International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-1, Issue-5, November 2012

Automated Entity Alias Evocation from Web

Snehal S. Shinde, P. R. Devale

Abstract : ***Identifying the correct reference to an entity among a list of references is required in lots of works such as information retrieval, sentiment analysis, person name disambiguation as well as in biomedical fields. More previous work had been done on solving lexical ambiguity here we proposed a method that is based on referential ambiguity. In this paper we proposed a method which is based on referential ambiguity to extract correct alias for a given name. Given a person name and/or with context data such as location, organization retrieves top K snippets and depth up to level two from a web search engine. With the help of Lexical pattern extract candidate aliases. As to find correct alias from a list of aliases we used n-depth crowling method. This method is useful to improve the precision and minimize the recall than the previous baseline method.***

Keywords: ***Web mining, web text analysis, text mining, n-depth crawling.***

I. INTRODUCTION

Finding a relevant, information of a particular entity on the web is very important task as it is helpful in information retrieval process. Retrieving information of a person simply by using his or her name is quite insufficient if the person has nick names. Now a day celebrities are known by 2 or more name in the web. Entities may be a person, an organization, a location, a festival name, etc. Identification of entities on the web is difficult for two basic reasons. First: different entities may share the same name (Lexical ambiguity). Second: One entity is known by different names (Referential ambiguity). The name disambiguation problem differs fundamentally from that of alias extraction because in name disambiguation 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 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, Mahendra Singh Dhoni is also known as Dhoni or Mahi. Similarly, entities are also referenced by drama, profession, etc.

II. RELATED WORK

Correct alias finding is important in information retrieval. In [1], Danushka Bollegala proposed a method which uses extraction techniques to automatically extract significant entities such as the names of other persons, organizations and locations on each webpage. In that method for given person name, it extract person name from the web by using lexical pattern matching method and anchor text analysis. They ranked the candidate alias from the list. For this they integrated various similarity measures scores and given to a single function to support vector machine. In [2] Dmitri proposed a method in which automatic entity extraction techniques are explained. In addition, it extracts and parses HTML and Web related data on each web page, such as hyperlinks and email addresses. Then this information is presented in an Entity Relationship Graph. This method is used to find relative information of a particular person on the web. In [3], A. Bagga proposed a method that summarizes the interested entities and ranks the similarity of summaries using various information metrics. In [4], T. Hokama proposed a method, especially for Japanese language. In [5], C. Galvez proposed a method for extraction of abbreviations of personal names that measures approximate string matching algorithms.

In [6], Christian Borgelt explained how text classification is done using graph mining and also explained different graph parameters.

III. PROPOSED METHOD

Fig. 1 shows the proposed system. In this sys if the name and alias pair is given it extracts the lexical pattern. With the help of this lexical pattern and given real name, candidate alias is extracted. Here we have considered n-depth crawling to extract the candidate aliases. Then extracted candidate aliases are ranked by using some approaches like degree distribution of candidate aliases by using hyperlink extraction on the web to identify correct aliases.

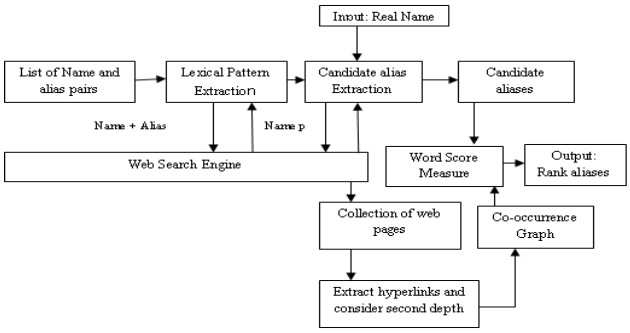

Fig. 1: Proposed System

A. Lexical Pattern Extraction

For lexical pattern extraction input is given as name and alias pair. This list gives frequently occurred lexical patterns between name and aliases. To retrieve the patterns query is given as input to web search engine. The query is in the form of name * alias. The wild operator * is used to perform NEAR query.

Revised Manuscript Received on 30 November 2012.
* Correspondence Author
Snehal S. Shinde*, Computer Engineering Department, Bharati Vidyapeeth's Deemed University College of Engineering, Pune, Maharashtra, India.
P. R. Devale, Information Technology Department, Bharati Vidyapeeth's Deemed University College of Engineering, Pune, Maharashtra, India.

B. Candidate Alias Extraction

Once the set of lexical pattern is extracted the patterns are used to extract the candidate aliases for a given name. 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 *'. Here 4 grams words are considered as candidate aliases. Thus we get a list of candidate aliases.

C. Ranking of Candidate Aliases

As all candidate aliases are not valid aliases for name, we must identify correct alias from list. The problem of alias identification is ranking of aliases with respect to given name as most closely alias assign a higher rank. For that purpose, we consider four different approaches such as lexical pattern frequency, co-occurrence frequency, page count on the web and graph mining method.

D. Lexical Pattern Frequency

Using lexical pattern extraction algorithm retrieves a list of patterns with the help of web search engine. We can use pattern frequency as one of the approach to calculate weight of the aliases. If the personal name under consideration and a candidate 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 name. the lexical pattern frequency of an alias is analogous to the document frequency (DF) popularly used in information retrieval.

E. Co-occurrence Frequency

This is the simplest of all association measures denotes the CF of a candidate alias x and a name n. the value of c is calculated as if there are many URLs, which are pointed to by anchor texts that contain a candidate alias x and a name n, then it is an indication that x is indeed a correct alias of the name p. So for CF we consider count of URLs which contain both.

F. Hub Discounting

If the majority of 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 a simple discounting measure for co-occurrences in hubs as follows:

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

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.

G. Page Count Based Measures

Web Dice: We compute the Dice, Web Dice between a name n and a candidate alias x using page counts as number of hits by giving query as,

$$WebDice(n, x) = \frac{2 * hits(n \cap x)}{hits(n) + hits(x)}$$

H. Graph Mining Association Measures

For forming a graph, we consider co occur between name and alias and also consider number of times alias appears with real name on that link. Graph-based representations of real-world problems have been helpful due to their improved clarity and efficient use in finding the solutions. The hash table scheme uses a hash function to map keys with their corresponding values.

Degree Distribution: It is defined as degree of a node in a network is the number of connections it has to other nodes. It is also called as probability distribution of these degrees over the whole network. Here we consider the probable distribution of link from the node. We consider real name as root node and probable candidate aliases are child node. The aliases are calculated by using hyperlink structure of the web.

$$Degree = \frac{Out - Link}{In - Link}$$

Clustering Coefficient

$$\Delta v = \{(u, w) \in E \mid (v, u) \in E \text{ and } (v, w) \in E$$

The number of triples at a node v is the number of paths of length tow in which v is the central node. Therefore, for a node v, the number of triples at node v is

$$C(u) = \frac{d(v)}{T(v)}$$

I. Normalization

We consider value of word score for each alias by using all possible approaches, the values we get are not in the same range. For that purpose we require to do normalization to get result in [0,1] range. Finally we sort all candidate aliases in descending order according to their rank for respective real name.

IV. CONCLUSION

Here we proposed name alias detection using n-depth crawling method. Here we proposed four different possible approaches such as lexical pattern frequency, co-occurrence frequency, web dice and graph mining measures. This method is useful in various tasks such as relation detection, information retrieval system and sentiment analysis system.

REFERENCES

1. Danushka Bollegala, YutakaMatsuo and IitsuruIshizuka, Member , IEEE, Automatic Discovery of Personal Name Aliases from the Web, *IEEE Transaction on knowledge and data engineering, vol. 23, no. 6*, June 2011.
2. Dmitri V. Kalashnikov Zhaoqu Chen Rabia Nuray – Turan Sharad Mehrotra Zheng Zhang, Web People Search via connection Analysis, *IEEE International Conference on Data Engineering*, 2009.
3. A. Bagga and B. Baldwin, Entity-Based Cross-Document Coreferencing using the vector space model, *Proc. Int's Conf. Computational linguistics (COLING '98), pp. 79-85*, 1998.
4. T. Hokama and H. Kitagawa, Extracting Mnemonic Names of People from the Web, *Proc. Ninth Int'l Conf. Asian Digital Libraries (ICADL ' 06), pp. 121-130*, 2006.
5. C. Galvez and Fg. Moya-Anegon, Approximate Personal Name Matching through Finite State Graphs, *J. Am. Soc. Fro Information Science and Technology, vol. 58, pp. 1-17*, 2007.
6. Christian Borgelt, Graph Mining: An Overview, *Proc, 19th GMA/GI Workshop Computational Intelligence, Germany*, 2009.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)