IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555
Vol. 2, No.6, December 2012

Efficient Extraction of Word Aliases from Web

Madhulika.Y[1], Dathathreya.P[2], Hanumantha Rao.N[3]

[1]Dept. of Computer Science & Engineering
Guru Nanak Engineering College, Hyderabad, India
madhulika.yarlagadda@gmail.com

[2]Dept. of Computer Science & Engineering
Guru Nanak Engineering College, Hyderabad, India
datta.gnec@gmail.com

[3]Dept. of Computer Science & Engineering
Visvesvaraya College of Engineering and Technology, Hyderabad, India
hanu.nadendlla@gmail.com

***Abstract*—On the web every individual is usually referred by different aliases names. Web related tasks will make use of exact aliases of given name in name disambiguation sentiment analysis etc. In this paper, we propose a cluster based method to extract aliases of a given name from the web. The proposed method extracts a set of candidate aliases and forms a cluster from the given name. Thus, reduces computation time to get correct aliases from the given input name. The results shown that the proposed cluster based scheme can provide significant time saving even for a small number of documents**.

Keywords-Text Mining; Web Text Analysis; Information Extraction; Cluster

I. INTRODUCTION

A cross-document coreference resolution algorithm based on the vector space model has been proposed [1], by performing each individual document coreference resolution to extract coreference chains, and clustering the coreference chains to identify all mentions of a name in the document set. The problem of cross-document coreference resolution is defined as when two mentions of a name refer to the same entity [12], [13], and this problem is closely related to the identification of alias. In general, firstly we extract aliases from individual documents, and secondly cluster the documents in order to find aliases from each document. In case of personal name disambiguation [14], [15], the goal is to remove the ambiguity when two different people share the same name. For a given an ambiguous name, where individual of ambiguous name from all the documents are grouped into a single cluster.

The web people search task (WePS) [2] has been provided an evaluation data set in case of web searching scenario, and compared various name disambiguation algorithms. However, the name alias extraction differs from name disambiguation because the objective of name disambiguation is to identify the different individuals that are referred by the same ambiguous name.

To extract abbreviations of names by using string matching algorithms has been proposed in [20] (e.g., matching Mahendra Singh Dohni with M. S. Dhoni). To compare names we can use regular expressions and edit-distance based methods. To detect duplicates in bibliography databases a string similarity measure has been proposed in [21]. Moreover, some patterns may not cover different ways that provides information about name aliases. In our alias extraction, we are more focused in extracting all references to the given word from the web.

The rest of the paper is organized as follows. Section II discusses the related work. Section III describes the proposed method. Section IV presents the results. Finally, Section V concludes the paper.

II. RELATED WORK

In [6], the authors stated that the databases frequently contain field-values and records that refer to the same entity but are not identical. In [18], the authors proposed how to extract symbolic knowledge from the web. In [19], the problem of text mining has been proposed to get the useful information from unstructured database based on the combination of Information Extraction (IE) and traditional Knowledge Discovery from Database (KDD). The problem of identifying duplicate records has been proposed in [4], [5], where it was referred as record linkage. The sorted neighborhood method has been proposed in [8] to solve the merge/purge problem. The merge/purge problem states that the task of merging data from different sources with an efficient manner while maximizing the accuracy of the obtained results.

The problem of finding the one of the best "hard" model from a set of soft facts has been proposed in [9]. Hardening defines the co-reference relationship among the references that are derived from the soft databases. A soft database may be formed by extracting required details from classified advertisements, or some newsgroup postings. Moreover, it may also create by merging the contents from various "hard" databases. Reference matching in [3] and entity name clustering and matching [4]. Moreover, vector-space similarity problem has been addressed [16] to identify whether two values or entities are alike enough to be duplicates. More recently, the use of paring functions that combines multiple standard metrics has been proposed [17].

Most traditional methods for calculating name similarities can be separated into two groups, namely, character-based techniques and vector-space based techniques. The character-based technique rely on character edit operations (i.e., substitutions, insertions, deletions, subsequence comparison, and etc.), while the vector-space based technique transform strings into vector representation based on similarity

computations are conducted. Levensthein metric is defined to transform from one name to another name as per minimum number of insertions, deletions, and etc, which is the best-known character-based string similarity metric. A general dynamic programing method for computing edit distance has been proposed in [7], which is an extension of Levensthein metric in order to create sequences of mismatched characters in the position of two strings. Most commonly the gap penalty is calculated using affine model as described in [10].

III. PROPOSED METHOD

In this section, we describe our proposed method as follows. Firstly, we provide input dataset which contains names and aliases. Secondly, we compute snippets form the given names and aliases. Form the retrieved snippets; we create patterns which further compute candidate aliases. Thirdly, extract the lexical patterns from all the retrieved candidate aliases, and rank candidate aliases of different lexical patterns. Here, the ranking is based on lexical pattern frequency, word co-occurrences in anchor texts, and page count on the web. Finally, we are able to identify most likely correct aliases for the given names. Figure 1 shows the proposed approach which contains different modules (i.e., Lexical pattern extraction, Ranking candidates, Co-occurrences in anchor text, Hub discounting, Page count association measures, and cluster discovery) described below.

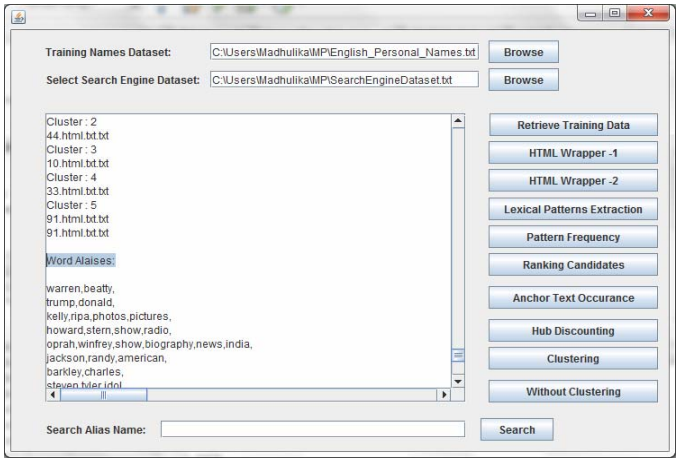

Figure 1: Framework for extraction of word aliases on the web

A. Lexical Pattern Extraction

Nowadays, all the search engines will return a brief text snippet for each query provided by the user in the web page. The return snippets consist of valuable information related to the given query. In order to express aliases of a name, the semantic clues can be used to extract lexical patterns that are provided by snippets.

The lexical pattern extraction method is used to get the information about aliases of names that are expressed on the web in different ways. The following steps describe about the lexical patterns extraction.

- Initially, we extract a list of lexical patterns from the given set of names and aliases.
- At the same time, we get snippets from a web search engine for the given query.
- We extract sequence of words that appear between the name and the alias from each snippet resulted in step 2.
- Finally, lexical patterns are created by replacing the real name and alias in the snippet.

B. Ranking Candidates

We compute candidate aliases from the above lexical patterns. The extracted candidates might consist of invalid aliases, when the web snippets contain noise. We must classify the candidates, which are most probable to be accurate aliases of given name from the extracted candidates. Ranking the candidates by means of given name, such that the candidates which are most probable to be accurate aliases are allocated higher rank. Initially, we describe various ranking scores to measure the relationship among a name and a candidate alias using different methods.

C. Pattern Frequency

To define aliases of a personal name, we described a procedure to extract various lexical patterns. A large number of lexical patterns can be extracted with the recommended pattern extraction procedure. An alias can be considered as good alias for the personal name, if the personal name under consideration and a candidate take place in numerous lexical patterns. As a result, a set of candidate aliases can be ranked in the descendent order of the number of various lexical patterns in which they appear with a name. Document frequency (DF) is widely used in information retrieval which is similar to the lexical pattern frequency of an alias.

D. Co-Occurrence in Anchor Text

Synonym extraction, ranking and classification of web pages are used in the anchor texts. Anchor texts have been considered widely in information retrieval. In addition, anchor texts can also be used to express a citation links even though they contain brief texts. Here, we considered anchor texts to measure the relationship between a name and its aliases on the web. An url pointed by anchor texts gives useful semantic clues linked to the resource denoted by the url. In this case, the term inbound anchor tests is used to denote a set of anchor texts pointing to the same url. If a name p and a candidate alias x appear in two different inbound anchor texts of a url u, then p and x are denoted as co-occurring. From the above statement we can define co-occurrence frequency (CF) as the number of different urls in which they co-occur.

Note that co-occurrences of an alias and a name in the same anchor text are not considered. Among all association measures CF is the simplest one. The CF of a candidate alias x and a name p is denoted by value k. Automatically, it is an indication that x is indeed an accurate alias of the name p, if there are numerous urls which are pointed by anchor texts that hold a candidate alias x and a name p. A high co-occurrence with the name can be reported by a word that has high frequency in anchor texts. To normalize this bias, the term tfidf is widely used in information retrieval.

The weight that is assigned to word that appear across various anchor texts can be reduced by tfidf. Both the name *p* and the candidate alias *x* are independent or dependent is determined based on the Log-likelihood ration (LLR), which is defined as the ratio between the likelihoods of two above alternative hypotheses. Collocation discovery often uses likelihood ratios as they are robust against sparse data and have more intuitive definition. To compute the similarity between words, we used the cosine measure.

E. Hub Discounting

We can observer frequently on the web that the many web pages contain links (i.e., called as hubs) like Facebook, Google, Yahoo, and etc. In some scenarios, for different reasons, two anchor texts might link to the same hub. Hubs are prone to noise because of co-occurrences. Two sets of anchor texts may point to a certain web page. In this case, the real name contains in one set of anchor texts, which we must find, and other candidate aliases can be found in other set of anchor texts. The confidence of the web site as a source increase when the majority of anchor texts pointed to that web site which is having the real name. For co-occurrences in hubs, we can compute simple discounting measure as defined in [11].

$$\alpha(h, p) = \frac{t}{d}$$

Where *t* is the number of inbound anchor texts of hub *h* that contain the real name *p*, and *d* is the total number of inbound anchor texts of hub *h*. The reliability of h as a source of information about *p* increases, if many anchor texts that link to *h* contain *p*. whereas it is likely to be a noisy hub and gets discounted, if *h* has many inbound links.

F. Page Count Association Measures

In this module various ranking scores have been defined using anchor texts. On the other hand anchor texts are not representing all names and aliases equally. Subsequently, we describe word association measures by considering co-occurrences not only in anchor texts but also in the overall web. The page counts can be retrieved from a web search engine, and it can also be considered as an approximation of their co-occurrences in the web. We use the page counts returned by a search engine to compute popular word association measures.

G. Clustering Discovery

The determination of clustering is basically different from that of ranking. We bring the matched services into groups that capture different trade-offs among all the considered request parameters as an alternative of choosing the *k* best matches. The proposed web service clustering framework effectively reduces the various trade-offs related with multiple matching parameters, while excluding irrelevant services. The framework satisfies the requirements that are based on dominance relationships. Further, the following high-level steps are comprised: The match that has a high possibility above a specified threshold will be selected. From the above set, choose representative set, and assign each of the close matches to its closest representative to form clusters. The defined two matches use the distance function between them. The derived set consists of those matches that have a nonzero probability to belong to the cluster. Thus we can select the representatives to be used as the seeds for forming clusters from the most interesting objects with respect to the existing trade-offs.

IV. RESULTS

The proposed paper is implemented in Java technology on a Pentium-IV PC with 20 GB hard-disk and 256 MB RAM. The proposed clustering method shows efficient results and has been efficiently tested with different documents sizes.

The Figure 2 shows the impact of clustering with number of documents as function of computation time (i.e., the time taken to compute the aliases from the given input set). As expected, the number of documents increases with the computation time, when using without clustering. Instead, with clustering, the computation time decrease with the number of documents increases. This is due to eliminating irrelevant information, hence, effectively reduces the various trade-offs related with multiple pattern matching. As a results, performance increase to provide efficient extraction of word alias on the web. Finally, the following index is used to measure the time savings (TS) provided by using clustering with respect to without clustering.

$$TS = \frac{T_1 - T_2}{T_1}$$

Where *T1* and *T2* represents computation time for without and with clusterings, respectively. Table 1 shows the relative time savings TS provided by clustering for different number of documents. As expected, time savings are more and more relevant as the number of documents increases. However, even for less number of documents (e.g., 2) the time saving provided with clustering is more than 25%.

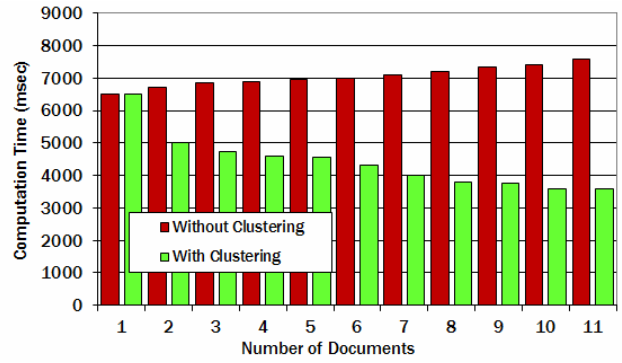

Figure 2: Impact of clustering in terms of computation time

TABLE I. TIME SAVINGS WITH CLUSTERING.

V. CONCLUSIONS

In this paper, we propsoed cluster method to extract a set of candidate aliases from the given name. Thus, reduces computation time to get correct aliases from the given input name. The results shown that the proposed cluster based scheme can provide significant time saving even for the small number of documents (e.g., 2). However, even for ten documents, the proposed method is able to provide above than 50% time savings.

Number of Documents	Time Saving (%)
2	25,37%
4	33,33%
6	38,57%
8	47,22%
10	51,35%

REFERENCES

[1] A.Bagga and B. Baldwin. "Entity-based cross-document coreferencing using the Vector Space Model," In Proceedings of the 17th international conference on Computational linguistics - Volume 1 (COLING '98), Vol. 1, Association for Computational Linguistics, Stroudsburg, PA, USA, 79-85.

[2] http://nlp.uned.es/weps.

[3] D. Gusfield. "Algorithms on strings, trees and sequences," Cambridge University Press, New York, 1997.

[4] B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P.James. "Automatic linkage of vital records," Science, 130:954–959, 1959.

[5] W. E. Winkler. "The state of record linkage and current research problems," Technical report, Statistical Research Division, U.S. Bureau of the Census, Wachington, DC, 1999.

[6] W. W. Cohen, H. Kautz, and D. McAllester. "Hardening soft information sources," In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000), Boston, MA, Aug. 2000.

[7] S. B. Needleman and C. D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequences of two proteins," Journal of Molecular Biology, 48:443–453, 1970.

[8] M. A. Hern´andez and S. J. Stolfo. "The merge/purge problem for large databases," In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD-95), pages 127–138, San Jose, CA, May 1995.

[9] W. W. Cohen, H. Kautz, and D. McAllester. "Hardening soft information sources," In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000), pages 255–259, Boston, MA, Aug. 2000.

[10] Bilenko, M., and Mooney, R. "Learning to combine trained distance metrics for duplicate detection in databases," Technical Report Technical Report AI 02-296, Artificial Intelligence Lab, University of Texas at Austin, 2002.

[11] 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.

[12] R. Guha and A. Garg, "Disambiguating people in search," technical report, Stanford Univ., 2004.

[13] J. Artiles, J. Gonzalo, and F. Verdejo, "A testbed for people searching strategies in the WWW," Proc. SIGIR '05, pp. 569-570, 2005.

[14] 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.

[15] P. Cimano, S. Handschuh, and S. Staab, "Towards the self-annotating web," Proc. Int'l World Wide Web Conf. (WWW '04), 2004.

[16] G. Salton. "Automatic text processing: the transformation, analysis and retrieval of information by computer," Addison-Wesley, 1989.

[17] V. N. Vapnik. "The nature of statistical learning theory," Springer-Verlag, Berlin, 1995.

[18] R. Ghani, R. Jones, D. Mladenic, K. Nigam, and S. Slattery, "Data mining on symbolic knowledge extracted from the web," in Proc. 6th Int. Conf. Knowledge Discovery Data Mining (KDD-2000) Workshop on Text Mining, Boston, MA, Aug. 2000, pp. 29–36.

[19] U. Y. Nahm and R. J. Mooney., "Using information extraction to aid the discovery of prediction rules from texts," In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining, pages 51–58, Boston, MA, Aug. 2000.

[20] 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.

[21] M. Bilenko and R. Mooney, "Adaptive duplicate detection using learnable string similarity measures," Proc. SIGKDD '03, 2003.

AUTHORS PROFILE

Ms.Y Madhulika is working as Assistant Professor in department of Information Technology, Guru Nanak Institute of Technology, Hyderabad, Andhra Pradesh, India. She completed Bachelor of Technology from Vignan's Engineering College, Guntur, Andhra Pradesh, India. She attended various International and National Conferences. Her research interests are Web Mining, Compiler Design, Computer Architectures, Theory of Computation and Programming Languages.

P Dathathreya is having over 21 years of experience in industry and teaching. He received his Bachelor of Engineering in Computer Science Engineering, and Master of Technology in Computer Science and Engineering from Mysore University Campus. He has extensive experience in IT industry working in MNC's as project leader/Manager and ODC in charge for data warehousing and data mining domains. His areas of interests are Project Management, Data Warehousing and Data Mining, SAN (storage area networks). He is currently working as professor in Guru Nanak Technical Campus, Hyderabad, Andhra Pradesh, India.

Mr. N. HanumanthaRao is working as Assistant Professor and Head of the department for Computer Science & Engineering, Visvesvaraya College of Engineering & Technology (VCET), Hyderabad, Andhra Pradesh, India. He completed his Bachelor of Technology from Vignan's Engineering College, Guntur, Andhra Pradesh and Master of Technology from Vellore Institute of Technology (VIT University), Vellore, Tamil Nadu, India. He is the mentor of CIT (Certificate in Information Technology) program organizing by IIIT-Hyderabad, received award in Art of teaching workshop conducted by EnhanceEdu, IIIT-Hyderabad. His research interests are Multi-core architectures, Operating Systems, Computer Networks, Cloud Computing, Data structures, Algorithms, and Data Mining. He is also a member of IEEE.