International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3– No.8, August 2012 – www.ijais.org

Automatic Extraction of Entity Alias from the Web

Sumitra A. Jakhete
Master of computer Engineering
Pune Institute of Computer Technology,
Pune University, Pune, India

Shweta C.Dharmadhikari
Associate Professor
Pune Institute of Computer Technology,
Pune University,Pune, India

ABSTRACT

An individual is known by more than one name on the web. Identifying the correct alias for an entity is playing a crucial role in the field of information retrieval, relation extraction, sentiment analysis, and entity name disambiguation as well as in biomedical fields. Traditional system provides the solution on solving lexical ambiguity, but it lagged on the problem of referential ambiguity. Through this paper we emphasis on referential ambiguity to extract correct alias for a given name. Given a name alias dataset retrieves lexical pattern from a web search engine. With the help of Lexical-pattern and using second level depth extract candidate aliases. As to identify correct alias from a list of aliases we used similarity measures as well as graph mining measures such as degree distribution and clustering coefficient. We integrate different word sore and calculate the final weight of each candidate alias. There by our method providing more promising result in terms achieving a statistically significant mean reciprocal rank (MRR) of 0.611 and improves the precision and minimize the recall that than the previous baseline method.

General Terms

8. II Database Management VIII Database Application, 8 III Information Storage and Retrieval III Information Search and Retrieval IX Artificial Intelligence VII Natural Language Processing.

Keywords

Graph Mining; Text Mining; Web Mining; Web Text Analysis.

1. INTRODUCTION

30% search on the web is on person name. An entity name can be identified by more than one reference or the same name for more than one entity name [1]. Entity can be a person name, location name or a famous temple or a thing in exists. If a person has more than one reference that means aliases then that is called as referential ambiguity. And same name for more than one person then it is called as lexical ambiguity. Previous research has done on person name disambiguation. In this paper, we focused on referential ambiguity. For example, the famous cricket player *Sachin Tendular* is also known as *Little Master* as a two word alias. *Mumbai* city is known as *Bombay* or an Indian festival like *deepavali* is known as the *festival of light* Many times various types of terms are used as alias such as doctor, name of a role or title of a drama .

Identifying aliases of a name is important in various tasks such as relation extraction, information retrieval, and sentiment analysis and name disambiguation. In information retrieval, to improve recall of a web search on a person name, a search engine can automatically expand the query using aliases of the name. 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 web pages [3].

We propose a fully automatic identification system for name alias. We proposed the task in following threefold ways.

- We propose lexical pattern extraction algorithm to retrieve pattern with the help of name alias dataset. This lexical pattern is useful for candidate alias extraction and which are independent of languages.
- To extract candidate aliases, we consider a set of patterns and real name taken as an input to the system. By considering all possible combination of name and pattern, for a given query, we get top-k URLs. Again extracting links, we get second level depth for web pages. After preprocessing we get final list of candidate aliases.
- To rank correct alias from list of candidate aliases, we propose four approaches such as lexical pattern frequency, word co-occurrence in an anchor text, page count on the web and graph mining method.
- We integrate word score from all approaches and consider normalized weight for each candidate aliases. We conduct a series of experiments to evaluate the various components of the proposed method.
- We evaluate our proposed method on different type of dataset such as person name data set, location name dataset and entity name dataset. Our proposed system improves the precision value.

The main contribution of this paper is a novel use of possible approaches for web people’s search and its integration into a search engine so that it can be efficiently used during query execution.

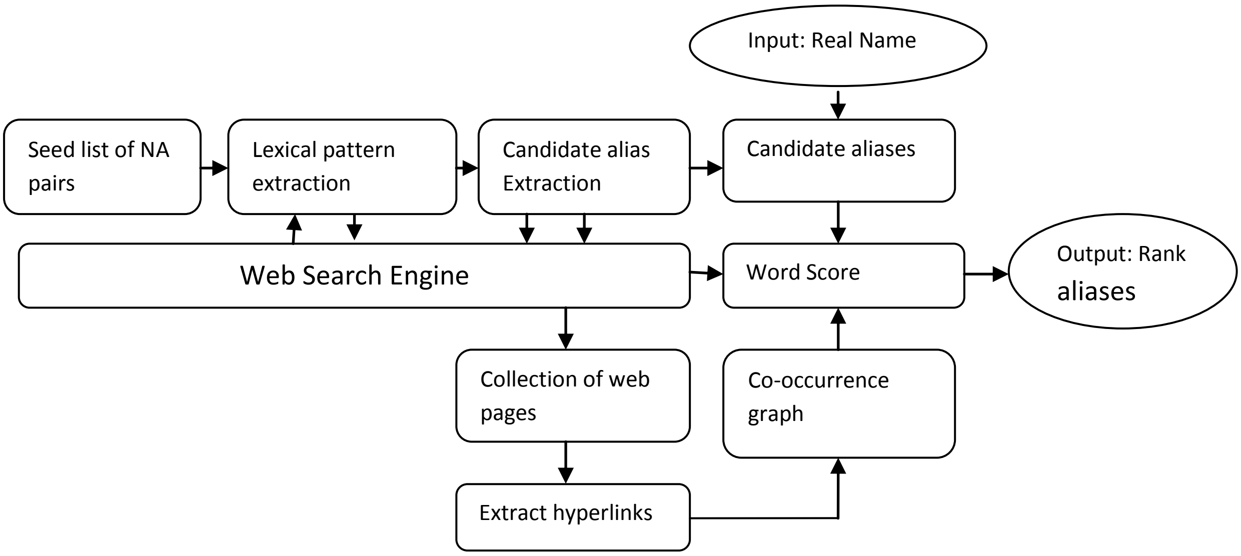

Fig 1: Proposed System Architecture

The remainder of this paper is organized as follows: A brief review of related work is given in section 2.Section 3 describes the mathematical model of proposed system architecture along with system design. And Section 4 consists of experimentation and results. Section 5 presents the conclusion.

2. RELATED WORK

In this paper, we focused on three main part of the people's search on web. We have done literature survey in the following way as name disambiguation, name alias detection, graph mining method.

First part of literature survey is on lexical ambiguity. Name disambiguation problem is similar to entity cross-document co references. In [5], Danushka Bollegala proposed a method which presents an unsupervised algorithm which produces key phrases for the different people with the same name. These key phrases could be used to further narrow down the search; leading to more people specific unambiguous information. In [13], A. Bagga proposed a method that take summaries about an entity of interest and used various information metrics to rank the similarity of the summaries. In [6], Dmitri V. Kalashnikov proposed a method in which they explained automatic extraction techniques to automatically extract 'significant' entities such as the names of other person's, organizations, and locations on each web page. In addition, it extracts and parses HTML and Web related data on each web page, such as hyperlinks and email addresses. The algorithm then views all this information in a unified way: as an Entity-Relationship Graph where entities (e.g., people, organizations, locations, WebPages) are interconnected via relationships (e.g., 'web page-mentions-person', relationships derived from hyperlinks, etc). This method is used to find relative information of a particular person on the web.

Second part of literature survey is on referential ambiguity. In [1], Danushka Bollegala proposed a method in which for given a person name it extract person name from the web by using lexical-pattern matching method and anchor text analysis. To rank a candidate alias form the list, they integrate various similarity measures scores and given to a single function to support vector machine. In [7], T. Hokama proposed a method, which is specific to Japanese language. For a given name p, they search for the query "* koto p" and extract the context that matches the asterisk In [8] C. Galvez proposed a new method to measure approximate string matching algorithms have been used for extracting variants or abbreviations of the personal names.

Third part from literature survey is based on graph mining method. In [2], Christian Borgelt explained how graph mining is useful for text classification and also explained different graph parameters. In [3], D. Kavitha has done survey on graph mining method which describes the type of graph mining algorithm. In [4], D. Chakrabarti has explained recursive use of graph mining method for application. In [14], I explained what the advantages of anchor text retrieval are and what are the preprocessing rules apply on anchor text to get final set of words.

3. MATHEMATICAL MODEL OF PROPOSED SYSTEM AND SYSTEM DESIGN

Let S be the System such as

S= { Ip, Op, RN,P,EC,C,U,W,V,E,,F_1,F_2,F_3,F_4,WS,S_u, F |Ø}

Ip=Input of the system

Ip=Domain D =NA= {na_1, na_2, na_3, na_n} //seed list of name and Alias pair.

3.1 Functional Description

3.1.1 Lexical Pattern Extraction

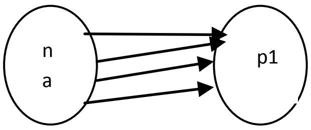

Fig. 2 : Venn Digaram of Lexical Pattern Extraction (one-to-many mapping)

$$\mathbf{f1(NA) \rightarrow P}$$

is a one to many (bijective) function since each name and alias, we get number of pattern such as 'known as', 'nicknamed as 'and same pattern will get from two different input.

We consider here input as seed list of name and alias pair. This seed list gives frequently occurred lexical patterns between name and aliases. Name * Alias query is used to retrieve data from web search engine [1].

3.1.2 Candidate Alias Extraction

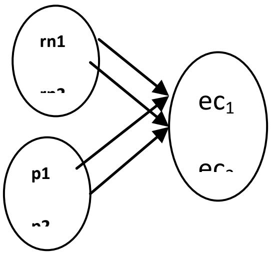

Fig. 3 : Venn Digaram of Candidate alias Extraction (overloading)

$$\mathbf{f2}(\mathbf{RN} \vee \mathbf{P}) \rightarrow \mathbf{EC}$$

Where

P = {P_1, P_2, P_3… $P_{n\}}$ // pattern returned by lexical pattern extraction function.

RN= {rn_1, rn_2, $rn_{3\ldots}$ rn_n}// RN is set of real name given as input to system.

EC= {n ∩ p | n ε N and p ε P}

Let EC is the set of candidate aliases.

Ec= {ec_1, ec_2, ec_3… ecn | ec Є Ec}

W= {a, an, the} //set of stop words in case of candidate aliases.

C=EC-W= {C_1, C_2, C_3…c_n|1<=n-grams (c) <=5}

'ngrams' is a function which extracts continuous sequences of words (n-grams) from the beginning of the part that matches the wildcard operator *.

Given an entity name, NAME and a set, P of lexical patterns, the function Extract_candidates returns a list of candidate aliases for the name. We associate the given name with each pattern, p in the set of patterns, P and produce queries of the form: "NAME p *".

3.1.3 Web page content retrieval

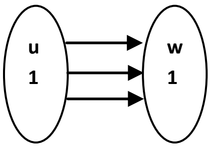

Fig. 4 : Venn Digaram of Web page content retrival (one-to-one mapping)

$$\mathbf{f3}(\mathbf{urls}) \rightarrow \mathbf{W}$$

U={u_1,u_2,u_3,….,u_n}//set of URLs returned for given name pattern by web search engine.

W={w_1,w_2,w_3,….,w_n}// set of Web page content for respective URLs.

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 [1]. For that purpose, we consider the text content of top-k web pages and applying the different similarity coefficient, we get the co-occurrence of name with aliases.

This function we are calling recursively to retrive second level depth page. For that we consider link extraction algorithm [4].

3.1.4 Graph-based data representation

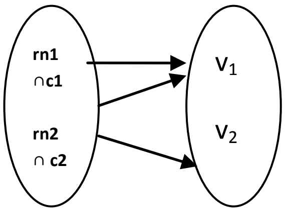

Fig. 5 : Venn Digaram of Graph-based data representation (one-to-many mapping)

$$\mathbf{f4}(\mathbf{rn1}, \mathbf{c1}) \rightarrow \mathbf{V}$$

Words in web page those co-occur with candidate alias are represented as nodes in the co-occurrence graph.

V={v_1,v_2,v_3,…..,vn}// set of vertices if they co-occur

Otherwise ⦰

IF v1 ⟶ U and v2 ⟶U Then *f4* a mapping fn : E -> v1 × v2 // edges if they found together.

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 [3].

3.1.5 Word Score Calculation

To find correct alias from alias set, we used various simlarity coefficient along with graph mining measures.

3.1.5.1 Lexical pattern frequency

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 entity like personal name.

3.1.5.2 Co-occurrence frequency

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 n.

3.1.5.3 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,

$$WebDice(n,x) = \frac{2*hits(n\ \mathrm{I}\ \ x)}{hits(n)+hits(x)} \tag{1}$$

3.1.5.4 *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,

$$\propto (h,n) = \frac{t}{d} \tag{2}$$

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*.

3.1.5.5 *Degree Distribution*

Degree Distribution is defined as degree of a node in a network is the number of connectionist has to other nodes. It is also called as probability distribution of these degrees over the whole network [2]. 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. We distribution of aliases are calculated by using hyperlink structure of the web.

$$Degree = \frac{Out-Link}{In-Link} \tag{3}$$

3.1.5.6 *Clustering Coefficient*

Δ v = | {(u,w) Є E | (v, u) Є E and (v,w) Є E}

The number of triples at a node v is the number of paths of length two in which v is the central node. Therefore, for a node v, the number of triples at node v is [2].

$$C(u) = \frac{d(v)}{T(v)} \tag{4}$$

3.2Normalization

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.

4. DATA SET

Here to get lexical pattern, we consider seed list of name alias pair. In the seed list, we consider data from different entities. We create name-alias pair from personal name data set, location name data set, entity name alias data set. The data set include people from various fields of cinema, sports, politics, and science. For location, we consider Indian city name and for entity name, we consider the name of festivals.

We crawl the URLs and retrieve the relevant data from web page. The data which we retrieved contains lots of noise. So we preprocessed data using seed rule such removal of stop words, navigation links, remove the number, hyper, for anchor text we used html parser to get required value from tag.

5. EXPERIMENT

System Configuration for proposed system required minimum dual core processor @1.8 GHZ, Minimum RAM 1 GB and 128 kbps internet connection and implement on JDK 1.7.

5.1Pattern Extraction

Using lexical pattern extraction algorithm, we extract various pattern. All the patterns are not giving sufficient data related with aliases. For this purpose, we rank patterns according to F-score value. F-score is nothing but harmonic mean between precision and recall of the pattern.

$$\text{Precision (p)} = \frac{\text{No.ofCorrect aliases retrieved by p}}{\text{No.of total aliases retrieved by p}} \tag{5}$$

$$\text{Recall (p)} = \frac{\text{No.of Correct aliases retrieved by p}}{\text{No.of total aliases in dataset}} \tag{6}$$

From this we calculated F-score as

$$\mathrm{F-score} = \frac{2 * \mathrm{Precision(p)} * \mathrm{Recall(p)}}{\mathrm{Precision(p)} + \mathrm{Recall(p)}} \tag{7}$$

Table 5.1 Overall Lexical pattern and F-score

Pattern	F-Score
better known as [NAME]	0.0869
is nicknamed as[NAME]	0.0786
[NAME] popularly known as *	0.0785
Also known as *	0.378
[NAME] blog *	0.2

4.1 Candidate alias extraction

We use here mean reciprocal mean (MRR) and AP to evaluate different approaches.

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{R_i} \tag{8}$$

Overall performance we get by using our proposed system for our given dataset is MRR= 0.75 and AP=0.5 If a method rank correct alias at the top then it receives a higher MRR and AP values. From this result our system gives better result than previous baseline method.

Our proposed method extracted various different aliases for the given entity. Here in this table we display Real name and most likely aliases we get by our method.

Table 5.2 Real name and its most likely candidate aliases

Real Name	**Candidate aliases**
Sachin Tendulkar	God of cricket, Master blaster, Tendlya
Mumbai	Bombay, economic city, chatrapati, web young
Mahendra singh Dhoni	MS Dhoni, mahendra, msd
Amitabh Bachchan	BigB, angry young man
William Howard Gates	Bill Gates

6. CONCLUSION

We proposed name alias detection using graph mining method. Here we proposed four different possible approaches such as lexical pattern frequency, co-occurrence frequency, web dice and graph mining measures. This method gives maximize recall and MRR and improve the precision in relation detection system .It is also useful in various tasks such as relation detection, information retrieval system and sentiment analysis system.

7. 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] Christian Borgelt 2009 Graph Mining: An Overview In Proc. 19th GMA/GI Workshop Computational Intelligence, Germany.

[3] D. Kavitha 2011 A Survey on Assorted Approaches to Graph Data Mining In International Journal of Computer Applications (0975 – 8887) Volume 14– No.1.

[4] Deepayan Chakrabarti,Yiping Zhany, Christos Faloutsos 2004 R-MAT: A Recursive Model for Graph Mining In Proceedings of the 2004 SIAM International Conference on Data Mining.

[5] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka 2006 Extracting Key Phrases to Disambiguate personal names on the web In CICLing'06 Proceedings of the 7th international conference on Computational Linguistics and Intelligent Text Processing.

[6] Dmitri V. Kalashnikov Zhaoqi Chen Rabia Nuray-Turan Sharad Mehrotra Zheng Zhang 2009 Web People Search via Connection Analysis In IEEE International Conference on Data Engineering.

[7] T. Hokama and H. Kitagawa 2006 Extracting Mnemonic Names of People from the Web In Proc. Ninth Int’l Conf. Asian Digital Libraries (ICADL ’06), pp. 121-130.

[8] C. Galvez and F. Moya-Anegon 2007 Approximate Personal Name- Matching through Finite-State Graphs In J. Am. Soc. for Information Science and Technology, vol. 58, page No.. 1-17.

[9] Md. Rafiqul Islam, Md. Rakibul Islam 2008 An Effective Term Weighting Method Using Random Walk Model for Text Classification In Proceedings of 11th International Conference on Computer and Information Technology (ICCIT 2008) 25-27, Khulna, Bangladesh.

[10] Michael Berry 2010 Text Mining Application and Theory, John Wiley and Sons Ltd.

[11] Soumen Chakrabarti 2003 Mining the web: Discover the web form hypertext data, ISBN 1558607544 Elsevier.

[12] G. Salton and M. McGill 1986 Introduction to Modern Information Retreival. McGraw-Hill Inc.

[13] A. Bagga and B. Baldwin, 1998 Entity-Based Cross-Document Coreferencing Using the Vector Space Model In Proc. Int’l Conf. Computational Linguistics (COLING ’98), Page No.. 79-85.

[14] Sumitra Jakhete, Shweta Dharmadhikari 2012 Analysis of Anchor text based on Pattern Growth Graph Algorithm for Name Alias Detection System In CiiT International Journal of Data Mining and Knowledge Engineering. DOI: DMKE062012003.

[15] Sumitra J., Shweta D., Madhuri C. 2012 Name Alias Detection system using graph mining method In 2nd international conference on computer application, Pondicherry, Volume 5,Page No. .125-127.

[16] Web search Engine such as www.googleapi.com //30-06-2012.