Efficient Document Retrieval using Annotation, Searching and Ranking

Sonal Kutade
Dept. of Computer Engg., G.H. Raisoni College of Engg. and Management, University of Pune, India

Poonam Dhamal
Dept. of Computer Engg., G.H. Raisoni College of Engg. and Management, University of Pune, India

ABSTRACT

It is always difficult to find relevant information in unstructured text documents. In this paper we study the methods of fuzzy search, instant search and proximity ranking and how they can be used in the process of annotation of documents. These various methods can be integrated to give better search results and to achieve efficient space and time complexities. We propose a novel alternative approach which facilitates the generation of the structured metadata automatically using OpenNLP, methods of Instant-fuzzy search and Proximity ranking. It is done by identifying documents which are likely to contain the information of interest. And this information will be subsequently useful for querying the database.

General terms

Annotation, Data mining, Text mining, Proximity ranking

Keywords

Document Retrieval, Document Annotation, Instant-fuzzy search, Proximity Ranking, OpenNLP**.**

1. INTRODUCTION

The document retrieval is referred as the matching of some stated query of user against free-text records set. These records could be any type of mainly unstructured text, such as paragraphs in a manual, real estate records or newspaper articles. The queries of user can be from a few words to multi-sentence descriptions of information need. Document retrieval is sometimes referred to as a branch of Text Retrieval, or Text Retrieval.

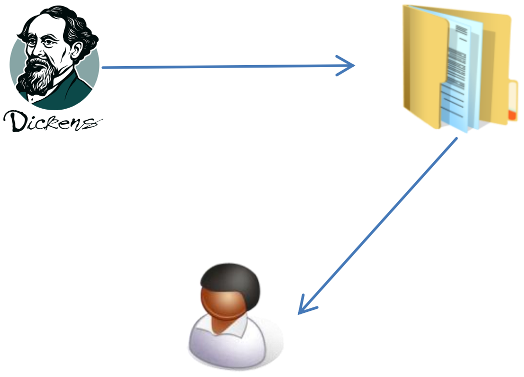

Fig 1: Document retrieval

If a user wants an efficient document retrieval process then annotation, document searching methods and ranking methods play a vital role in whole retrieval process. Here we discuss what these techniques are and how these different techniques are used in this document retrieval system.

Ranking: In the process of ranking every query answer is ranked based on its similarity or relevance to query, it is defined on various information pieces like co-occurrence of some keywords of query as a phrase in record and the query keywords frequencies in the record. Domain-specific features can play a vital role in ranking. E.g., for some publication, number of citations can be used as an indication in ranking because it is a good indicator of its impact. The Phrase matching effect in ranking gives better results. E.g., for the query q = bbrain, surgeryii, record containing the phrase “brain surgery” is more relevant than the record containing the keywords “brain” and “surgery” alone.

Basic Indexing: The three indexes are used to answer queries efficiently, a forward index, trie, and an inverted index as the methods described in [14] combines both methods i.e. fuzzy searching and instant search. In exacting, we can construct a trie for the terms in dictionary D. All paths to a leaf node from a root in the trie correspond to a term which is unique in dictionary D. Each and every leaf node stores an term’s inverted list. We can build a forward index too, which will include a forward list which contains the encoded integers of terms for every record. We can make use of this index for verifying if a record contains a keyword which matches a prefix condition.

Top-k Query Answering: If Given a integer which is positive say k, we find the k most query relevant answers. One way for finding these results is first determine all the results matching the query conditions, and then ranking them on their score. An another solution can be utilizing certain properties of the ranking function, and computing the k most relevant results using early termination techniques without computing all results.

OpenNLP: The Apache OpenNLP library is a machine learning based toolkit for the natural language text processing. It supports most common tasks of NLP, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are required to build more recent and advanced text processing services. Text annotation typically involves tasks at different linguistic levels, such as named entity recognition, part-of-speech tagging, tokenization, sentence boundary detection, semantic role labeling, co-reference resolution. Most of these are done with proper combinations of the OpenNLP tools. OpenNLP gives the annotations in the format of simple plain text. The OpenNLP tools really do a efficient job of annotations generation automatically.

Instant search: It is an emerging information-access model, it immediately returns the answers to user based on a partial query typed by user. E.g., the IMDB, (Internet Movie Database) has a search interface which offers the instant results to the users when they type queries. When the user types “uni”, the system returns answers such as “Unipune”, “Uniraj”, “Union bank of India”, and “Unishivaji”. More users prefer the experience of instantly seeing the search results and formulating the queries accordingly instead of being left in dark awaiting the hitting button of search. This new method helps users discover their answers with less efforts and quickly.

Fuzzy Search: Many users frequently make typing mistakes in the search queries. Mobile devices small keyboards, limited knowledge about data, lack of caution, can also be reason for mistakes. So in this case, by finding records with keywords matching with the query exactly, we cannot determine relevant answers. This can be solved by supporting the fuzzy search, in which we determine answers with keywords similar to query keywords. Combining instant search and fuzzy search can provide an better search experiences, especially for the mobile-phone users, who often have the problem of “fat fingers” i.e., each tap or keystroke is error prone and time consuming.

2. RELATED WORK

Instant Search: It is also known as *type-ahead search*. Many recent studies focused on the instant search. The studies in [8] presented the trie based techniques. Li et al. [9] studied the instant search on relational data which is modeled as a graph. The studies in [7] proposed query and indexing techniques to support the instant search.

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[10], [11], [13], [12]. In second class approaches keywords are indexed as the trie, they depend on a traversal on the trie to determine the similar keywords [8]. This trie based approach is specially suitable for instant and fuzzy search [8] 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 [15]. And, there are only few studies which increase proximity-aware searching query efficiency using techniques of early-termination [5], [6], [14]. The techniques which are discussed in [5], [6] generate an additional inverted index for each term pair, which results in a large space. For small or reducing index size, Zhu et al. [14] proposed to construct a compact index for a phrases subset. [6] studied only the problem for queries with two-keywords.

Annotation: An alternative approach is presented [1], [3] that facilitates the generation of metadata which is structured by identifying the documents that contain information of user’s interest and this information will be useful to query the database. In this the people will likely to assign the metadata related to the documents that they upload which will easily help users in documents retrieval.

3. IMPLEMENTATION

3.1 Information Extraction Algorithm (Annotation Process)

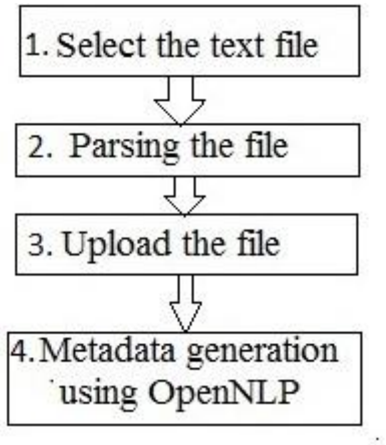

Fig 2: Information Extraction algorithm

3.2 Working and Theoretical Result

This system will use OpenNLP for automatically creation of metadata. As shown, the document publisher will select a text file then parsing of text file will be done. i.e...Ignoring stopwords from it and count frequency of high querying keywords, etc. Then at the time of uploading the document the metadata will be automatically generated using OpenNLP. At the other side end user will enter the query and he will get the ranked documents which will be searched using fuzzy search .User will receive only documents of his interest and this will be the best retrieval of documents.

4. CONCLUSION

This system proposes a new approach for annotating a document, and tries to satisfy querying needs of user efficiently. We studied how the fuzzy search and proximity ranking will improve efficiency of searching. Users will get less and distinct results due to automatic generation of metadata using OpenNLP, proximity ranking and instant-fuzzy search. The text mining will be highly boosted due to this system. In future we can enhance this system for any type of documents other than pdf, text, word, etc.

5. ACKNOWLEDGEMENT

I would like to say thanks to my guide “Prof. Poonam Dhamal” who gave her knowledge and time in order to complete this paper. This paper would never complete without her and the support of faculty members.

6. REFERENCES

[1] Vagelis Hristidis, Panagiotis G. Ipeirotis, Eduardo J. Ruiz, “Facilitating Document Annotation Using Content and Querying Value”, IEEE Transactions On Knowledge And Data Engineering, volume 6, no 2, IEEE 2014

[2] Cetindil, I., Esmaelnezhad, J., Taewoo Kim, Chen Li, “Efficient instant fuzzy search with proximity ranking”, Data Engineering (ICDE), 30th International Conference , IEEE 2014

[3] Akshay Shingote, Nikhil Vispute, Priyanka Dhikale, “Facilitating Document Annotation Using Content & Querying Value”, IJCTT, vol 9, March 2014

[4] Vagelis Hristidis, Eduardo Ruiz,” CADS: A Collaborative Adaptive Data Sharing Platform”, SCIS, International University, Florida, 2009

[5] R. Schenkel, A. Broschart, S. Won Hwang, G. Weikum, M. Theobald, “Efficient text proximity search,” SPIRE, 2007, pp. 287–299.

[6] H. Yan, S. Shi, F. Zhang, T. Suel, and J.-R. Wen, “Efficient term proximity search with the term-pair indexes,”CIKM, 2010, pp. 1229– 1238.

[7] H. Bast, F. Suchanek, A. Chitea, I. Weber, , “Ester : efficient search on text, entities, and relations,” SIGIR, 2007, pp. 671– 678.

[8] J. Feng, G Li, S. Ji, C. Li, , “Efficient interactive fuzzy keyword search,” WWW, 2009, pp. 371–380.

[9] C. Li, G. Li, J. Feng S. Ji, and, “Efficient type-ahead search on the relational data: a tastier approach” , SIGMOD, 2009, pp. 695–706.

[10] M. Hadjieleftheriou and C. Li, “Efficient approximate search on string collections,” PVLDB, vol. 2, no. 2, pp. 1660–1661, 2009.

[11] D. Xin, K. Chakrabarti, V. Ganti, S. Chaudhuri, “An efficient filter for approximate membership checking,” SIGMOD Conf, 2008, pp.805–818.

[12] R. Motwani S. Chaudhuri, and V. Ganti, “Robust identification of fuzzy duplicates,” ICDE, 2005.

[13] J. Lu, S. Ji, A. Behm, , C. Li, “ Space- constrained gram-based indexing for efficient approximate string search,” ICDE, 2009, pp.604–615.

[14] M. Zhu, S. Shi, J.-R. Wen, and N. Yu, “Can phrase indexing help to process non-phrase queries?” CIKM, 2008, pp. 679–688.

[15] R. Song, M. J. Taylor Y. Yu, , J. R. Wen, H. Hon, “Viewing term proximity from a different perspective,” ECIR, 2008, pp. 346–357