International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 2, Issue 2, February 2014)

Effective Data Retrieval Mechanism Using AML within the Web Based Join Framework

Usha Nandini D[1], Anish Gracias J [2]

[1]*ushaduraisamy@yahoo.co.in*
[2]*anishgracias@gmail.com*

***Abstract*— A vast amount of assorted information are posted and retrieved on web by its users and administrators. For the users of web, the main issue is to browse through the exact data they are looking for. This web content mining has attracted the interest of researchers towards developing an efficient technique for retrieving the exact web contents for users query. There are several techniques in existence, which provides a reasonably good performance in web content mining. But it is clearly evident that, these existing techniques could be replaced by improved web content mining techniques, which could be utilized for real-world applications. Approximate Membership Localization (AML) has been used previously to retrieve true matches for clean references. In this paper, an improved AML technique is used for extracting those true matches for clean reference which are non-overlapped. The main objective of this improved AML introduced in this paper is to retrieve exact true matches for clean reference and also avoid the redundant or overlapped matches. This improved AML combines two effective algorithms namely, stemming algorithm and p-prune algorithm to achieve its objective in web content mining. The simulation results suggests that this proposed improved AML retrieves more exact match for clean reference with less computational complexity and also reduces redundancy among retrieved true match for clean reference.**

***Keywords*— Approximate Membership Localization (AML), web content mining, stop-word, Stemming algorithm, P-Prune algorithm.**

I. INTRODUCTION

The task of Entity Recognition in a document is to recognize the entities that are predefined in that document. The predefined entities of a document may be the names of persons, products etc. Finding these entities is an easy task where the document size is minimal. But in case of larger documents, the task of entity recognition transforms into a Dictionary-based Membership Checking problem. There are enormous amount of textual data available online which are utilized for several educational and other purposes as information data sources.

Hence, information's should be retrieved effectively from web and this requirement has encouraged a study of different information retrieval technologies and development of effective algorithms to retrieve effective relevant information from web based on user queries. In order to retrieve relevant information's from web, these data must be organized.

Also, with the increasing amount of documents on web, the deterioration of these documents also increases. Hence retrieving these documents exactly becomes a trivial issue, as the references for these documents can be just approximate. There are even possibilities for retrieving non-relevant documents as a result of this deterioration. This dictionary-based approximate membership checking has been expressed be Approximate Membership Extraction (AME), which finds all the substring in a document that approximately matches the clean reference. AME guarantees a complete coverage of all the matched substrings in the document, but it results in a lot of redundancy which will not be suitable for real-world application. To overcome these problems, Approximate Membership Localization (AML) was used, which aimed at reducing the redundancy in AME results. Hence in this work, an improved AML is proposed to boost the performance of using AML for web-based join framework.

The improved AML has combined two effective techniques namely stemming algorithm and p-prune algorithm to obtain the objective. The following sections of this paper are categorized as follows. Chapter 2 briefs some of the related works that provides a clear picture for this proposed work. In chapter 3 the existing system is explained and it is followed by the proposed system explained in chapter 4.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 2, Issue 2, February 2014)

Finally in chapter 5, the experimental evaluation of proposed system is detailed and chapter 6 and 7 provides the conclusion and future enhancements for this proposed work.

II. Related Works

Eleni Mangina *et al* [1] have proposed a keyphrase extraction algorithm along with a tiling process for a document in an e-learning environment. Their technique has applied IUI techniques for an online e-learning environment. For keyphrase extraction technique, they have employed user modelling techniques, information retrieval techniques and extraction mechanisms and collaborative filtering methods. The main aim of their system was to recommend documents for the users of e-learning environment exactly based on their requirement and query with minimal inconvenience and non-intrusive manner. For this, the Key Extraction Algorithm was used to automatically extract queries and then those extracted results were filtered and provided to users.

Samhaa *et al* [2] have proposed another keyphrase extraction method for English and Arabic documents based on KP-miner. The KP-miner algorithm for extracting keyphrases was implemented in three steps. The initial step was to generate candidate keyphrases. In the second step, for the generated candidate keyphrase, weights were calculated. Finally based on the weight, final candidate phrase list was refined and generated.

Yashaswini *et al* [3] have developed a suffix stripping algorithm to retrieve Kannada information from web. Their work was mainly focused on extracting suffixes from Kannada languages for retrieving Kannada text available in online on Unicode. Using their algorithm, they have stripped fourteen different major classification of Kannada suffix and few other sub classes of Kannada suffix. Also the suffixes associated with nouns, adjectives and stop words were also stripped using their algorithm. This algorithm was used for text extraction, and other text recognition and speech recognition techniques. This algorithm for stripping suffixes was implemented along with a stemming algorithm.

Zhixu Li *et al* [4] proposed an efficient approximate membership localization method for a web-based join framework. This work was intended to minimize the demerits in approximate membership extraction. It uses the result of AME as base and has reduced the redundancies in AME results using AML. A dictionary-based approximate membership checking was performed by means of AME and non-overlapped matches were generated using AML by applying p-prune optimization algorithm.

III. Existing System

The existing system has made use of Approximate Membership localization (AML) which is an extension of Approximate Membership Entity (AME) to find out all the matched substrings from the documents. In existing system, the main objective of AML is to find the true and non-overlapped match for clean reference in a web-based join framework. AML mainly aims at finding the true match for clean reference. These true match for clean reference were found by using scoring correlation values. Document Retrieving based on Semantic words was performed.AML in existing system has used results of AME as base and to avoid the redundant results in it, Potential Pruning (P-Pruning) algorithm was used.

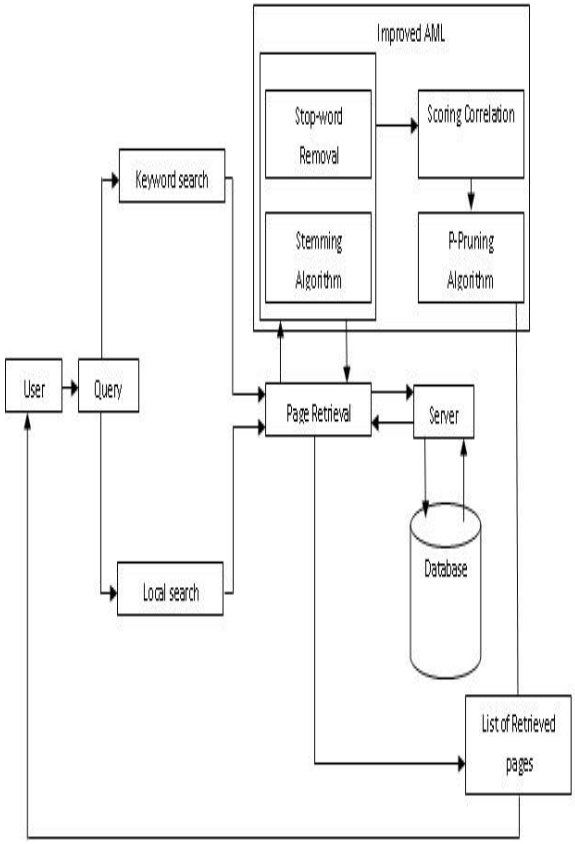
In the existing system, we are not able to get the exact matched substrings from the documents. The redundant of data will be the outcome of the existing system.

IV. Proposed System

The proposed system is an Improved AML, intended to improve the performance of existing AML technique for a web-based join framework. In this technique, two major techniques namely stemming algorithm and p-pruning algorithm are used to extract exact matches for clean reference and to reduce redundancy of match substring for clean reference. Stemming algorithm is performed by removing the stop words in the query provided by the user. This improved AML uses stemming algorithm to improve the performance of existing AML system in extracting the true match for clean reference.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 2, Issue 2, February 2014)

The architecture diagram for proposed system in a web-based join framework is given in figure



.

Fig 1. System Architecture

Being an extension of AME, the AML in existing system has simply used the results of AME, and later on it has involved in removing the redundancies in AME results. In our existing system, instead of using AME results, stemming algorithm is used as specified above. This proposed system consists of two main modules namely (i) Keyword search and (ii) local search.

The local search is the result of normal search operation performed using search engines like Google, Yahoo, Bing etc. The Keyword search is our proposed improved model for retrieving results exactly for the user query with less overlapped and more relevant data the user requires.

This proposed technique is applied in an web-based join framework which is based on approximate join. But opposed to the traditional approximate join, this proposed web-based join framework does not use approximate join. Instead it generates an intermediate table that contains the correlation values for all the extracted pages. Before extracting pages, initially stemming of stop words is performed.

A. Stop Words Removal:

Stop words are words which are filtered the stemming words prior to, or, after processing of text language data. These stop words are usually ignored by search engines. The purpose of stop words can be chosen by any group of words. For some search_mechines, these are some of the most common, short function words such as *the, is, at, which, on etc*. In this case stop words can cause problems when searching for phrases that include them, particularly with names such as 'The Who', ' The', or 'Take That', Other search engines remove some of the most common words- including lexical words, such as 'want'-from query in order to improve performance.

To remove the topic neutral words from the documents. Such as,

Articles (a, an, the)

Prepositions (in, of, at)

Conjunctions (and, or, nor)

B. Word Stemming:

A stem is a part of a word. In any instance, a word may occur in any one of its morphological form. Stemming is the process for reducing inflected words to the stem or root form. For example: 'argue','argued''arguing' are reduced to the stem 'argue'. A stem is a form to which affixes can be attached. Thus, in this usage, the English word *friendship* contains the stem *friend, to* which the derivational suffix *–ship* is attached to form a new stem *friendship*, to which the inflectional suffix –s is attached.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 2, Issue 2, February 2014)

Thus the key terms of a query or document are represented by stems rather the by the original words.

The stemming algorithm used here is an iterative algorithm and it recursively stems words until they cannot be reduced further. Based on these stemmed queries, reference list is generated and correlation values are found in the following steps of proposed technique.

C. Scoring Correlation:

In order to find the clean references approximately, there are two solutions in AML. One method is to link the patterns of association between query and reference. Second solution is to make use of correlation scoring. In this proposed technique, the scoring correlation is used with an unsupervised approach as in [5].

This scoring is generated for each best matched clean reference to perform join by setting a threshold value. The scoring value is based on three relevant parameters for evaluating the correlation among the best matched clean references. The three parameters of scoring correlation are *frequency*, *distance* and *document importance*. Given a list of elements *T* with an attribute *T.X* and a clean reference list *R*, for each clean reference *r* in *R.A*, the probability that *r* is correlated to a value *T.x* of *T.X* can be measured by equation 1.

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

Where, *imp(d)* is the importance of the document and is calculated using equation

$$2.\ imp(d) = \frac{\log(2)}{\log(1+\frac{[rank(d)]}{B})} \quad (2)$$

And *score(r,d)* is the local score of clean reference *r* in document *d* and the equation for finding score is given in equation 3.

$$score(r,d) = \omega_a . \frac{freq}{N} + (1-\omega_a) . \sum_{1 \le i \le freq} \frac{|d| - dist_i}{freq.|d|} \quad (3)$$

Where $|d|$ is the length of document *d*, *freq* is the frequency that *r* mentioned in *d*, $dist_i$ is the distance between the i^{th} mention of *r* and the query entity *T.x* in *d*. *N* is a normalization factor, ω_a is the weight given to the frequency of a reference mentioned in *d*, and 1-ω_a is the weight given to the distance between each mention of the reference and the position of *T.x* in *d*.

D. Approximate Membership Localization (AML):

The AML problems can be solved based on two assumptions and those are as follows.

Assumption 1: any approximate mention m that matched with a reference consists of consecutive words in a document, i.e., each m is a substring.

Assumption 2: only substrings whose length is up to a length threshold *L* are of interest, so we may as well require that $|m| \le L$.

Based on these assumptions two constraints are generated. The two constraints are,

1. Boundary Constraint.
2. Non-overlapped Constraint.

Further, using these constraints the document is divided into domains for convenience.

Domain: A domain D is a subdocument of M where there is at most one bestmatch substring in D. If a given entity r is the only possible reference that corresponds to the bestmatch substring in D, then D is one of r's domain in M.

From each domain, segments i.e. sub-domains are generated. These segments may be either overlapped or non-overlapped. According to indivisible segment constraint and the boundary constraint, we generate substrings with segments and intervals instead of single words.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 2, Issue 2, February 2014)

E. P-Prune Algorithm:

In this section the p-prune optimization algorithm used to avoid the redundancies is explained in detail. Basically there are three pruning strategies. Those three are given below.

Prune 1 (Weight Pruning). A domain D of r should be removed, if the sum weight of all segments in D is smaller than $\delta.\omega t(r)$.

Prune 2 (Interval Pruning). In a domain of r, if there is an interval t whose weight is larger than $\frac{1-\delta}{\delta}.\omega t(r)$ *on the left (right) side of the strong segment, then this interval and other segments and intervals on the left (right) side of t should be removed from the domain.*

Prune 3 (Boundary Pruning). The leftmost and rightmost partitions of a domain of r should be two segments of r.

The steps of *P-Prune optimization algorithm* is as follows.

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. Window domain is generated for each document *M* and each domain is represented by (pos_b, pos_e, r), where pos_b and pos_e refer to the starting and ending position of the domain in *M*, *r* is the unique reference it corresponds to. All domains are sorted according to their starting positions and stored in a set D_{sort}. We also use set D_{act} to store active domains being processed. Starting from the beginning position of *M*, we do:

1. We step to the next new word position pos_{cur}, with the word w_{cur} in that position.
2. We remove the first domain from D_{sort} and put it into D_{act}, until it can't satisfy $D_{sort}[1].pos_b = pos_{cur}$.
3. For each domain $D = (pos_b, pos_e, r)$ in D_{act}, if w_{cur} 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 $pos_{cur} = D.pos_e$, the Boundary Pruning and Weight Pruning are applied as well.
4. If pos_{cur} 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 D_{act}.
5. We do step 1 to step 4 iteratively, until pos_{cur} reaches the ending position of M. We output all bestmatch substrings we learn.

V. Conclusion

We have used two effective techniques namely stemming algorithm and p-prune algorithm to effectively improve the performance of AML in existing technique by reducing the overlapped matches for clean reference. Unlike existing AME and AML, this proposed improved AML has showed better performance in means of retrieving non-overlapped true match for clean reference in a web-based join framework. This technique has initially stemmed the query to extract true matches and later it has used the p-prune optimization algorithm to reduce redundancy. The performance of this improved AML suggests that it could be applied for real world applications, replacing existing methodologies.

VI. Future Enhancements

There are several problems that need to be investigated in the future. An alternate optimization technique could improve the performance to a greater extend. Also studies must be made to retrieve web pages faster than present.

Reference

[1] Eleni Mangina and John Kilbride, "Evaluation of keyphrase extraction algorithm and tiling process for a document/resource recommender within e-learning environments", Computers & Education, Vol. 50, pp. 807–820, 2008.
[2] Samhaa R. El-Beltagy and Ahmed Rafea, "KP-Miner: A keyphrase extraction system for English and Arabic documents", Information Systems, Vol. 34, pp. 132– 144, 2009.
[3] Yashaswini Hegde, Shubha Kadambe and Prashantha Naduthota, "Suffix Stripping Algorithm for Kannada Information Retrieval", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013.
[4] Zhixu Li, Laurianne Sitbon, Liwei Wang, Xiaofang Zhou and Xiaoyong Du, "AML: Efficient Approximate Membership Localization within a Web-Based Join Framework", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 2, February 2013.
[5] S. Chaudhuri, V. Ganti, and D. Xin, "Exploiting Web Search to Generate Synonyms for Entities," Proceedings of 18th International Conference World Wide Web (WWW), pp. 151-160, 2009.