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.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Entity Recognition in a Web Based Join Structure

J.Kavitha M.Tech[1], A.Pasca Mary[2]

ABSTRACT: Given a document, the task of Entity Recognition is to identify predefined entities such as person names, products, or locations in this document. With a potentially large dictionary, this entity recognition problem transforms into a Dictionary-based Membership Checking problem called Approximate Membership Extraction (AME) which aims at finding all possible substrings from a document that match any reference in the given dictionary. It generates many redundant matched substrings, thus rendering AME unsuitable for real-world tasks based on entity extraction. Approximate Membership Localization (AML) only aims at locating true mentions of clean references. An important observation is as follows: in real world situations, one word position within a document generally belongs to only one reference-matched substring, meaning that the true matched substrings should not overlap. Therefore, AML targets at locating non-overlapped substrings in a given document that can approximately match any clean reference. In the event where several substrings overlap, only the one with the highest similarity to a clean reference qualifies as a result. Web-based join Structure which is a search-based approach joining two tables using entity recognition from web documents and it is a typical real-world application greatly relying on membership checking. Membership checking is performed by using correlation, Inverse Document Frequency (IDF), Jaccard Similarity, P-Pruning Technique.

KEYWORDS —Web-based join, approximate membership location, AML

I. INTRODUCTION

Named entity recognition (NER) aims at finding named entities in unstructured text. It is an important task in information extraction and integration, and serves many applications, including identifying geographical locations for geo tagging, identifying gene and protein names from MEDLINE abstracts for text mining, identifying names and their categories to improve Web search. Given a document, the task of Entity Recognition is to identify predefined entities such as person names, products, or locations in this document. With a potentially large dictionary, this entity recognition problem transforms into a Dictionary-based Membership Checking problem, which aims at finding all possible substrings 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 mention of 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.

The dictionary-based approximate membership checking process is now expressed by the Approximate Membership Extraction (AME), finding all substrings in a given document that can approximately match any clean references. The objective of AME guarantees a full coverage of all the true matched substrings within the document, where the true matched substring is a true mention of the clean reference semantically. On the other hand, it generates many redundant matched substrings, thus rendering AME unsuitable for real-world tasks based on entity extraction.

The new type of membership checking problem is the Approximate Membership Localization (AML) which only aims at locating true mentions of clean references. An important observation is as follows: in real world situations, one word position within a document generally belongs to only one reference-matched substring, meaning that the true matched substrings should not overlap. Therefore, AML targets at locating non-overlapped substrings in a given document that can approximately match any clean reference. In the event where several substrings overlap, only the one with the highest similarity to a clean reference qualifies as a result.

To inspect the improvements of AML over AME, I apply both approaches into a proposed web-based join structure, which is a typical real-world application greatly relying on membership checking.

II. EXISTING APPROACH

Approximate Membership Extraction (AME), finding all substrings in a given document that can approximately match any clean references. The objective of AME guarantees a full coverage of all the true matched substrings within the document, where the true matched substring is a true mention of the clean reference semantically. On the other hand, it generates many redundant matched substrings, thus rendering AME unsuitable for real-world tasks based on entity extraction. Indeed, redundant pairs are qualified to be part of AME results, but are unlikely to be true matches in real-world situations.

Problem Statement

Given a dictionary R of strings and a similarity threshold $\delta \in [0,1]$, then a query M is submitted. Here M represents a relatively long string (e.g. a text file). The task of AME is to extract all M's substrings m, such that there exists some $r \in R$ satisfying $Sim(m,r) \geq \delta$.

The system solves this problem in two steps.

- In the first step, for each substring in the text, it filters away the strings in the dictionary that are very different from the substring.
- In the second step, each candidate string is verified to decide whether the substring should be extracted. It developed an incremental algorithm using signature-based inverted lists to minimize the duplicate list-scan operations of overlapping windows in the text. The experimental study of the proposed algorithms on real and synthetic datasets showed that this solution significantly outperform existing methods in the literature.

The Filtration-Verification Framework

To efficiently solve AME and other related problems, have to design methods following two phases of filtration and verification. In the AME problem, employing this framework usually requires building an indexing structure for the dictionary R. For each approximate member m extracted, it define the string r in R that is similar enough with m to be m's evidence. Thus, the task of extracting all approximate members from M can be simply reduced to determining whether there exists any evidence for each substring of M, and filtration-verification is actually referred to as evidence filtration and evidence verification.

Generally, the foundation of filtration is based on some necessary condition (denoted as NC) of our matching criterion $Sim \geq \delta$, that is, if some candidate evidence is real evidence, it must satisfy NC. With the dictionary R given offline, we build an index that quickly recommends for a query m ALL potential evidence that meets NC, so that true evidence is never missed. Then the evidence is verified against the actual matching criterion to determine whether the string m is a true approximate member.

NC plays a key role in our whole framework. It ensures the correctness of the whole algorithm. Moreover, it determines how balanced this framework is and can evaluate it through the following two categories

- powerful that is, does it eliminate as much false evidence as possible
- easy-going that is, can build a quick index to test it at low cost

Although syntactic similarity is an important indicator of relationship between strings, it is not effective when representations of the same real-world entity are syntactically far apart from each other this textual similarity may put AME at risk of mistakenly discarding some true matched substrings.

- The matched substrings with many redundancies cause a low efficiency of the AME process and deteriorate the performance

- The problem is to locating non-overlapped references approximately mentioned in a given document.
- The matched pair of results are not much closer to the true matched pairs.
- AME and AML are not applied in the membership checking sub-module.
- Textual similarity may put AME at risk of mistakenly discarding some true matched substrings

III. PROPOSED APPROACH

Approximate Membership Localization (AML) only aims at locating true mentions of clean references. An important observation is as follows: in real world situations, one word position within a document generally belongs to only one reference-matched substring, meaning. Web-based join Structure which is a search-based approach joining two tables using entity recognition from web documents and it is a typical real-world application greatly relying on membership checking.The techniques used in the proposed system are

- Correlation
- Inverse document Frequency
- Jaccard similarity
- Pruning Technique

At the initial stage Multi-pattern Matching is performed, which aims at finding all occurrences of patterns from a given set within a document. The technique for solving the Multi-pattern matching problem is to build a tree over all pattern. This technique reduces the number of comparisons between substrings and pattern.

Based on the selected pattern approximate string matching is performed by using alpha-beta pruning and P-pruning.

Basic prefix signature scheme is used to find the prefix Signature set or strong words of each entity. For example string s="ieee transaction on knowledge and data Engineering" from here the prefix signature set of s is {Engineering, Knowledge}.

Available M documents of keyword are divided into several subdocuments or domain and then candidate match generation is performed at each subdocument by applying similarity function. Each subdocument should have one best match substring in case where several substring overlaps the one with the highest similarity will qualifies as result

In the result of pruning correlation scoring is performed. The parameters used to score correlation are as follows

- Frequency

  Frequency is the number of times each reference is mentioned in the document Docs.
- Distance

  Distance is the distance between the mention of each clean reference and the position
- Document importance

  Documents retrieved on the web are of different importance w.r.t. their relevance to the query, (i.e) their ranks in a web search engine result.

$$\text{Imp(d)} = \frac{\log(2)}{\log(1+\frac{(rank(d))}{B})}$$

$$\text{Score(r,d)} = w_a \frac{freq}{N} + (1 - w_a).\sum_{1 \le i \le freq} \frac{|d| - dist_i}{freq.|d|}$$

Where |d| is the length of document d, w_a is the weight given to frequency of reference

Strings are the set of words. For any word w, use wt(w) to denote its Inverse Document Frequency(IDF)weight. For example, Given a string s=w1,w2…wk, the weight of s is the sum of weights of all its constituents and can be denoted as wt(s)=∑1≤i≤k(wt(wi))

The weighed Jaccard similarity of two strings s1 and s2 can be calculated as WJS(s1,s2)=$wt(s1 \cap s2) \div wt(s1 \cup s2)$

At the final stage Boundary pruning and weight pruning are performed. Therefore only one best match substring is qualified as result.

- Advancement of this system is to get the matched result from multiple tables is clearly possible.
- The matched pair results of the AML are much closer to the true matched pairs than AME results.
- In addition with similarity correlation score is used to generate Best Match Result.

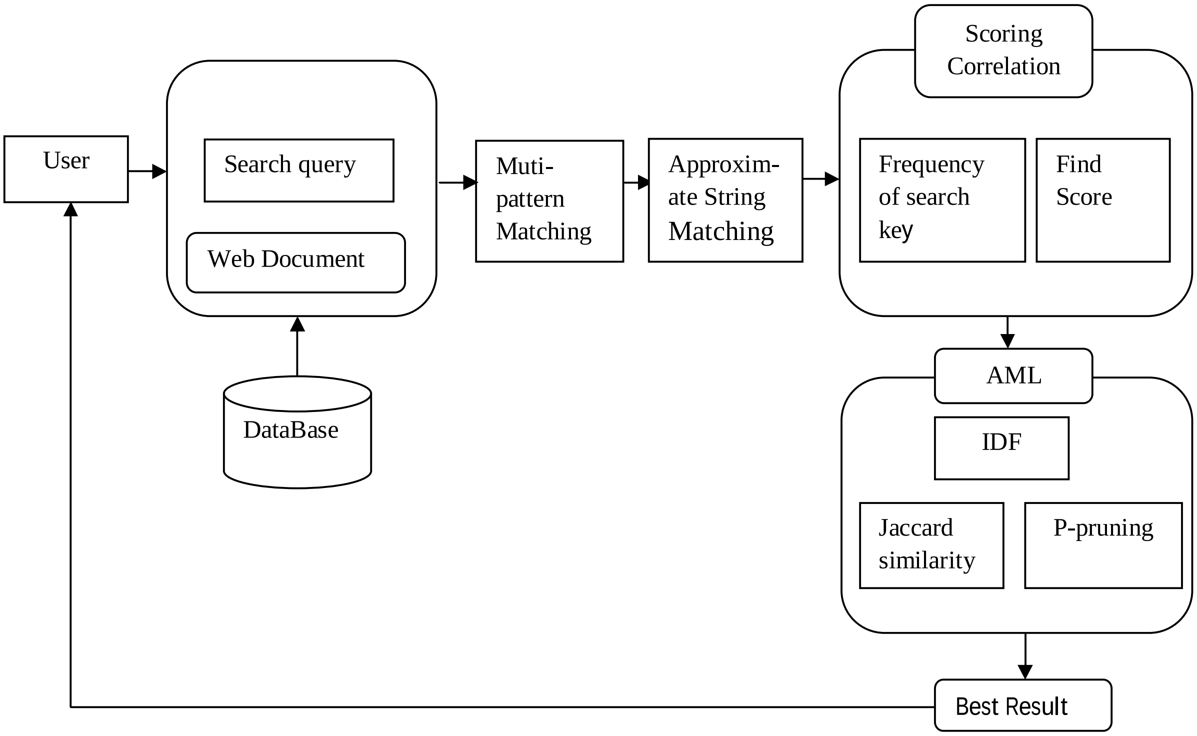

Fig. System Architecture Design.

a. Multi-Pattern Matching

At the initial stage Multi-pattern Matching is performed, which aims at finding all occurrences of patterns from a given set within a document. The technique for solving the Multi-pattern matching problem is to build a tree over all patterns. This technique can significantly reduces the number of comparisons between substrings and pattern.

b. Approximate String Matching

Based on the selected pattern approximate string matching is performed by using alpha-beta pruning and P-pruning.

Basic prefix signature scheme is used to find the prefix Signature set or strong words of each entity. For example string s="ieee transaction on knowledge and data Engineering" from here the prefix signature set of s is {Engineering, Knowledge}.

Available M documents of keyword are divided into several subdocuments or domain and then candidate match generation is performed at each subdocument by applying similarity function. Each subdocument should have one best match substring in case where several substring overlaps the one with the highest similarity will qualifies as result

c. Scoring Correlation

In the result of pruning correlation scoring is performed. The parameters used to score correlation are as follows

- Frequency freq:
  Frequency is the number of times each reference is mentioned in the document Docs.
- Distance dist:
  Distance is the distance between the mention of each clean reference and the position
- Document importance imp(d):
  Documents retrieved on the web are of different importance w.r.t. their relevance to the query, (i.e) their ranks in a web search engine result.

$$\mathrm{Imp}(d) = \frac{\log(2)}{\log(1+\frac{(rank(d))}{B})}$$

$$\mathrm{Score}(r,d) = w_a \frac{freq}{N} + (1-w_a).\sum_{1\le i\le freq} \frac{|d|-dist_i}{freq.|d|}$$

Where |d| is the length of document d, w_a is the weight given to frequency of reference

d. AML Results

Strings are the set of words. For any word w, use wt(w) to denote its Inverse Document Frequency(IDF)weight. For example, Given a string s=w1,w2…wk, the weight of s is the sum of weights of all its constituents and can be denoted as wt(s)=$\sum_{1\le i\le k}(wt(w_i))$

The weighed Jaccard similarity of two strings s1 and s2 can be calculated as

$$\mathrm{WJS}(s1,s2)=\frac{wt(s1\cap s2)}{wt(s1\cup s2)}$$

At the final stage Boundary pruning and weight pruning are performed and only one best match substring is qualified as result in case several substring overlaps the one with the highest similarity will qualifies as result.

V. CONCLUSION

Approximate Membership Localization (AML) only aims at locating true mentions of clean references. According to experimental results on several real-world data sets, P-Prune is proved to be several times faster, sometimes even tens or hundreds of times faster, than simply adapting formerly existing AME methods.Web-based join Structure is developed which is a search-based approach joining two tables using entity recognition from web documents and it is a typical real-world application greatly relying on membership checking. The results prove that the precision and recall of web-based join with the AML results can be as good as 0.873 and 0.831, respectively, largely outperforming AME (where results are 0.5 and 0.8, respectively).

The web-based join framework in joining publication titles with venue names from the conference and journal list, thus demonstrating that our method can reach a higher precision and recall than the previous search-based one proposed in textual-based similarity metrics that use a unique join attribute.

Future work will apply the P-Prune algorithm for AML to some other scenarios, such as returning the top-k most popular singers or movies among blogs or newswires by counting the number of time a given entity is approximately mentioned.This AML-targeted solutions are more appropriate than the AME-targeted solutions for this type of real-world applications, since the matched pair results of the AML are much closer to the true matched pairs than AME results.

REFERENCES

1. S. Agrawal, K. Chakrabarti, S. Chaudhuri, V. Ganti, A. Konig, and D. Xin, "Exploiting Web Search Engines to Search Structured Databases," Proc. 18th WWW Int'l Conf. World Wide Web, pp. 501- 510, 2009.
2. Aho and M. Corasick, "Efficient String Matching: an Aid to Bibliographic Search," Comm. ACM, vol. 18, no. 6, pp. 333-340, 1975.
3. 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. Borthwick, "A Maximum Entropy Approach to Named Entity Recognition," PhD thesis, New York Univ., 1999.
5. G. Brodal and L. Gasieniec, "Approximate Dictionary Queries," Proc. Seventh Symp. Combinatorial Pattern Matching, vol. 1075, pp. 65-74, 1996.
6. 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. 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.
8. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," Comm. ACM, vol. 13, no. 7, pp. 422-426, 1970.
9. W. Cohen, "Integration of Heterogeneous Databases Without Common Domains Using Queries Based on Textual Similarity," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 201-212, 1998.
10. 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.
11. S. Chaudhuri, V. Ganti, and R. Kaushik, "A Primitive Operator for Similarity Joins in Data Cleaning," Proc. 22nd Int'l Conf. Data Eng., p. 5, 2006.
12. S. Chaudhuri, V. Ganti, and D. Xin, "Exploiting Web Search to Generate Synonyms for Entities," Proc. 18th Int'l Conf. World Wide Web (WWW), pp. 151-160, 2009.
13. 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.
14. Chandel, P. Nagesh, and S. Sarawagi, "Efficient Batch Top-K Search for Dictionary-Based Entity Recognition," Proc. 22nd Int'l Conf. Data Eng., p. 28, 2006.
15. H. Chieu and H. Ng, "Named Entity Recognition: A Maximum Entropy Approach Using Global Information," Proc. 19th Int'l Conf. Computational Linguistics, p. 7, 2002.
16. W. Cohen, P. Ravikumar, and S. Fienberg, "A Comparison of String Distance Metrics for Name-Matching Tasks," Proc. IJCAI '03 Workshop Information Integration on the Web (IIWeb '03), pp. 9-10, 2003.
17. W. Cohen and S. Sarawagi, "Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods," Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 89-98, 2004.
18. Dagan, S. Marcus, and S. Markovitch, "Contextual word Similarity and Estimation from Sparse Data," Proc. 31st Ann. Meeting on Assoc. for Computational Linguistics, pp. 164-171, 1993.
19. Elmagarmid, P. Ipeirotis, and V. Verykios, "Duplicate Record Detection: A Survey," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 1, pp. 1-16, Jan. 2007.
20. L. Getoor and C. Diehl, "Link Mining: A Survey," ACM SIGKDD Explorations Newsletter, vol. 7, no. 2, pp. 3-12, 2005.