS System uses an ontology to derive the disambiguated search terms and suggested search completions based on the knowledge in the ontology (Fig. 1). The ontology was built by using search suggestions retrieved from Google, together with information extracted from DBpedia.

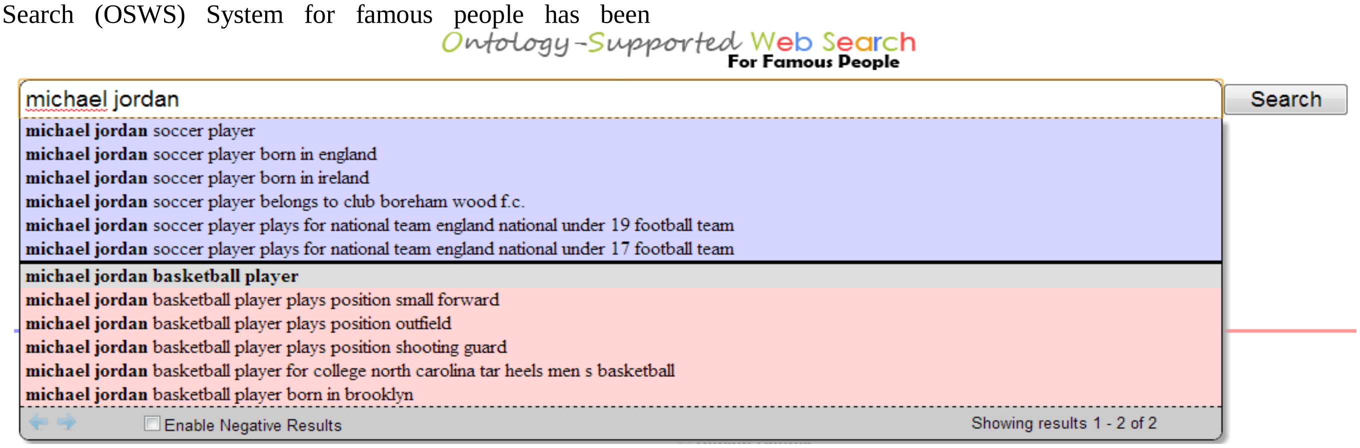

Figure 1. Interface of the Ontology Supported Web Search System.

The candidate list of famous people was mined from Google's suggested completions [4]. We then passed the concepts in the list to DBpedia and extracted the related knowledge. Various methods have been applied to clean the extracted DBpedia information [4]. Despite its massive multi-domain coverage, many concepts in the candidate list were not found in DBpedia. Furthermore, new DBpedia releases appear only every couple of months [3], but some people become famous overnight. Therefore, we used social networks as the secondary, fast-changing resource to create a new famous People Ontology.

Using Facebook as an example of the "social network to ontology approach," concepts were checked against Facebook's Graph Search and the ones belonging to the "people categories" were selected as targets. A threshold was used to identify who qualifies as "famous" person. We then extracted relevant information regarding the famous persons. After data cleaning, the mined knowledge was integrated into the People Ontology.

III. SOCIAL NETWORK MINING FOR DEVELOPING THE PEOPLE ONTOLOGY

A. *Social Networks*

Turning to social networks as sources of information about famous people is a natural choice, as social networks utilize people as the primary topic of representation. Thus, we can view a social network as

- a set of concept nodes where each node represents a person or an organization;
- a set of semantic relationships between those nodes, expressing how different nodes relate to each other;
- one or more categorization relationships assigning person or organization concepts to different classes;
- a set of attributes of each concept node that characterizes and distinguishes different person/organization concepts from each other.

The described structure of a social network is remarkably similar to the structure of an ontology, as (some flavors of) ontologies are also based on concepts that are interconnected by IS-A relationships and semantic relationships and have additional attributes describing the concepts.

Examples of attributes in Facebook include "id," "name," "picture," "website," "birthday," "description," "likes," etc. "Likes" is an especially useful attribute, which represents the number of people that like a specific page.

Facebook users are linked to exactly one "category." The category information is mandatory to fill when the user creates a Facebook public page. There are 24 possible categories that a Facebook person page may belong to, including "actor/director," "artist," "athlete," "politician," "writer" etc. Mining of social network pages is possible because users can access category and attribute information by program.

B. *Identifying People in a Social Network*

One can search people in social networks by using their APIs or sending Web queries. Facebook provides such searches through http requests. The url below returns the first 10 Facebook pages with "Barack Obama" in the name:

https://graph.facebook.com/search?q=barack%20obama&type=page&limit=10.

Figure 2. Top five Facebook ressults of query “Barack Obama.”

As Fig. 2 shows, several public pages have been created for Barack Obama. One can choose to consider all of them as valuable sources of information, or one can decide to use only the “authorized” page.

In previous research [4], we discussed how to create a small list of a few thousand very famous people, a larger list of famous people and a much larger list of somewhat famous people. We called these lists the A-List, B-List and C-List. The name lists were retrieved from the Google’s suggested completions. The A-List contains the most famous people, as they are the suggestions returned by giving a first name [4]. The B-List was retained by entering a first name with a letter. The C-List includes the least famous people by giving a first name with two letters to the search engine.

We began with a sublist of 2,564 names in the A-List that do not exist in DBpedia. We name it the “reduced-A-List.” The first 10 Facebook results were collected. The one page with the largest “likes” was chosen as the selected page, as a page with more attention tends to be more authoritative.

We checked the category information of the selected pages and identified 954 of them as people. However, some had very few fans. We found it necessary to define a threshold to determine which people are important enough to be considered famous, or which page(s) of a celebrity is (are) popular enough so that this person should be stored in the PO. Statistics show that median Facebook page has 218 fans [8]. In this study, 218 was used as the minimum number of “likes” for a Facebook page to be selected for analyzing and storing its namesake in the PO. Note that the number of Facebook fans is not the only measure in evaluating a person’s popularity. The names were first selected through Google’s search completions. Facebook then was used to validate if the names refer to people.

626 people were found in the “reduced-A-List” with over 218 “likes”. The pseudocode below shows the procedure of identifying famous people from Facebook given a name list.

```
PEOPLE_IDENTIFICATION(list){
  FOR each name in list{
    Search name in Facebook Graph Search
    Save the top 10 pages returned
    max_likes = -infinity
    FOR each returned page{
      IF (page.likes > max_likes){
        page_with_max_likes = page
        max_likes = page.likes
      }// End of IF
     }// End of inner FOR
     IF  (page_with_max_likes.category!=PERSON  OR
max_likes < THRESHOLD)
       Remove page_with_max_likes
  }// End of outer FOR
}
```

C. *Mining Knowledge from a Social Network*

Most social network sites require users to establish their profiles when creating their accounts. Such a user profile may contain valuable information regarding the person. This section presents the process of extracting useful profile attributes from social networks. We save the mined attributes in a database and map them to the People Ontology.

In total, 33 different kinds of Facebook attributes were returned among the selected people. However, some attributes are Facebook-centric and have no use in suggesting search completions in the OSWS system. Thus, considering the usefulness of the attributes and after manually checking the quality and trustworthiness of the returned values of the attributes, a number of person attributes were chosen to be transferred into the PO. They include “name,” “category,” “likes,” “birthday,” “location,” “hometown,” “affiliation” of athlete, and “genre” and “record_label” of musician.

In a few cases, “location” stores irrelevant or even “wrong” information. Some examples include location data like “in the kitchen,” “in the world” and “home.” There are two ways to solve this problem. One is to query the returned location in a search engine and check if the first page of hits contains any url from mapping services. This method works fine for the “reduced-A-List,” but would cause delays when dealing with the much larger B-List [4]. In this study, yet another solution was applied. We used the “A-List” as the training data to extract stop words appeared in attribute “location.” Location data in the “B-List” was then automatically filtered with the stop word list.

“Category” is the most important attribute in disambiguating homonymous names. Most Facebook categories can be mapped directly to the PO. However, a few of them need further processing in order to provide better precision. They include “athlete,” “actor/director” and “public figure.”

An athlete could be a sportsman of different kind. A search suggestion of “Michael Jordan basketball player” is more informative than suggestion “Michael Jordan athlete.” By analyzing the additional Facebook attributes, such as “bio,” “description” and “personal_info,” and checking for matches with the 22 subdomains of athlete in the ontology, it was possible to specify 51 people playing specific sports among the 112 athletes found.

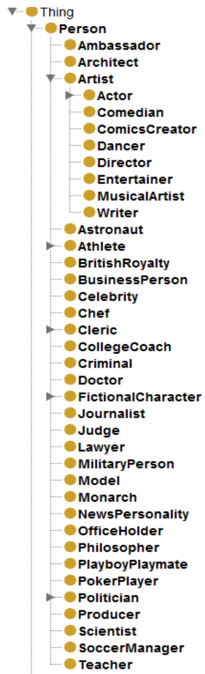

Figure 3. Partial view of the expanded “Person” hierarchy in Protégé.

Facebook combines actors and directors in one category: “actor/director.” To provide better specification, we parsed other descriptive attributes to determine if the person belongs to the “actor” or the “director” category.

“Public figure” is one of the biggest categories found in the “reduced-A-List.” However, they do not provide much valuable information in disambiguating homonymous names. In order to assign those people with more concrete categories, we checked their Facebook attributes with a list generated with all Facebook person categories, classes in the PO and their synonyms in WordNet [9]. We used a Synonym API [10] to collect synonyms. The API is based on REST calls, which return well-formatted XML results, providing synonyms based on the WordNet database.

For the category-specific attributes, “affiliation” of athlete carries information about the team the athlete is playing for. A list of stop words was built to remove noise from the data. “Record_label” and “genre” were processed with the same filtering method. “Record_label” provides information about the company that manages the musician. “Genre” describes the type of music the musician plays. Fig. 3 shows a partial view of the final Person hierarchy in Protégé format.

D. Mapping Social Network Profiles to the People Ontology

The previous section described how the relevant social network attributes were selected and cleaned. In this section, we explain how we mapped and integrated the social network profiles to the PO.

Many Facebook categories exist in the People Ontology, such as “artist,” “athlete,” “journalist,” etc. Therefore, these categories can be directly mapped to the PO. For categories that do not exist in the People Ontology, we expanded the ontology by adding the new classifications in the hierarchy.

The other Facebook attributes were also mapped to the People Ontology, as seen in Table 1. The first column shows the Facebook attribute and the second column represents the corresponding attribute in the PO. The third column in the table shows the type of property (data type or object) used in the ontology. Attributes “name,” “likes” and “birthday” were stored as data properties. The remaining attributes were mapped to the PO as objects, thus, it was necessary make sure that no repetition of objects occurs in the ontology. An object property was only added if it did not already exist in the People Ontology.

TABLE I. FACEBOOK ATTRIBUTES TO FAMOUS PEOPLE ONTOLOGY MAPPING

Facebook Attribute	Ontology Mapping	Property Type
name	name	datatype
birthday	dateofBirth	datatype
likes	facebookLikes	datatype
location	currentPlace	object
current_location	currentPlace	object
hometown	placeofBirth	object
affiliation	playsForTeam	object
genre	musicalGenre	object
record_label	recordLabel	object

The processing of the previous OSWS Ontology used the number of relationships and attributes to determine the popularity of a famous person [4]. The Facebook number of “likes” provides the same measurement, but at a different scale, meaning the numbers cannot be combined. Therefore, a separate data type property “facebookLikes” was created in the PO.

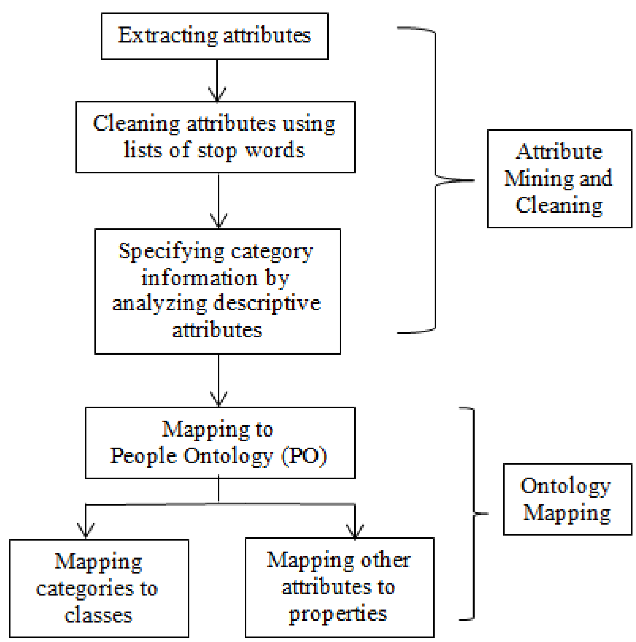

Figure 4. Algorithm flow of mining and processing the A-List data .

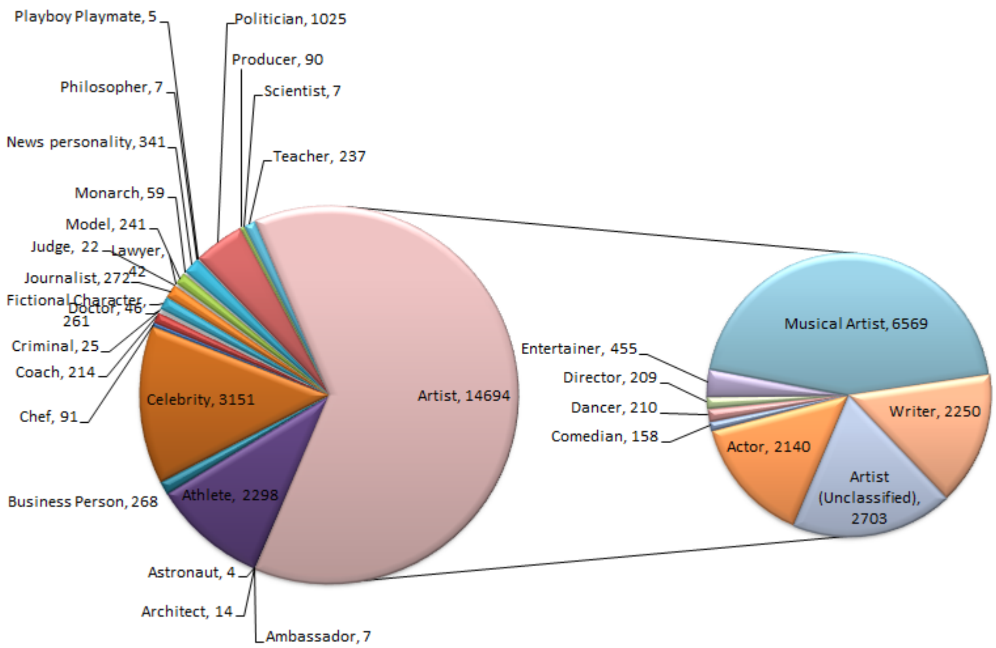

Figure 5. Distribution of the newly added B-List famous people among categories in the Famous People Ontology.

In total, 622 additional people who did not exist in the system were added to the PO by mining Facebook. Among the 626 names we analyzed in the A-List, 4 are musical band names instead of individuals. Since our ontology is about famous persons, we removed information representing groups of people from the results.

The new entities include 279 artists, 112 athletes, 113 celebrities and many other famous people from the remaining categories. For example, Sam Adams the singer is newly added to the ontology. His homonyms include Sam Adam the politician. Fig. 4 shows the algorithm flow of mining and processing the A-List data.

Using the A-List as the training set, we applied the same method of data extraction and data cleaning onto the B-List. Processing the B-List data was a complete automatic procedure. Among the 155,403 names in the B-List, only 35,626 were found in DBpedia. To expand the PO, the rest 119,777 names were stored in the "reduced-B-List" and queried against Facebook. In total, 40,360 people from the "reduced-B-List" were found in Facebook. Among them, we were able to identify 23,421 famous people with more than 218 likes on their Facebook pages.

Artist is the largest category among all with 14,694 people, including musical artists, actors, dancers, writers, etc. Fig. 5 shows the distribution of the famous people in the B-List among different categories. Each colored area is marked with the name of the category and the number of people that were found in this category.

This part of work was developed using the Facebook Graph API in Java. Unfortunately, Facebook allows only a limited number of Graph API calls per minute. Thus, a timer was set in the programs to send out one API call every two seconds. This slowed down our work considerably.

E. Other Social Networks

Twitter has become one of fastest growing online social networking services [11]. It has gained worldwide popularity with over 500 million users, including many public figures.

Twitter stores the following attributes for users: "user_id," "screen_name," "name," "profile_image_url," "location," "url," "description," "created_at," "followers_count," "friends_count," "statuses_count," "time_zone," and "last_update." In addition, Twitter stores information about every single tweet that is not statically associated with the user, e.g., "geo_lat" and "geo_long."

Twitter users are invited to follow people from certain categories. For example, when logging in, Twitter suggests people from "music," "sports," and "entertainment." Currently the following categories are supported: Music, sports, entertainment, twitter, funny, fashion, family, technology, food&drink, news, art&design, books, business, science, health, travel, government, staff picks, charity, nascar, pga, mtv movie awards, mlb, faith and religion, NBA, television, CMT awards, billboard music awards, US election 2012, NHL.

Twitter data can be mined by using the "phirehose" library [12] or with one of the built-in Twitter APIs [13]. Twitter's API provides the GET users search, which searches for users similar to Find People button on Twitter's official site [14]. The GET search returns a json object with all

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-311-7

associated properties of the person, which includes useful information, such as name, location, id, etc.

It is common to have more than one Twitter user returned from a search request. The first returned result, in most cases, is the official Twitter account of the famous person. Another valuable attribute returned by the GET search is the account's "verified" value. The property identifies if the returned Twitter profile is a verified account. Verification has been used to establish authenticity of identities on Twitter, including highly sought users in music, acting, fashion, government, politics, religion, journalism, media, advertising, business, etc. [15].

Twitter's GET search also returns several attributes associated with the user's account, including "name," "location," "description," "followers_count," etc. Compared to Facebook, Twitter has fewer profile attributes that can contribute to our People Ontology. Mapping can be built using string matching techniques.

Twitter limits its GET search to 60 calls per hour [14], which will be a major obstacle in this study.

IV. Related Work

Previous research has been reported on extracting data from social networks. Thelwall et al. have mined MySpace comments to detect the emotions [16]. Chu et al. have mined Facebook live feeds regarding social networking forensics [17]. Xu et al. investigated retrieving user opinions in social network services [18]. SONAR is an API for gathering and sharing social network information [19]. POLYPHONET was built as a social network extraction system [20].

Shibaki et al. have constructed a person ontology from Wikipedia by extracting person categories and the IS-A relationships among them [21]. However, the ontology does not contain other relationships or attributes other than the parent-child relations.

Mika [22] presents an approach to construct an ontology (folksonomy), based on the sub-community of an actor who interact with other actors, the semantic annotations (tags) the community use to describe documents, using tripartite graph. The concepts and ontology emerge from the associative relations within each sub-community and its interacting actors. He argues that the incorporation of the social context into the ontology models captures the idea that ontologies are inseparable from the context of the community in which they are created and used. It also highlights the emerging nature of ontologies, as opposed to the slow growing knowledge base such as WordNet. It is an algorithmic approach to construct folksonomy. Similar approach is proposed by Himanshu et al. [24] to construct ontology from social network using semantic tags used in a sub-community. However, the folksonomy constructions is a general approach to identify the concepts and disambiguation of concepts using social network, but not necessarily focus on a person ontology.

Finin et al. [23] have proposed the use of FOAF ontology (i.e., FOAF documents) to identify person, link and fuse distributed personal information using RDF, and develop a social network based on foaf:knows relations. It suggests the potential use cases of the person ontology represented in FOAF, but it does not address how to construct the ontology from the Web site.

In Information Retrieval community, the entity disambiguation is approached based on textual occurrences of names and its context. Bhattacharya and Getoor [25] use mutual relations between authors for entity resolution. In the context of citations we may conclude that "R. Srikant" and "Ramakrishnan Srikant" are the same author, since both are coauthors of another author. They consider the mutual relations between authors, paper titles, paper categories, and conference venues. Hassel et al. [26] uses the attribute information such as affiliation, topics of interests, etc. contained in an ontology derived from the DBLP [27], while Pilz [28] exploits the category information from Wikipedia for disambiguation. However, the focus is not to build an ontology of entities but utilizing them to disambiguate the names in a text.

We expand our previous approaches on exploiting social networks and DBpedia to construct and enrich a people ontology with more relationships [1] [2] [29]. To our knowledge, little research has been done in constructing ontologies from the social networking sites.

V. Conclusions and Future Work

This paper presented the process of mining a social network as a secondary resource to enrich the People Ontology, since the primary source of DBpedia had missing information of candidate people we extracted from Google Search suggestion. Using the social network mining approach we presented, we were able to classify 954 names in the A-List whose information was lacking in DBpedia.. A series of data extraction and data cleaning steps were performed to mine the Facebook public pages of the selected people. The standardized data was then mapped to the PO.

Using the same automated method, we were able to mine and map more than 23,000 people in the B-List that were located in Facebook to the People Ontology. Our approach shows a potential to develop ontology that can be scalable. The People Ontology can be utilized in semantic disambiguation of entities, and the linking of separate references. Our prototype semantic search system shows one utility of the People Ontology.

Currently Facebook is used for static information gathering. We plan to develop a mechanism in the future to automatically detect updates in the pages. In addition, we plan to investigate the friendship relations between people in the social network to further enrich the ontology.

In the future, we plan to extend the People Ontology by mining knowledge from other social networks, such as Twitter, LinkedIn and MySpace.

Acknowledgment

Christopher Ochs has contributed to the implementation of the OSWS system.

References

[1] T. Tian, J. Geller, and S. A. Chun, "Improving web search results for homonyms by suggesting completions from an ontology," 2nd ICWE Workshop on Semantic Web Information Management (SWIM). Lecture Notes in Computer Science (LNCS), 2010, issue 6385, pp. 175-186, Springer.

[2] T. Tian, J. Geller, and S. A. Chun, "Enhancing interface for ontology-supported homonym search," CAiSe'11 Workshop on Semantic Web Search (SSW). Lecture Notes in Business Information Processing (LNBIP), 2011, issue 83, pp. 544-553, Springer Verlag, Berlin.

[3] DBpedia, http://dbpedia.org/About, retrieved 07/08/2013.

[4] C. Ochs, T. Tian, J. Geller, and S. A. Chun, "Google knows who is famous today: Building an ontology from search engine knowledge and DBpedia," 5th IEEE International Conference on Semantic Computing (ICSC), Palo Alto, CA, 2011, pp. 320-327.

[5] Facebook, www.facebook.com, retrieved 07/08/2013.

[6] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," Journal of Computer-Mediated Communication, 2007, vol. 13, issue 1, pp. 210-230.

[7] Facebook Graph API, http://developers.facebook.com/docs/reference/api/, retrieved 07/08/2013.

[8] T. McCorkindale, "Can you see the writing on my wall? a content analysis of the fortune 50's Facebook social networking sites," Public Relations Journal, 2010, vol. 4, no. 3, pp. 1-10.

[9] WordNet, http://wordnet.princeton.edu/, retrieved 07/08/2013.

[10] Stands4 API, http://www.abbreviations.com/api.asp, retrieved 07/08/2013.

[11] Twitter, www.twitter.com, retrieved 07/08/2013.

[12] The "Phirehose" Library, https://github.com/fennb/phirehose, retrieved 07/08/2013.

[13] Twitter REST API, https://dev.twitter.com/docs/api, retrieved 07/08/2013.

[14] Twitter GET User/Search, https://dev.twitter.com/docs/api/1/get/users/search, retrieved 07/08/2013.

[15] Twitter Account Verification, https://support.twitter.com/groups/31-twitter-basics/topics/111-features/articles/119135-about-verified-accounts, retrieved 07/08/2013.

[16] M. Thelwall, D. Wilkinson, and S. Uppal, "Data mining emotion in social network communication: Gender differences in MySpace," Journal of the American Society for Information Science and Technology, 2010, vol. 61, issue 1, pp. 190-199.

[17] H. Chu, D. Deng, and J. H. Park, "Live data mining concerning social networking forensics based on a Facebook session through aggregation of social data," IEEE Journal of Selected Areas in Communications, 2011, vol. 29, issue 7, pp. 1368-1376.

[18] K. Xu, S. S. Liao, Y. Song, and L. Liu, "Mining user opinions in social network webs," The Fourth China Summer Workshop on Information Management, Wuhan, China, 2010, pp. 39-49.

[19] I. Guy, M. Jacovi, E. Shahar, N. Meshulam, and V. Soroka, "Harvesting with sonar - the value of aggregating social network information," CHI, Florence, Italy, 2008, pp. 1017-1026.

[20] Y. Matsuo, J. Mori, and M. Hamasaki, "POLYPHONET: An advanced social network extraction system from the web," International World Wide Web Conference (WWW), Edinburgh, Scotland, 2006, pp. 262–278.

[21] Y. Shibaki, M. Nagata, and K. Yamamoto, " Constructing large-scale person ontology from Wikipedia," 2nd Workshop on Collaboratively Constructed Semantic Resources, Beijing, China, 2010, pp. 1-9.

[22] P. Mika, "Ontologies are us: a unified model of social networks and semantic," Web Semantics: Science, Services and Agents on the World Wide Web, 2007, vol. 5, issue 1, pp. 5-15.

[23] T. Finin, L. Ding, L, Zhou, and A. Joshi, "Social networking on the semantic web," Learning Organization, 2005, vol. 12, issue 5, pp. 418-435.

[24] M. Hamasaki, Y. Matsuo, T. Nisimura, and H. Takeda, "Ontology extraction using social network," International Workshop on Semantic Web for Collaborative Knowledge Acquisition, Harderabad, India, 2007.

[25] I. Bhattacharya and L. Getoor, "Relational clustering for multitype entity resolution," Fourth International Workshop on MultiRelational Data Mining, 2005, pp. 3-12.

[26] J. Hassel, B. Aleman-Meza, and I. B. Arpinar, "Ontology-driven automatic entity disambiguation in unstructured text," Lecture Notes in Computer Science (LNCS), 2006, pp. 44-57.

[27] DBLP, http://www.informatik.uni-trier.de/~ley/db/, retrieved 10/22/2013.

[28] A. Pilz, "Entity disambiguation using link based relations extracted from Wikipedia," 26th International Conference on Machine Learning, Haifa, Israel, 2010.

[29] S. A. Chun, T. Tian, and J. Geller, "Enhancing the famous people ontology by mining a social network," 2nd International Workshop on Semantic Search over the Web, Istanbul, Turkey, 2012.