Ranking Model Adaptation for Domain Specific Search

Nisha.K, Adhithyaa.N, Stephen.B and Muthu Kumar.P

Abstract:- An individual is typically referred by numerous name aliases on the web. Accurate identification of aliases of a given person name is useful in various web related tasks such as information retrieval, sentiment analysis, personal name disambiguation, and relation extraction. Automatic Discovery of Personal Name Aliases is a method to extract aliases of a given personal name from the web. Given a personal name, the method first extracts a set of candidate aliases. Second, rank the extracted candidates according to the likelihood of a candidate being a correct alias of the given name. Automatically extracted lexical pattern-based approach will efficiently extract a large set of candidate aliases from snippets retrieved from a web search engine. Define numerous ranking scores to evaluate candidate aliases using three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web. To construct a robust alias detection system, integrate the different ranking scores into a single ranking function using ranking support vector machines. Evaluate the proposed method on three data sets: an English personal names data set, an English place names data set, and a Japanese personal names data set. The method outperforms numerous baselines and previously proposed name alias extraction methods, achieving a statistically significant mean reciprocal rank (MRR) of 0.67. Experiments carried out using location names and Japanese personal names suggest the possibility of extending the method to extract aliases for different types of named entities, and for different languages. Moreover, the aliases extracted using the method are successfully utilized in an information retrieval task and improve recall by 20 percent in a relation detection task.

1. INTRODUCTION

Searching for information about people in the web is one of the most common activities of internet users. Around 30 percent of search engine queries include person names. Retrieving information about people from web search engines can become difficult when a person has nicknames or name aliases. For example, the famous Japanese major league baseball player Hideki Matsui is often called as Godzilla on the web. A newspaper article on the baseball player might use the real name, Hideki Matsui, whereas a blogger would use the alias, Godzilla, in a blog entry. We will not be able to retrieve all the information about the baseball player, if we only use his real name. Identification of entities on the web is difficult for two fundamental reasons: first, different entities can share the same name (i.e., lexical ambiguity); second, a single entity can be designated by multiple names (i.e., referential ambiguity). For example, the lexical ambiguity consider the name Jim Clark. Aside from the two most popular namesakes, the formula-one racing champion and the founder of Netscape, at least 10 different people are listed among the top 100 results returned by Google for the name. On the other hand, referential ambiguity occurs because people use different names to refer to the same entity on the web. For example, the American movie star Will Smith is often called the Fresh Prince in web contents. Although lexical ambiguity, particularly ambiguity related to personal names has been explored extensively in the previous studies of name disambiguation, the problem of referential ambiguity of entities on the web has received much less attention. In this project, we specifically examine on the problem of automatically extracting the various references on the web of a particular entity. For an entity e, we define the set A of its aliases to be the set of all words or multiword expressions that are used to refer to e on the web. For example, Godzilla is a one-word alias for Hideki Matsui, whereas alias the Fresh Prince contains three words and refers to Will Smith. Various types of terms are used as aliases on the web. For instance, in the case of an actor, the name of a role or the title of a drama (or a movie) can later become an alias for the person (e.g., Fresh Prince, Knight Rider). Titles or professions such as president, doctor, professor, etc., are also frequently used as aliases. Variants or abbreviations of names such as Bill for William, and acronyms such as JFK for John Fitzgerald Kennedy are also types of name aliases that are observed frequently on the web.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. In our previous example, a user who searches for Hideki Matsui might also be interested in retrieving documents in which Matsui is referred to as Godzilla. Consequently, we can expand a query on Hideki Matsui using his alias name Godzilla. The semantic web is intended to solve the entity disambiguation problem by providing a mechanism to add semantic metadata for entities. An issue that the semantic web currently faces is that insufficient semantically annotated web contents are available. Automatic extraction of metadata can accelerate the process of semantic annotation. For named entities, automatically extracted aliases can serve as a useful source of metadata, thereby providing a means to disambiguate an entity. Identifying

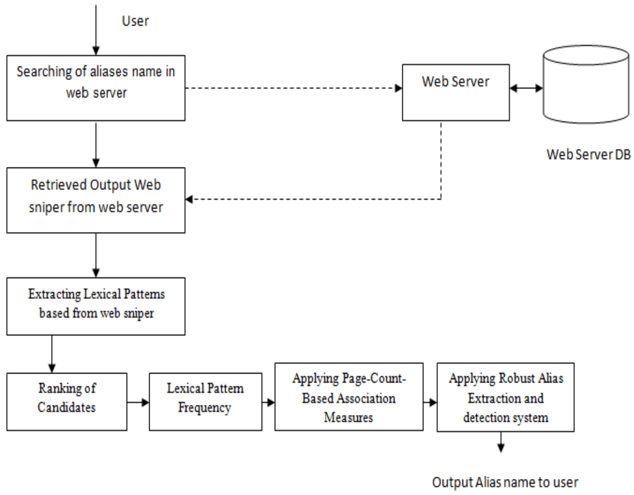
Nisha.K is working as Assistant Professor, Department of IT, Adhithyaa.N, Stephen.B and Muthu Kumar.P are Final Year Students, Department of IT, SNS College Of Engineering, Coimbatore

aliases of a name are important for extracting relations among entities. For example, Matsuo et al. Propose a social network extraction algorithm in which they compute the strength of the relation between two individuals A and B by the web hits for the conjunctive query, “A” and “B”. However, both persons A and B might also appear in their alias names in web contents. Consequently, by expanding the conjunctive query using aliases for the names, a social network extraction algorithm can accurately compute the strength of a relationship between two persons. Along with the recent rapid growth of social media such as blogs, extracting and classifying sentiment on the web has received much attention . Typically, a sentiment analysis system classifies a text as positive or negative according to the sentiment expressed in it. When people express their views about a particular entity, they do so by referring to the entity not only using the real name but also using various aliases of the name. By aggregating texts that use various aliases to refer to an entity, a sentiment analysis system can produce an informed judgment related to the sentiment.

We propose a fully automatic method to discover aliases of a given personal name from the web. Our contributions can be summarized as follows:

- 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 name alias data. We evaluate the confidence of extracted lexical patterns and retain the patterns that can accurately discover aliases for various personal names. Our pattern extraction algorithm 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.
- To select the best aliases among the extracted candidates, we propose numerous ranking scores based upon three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web. Moreover, using real-world name alias data, we train a ranking support vector machine to learn the optimal combination of individual ranking scores to construct a robust alias extraction method.
- We conduct a series of experiments to evaluate the various components of the proposed method. We compare the proposed method against numerous base lines and previously proposed name alias extraction methods on three data sets: an English personal names data set, an English place names data set, and a Japanese personal names data set. Moreover, we evaluate the aliases extracted by the proposed method in an information retrieval task and a relation extraction task.

Block Diagram:


User
Searching of aliases name in web server
Web Server
Web Server DB
Retrieved Output Web sniper from web server
Extracting Lexical Patterns based from web sniper
Ranking of Candidates
Lexical Pattern Frequency
Applying Page-Count-Based Association Measures
Applying Robust Alias Extraction and detection system
Output Alias name to user

1. Ranking of Candidates
2. Applying Page-Count-Based Association Measures.
3. Applying Robust Alias Extraction and detection system

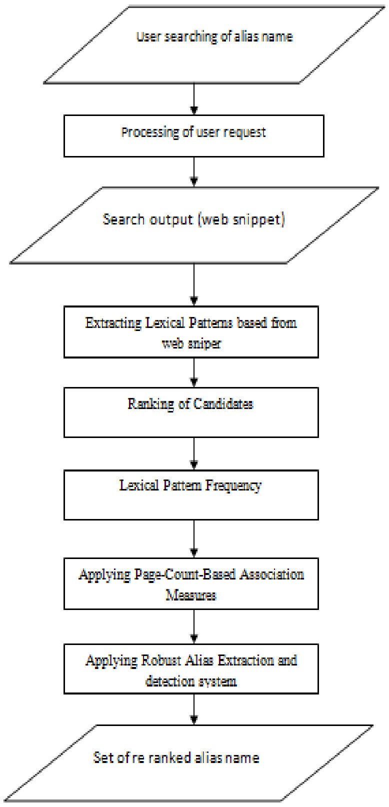
2. Ranking of Candidates:

Considering the noise in web snippets, 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 a given name. We model this problem of alias recognition as one of ranking candidates with respect to a given name such that the candidates, who are most likely to be correct aliases are assigned a higher rank. First, we define various ranking scores to measure the association between a name and a candidate alias using three different approaches: lexical pattern frequency, word co-occurrences in an anchor text graph and page counts on the web.

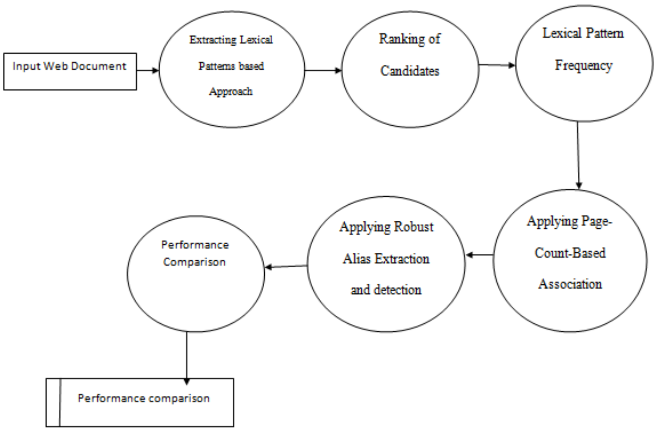
3. Applying Page-Count-Based Association Measures:

We defined various ranking scores using anchor texts. However, not all names and aliases are equally well represented in anchor texts. Consequently, in this section, we define word association measures that consider co-occurrences not only in anchor texts but in the web overall. Page counts retrieved from a web search engine for the conjunctive query, “p and x,” for a name p and a candidate alias x can be regarded as an approximation of their cooccurrences in the web. We compute popular word association measures using page counts returned by a search engine.

System Flow Diagram:


User searching of alias name
Processing of user request
Search output (web snippet)
Extracting Lexical Patterns based from web sniper
Ranking of Candidates
Lexical Pattern Frequency
Applying Page-Count-Based Association Measures
Applying Robust Alias Extraction and detection system
Set of re ranked alias name

Data Flow Diagram:


Input Web Document
Extracting Lexical Patterns based Approach
Ranking of Candidates
Lexical Pattern Frequency
Applying Page-Count-Based Association
Applying Robust Alias Extraction and detection
Performance Comparison
Performance comparison

3.1WebDice:

We compute the Dice, webDice(p,x) between anamep and a candidate alias x using page counts as

$$\mathrm{WebDice}(p,x) = \frac{2 \times \mathrm{hits}(``p\ AND\ x")}{\mathrm{hits}(p) + \mathrm{hits}(x)}.$$

Here, hits(q) is the page counts for the query q.

3.2. WebPMI:

We compute the PMI, WebPMI(p,x) using page counts as follows:

$$\mathrm{WebPMI}(p,x) = \log_2 \frac{L \times \mathrm{hits}(``p\ AND\ x")}{\mathrm{hits}(p) \times \mathrm{hits}(x)}.$$

Here, L is the number of pages indexed by the web search engine, which we approximated as $L=10^{10}$ according to the number of pages indexed by Google. It should be noted however that the actual value of L is not required for ranking purposes because it is a constant and can be taken out from the definition of WebPMI as an additive term. Both WebDice and WebPMI measures are described.

3.3. Conditional Probability:

Using page counts, we compute the probability of an alias, given a name as

$$Prob(x|p) = \frac{\mathrm{hits}(``p\ AND\ x")}{\mathrm{hits}(p)}.$$

Similarly, the probability of a name, given an alias is

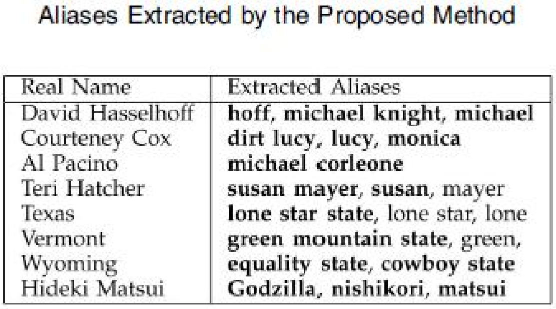
$$Prob(p|x) = \frac{\mathrm{hits}(``p\ AND\ x")}{\mathrm{hits}(x)}.$$

Unlike PMI and the Dice, conditional probability is an asymmetric measure.

4. Applying Robust Alias Extraction and detection system:

We compare the proposed SVM-based method against various individual ranking scores (baselines) and previous studies of alias extraction (HK [13]) on Japanese personal names data set. We used linear, polynomial (quadratic), and radial basis functions (RBF) kernels for ranking SVM. Mean reciprocal rank (MRR) and AP [26] is used to evaluate the different approaches. Table 1 presents the aliases extracted for some entities included in our data sets. Overall, the proposed method extracts most aliases in the manually created gold standard (shown in bold). It is noteworthy that most aliases do not share any words with the name nor acronyms, thus would not be correctly extracted from approximate string matching methods. It is interesting to see that for actors the extracted aliases include their roles in movies or television dramas. We evaluate the effect of aliases on a real-world relation detection task as follows: First, we manually classified 50 people in the English personal names data set, depending on their field of expertise, into four categories: music, politics, movies, and sports. Following earlier research on web-based social network extraction and we measured the association between two people using the PMI between their names on the web.

Table 1



Aliases Extracted by the Proposed Method

Real Name	Extracted Aliases
David Hasselhoff	**hoff, michael knight, michael**
Courteney Cox	**dirt lucy, lucy, monica**
Al Pacino	**michael corleone**
Teri Hatcher	**susan mayer, susan**, mayer
Texas	**lone star state**, lone star, lone
Vermont	**green mountain state**, green,
Wyoming	**equality state, cowboy state**
Hideki Matsui	**Godzilla, nishikori, matsui**

We then use group average agglomerative clustering (GAAC) [18] to group the people into four clusters. Initially, each person is assigned to a separate cluster. In subsequent iterations, GAAC process merges the two clusters with the highest correlation

5. CONCLUSION

We proposed a lexical-pattern-based approach to extract aliases of a given name. We use a set of names and their aliases as training data to extract lexical patterns that describe numerous ways in which information related to aliases of a name is presented on the web. Next, we substitute the real name of the person that we are interested in finding aliases in the extracted lexical patterns, and download snippets from a web search engine. We extract a set of candidate aliases from the snippets. The candidates are ranked using various ranking scores computed using three approaches: lexical pattern frequency, co-occurrences in anchor texts, and page counts-based association measures. Moreover, we integrate the different ranking scores to construct a single ranking function using ranking support vector machines. We evaluate the proposed method using three data sets: an English personal names data set, an English location names data set, and a Japanese personal names data set. The proposed method reported high MRR and AP scores on all three data sets and outperformed numerous baselines and a previously proposed alias extraction algorithm. Discounting co-occurrences from hubs is important to filter the noise in co-occurrences in anchor texts. For this purpose, we proposed a simple and effective hub discounting measure. Moreover, the extracted aliases significantly improved recall in a relation detection task and render useful in a web search task.

REFERENCES

[1] R. Guha and A. Garg, “Disambiguating People in Search,” technical report, Stanford Univ., 2004.

[2] J. Artiles, J. Gonzalo, and F. Verdejo, “A Testbed for People Searching Strategies in the WWW,” Proc. SIGIR ’05, pp. 569-570, 2005.

[3] G. Mann and D. Yarowsky, “Unsupervised Personal Name Disambiguation,” Proc. Conf. Computational Natural Language Learning (CoNLL ’03), pp. 33-40, 2003.

[4] 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.

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

[6] M. Mitra, A. Singhal, and C. Buckley, “Improving Automatic Query Expansion,” Proc. SIGIR ’98, pp. 206-214, 1998.

[7] P. Cimano, S. Handschuh, and S. Staab, “Towards the Self-Annotating Web,” Proc. Int’l World Wide Web Conf. (WWW ’04), 2004.

[8] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H.Takeda, K. Hasida, and M. Ishizuka, “Polyphonet: An Advanced Social Network Extraction System,” Proc. WWW ’06, 2006.

[9] 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.

[10] 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.

[11] C. Galvez and F. Moya-Anegon, “Approximate Personal Name- Matching through Finite-State Graphs,” J. Am. Soc. for Information Science and Technology, vol. 58, pp. 1-17, 2007.

[12] M. Bilenko and R. Mooney, “Adaptive Duplicate Detection Using Learnable String Similarity Measures,” Proc. SIGKDD ’03, 2003.

[13] 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.

[14] M. Hearst, “Automatic Acquisition of Hyponyms from Large Text Corpora,” Proc. Int’l Conf. Computational Linguistics (COLING ’92), pp. 539-545, 1992.

[15] 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.

[16] S. Chakrabarti, Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann, 2003.

[17] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” Information Processing and Management, vol. 24, pp. 513-523, 1988.

[18] C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[19] T. Dunning, “Accurate Methods for the Statistics of Surprise and Coincidence,” Computational Linguistics, vol. 19, pp. 61-74, 1993.

[20] K. Church and P. Hanks, “Word Association Norms, Mutual Information and Lexicography,” Computational Linguistics, vol. 16, pp. 22-29, 1991.

[21] T. Hisamitsu and Y. Niwa, “Topic-Word Selection Based on Combinatorial Probability,” Proc. Natural Language Processing Pacific-Rim Symp. (NLPRS ’01), pp. 289-296, 2001.

[22] F. Smadja, “Retrieving Collocations from Text: Xtract,” Computational Linguistics, vol. 19, no. 1, pp. 143-177, 1993.

[23] D. Bollegala, Y. Matsuo, and M. Ishizuka, “Measuring Semantic Similarity between Words Using Web Search Engines,” Proc. Int’l World Wide Web Conf. (WWW ’07), pp. 757-766, 2007.

[24] T. Joachims, “Optimizing Search Engines Using Clickthrough Data,” Proc. ACM SIGKDD ’02, 2002.

[25] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying Conditional Random Fields to Japanese Morphological Analysis,” Proc. Conf. Empirical Methods in Natural Language (EMNLP ’04), 2004.

[26] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. ACM Press/Addison-Wesley, 1999.

[27] P. Mika, “Ontologies Are Us: A Unified Model of Social Networks and Semantics,” Proc. Int’l Semantic Web Conf. (ISWC ’05), 2005.