International Journal of Engineering Research and General Science Volume 3, Issue 5, September-October, 2015
ISSN 2091-2730

Efficient Document Retrieval using Content and Querying Value based on Annotation with Proximity Ranking

[1]Ms. Sonal Kutade, [2]Prof. Poonam Dhamal

[1] ME Student, Department of Computer Engg, G.H. Raisoni College of Engineering and Management, Savitribai Phule Pune University, Pune.

[2]Assistant Professor, Department of Information Technology, G.H. Raisoni College of Engineering and Management, Savitribai Phule Pune University, Pune.

sonal.kutade@gmail.com

Abstract— Always it is hard to discover the relevant information in unstructured text documents. The structured information remains buried in unstructured text. Annotations in the form of Attribute name-value pairs are more expressive for retrieval of such documents. This system proposes a novel, different, alternative approach for document retrieval which includes annotations identification and also extends the existing system using fuzzy search with proximity ranking. This system identifies the values of structured attributes by reading, analyzing and parsing the uploaded documents. Searching process will make use of fuzzy search with proximity ranking for searching the user interested documents only. Thus this system proposes an approach for efficient document retrieval using effective methods.

Keywords— Document retrieval, instant-fuzzy search, proximity ranking, document annotation, OpenNLP, content, querying value, natural language processing.

INTRODUCTION

There are many application areas where users create and share their information; for instance, online job portal websites, news blogs, disaster management networks, scientific networks, social networking groups. Usually such data exists in unstructured text format. It also contains structured information but it remains buried in the presence of unstructured text. Current tools of information sharing allow the users for documents sharing and annotating/tag them in the ad hoc way, like the software of content management (e.g. MS Share-Point). Likewise, Google Base allows the users to define the attributes for their objects or choose from the predefined templates. This process of annotation can facilitate later information discovery. Various annotation systems allow only the "untyped" keyword annotation: e.g., a user may annotate a resume using a tag such as "Profile Computer Engineer"

Annotation strategies which use "attribute name-value" pairs are usually more expressive, because they contain more information than the untyped approaches. The above information can entered as (Profile, Computer Engineer) in such case. Existing system facilitates the structured metadata generation by identifying the documents which are likely to contain user interested information and this information is subsequently used for querying of the database. It uses CADS which stands for Collaborative Adaptive Data Sharing platform and which is used as an "annotate-as-you-create" infrastructure for facilitating the fielded data annotation. And later document owner modifies them by adding more annotation fields i.e. attributes. So here it requires more efforts of document owner which become time consuming process. Another limitations of existing system are no use of any searching and ranking technique.

So we propose an alternative, different and innovative approach which facilitates the identification of structured "attribute values". Later these values will be subsequently useful at the time of querying the database. It also uses Instant-fuzzy search with proximity ranking for searching the user interested documents only. The resultant documents will be ranked using keyword weightage.

The main Objectives of this system are to save the time by minimizing the user efforts in filling the information, to identify the attribute values i.e. content for attributes names when such information actually exists in document instead of prompting users to fill it, and to retrieve only the documents of user interest.

RELATED WORK

This system presented an annotation approach [1] which facilitates the structured metadata generation using CADS. It is done by identifying the documents that are likely to contain needed information and later this information will be useful for database querying. They presented the algorithms to identify the structured attributes which are likely to appear in the document, by utilizing both the content of text and query workload. The idea behind this approach is that humans are more expected to add the metadata during time of creation, if prompted by some interface or/and that it is much easier for the algorithms and/or humans to identify the metadata when such kind of information is actually existing in document, instead of fill up forms by naively prompting users with information which is not present in the document.

*CADS***:** This paper [1] proposed CADS system, which is used as a Collaborative Adaptive Data Sharing platform, and is a data sharing platform where the integration and annotation take place at the time of data insertion i.e. production and querying i.e. consumption actions. A main goal of CADS [3] is to influence the information demand for creation of adaptive insertion and query forms.

Instant Search: The integration of proximity information in instant fuzzy search for achieving the better complexities is explained in [2]. Many recent studies focused on the instant search. The studies in [6] proposed query and indexing techniques to support the instant search. Li et al. [8] studied the instant search on relational data which is modeled as a graph.

*Fuzzy Search***:** Fuzzy search studies can be categorized into two categories, first gram-based and second are trie- based approaches. In former approach, the data sub-strings are used for matching the fuzzy string. In second class approaches keywords are indexed as the trie, they depend on a traversal on the trie to determine the similar keywords [7]. This trie based approach is specially suitable for instant and fuzzy search [7] since every query is a prefix and trie supports efficient incremental computation.

Proximity Ranking: The Recent studies show that the term proximity is highly correlated with relevancy of document, and proximity-aware ranking increases the top results precision significantly. And, there are only few studies which increase proximity-aware searching query efficiency using techniques of early-termination [4], [5]. The techniques which are discussed in [4], [5] generate an additional inverted index for each term pair, which results in a large space. [5] studied only the problem for queries with two-keywords.

IMPLEMENTATION

A. Proposed System Architecture:

This system will use OpenNLP for stopword removal, checking of identification of attribute values. As shown in fig 1, here we have dataset of newsgroups containing thousands of documents. The Structured attribute names are stored in the database. The user can search by using either content i.e. attribute name of document or query containing attribute name and value. As user enters the query, the attribute name and value will be separated and identified by Preprocessing (OpenNLP). Then analysis and parsing of text file will be done using parser.. It will read, analyze & parse the whole document. At the other side these attribute names and values of content and queries i.e annotations will be useful to user for querying the database. At another side user will enter his query and finally he will get the resultant documents that are searched and ranked using instant fuzzy search and ranked with ranking based on calculation of keyword weightage (1) in documents. So user will receive only documents of his interest. In this way, this system is trying to prioritize document annotations that are many times used by users that are querying. After searching the documents, we can download required document and can view the annotations as per query in respective documents. And another main advantage of this system is that the resultant documents will be searched using fuzzy search and ranked using advanced technique of modified proximity ranking. The natural language processing tasks of OpenNLP are as shown in figure 2.

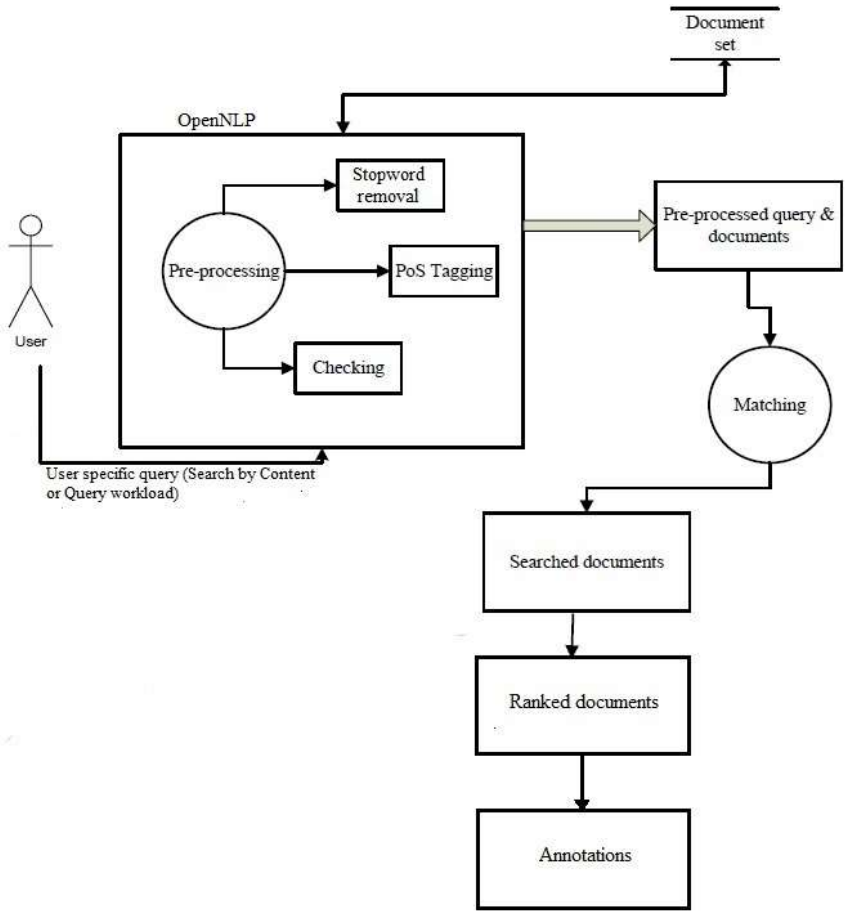

Fig. 1. Proposed System architecture

B. OpenNLP:

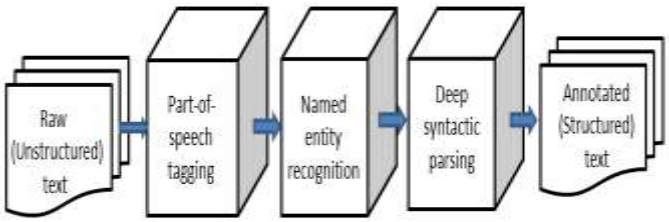

Fig. 2. OpenNLP tasks

The Apache OpenNLP library is a machine learning based toolkit for text processing of text of the natural language. It supports most of the tasks of NLP, like tokenization, sentence segmentation, named entity recognition, part-of-speech tagging, as well as parsing, chunking, and coreference resolution. These tasks are required to build more latest and advanced services of text processing. Text annotation usually involves these tasks at different linguistic levels. These tasks are done with right combinations of Open NLP tools. Splitter determines the sentences. It can find that a punctuation character marks the sentence end or not. Tokenizer segments the input character sequence into tokens such as words, punctuations and numbers.

POS tagger does the identification of the part of speech is done such as a noun, verbs, adverb for each word of sentence helps in analyzing role of each rule in sentences. So here "tag" method is used for tagger class of Open NLP. Example: Input – Tokens and Output – tag to each token. OpenNLP POS Tagger uses a probability model for predicting the right pos tag from tag set. A token can have many POS tags which depends on token and context.

Some tag set examples – DT:- singular determiner/ quantifier (e.g.that), NN:- singular noun / mass noun, IN :- preposition, NNS:- plural noun, VBZ:- verb, etc.

Instant search: It is referred as an emerging information-access model. Based on a partial query typed by user, it returns the answers instantly to user. E.g., Internet Movie Database has a search interface which offers the instant results to the users while they type their queries. When the user types "maha", the system returns answers such as "mahadiscom", "mahanews", "maharashtra times". Most of the users prefer to see search results instantly and they formulate their queries accordingly instead of being left in dark awaiting hitting the search button. This new technique helps users for discovering their answers with fewer efforts.

Fuzzy Search: Many of the users normally make typing mistakes in the search queries. The reasons for the same can be lack of caution, small keyboards of mobile, limited knowledge about data. So in this case, we cannot determine relevant answers. This problem can be solved by supporting the fuzzy search, in that we determine answers with keywords which are similar to query keywords. Combination of instant search and fuzzy search can provide better search experiences, particularly for the users of mobile-phone, who frequently having problem of "fat fingers" i.e., each keystroke is error prone and is time consuming.

Proximity ranking: Proximity ranking looks for document where two or more independent occurrences of matching terms are within a specified distance, where the distance is equal to the number of in-between words/characters. Here ranking will use the function for ranking which can be called as modified proximity ranking function which is defined in mathematical model.

C. Mathematical Model

Let S be the system which contains inputs, functions, and outputs.

S = {I, F, O} where

1)I = {I1, I2, I3. . . In}

Where, `I' is the set of documents that user wants to upload in

text, pdf, word format and there can be multiple files uploaded on server by multiple users or dataset of documents.

2) F = {F1, F2}

Here, two functions are defined which forms the system where

F1= Identification, separation of attribute values from attribute names and their insertion in csv file.

F2= Instant-fuzzy search with proximity ranking

3) O = {O1, O2, O3. . . O}

Where, `O' is the set of outputs which contain:

O= Set of resulted documents

- Ranking function:

Ranking will use following function to rank the resultant documents:

For each document d,

$$W=\sum_1^n i \qquad (1)$$

Where,

1) W = Weightage of query keywords in documents

2} i = weightage of each word in the document
=1/total no. of words in the document

3) n = total no. of query keywords

D. Algorithms

Algorithms used for fuzzy searching and ranking relevant documents:

Inputs: Documents in dataset D,

Query entered by user Q.

Output: Ranked relevant documents list

Let n be the total no. of documents in dataset.

I. When user enters a valid query,

1. for i =1 to n

2. Read document content

3. Compare query keyword with content of document

4. If (70% word match found)

Display the document

5. Else

Ignore and Go to next document.

II. Ranking function:

Finally, the valid segmentations are ranked using (1).

RESULTS

We test the system using the dataset of newsgroups containing thousands of documents. The system is built using ASP.NET using C# and MS SQL Server 2008. The maximum size of document is 32kb. Following graphs show result of searching of system annotated documents and ranking of them.

Figure 3 shows the graph of time taken for searching thousand no. of documents using content based search, query based search and their ranking.

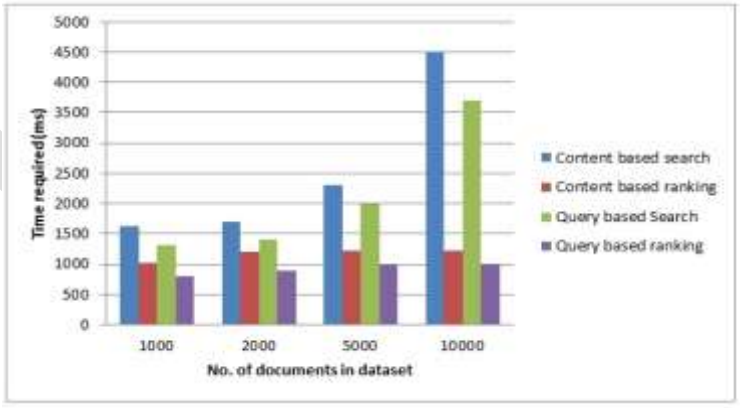

Fig. 3. Graph of time taken for searching/ranking relevant documents Vs total no. of documents

Figure 4 shows the graph of total no. of documents found by searching whole documents using content based search, query based search and more specific query based search.

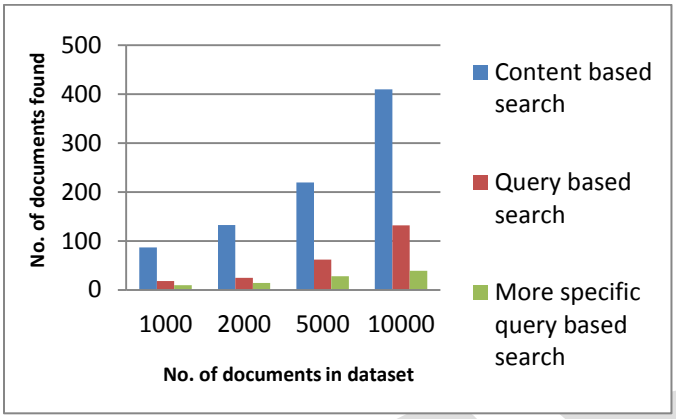

Fig. 4. Graph of total no. of documents Vs no. of documents found

After searching the documents, we can download required document and can view the annotations as per query in respective documents as shown in figure below.

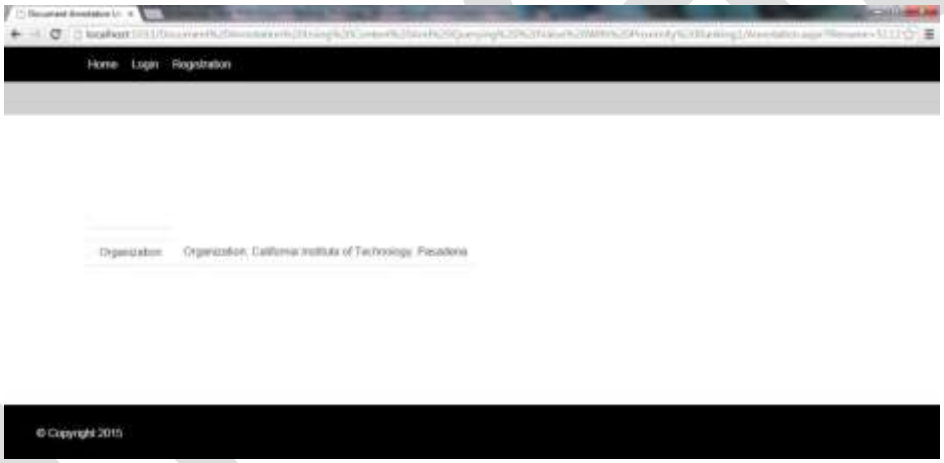

Fig 5. Annotation in required searched document

CONCLUSION AND FUTURE SCOPE

This paper proposes a new approach for efficient document retrieval including smart annotation, searching & ranking techniques. The system tries to satisfy querying needs of user efficiently. This system gives different ways for searching: the values of Content and Query. Using these techniques, we can increase chances of documents visibility up to maximum percent. Also using the fuzzy search and proximity ranking will achieve efficient time and space complexities and improve the overall performance of system. Users will get less and distinct results of documents. The text mining will be highly boosted due to this system.

Query based searching can be said as the future in information retrieval. Documents with complicated formats like can also be used in future. Document clustering can also be used in future. This system can surely give a huge boost in text mining and can be thought of as a changing trend.

ACKNOWLEDGEMENT

This is to acknowledge and thank all the individuals who played defining role in shaping this paper. Without their constant support, guidance and assistance this paper would not have been completed. Without their Coordination, guidance and reviewing this task could not be completed alone.

I avail this opportunity to express my deep sense of gratitude and whole hearted thanks to my guide Prof. Poonam Dhamal for giving her valuable guidance, inspiration and encouragement to embark this paper.

I would personally like to thank all staff members and my friends for their support.

REFERENCES:

1. Vagelis Hristidis, Eduardo J. Ruiz, Panagiotis G. Ipeirotis, , "Facilitating Document Annotation Using Content and Querying Value", volume 6, no 2, IEEE 2014
2. Chen Li , Cetindil, I., Taewoo Kim , Esmaelnezhad, "Efficient instant fuzzy search with proximity ranking", Data Engineering (ICDE), 30th International Conference ,IEEE 2014
3. V. Hristidis, E. Ruiz, " CADS: A Collaborative Adaptive Data Sharing Platform", SCIS, International University, Florida, 2009

A. Broschart, R. Schenkel, , S. Won Hwang, G. Weikum, M. Theobald, "Efficient text proximity search," SPIRE, 2007.

4. H. Yan, J. Wen, S. Shi, F. Zhang, T. Suel,, "Efficient term proximity search with the term-pair indexes,"CIKM, 2010, pp. 1229-1238.
5. H. Bast, , A. Chitea, F. Suchanek,Weber, "Ester : efficient search on text,entities, and relations," SIGIR, 2007.

B. Li, J. Feng, G Li, S. Ji, "Efficient interactive fuzzy keyword search," WWW, 2009.

C. Li, G. Li, J. Feng S. Ji, "Efficient type-ahead search on the relational data: a tastier approach" , SIGMOD, 2009.

6. Sonal Kutade, Poonam Dhamal, "Efficient Document Retrieval using Annotation, Searching and Ranking", IJCA, (0975 – 8887) Vol 108, No. 5, December 2014
7. Harshal J. Jain, M. S. Bewoor, S. H. Patil, "Context Sensitive Text Summarization Using K Means Clustering Algorithm", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012
8. Md. Abu Nisar Masud, Md. Munasir Mamun, "A General Approach to Natural Language Generation" In Proceeding of IEEE, INMIC, 2003.
9. Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng, "Annotating Search Results from Web Databases", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 3 YEAR 2013.
10. Akshay Shingote Nikhil Vispute Priyanka Dhikale," Facilitating Document Annotation Using Content & Querying Value", International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 4– Mar 2014
11. M. J. Cafarella, J. Madhavan, and A. Halevy, "Web-scale extraction of structured data," SIGMOD *Rec.*, vol. 37, pp. 55–61, March 2009.
12. M. Hadjieleftheriou and C. Li, "Efficient approximate search on string collections," PVLDB, vol. 2, no. 2, pp. 1660–1661, 2009.
13. D. Xin, K. Chakrabarti, V. Ganti, S. Chaudhuri, "An efficient filter for approximate membership checking," SIGMOD Conf pp.805–818, 2008.