Entity Recognition by Extracting Properties from Web Documents

P.Pandiyan[1], Prof.G.Ilanchezhiapandian[2]

[1]P.G Scholar, Computer Science and Engineering, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore, Tamilnadu, India

[2]Head of Computer Science and Engineering Department, Ganadipathy Tulsi's Jain Engineering College, Kaniyambadi, Vellore, Tamilnadu, India

Abstract

Entity Recognition is the process of identifying predefined entities such as person names, products, or locations in a given document. This is done by finding all possible substrings from a document that match any reference in the given entity dictionary. Approximate Membership Extraction (AME) method was used for finding all sub-strings in a given document that can approximately match any clean references but it generates many redundant matched sub-strings because of approximation, thus rendering AME is not suitable for real-world tasks based on entity extraction. I propose a web-based join framework which combines a web search along with the approximate membership extraction. Our process first provides a top n number of documents fetched from the web using a general search using the given query and then approximate membership extraction is applied on these documents using the clear reference table and extracts the entities from the document to form the intermediate reference table.

Keywords: *Web-based join, Approximate Membership Extraction, Reference table, Approximate Membership Localization.*

1. Introduction

Given a document, the task of Entity Recognition is to identify predefined entities such as person names, products, or locations in the given document. With a large dictionary, this entity recognition problem transforms into a Dictionary-based Membership Checking problem, which aims at finding all possible sub-strings from a document that match any reference in the given dictionary. With the growing amount of documents and the deterioration of documents quality on the web, the membership checking problem is not trivial given the large size of the dictionary and the noisy nature of documents, where the references can be approximate and there may be mentions of non-relevant references. The approximation is usually constrained by a similarity function (such as edit distance, jaccard, cosine similarity, etc.) and a threshold within [0, 1], such that slight mismatches are allowed between the substring and its corresponding dictionary reference.

For instance, given a list of conference names like "ACM SIGMOD Conference", "VLDB Conference", "IEEE ICDE Conference" as shown on the left part of Fig. 1.1, the task is to find matches from the text on the right, such as "VLDB 2010 Conference" and "ICDE Conference" although they do not match the string "VLDB Conference" and "IEEE ICDE Conference" in the dictionary exactly. The dictionary-based approximate membership checking process is now expressed by the Approximate Membership Extraction (AME), finding all sub-strings in a given document that can approximately match any clean references.

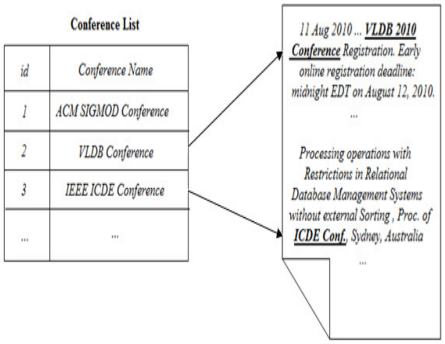
Conference List
id
Conference Name
1
ACM SIGMOD Conference
2
VLDB Conference
3
IEEE ICDE Conference
...
...
11 Aug 2010 ... VLDB 2010 Conference Registration. Early online registration deadline: midnight EDT on August 12, 2010.
...
Processing operations with Restrictions in Relational Database Management Systems without external Sorting, Proc. of ICDE Conf, Sydney, Australia
...

Fig. 1 Example of the approximate membership checking

In this project, i proposed a new type of membership checking problem: Approximate Membership Localization (AML). AML only aims at locating true mentions of clean references. AML targets at locating non-overlapped sub-strings in a given document that can approximately match any clean reference. I also increase the performance of the AML process making it as a Web-Based Join Framework.

2. System Description

I propose the project as a web-based join framework which uses a list of elements T with an attribute T:X and a clean reference list R with an attribute R:A, the problem is to create an intermediary table RT from the web containing valued correlations between two attributes T:X and R:A in order to perform a join between T and R.

Given that the information available on the web can be noisy, RT shall contain the likelihood associated with its contents. Based on the hypothesis that there exist web documents containing elements of T:X that also contain the elements of R:A, I use the elements of T:X as a query for a search engine to retrieve the ranked list of documents.

First, given that the mention attribute in web pages might not be exact, I need to locate clean references approximately mentioned in given documents, for this AML method is presented. Second, besides given attribute, some other attributes might also occur in the web pages. To link given attribute with them, i use score correlation. I use Edit Distance Vector which is the simple function for finding the similarity when compare with cosine similarity. Edit Distance Vector reduce the extra complexity in computation because it is secondary step of extraction there is no need of high evaluation.

Finally, scoring correlation is the approach which is applied. It provides a score that can be used by setting a threshold (either for the value or for the number of best matched clean references) to perform the join.

3. Methodology

3.1 Approximate Membership Localization

AML divides documents with non-overlapping substring with at most one best match substring they are called domains. These domains are sub divided into segments (string without divided symbols between them) and intervals. Potential Redundancy Prune is applied on each domain to extract best matched candidate. Then i apply Edit Distance Vector a similarity function on the redundancy pair to extract the final best match substring.

3.2 P-Prune Algorithm

In the algorithm of P-Prune (potential redundancy prune) given a reference list R, an inverted index is built over the words of all references in R and the strong words of each reference are also selected. It is also called the Decrease and Conquer algorithm, first it prune the redundant and/or unrelated information. Then, it computes on the unique information.

For a coming document M, i generate window domains from it. Each domain is represented by the starting and ending position of the domain in M. All domains are sorted according to their starting positions.

3.3 Edit Distance Vector

Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other string. Edit distance found in applications like natural language spelling processing, where spelling correction can be done automatically to determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low distance to the word in question.

Example

The Levenshtein distance between "kitten" and "sitting" is 3. The minimal edit/modify script that transforms the former into the latter is:

1. **k**itten → **s**itten (substitution of "s" for "k")
2. sitt**e**n → sitt**i**n (substitution of "i" for "e")
3. sittin → sittin**g** (insertion of "g" at the end).

3.4 Scoring Correlation

Scoring correlation is the approach which is applied finally. It provides a score that can be used by setting a threshold (either for the value or for the number of best matched clean references) to perform the join. Scoring correlation for each document depends upon three relevant parameters frequency, distance and document importance.

The document importance main depends upon the page ranking assigned to document during the web search which is not focused in the paper. I use an inverse document based page ranking to assign rank to the documents which enhance the process of scoring correlation. The following equation is used to find the score correlation.

$$P(r,T.x) = \frac{\sum_{d \in Docs} imp(d).score(r,d)}{\sum_{d \in Docs} imp(d)} \qquad (1)$$

where, r is clean reference in R.A

$$imp(d) = \frac{\log(2)}{\log\left(1 + \frac{\left[rank(d)\right]}{B}\right)} \qquad (2)$$

N documents of Doc should be partitioned into $B(1 \leq B \leq N)$

4. System Design

4.1 Architecture Diagram

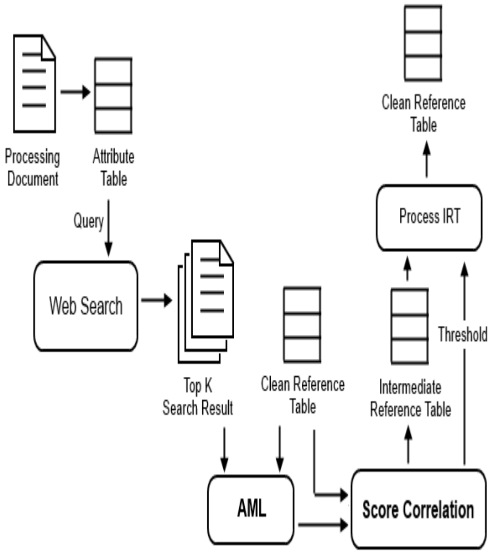

Fig. 2 Architecture Diagram

4.2 Web Document Search

This module is used to extract the documents from the web using a content based search processed using the given document attributes as query input. Content based document retrieval is process of retrieving a document by search for the given keyword within each document in the document set. I select top K resultant documents from the search results for our next level of processing. Each document contains certain rank value assigned to it based on the content matching the given query attributes.

4.3 Approximate Member Localization (AML)

This module is helps to remove the AML problem from our process which is to extract the clean reference approximately. This module uses an optimized P-Prune algorithm, which can prune potential redundant sub-strings from documents before generating clean reference approximately. This algorithm shows a much higher efficiency than the AME-based algorithm. After the pruning process I used a similarity function for extracting the approximate clean reference from the documents.

4.4 Scoring Correlation

Scoring correlation is the approach which is applied finally. It provides a score that can be used by setting a threshold (either for the value or for the number of best matched clean references) to perform the join. Scoring correlations for each document depends upon three relevant parameters frequency, distance and document importance. The document importance mainly depends upon the page ranking assigned to document during the web search which is not focused in the existing system. I use an inverse document based page ranking to assign rank to the documents which enhance the process of scoring correlation.

5. Conclusion

In this paper, I formalized the AML problem and proposed to solve it with an efficient P-Prune algorithm. P-Prune is proved to be several times faster, sometimes even hundreds of times faster, than simply adapting formerly existing AME methods. To find out the improvement of AML over AME, we apply both approaches within our proposed web-based join framework, which is a real-world application that greatly relies on the results of membership checking. The results prove that the precision and recall of web-based join with the AML results can be as good as AME. We also apply the web-based join framework in joining publication titles with venue names from the ERA conference and list of journals, thus demonstrating that our method can reach a higher precision and recall than the search-based and textual-based similarity metrics that use a unique join attribute.

References

[1] S. Agrawal, K. Chakrabarti, S. Chaudhuri, V. Ganti, A. Konig, andD. Xin, “Exploiting Web Search Engines to Search Structured Databases,” Proc. 18th WWW Int’l Conf. World Wide Web, pp. 501-510, 2009.

[2] Z. Li, L. Sitbon, L. Wang, X. Zhou, and X. Du, “Approximate Membership Localization (AML) for Web-Based Join,” Proc. 19th CIKM Int’l Conf. Information and Knowledge Management, 2010.

[3] A. Arasu, V. Ganti, and R. Kaushik, “Efficient Exact Set-Similarity Joins,” Proc. 32nd VLDB Int’l Conf. Very Large Data Bases, pp. 918-929, 2006.

[4] R. Bayardo, Y. Ma, and R. Srikant, "Scaling Up All Pairs Similarity Search," Proc. 16th WWW Int'l Conf. World Wide Web, pp. 131-140, 2007.

[5] B. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," Comm. ACM, vol. 13, no. 7, pp. 422-426, 1970.

[6] B. Bocek, E. Hunt, and B. Stiller, "Fast Similarity Search in Large Dictionaries," Technical Report ifi-2007.02, Dept. of Informatics Univ. of Zurich, 2007.

[7] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, 2001.

[8] G. Brodal and L. Gasieniec, "Approximate Dictionary Queries,"Proc. Seventh Symp. Combinatorial Pattern Matching, vol. 1075, pp. 65-74, 1996.

[9] K. Chakrabarti, S. Chaudhuri, V. Ganti, and D. Xin, "An Efficient Filter for Approximate Membership Checking," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 805-818, 2008.

[10] H. Chan, T. Lam, W. Sung, S. Tam, and S. Wong, "A Linear Size Index for Approximate Pattern Matching," Proc. 17th Ann. Symp. Combinatorial Pattern Matching, pp. 49-59, 2006.