ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Avoid Redundant Matched Substring Using AML Prunes Within Web-Based Framework

[I]Vaishnavi.N, [II]Arockia Jesuraj.Y
[I]PG Student, [II]Associate Professor
[I,II]Dept. of Computer and Communication Engg., Sethu Institute of Tech., Madurai, India

Abstract
Approximate Membership Localization algorithm is a dictionary based entity recognition. AML only aims at locating true mentions of clean references. 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. AML targets at locating non overlapped substrings in a given document that can approximately match any clean reference. The results of AML are expected to be much closer to the true matched pairs but not involving overlapped redundant substrings. Web-based framework is a search based approach joining two tables using dictionary based entity recognition from web document. P-Prune technique is used to reduce redundancy, and shows a much high efficiency than the AME.

Keywords
Approximate Membership Localization (AML), Pruning Technique, Web-based framework

I. Introduction

For a given document Entity Recognition can be identified using predefined entities like person names, products or locations in the specified documents. With an impending large dictionary, the entity recognition is transformed into a Dictionary-based Membership Checking problem. This problem 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 approximation is usually constraint by a similarity function and a threshold within [0, 1] such that slight mismatches are allowed between the substring and its corresponding dictionary reference. For example given a list of engineering department names like "computer science engineering", "civil engineering", "electrical engineering" that task is to find matches from the text, such as "computer and communication engineering" and "civil and structural engineering" although they do not match the string "computer science engineering" and "civil engineering" in the dictionary exactly.

The dictionary-based approximate membership checking process is now expressed by the Approximate Membership Extraction (AME). It will find all substrings in a given document that can be approximately match any clean references. The aim of AME guarantees a full coverage of all the true matched substrings within the document. It also generates many redundant matched substrings, thus rendering AME is unsuitable for real world task based on entity extraction.

This paper proposes a new type of membership checking problem: Approximate Membership Localization (AML). AML only concentrates at locating true mentions of clean references. AML targets at locating non-overlapped substrings in a given document that can approximately match any clean reference. The results of AML are expected to be much closer to the true matched pairs by not involving overlapped redundant substrings.

A. Potential Redundancy Prune

This method avoids redundancy. The AME-based method for AML uses time resources for generating and identifying unqualified redundant matches. In order to efficiently solve AML, this paper put forward an optimized algorithm P-Prune, which can prune potential redundant substrings before generating them. This technique shows a much higher efficiency than the AME-based technique.

1. General Idea of P-Prune

For an input document M, AML only requires best match substrings. Assuming first divide M into subdocuments, where each subdocument is a consecutive substring of M and subdocuments may overlap with each other such that all best match substrings of M are located within these subdocuments, the problem of finding all best match substrings from M becomes a problem of finding all best match substrings from M becomes a problem of finding the best match substrings from these subdocuments of M. If there is atmost one best match substrings in a subdocument it becomes faster to judge whether there is a best match substring in each subdocument. This kind of subdocument is defined as a domain. A domain D is a subdocument of M where there is atmost one best match substring in D.

II. Literature Review

Sanders et al., Suggested [17] An algorithm to solve the approximate dictionary matching problem. Given a list of words maximum distance d fixed at preprocessing time and a query word q, we would like to retrieve all words from w that can be transformed into q with d or less edit operations. The present data structure that support fault tolerant by generating an index. The most trivial algorithm to solve the problem is scanning sequentially through the input list and noting the best matches at each query.

Ganti et al., Suggested [12] Each structured database is searched individually and the relevant structured data items are returned to the web search engine. The search engine gathers the structured search results and display them along
side the web search results. Typically, these structured databases contain information about named entities like products, people, movies and locations. The results from the structured database search are therefore independent of the results from web search.

Xin et al., Suggested [14] Tasks recognizing named entities such as products, people name from documents have recently received significant attention in the literature. Many solutions to these tasks assume the existence of reference entity extraction tables. Then

develop efficient techniques to facilitate approximate matching in the context of our similarity functions.

Kaushik et al., Suggested [8] Given two input collections of sets, a set-similarity join (SSJoin) identifies all pairs of sets, one from each collection, that have high similarity. Recent work has identified SSJoin as a useful primitive operator in data cleaning. A large number of different similarity functions such as edit distance, jaccard similarity, and generalized edit distance have been traditionally used in similarity joins. It is well-known that no single similarity function is universally applicable.

III. Existing System

In existing system, 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 un-suitable 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. Lower efficiency of the entity extraction process. Accuracy of the matched pair extraction is low. This technique cause many redundancies.

IV. Proposed System

An AML problem target at locating non overlapped substrings which is a better approximation to the true matched substrings without generating overlapped redundancies. In order to perform AML efficiently, propose the optimized algorithm P-Prune that prunes a large part of overlapped redundant matched substrings before generating them. Our study using several real-world datasets demonstrates the efficiency of P-Prune over a baseline method.

A. Document Retrival

The web-based framework requires to retrieve web documents. Intuitively a web document is the webpage returned by a web search engine. Extract of the original webpage generally presented in the ranking list returned by search engine. For example, manually extracted about 60k publication records from more than 500 researcher's home pages. Each record contains atleast the paper title and the paper domain from the ERA list. The document is retrieved from the domain. To generate domains from M, all reference entities have to be considered. For this, here influence the basic prefix signature scheme. Words in the prefix signature set of r as r's strong words.

B. Scoring Calculation

This approach provides a score that can be used by setting a threshold to perform join. To determine the number of times and the locations where a clean reference is mentioned in documents [14]. Given that these references may be approximately mentioned in the documents, we need to find non overlapped substrings that can approximately match any clean reference in documents. For a given value T. x of T.X, the three relevant parameters of the evaluation of correlations are for each document.

- Frequency freq: The number of times each reference is mentioned in each document of Docs.
- Distance dist: The distance between the mention of each clean reference and position of T. x.
- Document importance imp(d): documents retrieved on the web are of different importance with respect to their relevance to the query, i.e, their ranks in a web search engine results.

C. Similarity Function

Approximate membership localization is to find all match substrings for each reference. Similarity calculation gives the similarity value between the strings. There is no overlap between the input values means it does not give the similarity value. Given an input document M and an entity reference list R, the task of approximate membership localization is to find all match substrings m in M for each reference r in R, such that:

1. sim (m, r) $\geq \delta$, where sim( ) is a given similarity function, and δ is a given threshold between [0, 1].
2. There is no substring m1 that overlaps with m in any position of M which satisfies sim(m1, r1) > sim(m, r), where r1 is also reference from R.

D. AML

In AML, the similarity functions first do the inverse document frequency. Then find the Jaccard similarity between the document strings. Apply the boundary constraint to prevent the generation of boundary redundancies and apply the non overlapped constraint to remove all the overlapped redundancies. AML problem based on two assumptions:

1. Assumption 1: Any approximate mention m that matched with a reference consist of consecutive words in a document i.e each m is a substring.
2. Assumption 2: Only substrings whose length is upto a length threshold L are of interest, so it require $|m| \leq L$.

For the calculation of similarity functions, it observe strings as sets of words. For any word w, wt(w) is used to denote its Inverse Document Frequency (IDF) weight

E. Pruning Technique

Optimized technique P-Prune, which can prune potential redundant substrings before generating them. This algorithm shows a much higher efficiency than the AME based algorithm. Pruning technique is used to contain a best match substring, an how to minimize the size of the remaining domains. The domain D is divided into several consecutive partitions: segments or intervals. In a domain D of r, the consecutive words which are all present in r compose a segment in D; the consecutive words which are all absent from r compose an interval in D. The segment that contains the strong word of D is the strong segment in D.

F. Performance Evaluation

In order to exhibit the appeal for AML versus AME in search based method, it compare the join precision and recall of the search-based method with using AME results or AML results as the locations of the reference in the retrieved documents, respectively.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

1. Redundancy Calculate

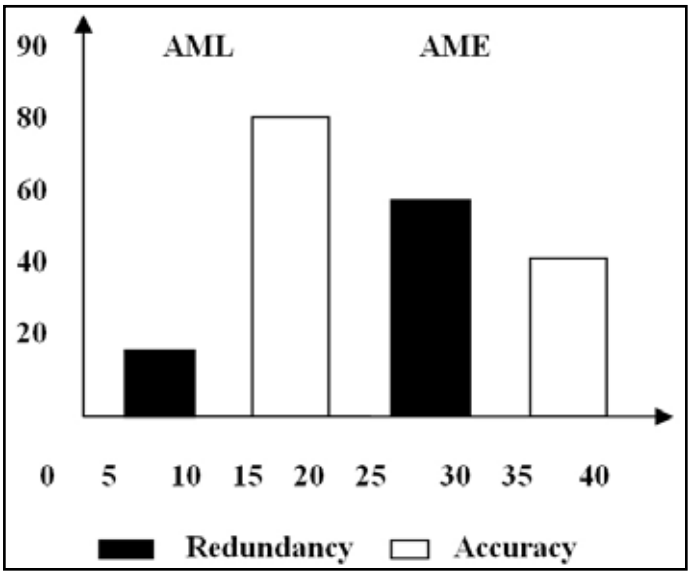

Fig. 1: Redundancy and accuracy comparison between AME and AML

Identifying and removing redundancies by using pruning technique, the matched pair results of the AML are much closer to the true matched pairs than AME results. In the above figure it shows that for AME the redundancy is high but accuracy is low. For AML, the redundancy is low but accuracy is high. So here AML achieved its aim.

2. Result Evaluation

The results of AML are expected to be much closer to the true matched pairs but not involving overlapped redundant substrings. AML targets at locating non-overlapped substrings in a given document that can approximately match any clean reference.

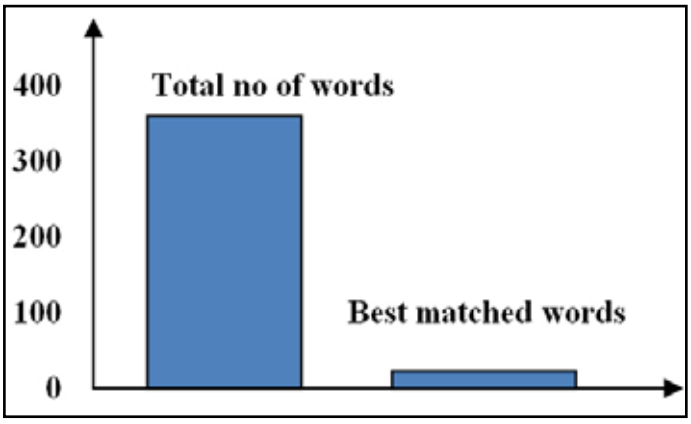

Fig 2: Selecting the best matched words

V. Conclusion

An efficient P-Prune algorithm is used to avoid redundancy. Prune is proved to be several times faster, sometimes even tens or hundreds of times faster, than simply adapting formerly existing AME methods. To inspect the improvement of AML over AME, we apply both approaches within our proposed web-based framework, which is a typical real-world application that greatly relies on the results of member checking.

VI. Future Enhancement

In Future enhancement, Genetic algorithm (GA) is routinely used to generate useful solutions to optimization and search problems. Genetic algorithm belong to larger class of evolutionary (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and cross over and also Music Recognition-singers, movies list datasets are used.

References

[1] H.Chieu, Hang,"Named Entity Recognition: A Maximum Entropy Approach Using Global Information," 2002.
[2] G.Zhou and J.Su, "Named Entity Recognition Using an HMM-based Chunk Tagger," 2002.
[3] W.Cohen, P.Ravikumar, and S.Fienberg, "A Comparison of String Distance Metrics for Name-Matching Tasks." 2003.
[4] W.Cohen and S.Sarawagi, "Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods," 2004.
[5] A.Arasu, V.Ganti, and R.Kaushik, "Efficient Exact Set-Similarity Joins," 2006.
[6] H.Chan, T.Lam, W.Sung, S.Tam, and S.Wong, "A Linear Size Index for Approximate Pattern Matching," 2006.
[7] A.Chandel, P.Nagesh, and S.Sarawagi, "Efficient Batch Top-K Search for Dictionary-Based Entity Recognition," 2006.
[8] S.Chaudhuri, V.Ganti, and R.Kaushik, "A Primitive Operator for Similarity Joins in Data Cleaning," 2006.
[9] R.Bayardo, Y.Ma, and R.Srikant, "Scaling Up All Pairs Similarity Search," 2007.
[10] B.Bocek, E.Hunt, and B.Stiller, "Fast Similarity Search in Large Dictionaries," 2007.
[11] A.Elmagarmid, P.Iperiotis, and V.Verykios, "Duplicate Record Detection: A Survey," 2007.
[12] K.Chakrabarti, S.Chaudhuri, V.Ganti, and D.Xin, "An Efficient Filter for Approximate Membership Checking," 2008.
[13] S.Agrawal, K.Chakrabati, S.Chaudhuri, "Exploiting Web Search Engines to Search Structured Databases," 2009.
[14] S.Chaudhuri, V.Ganti, and D.Xin, "Exploiting Web Search to Generate Synonyms for Entities," 2009.
[15] W.Wang, C.Xiao, X.Lin, and C.Zhang, "Efficient Approximate Entity Extraction with Edit Distance Constraints," 2009.
[16] J.Lu, J.Han, and X.Meng, "Efficient Algorithms for Approximate Membership Extraction Using Signature-Based Inverted Lists," 2009.
[17] D.Karch, D.Luxen, and P.Sanders, "Improved Fast Similarity Search in Dictionaries," 2010.
[18] W.Hon, T.Lam, R.Shah, S.Tam, "Cache Oblivious Index for Approximate String Matching" 2011.
[19] G.Navarro, R.Baeza-Yates, E.Sutinen, "Indexing Methods for Approximate String Matching," 2011.
[20] Liwei Wang, Xiaofang Zhou, Zhixu Li, "Efficient Approximate membership Localization Within a Web-Based Framework," 2012.

N.Vaishnavi received her B.Tech, degree in Information Technology from Anna University at Syed Ammal Engineering College, Tamilnadu, India, in 2012. She is currently pursuing her M.E in Computer and Communication engineering from Anna University at Sethu Institute of Technology, Tamilnadu, India.

Y.Arockia Jesuraj received his M.Sc degree in Computer Science from Madurai Kamaraj University, Tamilnadu, India, in 1997. He has received his M.Tech degree in Computer Science and Engineering from Anna University, Tamilnadu, India, in 2005.