An Efficient Utilisation of AML Using P-Prune Technique

Authors

Anju S Oommen¹, Marvin Philip Polson²

¹P.G Scholar, Department of Computer Science, Musaliar College of Engineering and Technology, Pathanamthitta, India
²Assistant Professor, Department of Computer Science, Musaliar College Of Engineering And Technology, Kerala, India

Email id:anjusoommen@gmail.com, marvinppolson@gmail.com

ABSTRACT

Nowdays tremendous growth of the amount of data or information available on internet or web. 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. The main objective of this AML introduced in this paper is to retrieve exact true matches for clean reference and also avoid the redundant or overlapped matches Approximate Membership Localization (AML) is the process that provides user with most relevant matched substrings. In a document, one word position belongs to only one reference matched substring.. In the Approximate Membership Extraction (AME) technique when searching a document in a web it displays all coordinated substring .So redundancy occurs and it causes less efficiency. For this,AML is used. It provides non overlapped substring during searching process and avoids redundancy by using optimized algorithm called P-prune algorithm. The pruning algorithm eliminates unwanted data that is overlapped data and increases the efficiency of searching process

Key Words: Approximate Membership Localization (AML), Approximate Membership Extraction (AME), Pruning algorithm

1. INTRODUCTION

Data mining automates the detection of relevant patterns in a database, using clear approaches and algorithms to look into current as well as historical data that can then be analyzed to predict future trends. Because data mining tools predict not only future trends but also behaviors by reading through databases for
hidden patterns, it allows organizations to make accurate, proactive, knowledge-based decisions, and answer questions that were previously too time-consuming to resolve.

Named entity extraction is a process of information extraction that is used to locate and classify elements in text into predefined categories such as names of persons, organizations, locations. As the domain knowledge encoded in the dictionary helps to improve the extraction performance.

The dictionary based entity recognition is that the terms in the documents are referred in a very big dictionary. The search takes little time and thus finds all matches against dictionary it means the number of matches or the size of the dictionary. The data matching process compare two sets of collected data. This matching is done in order to discard duplicate content.

A string metric is a metric that calculates the similarity or dissimilarity between two text strings for approximate string matching. The string searching is to find the location of a specific text pattern within a large body of text. For example sentence, paragraph.

Approximation is defined as similar but not exactly equal. The estimated string matching is a technique for discovering strings that match a pattern nearly rather than accurately. The difficulty of approximate string matching typically separated into two sub-problems finding approximate substring matches inside a given string and finding dictionary strings that match the model approximately.

The approximate membership Extraction (AME) is a dictionary based entity search process. It takes more searching time and it causes many redundancies. To overcome this problem approximate membership localization is proposed. We propose a web-based join framework which consists of a web search along with the approximate membership localization. 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..ie 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 localization(AML) is applied on these documents using the clear reference table and extracts the entities form the document to form the intermediate reference table using jaccard similarity, Score Correlation.

2. RELATED WORKS

In [1] Agarwal et al. suggested a technique that use a That is it matches the query terms only against the information in its own database. combination of pre-processing and web search engine adaptations in order to implement entity search functionality at very low space and time overhead. The main tasks are to identify
relevant information in a structured database using a web search query very efficiently and effectively. The search in each structured database is “soiled” in that it exclusively uses the information in the specific structured database to find matching entities. The result from the structured database search is therefore independent of the result from web search. The major drawback is that it takes a high processing time to search from a structured database.

In [2] Arasu et al. suggested a similarity join operation for reconciling representation of an entity. Set similarity join algorithm define that given two record compilation of sets recognize the entire couple of set, individual from every assortment that are extremely related. The information anthology frequently has different contradiction which have to be predetermined earlier than the data can be worn for exact data examination. The conception of resemblance is captured numerically using a string based similarity. Apart from string based similarity semantic relationship flanked by entities can be subjugated to recognize diverse representation of the identical thing. The algorithm is characterized as signature based algorithms that first generate signature for record sets, subsequently find every one of twosome sets whose signature overlie, and finally acquiesce the division of these applicant brace that gratify the set- relationship predicate. The major drawback is that it just compare with minimum amount of database so that it does not give exact similarity.

In [7] Karachi et al. describes an algorithm to resolve the fairly accurate thesaurus matching quandary. Given a directory of words w, highest distance d, preset at pre-processing instance and a question word q to retrieve all words from w that can be transformed into q with d or less edit operations. Each word is represented by a string of characters over a finite alphabet Σ. The Levenshtein distance ed (a,b) defines a metric between two words a,b and is used to compute distance between two words. The most frivolous algorithm to crack the trouble is scrutinize consecutively through the input list and noting the best match at each entry. The major drawback is that this distance computations are expensive and takes more time so processing is low.

In [3] Chan et al. describes the difficulty of directory a wording to support examination of substrings that match a given prototype with major errors. A naive result either has a worst case matching time complication or requires space. Developing a resolution with enhanced performance has been a contest for calculating the distance index that can hold up error matching in respect to point where occ is the amount of happenings. The major concern is how to archive efficient matching without large amount of space for indexing, one can improve the matching instance by counting all probable incorrect substrings however this seems to need o(nk)space. They are able to avoid brute force matching of patterns with a moderate increase in the index size. The major drawback is that it take long time and space complexity is high.

In [4] Chaudhuri et al. describes about the entity matching task identifies entity pairs one from a reference entity table and other from an external entity list. The task is to identify whether or not a candidate string
matches with member of reference table. However the challenge is that it is quite hard to obtain a large number of documents containing string unless large portion of the web is crawled and indexed as done by search engines. The approach is mainly applied to calculate string resemblance score between the candidate and the reference strings. The major problem is that the excellence of the id token set is low.

3. EXISTING SYSTEM

The Approximate membership Extraction (AME) is a dictionary based entity search process. AME aims at identifying all substrings approximately matching any reference. The main objective of AME guarantees a full coverage of all true matched substrings within the document. But it generates many redundant matched substrings and it also lower efficiency and accuracy. The major limitations of AME are that causes redundancy and lower the performance efficiency. For example if there exists a dataset contains various names such as {abi, asha, rani, ram, anuskha, ramanathan}. If the input string is “ram”. The AME retrieve all the substring that match that input string. It will retrieve names such as {ram, ramkumar, anusharam, ramanathan}. The AME does not match the exact data it is not suitable for the real world entities.

The process of AME is described detail in a diagrammatical format. AME process takes more time to remove the the redundancies so its performance degrades. The work flow of AME is shown.

![Flowchart of AME process](image)

**Fig.1** Work flow of AME process
4. PROPOSED SYSTEM

The proposed approach is to overcome the problems in approximate membership extraction (AME). Approximate membership localization propose at situating non overlapped substrings references approximately mentioned in a given record, generally in documents each string can be mentioned more than once. This will create data redundancies. Similarly for exactly matched strings there will be always only one true value that means the true mentioned strings should not overlap. In AML by using the score value and the similarity value the data redundancies should be avoided. In order to discover the non-overlapped substring pruning concept is used. Pruning is an optimization algorithm; it prunes redundant matched substring before generating them.

In this system the documents are retrieved from web. The document taken is the journal data and kept it in the certain location as a dataset. The dataset is encumbered in to the database. After entering the dataset in to database it splits the content of each document by defining a field to it. Such as ID, Name, Domain, ISSN.

The three main steps is required for AML process

Step 1: Search Keyword Scoring
Enter querying word
Score the Word( Scoring Correlations)

Step 2: Similarity Matching
Apply top-k matching algorithm
And search for relevant words present in the database

Step 3: Membership Localization
   3.1 Apply AML algorithms
      3.1.1 Probable Redundancy Prune

3.1 Apply Boundary Redundancy
3.3 Apply Multiple-match Redundancy
   3.4 Produce Best Match Output
Scoring Correlations

Scoring correlations for each document depends upon three appropriate parameter frequency, distance and document relevance.

**Frequency freq**: the number of times each reference is mentioned in each document of Docs.

---

*Fig. 2 Design of AML process*
Distance **dist**: the distance between the mention of each clean reference and the position of T.x.

**Document importance imp**: documents retrieved on the web are of different magnitude w.r.t. their significance to the query, i.

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.

\[
P(r, T.x) = \frac{\sum_{d \in Doc} imp(d) \cdot score(r, d)}{\sum_{d \in Doc} imp(d)} \quad (1)
\]

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

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

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.

\[
score(r, d) = \omega_a \cdot \frac{\text{freq}}{N} + (1 - \omega_a) \sum_{i \leq \text{freq}} \frac{|d| - \text{dist}_i}{\text{freq} \cdot |d|} \quad (3)
\]

Where is the length of document d, freq is the frequency that r mentioned in d, is the distance between the i\textsuperscript{th} mention of r and, \(w_a\) is the weight given to the frequency of a reference mentioned in d, and the query entity T.x in d. N is a normalization factor 1-\(w_a\) is the weight given to the distance between each mention of the reference and the position of T.x in d.

### 4.2 Similarity Calculation

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 inputs values means it does not give the similarity value. This similarity process is done using word net dictionary. The word net matches all entries of the database with the given input query. First it performs syntactic checking and then checks for linked synonyms. Using the result of sentence structure checking, linked synonyms it calculates the value of similarity. The one with largest similarity to its matched entity is the best matched substring.
**Inverse Document Frequency**

Inverse document frequency (IDF) is a mathematical gauge that indicates how essential a word is to a document in a collection. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases relatively to the number of times a word materialize in the document. Where N is the overall quantity of document .\(d_i\) is defined as document frequency.

The inverse document frequency can be ascertained by

\[
\text{IDF} = \log \left( \frac{N}{d_i} \right)
\]

Where N=Total number of document \(d_i\)=Document frequency

**4.4 Jaccard Similarity**

Jaccard similarity is a gauge used for evaluating the correspondence and multiplicity of experiment sets. The Jaccard coefficient process likeness between limited example sets, and is distinct as the amount of the juncture separated by the amount of the unification of the example sets that is comparing two strings \(S_1, S_2\). Jaccard similarity addresses finding of textually similar documents in a large corpus such as the Web or a collection of news articles. Character-level similarity is done here and not similar meaning.

Jaccard similarity

\[
\text{WJS} (S_1, S_2) = \frac{\text{wt}(S_1 \cap S_2)}{\text{wt}(S_1 \cup S_2)}
\]

Where \(S_1, S_2\) is string_1, string_2

**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 \(m\) can be 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 \(m \leq L\)

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

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

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 \(r\)'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.

**Pruning Algorithm**

Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. The twin goal of pruning is minimised complexity of the final classifier as well as better predictive accuracy by the reduction of over fitting and removal of sections of a classifier that may be based on noisy or erroneous data.

By pruning technique it avoid redundancies thus avoid the problem of AME (Approximate membership extraction). 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 $\delta \cdot \omega_t(r)$

Prune 2 (Interval Pruning). In a domain of r, if there is an interval t whose weight is larger than $1 - \delta / \delta \cdot \omega_t(r)$ 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.

**5. CONCLUSION**

The Approximate Membership Localization (AML) is used to overcome the redundancy problem of Approximate Membership Extraction (AME). An efficient P-Prune algorithm is used to avoid redundancy. Prune is proved to be quite a lot of times faster, sometimes even hundreds of times faster, than existing AME methods. To inspect the changes of AML over AME, we apply both approaches within our proposed web-based framework. This is a typical real-world application that greatly relies on the results of member checking.

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


