International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 5, Sep - Oct. 2013 pp-2767-2769 ISSN: 2249-6645

Lexical Pattern- Based Approach for Extracting Name Aliases

Mohammad Khasim[1], Sayeed Yasin[2]
[1]*M. Tech, Nimra College of Engineering & Technology, Vijayawada, A.P., India.*
[2]*Asst. Professor, Dept.of CSE, Nimra College of Engineering & Technology, Vijayawada, A.P., India.*

ABSTRACT: Searching for information about people in the web is one of the most common activities of many internet users. Around 30% of search engine queries include person names. Retrieving information about people from web search engines can become difficult when a person has nick names or name aliases. We will not be able to retrieve all the information about a cricket player, if we only use his real name. For example we can't retrieve all the information regarding the cricketer sachin tendulkar by using his real name only. So we need his nick names too. Identification of entities on the web is difficult mainly for two fundamental reasons: The first one is that different entities can share the same name (i.e., lexical ambiguity) and the second one is that a single entity can be designated by multiple names (i.e., referential ambiguity). In this paper, we propose a lexical pattern-based approach to extract aliases of a given name using snippets returned by a web search engine. The lexical patterns are generated automatically using a set of real world nick names or name alias data. The proposed work does not assume any language specific preprocessing such as part of speech tagging or dependency parsing, etc., which can be both inaccurate and computationally costly in web scale data processing.

Keywords: *Alias, Anchor text, Lexical analysis, Pattern.*

I. INTRODUCTION

Finding a relevant, information of a particular entity on the web is very important task as it is helpful in the information retrieval process. Retrieving information of a person simply by using his/her name is quite insufficient if the person has nick names [1]. Now- a- days celebrities are known by two or more name in the web. Entities may be a person, a location, an organization, a festival name, etc. Identification of entities on the web is difficult for two basic reasons- The first one is that different entities may share the same name (Lexical ambiguity) and the second one is that one entity is known by different names (Referential ambiguity). The name dis- ambiguation problem differs fundamentally from that of alias extraction because in name dis- ambiguation the objective is to identify the different entities that are referred by the same ambiguous name; in alias extraction, we are only interested in extracting all references to a single entity from the web. For example: "Diwali" is also known as Deepavali as a one word alias or Festival of Lights as a three word alias. The cricketer, Sachin Tendulkar is also known as Master blaster or little master. Similarly, entities are also referenced by profession, drama, etc.

Identifying aliases of a name is important in various tasks such as information retrieval, relation extraction and sentiment analysis and name disambiguation. In information retrieval, to improve recall of the web search on a person name, a search engine can automatically expand the query using aliases of the name. In this paper, we propose an alias identification method that is based on two main things such as links extraction and association measures used. For link extraction we used extract link and also consider the second level depth of the web pages [2]. We propose lexical pattern extraction algorithm to retrieve pattern with the help of name alias datasets. This lexical pattern is useful for the candidate alias extraction and which are independent of languages.

II. RELATED WORK

Alias identification is closely related to the problem of cross document co- reference resolution in which the objective is to determine whether two mentions of a name in different documents refer to the same entity. In [3], the authors proposed a cross document co- reference resolution algorithm by first performing within document co- reference resolution for each individual document to extract co- reference chains, and then, clustering the co- reference chains under a vector space model to identify all mentions of a name in the document set. However, the vastly numerous documents on the web render it impractical to perform within document co- reference resolution to each document separately, and then, cluster the documents to find aliases.

In personal name dis- ambiguation the goal is to dis- ambiguate various people that share the same name (namesakes) [4][5]. Given an ambiguous name, most name dis- ambiguation algorithms have modeled the problem as one of document clustering in which all documents that discuss a particular individual of the given ambiguous name are grouped into a single cluster. The web people search task (WePS) provided an evaluation data set and compared various name dis- ambiguation systems. However, the name dis- ambiguation problem differs fundamentally from that of alias extraction because in name dis- ambiguation the objective is to identify the different entities that are referred by the same ambiguous name; in alias extraction, we are interested in extracting all the aliases to a single name from the web.

III. PROPOSED WORK

The proposed method is outlined in Figure 1 and comprises two main components: pattern extraction, and alias extraction and ranking. Using a seed list of name- alias pairs, we first extract lexical patterns that are frequently used to convey information related to aliases on the web. The extracted patterns are then used to find the candidate aliases for a given name.

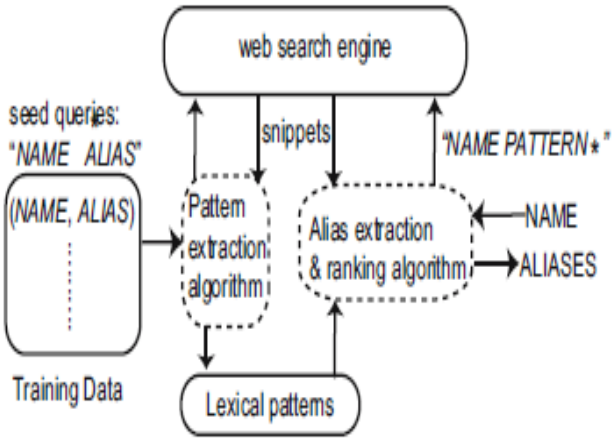

Figure 1: Proposed method

A. Lexical pattern extraction

For lexical pattern extraction input is given as name- alias pair. This list gives frequently occurred lexical patterns between the name and aliases. To retrieve the patterns query is given as input to the web search engine. The query is in the form of "name * alias". The wild operator "*" is used to perform NEAR query. Algorithm 1 is used to capture the various ways in which information about aliases of names is expressed on the web.

Algorithm 1: ExtractPatterns(S)
comment: S is a set of (NAME, ALIAS) pairs
P ← null
for each (NAME,ALIAS) ∈ S
do
D ← GetSnippets("NAME * ALIAS")
for each snippet d ∈ D
do P ← P + CreatePattern(d)
return (P)

B. Candidate Alias Extraction

Once the set of lexical pattern is extracted, then the patterns are used to extract the candidate aliases for a given name as in Algorithm 2. If an entity name, name and a set "P" of lexical patterns is given as input, the Extract_Candidates function returns a list of candidate aliases for the name. Given name is associated with each pattern "p" in the set of patterns P and produce queries of the form: "NAME p *". Thus we get a list of the candidate aliases.

Algorithm 2: ExtractCandidates(NAME,P)
comment: P is the set of patterns
C ← null
for each pattern p ∈ P
do D ← GetSnippets("NAME p * ")
for each snippet d ∈ D
do C ← C + GetNgrams(d,NAME, p)
return (C)

C. Ranking of Candidates

Considering the noise in web snippets, the candidates extracted by the shallow lexical patterns might include some invalid aliases. From among these candidates, we must identify those which are most likely to be correct aliases of the given name. We model this problem of alias recognition as one of the ranking candidates with respect to a given name such that the candidates, who are most likely to be correct aliases are scores to measure the association between a name and a candidate alias using three different approaches: lexical pattern frequency, word co-occurrences in anchor texts , and hub discounting.

D. Lexical Pattern Frequency

Using lexical pattern extraction algorithm retrieves a list of patterns with the help of a web search engine. We can use pattern frequency as one of the approach to calculate the weight of the aliases. If the personal name under consideration and a candidate name alias occur in many lexical patterns, then it can be considered as a good alias for the personal name. Consequently, we rank a set of candidate aliases in the descending order of the number of different lexical patterns in

which they appear with a given name. The lexical pattern frequency of an alias is analogous to the document frequency (DF) popularly used in information retrieval process.

E. Co- Occurrences in Anchor Texts

Anchor texts have been used extensively in information retrieval process and have been used in various tasks such as synonym extraction, query translation in cross language information retrieval, and ranking and classification of the web pages. Anchor texts are particularly attractive because they not only contain concise texts but also provide links that can be considered as expressing a citation. We revisit anchor texts to measure the association between a name and its aliases on the Internet. Anchor texts pointing to a url provide useful semantic clues related to the resource represented by that url. For example, if the majority of inbound anchor texts of an url contain a personal name, it is likely that the remainder of the inbound anchor texts contain information about aliases of the name. Here, we use the term inbound anchor texts to refer the set of the anchor texts pointing to the same url. For example, consider the picture of Sachin Tendulkar shown in Figure 2. Figure 2 shows a picture of Sachin Tendulkar being linked to by four different anchor texts. According to our definition of co-occurrence, Sachin Tendulkar and Tendlya are considered as co-occurring.

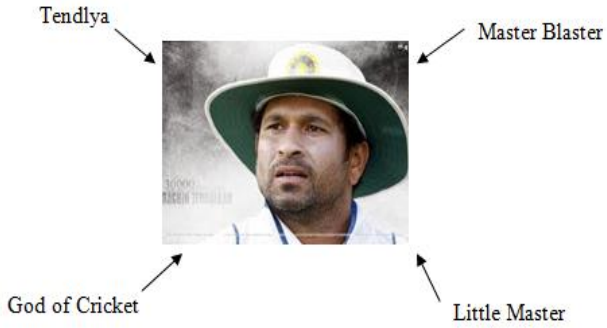

Figure 2: Picture of Sachin with different aliases

F. Hub Discounting

If the majority of the link contain person name in anchor text, then the confidence of that page as a source of information regarding the person whom we are interested in extracting aliases increases. We use this intuition to compute the simple discounting measure for co-occurrences in hubs as follows,

$$\alpha(h,n) = \frac{t}{d}$$

Where "t" is total number of inbound anchor text of "*h"* that contain real name n and d is total number of inbound anchor text of *h*.

IV. CONCLUSION

In this paper we proposed a lexical pattern-based approach to extract aliases of a given person name. We use a set of names and their aliases as the training data to extract lexical patterns that describe numerous ways in which information related to aliases of a name is presented on the web. An individual is typically referred by numerous nick names or name aliases on the web. Accurate identification of aliases of a given person name is useful in various web related tasks such as sentiment analysis, information retrieval, personal name disambiguation, and relation extraction. We propose a method to extract aliases of a given person name from the web. Given a person name, the proposed work first extracts a set of candidate aliases. Second, we rank the extracted candidates according to the likelihood of the candidate being a correct alias of the given name.

REFERENCES

[1] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka, Member, IEEE 2011 Automatic Discovery of Personal Name Aliases from the Web In IEEE Transaction on knowledge and data engineering, vol. 23, no. 6.
[2] D. Kavitha 2011 A Survey on Assorted Approaches to Graph Data Mining In International Journal of Computer Applications (0975 – 8887) Volume 14– No.1.
[3] M. Berland and E. Charniak, "Finding Parts in Very Large Corpora," Proc. Ann. Meeting of the Assoc. for Computational Linguistics (ACL '99), pp. 57-64, 1999.
[4] G. Mann and D. Yarowsky, "Unsupervised Personal Name Disambiguation," Proc. Conf. Computational Natural Language Learning (CoNLL '03), pp. 33-40, 2003.
[5] R. Bekkerman and A. McCallum, "Disambiguating Web Appearances of People in a Social Network," Proc. Int'l World Wide Web Conf. (WWW '05), pp. 463-470, 2005.