# ENTITY RECOGNITION IN A WEB BASED JOIN STRUCTURE

J.KAVITHA M.TECH, A.PASCA MARY (M.E)
Dept of cse,Er.Perumal Manimekalai college of Engineering, India
pascamary@ymail.com

***Abstract:*** **In this paper, we propose a new type of Dictionary-based Entity Recognition Problem, named Approximate Membership Localization (AML). The popular Approximate Membership Extraction (AME) provides a full coverage to the true matched substrings from a given document, but many redundancies cause a low efficiency of the AME process and deteriorate the performance of real world applications using the extracted substrings. The AML problem targets 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, we propose the optimized algorithm P-Prune that prunes a large part of overlapped redundant matched substrings before generating them. Our study using several real-word data sets demonstrates the efficiency of P-Prune over a baseline method. We also study the AML in application to a proposed web-based join framework scenario which is a search-based approach joining two tables using dictionary-based entity recognition from web documents. The results not only prove the advantage of AML over AME, but also demonstrate the effectiveness of our search-based approach.**

*Index Terms*—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.

In this paper, we propose a new type of membership checking problem: Approximate Membership Localization (AML). 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.

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

Being a subset of the AME results, a straightforward approach (AME-based) to obtain AML results is to a posteriori remove redundancies from the AME results. These redundant substring and reference pairs referred to as redundancies in the rest of the paper fall into one of the two following categories.

1. Boundary Redundancy: For example, if d = "... published in acm sigmod conference in 2009" and r = "acm sigmod conference," the AME may find not only "acm sigmod conference," but also other substrings such as "in acm sigmod conference," "acm sigmod," "sigmod conference" and "sigmod conference in" matching with r.

2. Multiple-match Redundancy: For example, if d = ". . . submitted to vldb journal..." and r1 = "vldb conference," r2 = "vldb journal," the AME may find substring "vldb" matched with r1, as well as "vldb journal" matched with r2.

## II. EXISTING APPROACH

Existing problem of identifying all sub-strings in an (long) input string which approximately match (according to one of several popular similarity measures) with some member string in a large dictionary characteristic of this scenario is that most input sub-strings do not match with any member of the dictionary.

Existing System developed a compact filter which efficiently filters out a large number of sub-strings which cannot match with any dictionary member. The sub-strings which pass our filter are then verified by checking for membership. At the same time, our filter is exact in that any input sub-string which matches with a dictionary member will not be filtered out.

- 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.

## III.NEW APPROACH

We propose a new type of membership checking problem: Approximate Membership Localization (AML). 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.

The AML provides non overlapped substrings which is a better approximation to the true matched substrings. In the event where several substrings overlap, only one with highest similarity qualifies as result. To perform AML We used optimized Alpha-Beta pruning and P-Prune that prunes a large part of overlapped redundant

We proposed web-based join framework is a typical real-world application greatly relying on membership checking. For instance, assuming a list of conference names and given a paper title, discovering which conference this paper was most probably published at is supported by the gathering of a certain number of web documents containing this paper title.

For each conference name, the number of its approximate mentions and their distance to the paper title are used as indicators of the probability for the paper to be related to this conference. AME and AML are applied in the membership checking submodule. The experimental study show that the best precision and recall of web-based join with the AML results can be as high as 0.873 and 0.831, respectively, which is much better than the AME results (0.5 and 0.8, respectively).

- 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.

*A.Proposed framework*

In this section, we present the web-based join framework. As opposed to the traditional approximate join, web-based join targets a different scenario: given 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 from the web an intermediary table RT 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 dirty and noisy, RT shall contain the likelihood associated with its entries. Based on the hypothesis that there exist web documents containing elements of T:X that also contain the elements of R:A, we use the elements of T:X as a query for a search engine to retrieve the ranked list of documents Docs.

*B.Scoring correlations*

The values of the links to clean references of R in Docs are measured with an unsupervised approach. This approach 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. 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 the position of T:x.

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 results.

*C.Potential redundancy prune*

The AME-based method for AML use time resources for generating and identifying unqualified redundant matches. In order to efficiently solve AML, we propose an optimized algorithm P-Prune, which can prune potential redundant substrings before generating them. This algorithm shows a much higher efficiency than the AME-based algorithm, as will be demonstrated in the experimental study.

General idea of P-Prune. For an input document M, AML only requires bestmatch substrings. Assuming we can divide M into subdocuments, where each subdocument is a consecutive substring of M and subdocuments may overlap with each other such that all bestmatch substrings of M are located within these subdocuments, the problem of finding all bestmatch substrings from M becomes a problem of finding the bestmatch substrings from these subdocuments of M. Furthermore, if there is at most one bestmatch substring in a subdocument, it becomes faster to judge whether there is a bestmatch substring in each subdocument. For convenience of presentation, we define this kind of subdocument as a domain below.

*D. Generating domains from document*

To generate domains from M, all reference entities have to be considered. Here, we leverage the basic prefix signature Scheme to find the prefix signature set of each entity. If a substring m from M matches with an entity r, it should contain at least one word from the prefix signature set of r. In this paper, we call these words in the prefix signature set of r as r's strong words.

*E.P-PRUNE algorithm*

Step 1: We step to the next new word position poscur, with the word wcur in that position.

Step 2: We remove the first domain from Dsort and put it into Dact, until it can't satisfy Dsort[1].posb = poscur.

Step 3: For each domain D = (posb; pose; r) in Dact, if wcur presents in r, it becomes a word in D's segment, otherwise it becomes a word in D's interval. The Interval Pruning strategy is applied to all generated intervals. If poscur = D:pose, the Boundary Pruning and Weight Pruning are applied as well.

Step 4: If poscur is not included in any range of domains or it already reaches the ending position of M, then the function LocateBestMatch is called to generate the bestmatch substrings from all domains in Dact.

Step 5: We do step 1 to step 4 iteratively, until poscur reaches the ending position of M. We output all bestmatch substrings we learn.

## IV ARCHITECTURE DIAGRAM

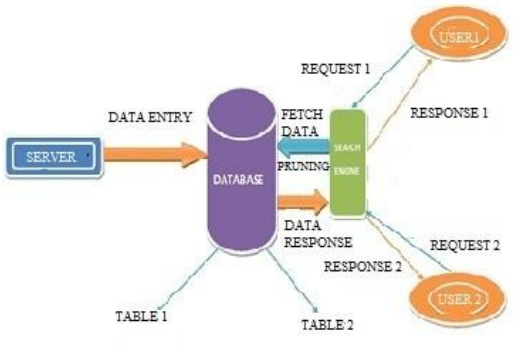


Fig.1 Architecture diagram

### *A. Multi-pattern matching*

Which aims at finding all occurrences of patterns from a given set within a document. A popular technique for solving the multi-pattern matching problem is to build a tree over all patterns as used in the well-known algorithm. This can significantly reduce the number of comparisons between substrings and patterns. However, this technique only works for exact matches

### *B. Approximate string matching*

If there is only one member in the reference list, then the AME or AML problem is reduced to another well m, "Approximate String Matching," which targets at finding a pattern string approximately in a text. If we model each reference into a pattern, the AME or AML problem reduces to another well-studied problem in the "String Matching" literature.

### *C. Approximate joining*

After "Approximate String Matching", This Module helps us to find a true matched substring from given two sets of tables.T1 and T2. The similarity function and a similarity threshold, the task of approximate join is to find all pairs of strings from the two tables(Like T1 and T2) and finally get the true matched substring.

Weighted Jaccard similarity of two strings s1 and s2 can be as follows

$$WJS(s1,s2)=\frac{wt(\text{number of tokens shared by string})}{wt(\text{total number of tokens})}$$

### *D. Aml results*

one way is to use patterns of the association between T1 and T2, which need to be learned with enough true matched data sets. The other way is to score the correlations between elements between T1:X and T2:X and also the references in R according to the results of AML. In our approach, we prefer the true matched AML and AME with an supervised approach and that will be shown in a Web document.

Three parameters are required for the evaluation of correlation

- Frequency
- Distance
- Importance

In the event where several substrings overlap, only one with highest similarity qualifies as result.

## V.CONCLUSION

In this paper, we formalize the AML problem and propose to solve it with an efficient P-Prune algorithm. 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.

We apply both approaches within our proposed web-based join framework, which is a typical 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 0.873 and 0.831, respectively, largely outperforming AME (where results are 0.5 and 0.8, respectively).

We also apply the web-based join framework in joining publication titles with venue names from the ERA 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.

## VI. FUTURE WORK

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. We believe that our 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.

## VII. REFERENCES

J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vision*, Oct. 2003, pp. 1470–1477.

D. Lowe, "Distinctive image features from scale-invariant key points," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 404–417.

H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *J. Comput. Vision Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.

D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2006, pp. 2161–2168.

O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2009, pp. 17–24.

W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial-duplicate web image search," in *Proc. 18th Int. Conf. Multimedea*, Oct. 2010, pp. 511–520.

X. Shi, W. Zhao, and Y. Shen, "Automatic license plate recognition system based on color image processing," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2005, pp. 307–314.

M. Zayed, J. Boonaert, and M. Bayart, "License plate tracking for car following with a single camera," in *Proc. IEEE Int. Conf. Intell. Trans. Syst.*, Oct. 2004, pp. 719–724.

S. Yohimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and N. Pedrycz, "License plate detection system by using threshold function and improved template matching method," in *Proc. IEEE Ann. Meet. Fuzzy Inform.*, Jun. 2004, pp. 357–362.

S. Yoshimori, Y. Mitsukura, M. Fukumi, N. Akamatsu, and R. Khosal, "License plate detection system in rainy days," in *Proc. IEEE Int. Symp.Comput. Intell. Robot. Autom.*, Jun. 2003, pp. 972–976.

Y.-Q. Yang, J. B. R.-L. Tian, and N. Liu, "A vehicle license plate recognition system based on fixed color collocation," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Aug. 2005, pp. 5394–5397.

 