HYBRID APPROACH FOR GENERATING NON OVERLAPPED SUBSTRING USING GENETIC ALGORITHM

Akila Rani.M[1], Shanthi.D[2], Farzhana.I[3]

[1]*Assistant Professor, Department of CSE, NPR college of Engineering and Technology, TamilNadu, India*
[2]*Professor and Head, Department of CSE, PSNA College of Engineering, TamilNadu, India*
[3]*PG Student, Department of CSE, NPR college of Engineering and Technology, TamilNadu, India*

Abstract

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. There should not be any overlap in a true 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. To overcome this problem, 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. The high comparison load and time taken for generating the result is minimized. This enhancement can be achieved by Genetic algorithm which helps in identifying true matched substring with the help of fitness function. The equivalent key term are compared instead of comparing all the terms and hence reduces the time taken.

Keywords- *Approximate Membership Localization (AML), Approximate Membership Extraction (AME), Pruning algorithm, Genetic algorithm.*

----------***----------

1. INTRODUCTION

Data mining is a process of collecting, searching and analyzing a huge amount of data in a database to find patterns or relationship. Data mining finds important information hidden in huge volumes of data. Technically it is the process of finding the correlations or patterns among dozens of fields in large relational database.

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 approximation is done based on similarity function such as semantic similarity, edit distance, jaccard similarity, cosine similarity.

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.

2. RELATED WORKS

In [1] Agarwal et al. suggested a technique that use a 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. That is it matches the query terms only against the information in its own database. 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 overlap, and finally yield 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 $\sum$. 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 check 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 used 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 redundancies so its performance degrades. The work flow of AME is shown.

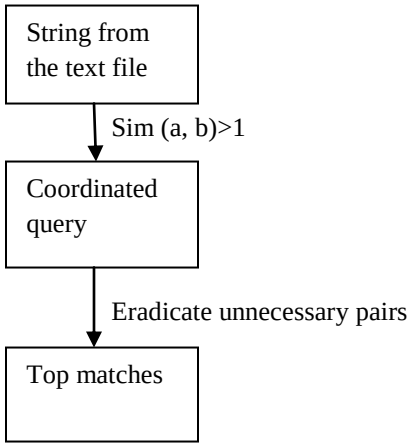


Fig 1 Work flow of AME Process

4. PROPOSED SYSTEM

The proposed approach is to overcome the problems in approximate membership Localization (AML). Approximate membership localization intend 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 redundancies. Similarly for true 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 redundancies should be avoided. In order to find the non-overlapped substring pruning concept was used. The pruning is an optimization algorithm it prunes redundant matched substring before generating them.Eventhough it eliminates unwanted data there still exist redundancies the matched pair results of the AML are much nearer to the real coordinated pairs of AME results. To avoid this redundancies genetic algorithm is being proposed. A genetic algorithm (GA) is a investigate heuristic that imitate the method of normal progression. This heuristic (also sometimes called a meta heuristic) is regularly used to produce useful resolution to optimization and search difficulty. Genetic algorithms fit in to the big group of development algorithms, which create answer to optimization problems using procedure encouraged by ordinary progress, such as legacy, alteration, choice, and cross over. In initialization process initially many individuals solution are randomly generated to form an initial population. Selection process discards the bad design and only keeps the best individuals in the population. In crossover process it cross checks the given query in all the documents. Mutation is an iterative process where sub child acts as a parent and the chain grows up to a level. Hence genetic algorithm is used to overcome the problem of AML. It also increases the performance, accuracy and reduces time of searching process.

In this system the documents are retrieved from web. The document taken is the journal data and storing it in the certain location as a dataset. The dataset is loaded in to the database. After loading 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 architecture diagram AML process describes the process involved in it.

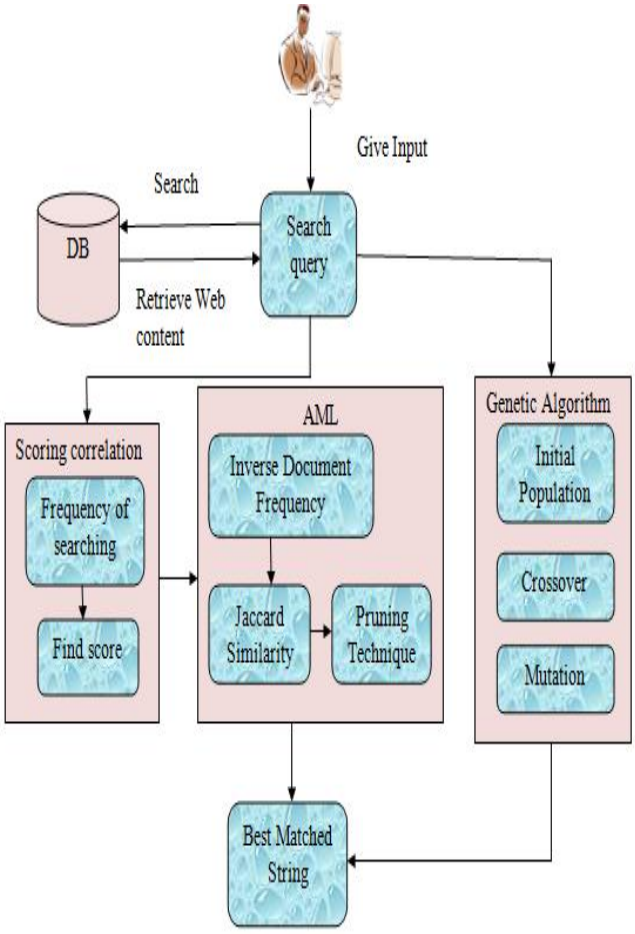


Fig 2 Architecture diagram of AML process

4.1 Scoring Correlation

The scoring method is used to determine the amount of instances and the locations where a clean orientation is indicated in a document. Then for each word in the database calculate score value. For calculating score three parameters are important such as frequency, distance, document importance. Frequency (f) is defined as number of times each reference is mentioned in each document. Distance (d) is calculated between the mention of each reference and the position .The importance of the document imp (d) is based on their relevance to their query.

Score correlation=imp (d)*score (d)/imp

Where imp (d) =log 2/log1

Score (d) =weight*freq*w1*s1

Where weight=freq/length of query
Freq=list of files
W1=1-weight
S1=∑(1 to n) [∴ n=number of occurrence]

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 related synonyms. Using the result of syntax checking, related synonyms it calculates the value of similarity. The one with largest similarity to its matched entity is the best matched substring.

4.3 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 .df is defined as document frequency. The inverse document frequency calculated as

$$IDF=\log (N/df)$$

Where N=Total number of document
df=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 S1,S2. Jaccard similarity addresses finding of textually similar documents in a large corpus such as the Web or a collection of news articles. Here character-level similarity is done and not similar meaning.

$$\text{Jaccard similarity WJS (s1, s2)} = \frac{wt\,(s1 \cap s2)}{wt\,(s1 \cup s2)}$$

Where S1, S2 is string1, string2

4.5 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 dual goal of pruning is reduced 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). This method is just retrieving the data based on the similarity comparison. So there will be redundancies AML reduces redundancy. The overall searching process is effective when compared with others. It produces approximately exactly matched substring.

4.6 Genetic Algorithm

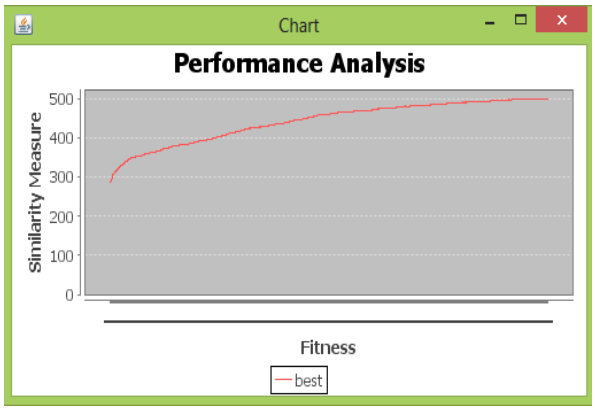
Genetic algorithm objective is situating non-overlapped substrings in a given record that can roughly equivalent to any sterile reference with the best matched pairs. Genetic algorithm performs some basic process. First create initial population, this populace is arbitrarily produced and can be any needed size, from only some individuals to thousands. Next to it cross over is done. In crossover process the contents in the given documents are cross checked to get the related matched string.

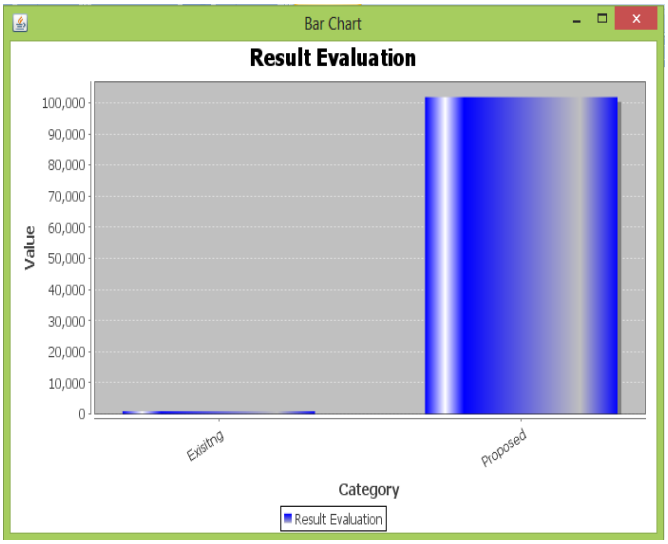
By doing this process the comparison load is reduced and makes the searching process easy and effective. In mutation process for the given query the matches in each document is calculated separately as taking the query as parent and their similar matches as sub child it is an iteration process . Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. Finally the best match is provided by comparing both pruning and genetic technique.

4.7 Pruning vs. Genetic Algorithm

In the existing system pruning concept is used to eliminate the redundancies. Generally pruning means eliminating unwanted things. So by using pruning concept in AML the fake strings, redundant matches are eliminated. Thus the best matched non-overlapped strings can be found easily.

Though pruning avoids redundancies the number of comparison will be large this will take huge amount of time to find the best match. In order to reduce comparison genetic algorithm is proposed. In genetic algorithm by using cross over technique the related terms for the input string was found and then the best match for the query word is found from the related terms. Hence genetic algorithm is more accurate than pruning and also less time consuming.





Comparison graph of pruning and genetic algorithm

3. CONCLUSIONS

The Approximate Membership Localization (AML) was used to overcome the redundancy problem of Approximate Membership Extraction (AME). Genetic Algorithm intention is positioning non-overlapped substrings in a specified document that can roughly match any sterile reference with the best matched pairs. The result of GA are estimated to be more nearer to the correct matched couple without relating overlapped superfluous substrings and also reduces the processing time when compared to other existing methods. Finally proposed method improved the efficiency and performance of predicting the best matches for big datasets.

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]. A. 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

[3]. 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.

[4]. 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.

[5]. 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.

[6]. A. 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.

[7]. D. Karch, D. Luxen, and P. Sanders, "Improved Fast Similarity Search in Dictionaries," Proc. 17th Int'l Conf. String Processing and Information Retrieval, pp. 173-178, 2010.

[8]. W. Hon, T. Lam, R. Shah, S. Tam, and J. Vitter, "Cache-Oblivious Index for Approximate String Matching," Theoretical Computer Science, vol. 412, pp. 3579-3588, 2011.

[9]. L. Gravano, P. Ipeirotis, H. Jag dish, N. Koudas, S. Muthukrishnan, and D. Srivastava, "Approximate String Joins in a Database (Almost) for Free," Proc. 27th VLDB Int'l Conf. Very Large Data Bases, pp. 491-500, 2001.

[10]. N. Koudas, S. Sarawagi, and D. Srivastava, "Record linkage: Similarity Measures and Algorithms," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 802-803, 2006.

[11]. Z. Li, L. Sitbon, L. Wang, X. Zhou, and X. Du, "Approximate Membership Localization (AML) for Web-Based Join," Proc. 19th CIKM Int'l Conf. Information and Knowledge Management, 2010.