INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

AN APPROACH TO EXTRACT ALIASES OF A GIVEN PERSONAL NAME

MS. N. S. CHAPKE[1], PROF. P. K. BHARNE[2]

1. Department of CSE, PG student, SSGMCE, Shegaon (M.S.) India
2. Department of CSE, Assistant Professor, SSGMCE, Shegaon (M.S.) India

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: A person may have multiple personal name aliases i.e. nick names on web. It is important to find aliases in order to get the complete information about particular entity. In various task such as relation extraction, information retrieval, web search, entity disambiguation identifying aliases of a given personal name is useful. The main objective here is to extract the aliases of a given personal name. As a training data, set of well known name alias pair will be given as an input to the method and it will generate the patterns. This extracted patterns along with personal name further given as an input to get the aliases. At the end method comes up with different aliases so, it is equally important to find the most likely aliases. To find that, various ranking scores are used like lexical pattern frequency, word co-occurrences and page counts to measure the association between a name and a candidate alias. These different ranking functions are then integrated into a single ranking function using ranking support vector machine (SVM).

Keywords: Information extraction, Aliases and lexical patterns.

Corresponding Author: MS. N. S. CHAPKE

PAPER-QR CODE

Access Online On:

www.ijpret.com

How to Cite This Article:

NS Chapke, IJPRET, 2014; Volume 2 (9): 363-368

INTRODUCTION

Retrieving information is the task where users might search for documents and information within documents on the web. Many celebrities and experts from various fields are referred by their original names as well as by their aliases on web. Most of the queries to web search engines include person names. Many web pages about person names might also be created by aliases. Extracting complete information about people from the web will be difficult if the person is referred by different aliases so, for that it is important to know the aliases of the person whom anyone interested in. A newspaper article might use real name but in a blog entry people might referred by their aliases instead of personal names. The user would not be able to retrieve all information from the web unless having top ranked aliases of that person. Different types of terms are used to denote aliases on the web like for actors, title of the drama or name of the role later become an alias for the person. Abbreviations of names, variants and acronyms such as JFK for John Fitzgerald Kennedy are also types of name aliases that are frequently observed on web.

To extract the aliases using snippets from the web, a lexical-pattern-based approach is used. By giving the real world name alias data, the aim is to find the lexical patterns. Lexical patterns are the words that appear between the name and alias pair so, with the help of patterns it will be easy to judge aliases. After that evaluate the confidence of extracted lexical patterns and retain the patterns that can accurately discover aliases for various personal names. To select the best aliases among the extracted candidates, numerous ranking scores applied based on three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the Web. It is not restricted to finding the aliases only for personal names. Anyone can find the aliases of other entity type also. For that, need the set of name alias pairs for each entity type. This is particularly attractive because anyone does not need to modify the pattern extraction algorithm to handle different types of named entities.

1. RELATED WORK

The namesake disambiguation problem focuses on identifying the different individuals who have the same name. The existing namesake disambiguation algorithm assumes the real name of a person to be given and does not attempt to disambiguate people who are referred only by aliases. So it is important to have the knowledge of aliases to identify a particular person from his or her namesakes on the web [6]. Hokama and Kitagawa [3] proposed an alias extraction method but that is specific to the only Japanese language. They have used manually crafted patterns. In contrast here patterns are generated automatically. Identifying aliases of a name are important in information retrieval. In information retrieval, to improve recall of a web search on a person name, a search engine can automatically expand a query using aliases of the name [5]. For example, a user who searches for Hideki Matsui might also be interested in retrieving documents in which Matsui is referred to as Godzilla. So, by expanding a query on Hideki Matsui using his alias name Godzilla user will get all the information.

2. LEXICAL PATTEN BASED APPROACH

The proposed method will work on the personal names and give the aliases. Main components are: pattern extraction, alias extraction and ranking. By giving the input as name alias pairs, lexical patterns will be extracted which appears between real name and alias of it. After that by giving the input as extracted patterns along with real name, aliases will be identified. Various ranking scored are used to find the correct aliases among the extracted candidates.

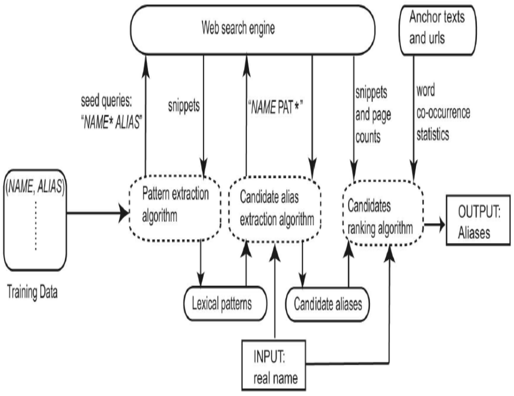
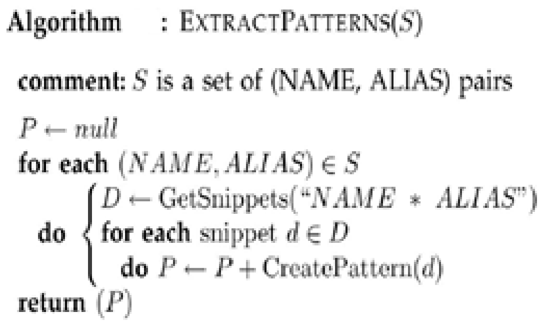


Fig. 1. Proposed method.[1]

3.1 Extract Lexical Patterns

The first step is to find the lexical patterns from snippets. Search engine provides the brief text snippets for every query. Patterns are nothing but the clues through which idea about aliases get clarified. For example, Will Smith, aka (i.e., also known as) the fresh prince, means fresh prince is an alias for will smith. Like aka numerous clues exist such as, nicknamed, alias, real name, which are used on the web to represent the aliases of a name. Not all patterns are equally informative about aliases of a real name. Hence, patterns are ranked according to their F-scores to identify the patterns that accurately convey information about aliases. F-score of a pattern s is computed as the harmonic mean between the precision and recall of the pattern. If all this patters are gathered then the task of finding aliases would get easier. It can be one word or multi words.

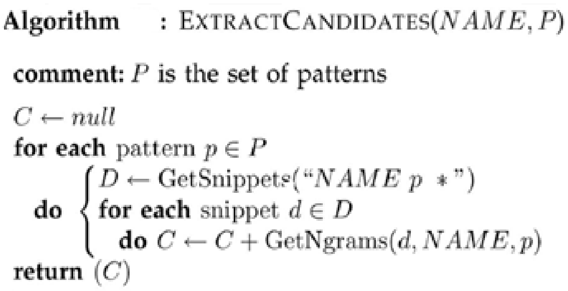


```
Algorithm    : EXTRACTPATTERNS($S$)
 comment: $S$ is a set of (NAME, ALIAS) pairs
 $P \leftarrow null$
 for each $(NAME, ALIAS) \in S$
   do { $D \leftarrow$ GetSnippets("$NAME * ALIAS$")
        for each snippet $d \in D$
          do $P \leftarrow P +$ CreatePattern($d$)
 return ($P$)
```

Fig. 2. An algorithm to extract patterns.[1]

Above algorithm shows the algorithm to extract the lexical patterns. The proposed pattern extraction algorithm can extract a large number of lexical patterns. 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. .An input will be name alias pair and output will be the value that matches the wildcard operator i.e. list of lexical patterns that frequently connect name and their aliases in web snippets. For each pair GetSnippet function downloads snippets from a web search engine for the query "NAME * ALIAS". After it will try to match the query and find out the list of words which appears between name and alias. Finally CreatPattern function extracts the sequence of patterns.

3.2 Extract Aliases



```
Algorithm    : EXTRACTCANDIDATES($NAME, P$)
 comment: $P$ is the set of patterns
 $C \leftarrow null$
 for each pattern $p \in P$
   do { $D \leftarrow$ GetSnippets("$NAME\ p\ *$")
        for each snippet $d \in D$
          do $C \leftarrow C +$ GetNgrams($d, NAME, p$)
 return ($C$)
```

Fig. 3. An algorithm to extract aliases.[1]

Once a set of lexical patterns are extracted it will be given as an input along with the personal name to ExtractCandiate function and it will returns a list of candidate aliases for the given name. The given name is associated with each pattern which is extracted and generates query like NAME p*. NAME is the personal name and p is the patterns which are extracted. By knowing two values it is possible to find the third one. The known name along with extracted pattern will be given as input and above algorithm try find the alias from snippets. The GetSnippets function downloads a set of snippets for the query and matches the value of wildcard operator. The matched value can be considered as an aliases .Finally, the GetNgrams function extracts continuous sequences of words from the beginning of the part that matches the wildcard operator *. Because aliases can have more than one word, just for the safe side it will take the combination of first five words. Candidates that contain only stop words such as a, an, and the, are directly removed. Different terms are used to represent aliases and it can be one word or multiword. In order to get the all possible combinations continuous sequence of words are considered.

3.3 Ranking

While matching the wildcard operator in ExtractCandidates it may take more than one words which is not an alias of name. Two anchor texts might link to a hub for entirely different reasons. Therefore, cooccurrences coming from hubs are prone to noise. If the majority of anchor texts linked to a particular web site use the real name then the confidence of that page as a source of information regarding the person whom we are interested in extracting aliases increases. In this case it is important to remove the invalid aliases and identify the correct one. Three different approaches are used to find the correct aliases which are lexical pattern frequency, word co-occurrences in an anchor text graph [2], and page counts on the web. In lexical pattern frequency, how many times name and alias occur in many lexical patterns is calculated while co-occurrence is defined as number of different urls in which they co-occur. Not all name aliases are equally well represented in anchor texts so in page count overall web is considered.

3. CONCLUSION

Proposed method will automatically generate the aliases of a given personal name by using lexical pattern based approach. For training data set of names and their aliases are used to extract lexical patterns that describe different ways in which information related to aliases of a name is presented on the web. The next step is to download snippets from a web search engine. The extracted patterns are then used to find the aliases for a given name. The candidates will be ranked using various ranking scores computed using three approaches: lexical pattern frequency, co-occurrences in anchor texts, and page counts. Lexical patterns can only be matched within the same document. In contrast, anchor texts can be used to identify aliases of names across documents hence by combining, performance can be improved. Proposed method is not specific to personal names. It can be used for location names and many more. In future, alias extraction can become a common tool for ‘non-celebrities’ and further would be extended to support different languages.

4. REFERENCES

1. D. Bollegala, Y. Matsuo, and M. Ishizuka," Automatic Discovery of Personal Name Aliases from the Web," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011 [2] A. Bagga and B. Baldwin, "Entity-Based Cross-Document Coreferencing Using the Vector Space Model," Proc. Int'l Conf. Computational Linguistics (COLING '98), pp. 79-85, 1998.

2. 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.

3. P. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," Proc. Assoc. for Computational Linguistics (ACL '02), pp. 417-424, 2002.

4. M. Mitra, A. Singhal, and C. Buckley, "Improving Automatic Query Expansion," Proc. SIGIR '98, pp. 206-214, 1998.

5. M. Bilenko and R. Mooney, "Adaptive Duplicate Detection Using Learnable String Similarity Measures," Proc. SIGKDD'03,