ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Improved Ranking Methodology for Alias Name Search

M. Vanitha Sheba[1], B. Sundar Raj[*2]

[1]Assistant Professor, Department of CSE, Jerusalem College of Engineering, Chennai, Tamil Nadu, India

[2] Assistant Professor, Department of CSE, Bharath University, Chennai, Tamil Nadu, India

[*]Corresponding Author

ABSTRACT: Identifying the details of both the aliases and the personal name of a person using a single query is generally a tiresome process. A solution for this problem can be achieved by improving the search process. In this project the set of patterns that describes how the aliases can be represented in different ways is first extracted. The name given as input is compared with the extracted patterns to find the aliases of a given name. The improvement made here is the introduction of a Ranking methodology for the candidate Name and Aliases. The methodology does the ranking based on the lexical pattern frequency and page count. Since considering page count can be less efficient sometime, in this project ranking uses the number of times the same name is repeated in the dataset. If the name count returned for a given input name is high, then the displayed files are more related to the same.

KEYWORDS:name alias, ranking, web mining, pattern extraction.

I. INTRODUCTION

In Internet people use to search the information for famous person. Retrieving information about people becomes difficult, if the particular person have different nicknames or alias names. Alias name means another name or assumed name. It will be difficult to find the correct information if a person has more alias names. That person will have the original or real name in which he is known by everyone. But his friends or fans may use different names. Bloggers will use different name for the same person. So retrieving all information stored in various names of same person with single query is difficult.

A.Overview of the Project

In previous methods web search engine was used for information retrieval. In training data the details of nicknames and alias names of a person are stored. Training data is that which is used to train the application for extracting patterns. Patterns are the one which it is used to connect two different names used for a single person. Some example for patterns are “also called as, also known as, otherwise called as, etc”.

Only single query is given for retrieving all the information stored in the data set for a single person who has more than one nick names or alias names. First the patterns are extracted from the training data which contains the name and alias of a famous person using pattern extraction algorithm. This algorithm contains the function that “Extract Patterns” returns a list of lexical patterns that frequently connect names and their aliases in web snippets .Then the input is given which is a name of a famous person. The given input is compared with those extracted patterns and the alias names of the candidates are generated using the candidate alias extraction algorithm that contains, the function “Extract Candidates” returns a list of candidate aliases for the name. Ranking is performed by two methods in older method. They are lexical pattern frequency and web page counts.

In this project ranking is performed on the basis of “number of times that particular name is repeated in the document” that is present in the dataset. If the computed rank is high then it is more related to that person. A ranking is a relationship between a set of items such that, for any two items, the first is either 'ranked higher than', 'ranked lower than' or 'ranked equal to' the second.

Web Mining

Web mining - is the application of data mining techniques to discover patterns from the Web. According to analysis targets, web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining.

Web Usage Mining

Web usage mining is the process of extracting useful information from server logs i.e. user's history. Web usage mining is the process of finding out what users are looking for on the Internet. Some users might be looking at only textual data, whereas some others might be interested in multimedia data.

Web Structure Mining

Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site. According to the type of web structural data, web structure mining can be divided into two kinds:

1. Extracting patterns from hyperlinks in the web: a hyperlink is a structural component that connects the web page to a different location
2. Mining the document structure: analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

Information Retrieval

Recovery of information, especially in a database stored in a computer has two main approaches are matching words in the query against the database index (keyword searching) and traversing the database using hypertext or hypermedia links. Keyword searching has been the dominant approach to text retrieval hypertext has so far been confined largely to personal or corporate information-retrieval applications. Evolving information-retrieval techniques, exemplified by developments with modern Internet search engines, combine natural language, hyperlinks, and keyword searching. Other techniques that seek higher levels of retrieval precision are studied by researchers involved with artificial intelligence.

Pattern Matching

Pattern matching is the act of checking some sequence of tokens for the presence of the constituents of some pattern. In contrast to pattern recognition, the match usually has to be exact. The patterns generally have the form of either sequences or tree structures. Uses of pattern matching include outputting the locations of a pattern within a token sequence, to output some component of the matched pattern, and to substitute the matching pattern with some other token sequence (i.e., search and replace). Sequence patterns (e.g., a text string) are often described using regular expressions and matched using techniques such as backtracking.

II. RELATED SEARCH

Alias identification is closely related to the string matching in which the objective is to determine whether two mentions of a name in different documents refer to the same entity**.** There are two algorithms[7] for string matching algorithms. They are the basic jaro algorithm and jaro-winkler algorithm. They used the Approximate String Matching (ASM) algorithms to detect the aliases of the names. It increases the similarity for both true and false aliases. It works only on Arabic names dataset.

Word co-occurrence graph is an undirected graph that is used to represent words that appear in anchor texts. Words are considered as co-occurring if two anchor texts contain these words to the same URL. Representing words that appear in anchor texts as a graph enables us to capture the complex inter-relations between the words. The problem of alias extraction is modelled [4]as one of the ranking node in the graph with represent to the given name.

The problem of ranking the authority of name aliases of the same user based on emails .A ranking method based [8]on email communication relation analysis and morphologically similar alias clustering is performed. It is must to scan each email in the dataset, which is time consuming. This proposed system includes the alias authority ranking. A string has similarity measure to detect duplicates in biography databases. However, an inherent limitation [2] of such string matching approach is that they cannot identify aliases, which share no words or letters with real names. For example, approximate string matching method would not identify fresh prince as an alias of Will Smith.

DOI: 10.15680/ijircce.2015.0307182

An alias extraction method [5], that is, specific to the Japanese language. For a given name p, they search for the query”*koto p” and extract the context that matches the asterisk. The Japanese word koto roughly corresponds to also known as in English. However koto is a highly ambiguous word in Japanese that can also mean incident, thing, matter, experience, and task.

In personal name disambiguation [3] the goal is to disambiguate various people that share the same name (namesakes). Given an ambiguous name, most name disambiguation algorithm 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 search task (WePS) provided an evaluation data set and compared various name disambiguation system. However, 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.

III. PROBLEM STATEMENT AND PROPOSED SYSTEM

A.Problem Statement

The existing system defines a numerous ranking scores. But the page count is also difficult if the user opens more pages in web that is not related to the input.

B.Proposed System

With the training data set (name, alias) a set of patterns is extracted using pattern extraction algorithm from the given dataset. Then the extracted pattern is compared with the input given by the user using candidate extraction algorithm. Set of candidate alias names are formed. Ranking method is improved by counting the number of times the same name is repeated in the dataset. We can conclude that if the name count is more, then the displayed files are more related to the given input.

IV. SYSTEM DESIGN

The design of the application is explained in the forth coming subdivisions. The application is designed so as to meet the specification.

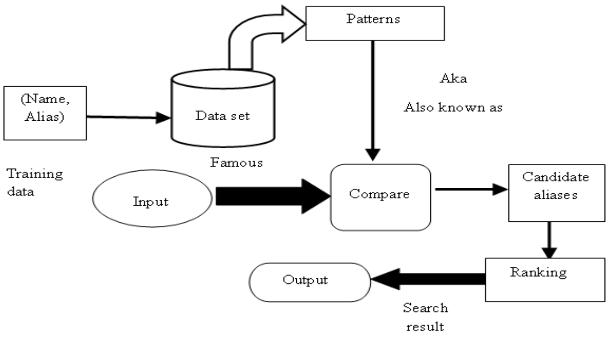

Fig. 1. System Architecture

A.Training Data

The training data (name,alias) in Fig. 1 is the portion of the initial data set that contains input values and target values that are used to develop a predictive model. Training data is that which is used to train the application for extracting patterns. In training data the details of nicknames and alias names of a person are stored. Training data is that which is used to train the application for extracting patterns.

B.Data Set

A data set is mentioned in Fig.4.1 is a named collection of data that contains individual data units organized (formatted) in a specific manner and accessed by a specific access method that is based on the data set organization.

Types of data set organization include sequential, relative sequential, indexed sequential, and partitioned. A data set is a named collection of data that contains individual data units organized (formatted) in a specific and is accessed by a specific access method that is based on the data set organization. Types of data set organization include sequential, relative sequential, indexed sequential, and partitioned. Access methods include the Virtual Sequential Access Method (VSAM) and the Indexed Sequential Access Method (ISAM).

A data set corresponds to the concept of a file in other operating systems such as Linux and Windows 2000. Data set organization and file format are terms that have a close correspondence. A data set generally contains a collection of business data (names, salaries, sale Figures, and so forth) whereas a file can contain many types of data (graphic images, audio data, video data, and so forth). For business data, the database is a newer alternative to the data set and the file[1].

C.Patterns

As mentioned in Fig. 4.1 Searching for patterns is one of the main goals in data mining. Patterns have important applications in many KDD domains like rule extraction or classification. Patterns are the one which it is used to connect two different names used for a single person. Some example for patterns are also called as, also known as, otherwise called as, etc. patterns are extracted from the training data which contains the name and alias of a famous person using pattern extraction algorithm. Then the input is given which is a name of a famous person[2].

D.Comparing

Comparing is when you try to find the similarities between two things while contrasting and trying to distinguish the differences. The input is compared with those extracted patterns and the alias names of the candidates are generated using the candidate alias extraction algorithm[3].

E. Ranking

A ranking is a relationship between a set of items such that, for any two items, the first is either 'ranked higher than', 'ranked lower than' or 'ranked equal to' the second .Ranking is performed on the basis of number of times that particular name is repeated in the document that is present in the dataset. If the computed rank is high then it is more related to that person[4].

V. SYSTEM DESCRIPTION

A.Pattern Extraction

We extract the set of patterns using name and alias of a known person. Patterns are extracted using pattern extraction algorithm.

Pattern Extraction Algorithm:

```
EXTRACT PATTERNS(S)
▪    S is a set of (NAME, ALIAS) pairs
▪    For each S do
▪    D ← GetSnippets(“NAME * ALIAS”)
▪    For each snippet d € D
▪    Do P ← P+CreatePattern(d)
▪    Return(p)
```

Many modern search engines provide a brief text snippet for each search result by selecting the text that appears in the web page in the proximity of the query. Such snippets provide valuable information related to the local context of the query. For names and aliases, snippets convey useful semantic clues that can be used to extract lexical patterns that are frequently used to express aliases of a name. For example, consider the snippet returned by Google2 for the query "Will Smith _ The Fresh Prince."

Here, the wildcard operator is used _ to perform a NEAR query and it matches with one or more words in a snippet. In addition to aka, numerous clues exist such as nicknamed, alias, real name is nee, which are used on the web to represent aliases of a name. Consequently, the shallow pattern extraction method Lexico - syntactic patterns have been used in numerous related tasks such as extracting hypernyms and meronyms[5].

Given a set S of (NAME, ALIAS) pairs, the function ExtractPatterns returns a list of lexical patterns that frequently connect names and their aliases in web snippets. For each (NAME, ALIAS) pair in S, the GetSnippets function downloads snippets from a web search engine for the query "NAME _ ALIAS." Then, from each snippet, the Create-Pattern function extracts the sequence of words that appear between the name and the alias. Results of our preliminary experiments demonstrated that consideration of words that fall outside the name and the alias in snippets did not improve performance. Finally, the real name and the alias in the snippet are, respectively, replaced by two variables [NAME] and [ALIAS] to create patterns. Our definition of lexical patterns includes patterns that contain words as well as symbols such as punctuation markers[6].

We extract the pattern [NAME], aka [ALIAS]. We repeat the process described above for the reversed query, "ALIAS _ NAME" to extract patterns in which the alias precedes the name. In the experiments, the number of matched words is limited with "_" to a maximum of five words. Because snippets returned by web search engines are very short in length compared to the corresponding source documents, increasing the matching window beyond five words did not produce any additional lexical patterns. Once a set of lexical patterns is extracted, we use the patterns to extract candidate aliases for a given name[7].

Pattern Extraction Diagram:

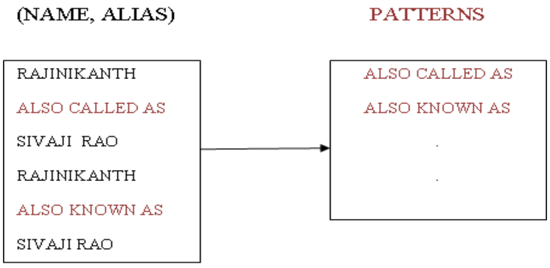

Fig.2. Example for patterns

Fig. 2 explains how the patterns are extracted from the training set. The set of name and alias of a famous person is given as input for the pattern extraction module. The patterns which are present between the given name and alias are extracted. The threshold value which counts the number of times the particular pattern is repeated. Because there may be other words or string that can be used in between two names other than patterns. So, only the repeated string is said to be the exact pattern. If the threshold value is equal to or greater than 5 then the extracted pattern is found as perfect and displayed[8].

B. Candidate Alias Extraction

In this module the candidate alias extraction algorithm to extract the alias name for the given input is used. We associate the given name with each pattern extracted above and the alias names are extracted.

Candidate Alias Extraction Algorithm

```
EXTRACT CANDIDATES (NAME, P)
▪        P is a set of patterns
▪        For each pattern p € P do
▪        D ← GetSnippets("NAME p *)
▪        For each snippet d € D
▪        do C ← C+GetNgrams(d, NAME, p)
```

- return(C)

Given a 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_.” Then, the GetSnippets function downloads a set of snippets for the query.

Finally, the GetNgrams function extracts continuous sequences of words (n-grams) from the beginning of the part that matches the wildcard operator. Experimentally, we selected up to five grams as candidate aliases[9].
For efficiency reasons, the number of snippets downloaded is limited by the function GetSnippets to a maximum of 100 in both Algorithms. In Google it is possible to retrieve 100 snippets by issuing only a single query by setting the search parameter num to 100, thereby reducing the number queries required in practice.

Candidate Extraction Diagram

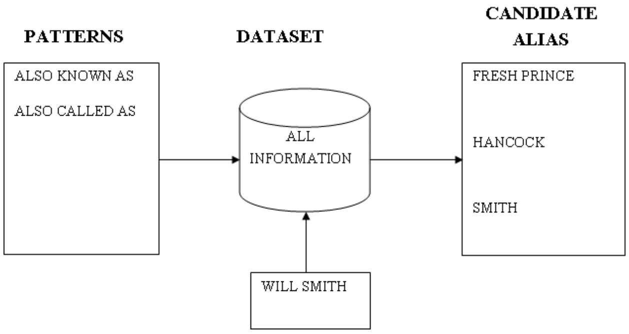

Fig. 3. Example for candidate alias

Fig. 3 explains how the alias names of a person are extracted. The input given which is the name of a famous person is compared with the patterns extracted and the alias names are extracted using candidate alias extraction algorithm[10]. The algorithm will extract all the strings after the patterns. There may be any other string other than the alias name. We set a threshold value which counts the number of times the particular alias name is repeated. If the threshold value is equal to or greater than 3 then the extracted alias is found as exact alias and it is displayed.

C. Ranking
In this module ranking is performed for the extracted candidate alias and their names by their profession.

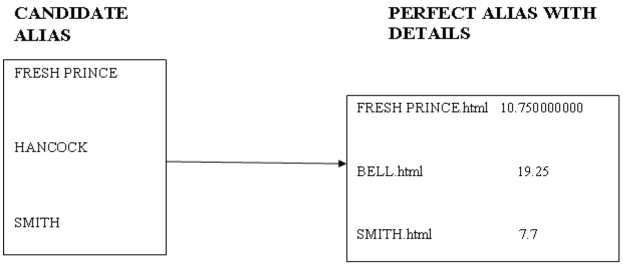

Fig.4. Example for ranking

Fig. 4 explains how the ranking is done and the final output is displayed. Ranking is performed for the candidate Name and Aliases. The ranking is performed by counting the number times the same name is repeated in the dataset[11]. We can conclude that if the name count is more, then the displayed files are more related to the given input. Thus the formula for ranking is

$$Rank = \frac{no.of\ times\ repeated}{total\ no.of\ words\ in\ the\ document} * total\ no.of\ document$$

VI. IMPLEMENTATION

Once the application is being designed, it can be implemented.

A.Pattern Extraction

The first step is to extract patterns that are frequently used to express aliases of a name. To extract these patterns a set of training dataset is prepared which contains the name and aliases of a famous person.

The training data which contains the name and aliases is loaded and it is used to extract the patterns from the given dataset[12].

The code(I) in appendix first coverts the name and alias to lower case and then it compares with the dataset and extract the words in between those name and aliases. Then it checks whether the extracted words are patterns or not by using the threshold value which is set as 5 here. The extracted words are checked if any of the particular patterns is repeated 5 times or more than that. The patterns which satisfy the threshold condition are displayed as patterns as shown below.

"Patterns: also known as, also called as, was known as, was known as"

B.Alias Extraction

The alias names of a given input (i.e. the name of a famous person) are identified with the help of extracted patterns. The given input is compared with the patterns extracted before and the name after that pattern is extracted as alias name as explained in code (II) in appendix. There may be any words other than the alias name after the patterns. To avoid the other words a threshold value is set to count the number of times the particular alias is repeated[13].

The threshold value here is set as 3 as shown in code (III) in appendix. The alias name which satisfies the threshold value is displayed as alias names in the output screen as shown below.

"Candidate Aliases: Fresh Prince, Bell, Manu, Willard Christopher Smith, Fresh Prince comeback, Melbourne, Bel, Serges."

C.Ranking

The ranking is performed for the input name based on the number of times the name and the alias name repeated in a particular document related to the given input name. It is concluded that if the ranking is more, then that particular document is more related to the given name when compared to the other documents.

VII.CONCLUSION

Lexical-pattern-based approach to extract aliases of a given name is implemented. Set of names and their aliases is used as training data to extract lexical patterns that describe numerous ways in which information related to aliases of a name is presented on the dataset. Next, the real name of the person in interest for finding aliases in the extracted lexical patterns, and download snippets from a dataset . A set of candidate aliases from the snippets is extracted. Ranking is performed for the candidate Name and Aliases. The ranking is performed by counting the number times the same name is repeated in the dataset. If the name count is more, then the displayed files are more related to the given input. In this paper pattern extraction and candidate alias extraction are performed with the help of dataset. But in future it can be performed in web search engine.

REFERENCES

1. Bagga and B. Baldwin, "Entity-Based Cross-Document Co- referencing Using the Vector Space Model," Proc. Int'l Conf.Computational Linguistics (COLING '98), pp. 79-85, 1998.
2. Sathyanarayana H.P., Premkumar S., Manjula W.S., "Assessment of maximum voluntary bite force in adults with normal occlusion and different types of malocclusions", Journal of Contemporary Dental Practice, ISSN : 1526-3711, 13(4) (2012) pp.534-538.
3. M.Bilenko and R.Mooney, "Adaptive duplicate detection using learnable string similarity measures,"Proc.SIGKDD'03,2003.
4. Selva Kumar S., Ram Krishna Rao M., Deepak Kumar R., Panwar S., Prasad C.S., "Biocontrol by plant growth promoting rhizobacteria against black scurf and stem canker disease of potato caused by Rhizoctonia solani", Archives of Phytopathology and Plant Protection, ISSN : 0323-5408, 46(4) (2013) pp.487-502.

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.
6. Subha Palaneeswari M., Abraham Sam Rajan P.M., Silambanan S., Jothimalar, "Blood lead in end-stage renal disease (ESRD) patients who were on maintainence haemodialysis", Journal of Clinical and Diagnostic Research, ISSN : 0973 - 709X, 6(10) (2012) pp.1633-1635.
7. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka, "A Co-occurrence Graph-Based Approach for Personal Name Alias Extraction From anchor texts",
8. Sukumaran V.G., Bharadwaj N., "Ceramics in dental applications", Trends in Biomaterials and Artificial Organs, ISSN : 0971-1198, 20(1) (2006) pp.7-11.
9. T.Hokama and E.Kitagawa, "Extracting Mnemonic Names Of People from the Web",Proc.Ninth Int'l Conf. Asian Digital Libraries (ICADL'06), pp.121-130,2006.
10. Selva Kumar S., Ram Krishna Rao M., Balasubramanian M.P., "Chemopreventive effects of Indigofera aspalathoides on 20-methylcholanthrene induced fibrosarcoma in rats", International Journal of Cancer Research, ISSN : ISSN: 1811-9727, 7(2) (2011) pp.144-151.
11. G. Mann and D. Yarowsky, "Unsupervised Personal Name Disambiguation," Proc. Conf. Computational Natural Language Learning (CoNLL '03), pp. 33-40, 2003.
12. Shaikh,M.;Memon,N.;Will,U.K., "Extended Approximate String matching Algorithms to detect Name Aliases", Intelligence and Security Informatics(ISI),2011 IEEE international Conference.
13. Yin Meijuan, Wang Qingxian, Chen Shuming, Liu Xiaonan, Luo Xiangyang, "Ranking the Authority of Name Aliases for Email Users", 2011 Third International Conference. on Multimedia Information.
14. Dr.K.P.Kaliyamurthie, D.Parameswari, Load Balancing in Structured Peer to Peer Systems, International Journal of Innovative Research in Computer and Communication Engineering, ISSN: 2249-2615,pp 22-26, Volume1 Issue 1 Number2-Aug 2011
15. Dr.R.Udayakumar, Addressing the Contract Issue,Standardisation for QOS, International Journal of Innovative Research in Computer and Communication Engineering, ISSN (Online): 2320 – 9801,pp 536-541, Vol. 1, Issue 3, May 2013
16. Dr.R.Udayakumar, Computational Modeling of the StrengthEvolution During Processing And Service Of9-12% Cr Steels, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801,pp 3295-3302, Vol. 2, Issue 3, March 2014
17. P.Gayathri, Assorted Periodic Patterns Intime Series Database Usingmining, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801, pp 5046- 5051, Vol. 2, Issue 7, July 2014.
18. Gayathri, Massive Querying For Optimizing Cost – CachingService in Cloud Data, International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801,pp 2041-2048, Vol. 1, Issue 9, November 2013