An Empirical Model of Clustering for Rank Oriented Results

[1]D. Jyothibhaskar, [2]Ch.Sunil
[1]*M.Tech Scholar,*[2]*Assisstant. Professor*
[1,2]*Department of CSE, Kaushik College of Engineering, Visakhapatnam, Andhra Pradesh, India.*

Abstract : Searching user interesting results in search engines is stiall an important research issue in the field of knowledge and data engineering, Even though various approaches available for finding the the results for user query they are m may not be optimal. Inthis paper we are proposing an efficient file relevance score mechanism followed by the clustering mechanism, In the clustering approach we cluster the retrieved documents based on the time relevance.

I.INTRODUCTION

TIME is an important dimension of relevance for a largenumber of searches, such as over blogs and news archives. So far, research on searching over such collectionshas largely focused on retrieving topically similar documentsfor a query. Unfortunately, ignoring or not fully exploitingthe time dimension can be detrimental for a large family ofqueries for which we should consider not only the documenttopical relevance but the publication time of the documentsas well .

For recency queries [2], the bulk of the relevant documents is,by definition, from recent days. For other families of queries,the relevant documents may be distributed differently overthe time span of a news archive. For example, the query[Madrid bombing] (Fig. 1) executed on a news archive mightbe after articles about the specific details of the Madrid train bombing at the time it happened, so this query might be considered a past query. More generally, relevant results forsome queries may exist in certain time

periods, in whichsudden, large-scale news coverage relevant to the queriestakes place and diminishes after a period of time. Otherqueries,

such as [Barack Obama], are likely to be after relevantresults from multiple "events."In addition to the temporal features presented above, we also useten entity-based features aimed at measuring the similarity betweena query and a document. The intuition is that a traditional termmatchingmethod that use only statistics, e.g., TFIDF, ignores thesemantic role of a query term. For example, consider the temporalquery Iraq 2001. A statistics-based model will rank a document having many occurrences of the terms Iraq or 2001 *higher than*a document with less frequency of the same terms without takinginto account *a semantic relationship* between query terms, whichcan be determined by, e.g., a term distance in a sentence.Entity-based features are computed for each entity *ej*in an annotated document ˆ *di*, and the proposed features includes *querySim*,*title*, *titleSim*, *senPos*, *senLen*, *cntSenSubj*, *cntEvent*, *cntEventSubj*,*timeDist*, and *tagSim*[13]. The first feature *querySim*is the termsimilarity score between *qj*and an entity *ej*in ˆ *di*. Here, we useJaccard coefficient for measuring term similarity. Feature *title* indicateswhether *ej*is in the title of *di*. Feature *titleSim*is the term similarity score between *ej*and the title. Feature *senPos*gives anormalized score of the position of the 1*st* sentence where *ej*occursin *di*, while the feature *senLen*gives a normalized score ofthe length of the 1*st* sentence of *ej*. Feature *cntSenSubj*is a normalizedscore of the number of sentences where *ej*is a subject.Feature *cntEvent*is a normalized score of the number of event sentences(or sentences annotated with temporal expressions) of *ej*, while the feature *cntEventSubj*a normalized score of the numberof event sentences that *ej*is a subject. Feature *timeDist*is a normalizeddistance score of *ej*and a temporal expression within asentence. Feature *tagSim*is the term similarity score between *ej*and an entity tagged in *di*. Note that the last feature is only applicablefor a document collection provided with tags (e.g., the NewYork Times Annotated Corpus).

II. RELATED WORK

A number of ranking models exploiting temporal informationhave been proposed, including [2, 7, 16, 18]. In [16], Li andCroft incorporated time into language models, called time-basedlanguage models, by assigning a document prior using an exponentialdecay function of a document creation date. They focused onrecency queries, where the more recent documents obtain higherprobabilities of relevance. In [7], Diaz and Jones also used documentcreation dates to measure the distribution of retrieved documentsand create the temporal profile of a query. They showed thatthe temporal profile together with the contents of retrieved documentscan improve average precision for the query by using a setof different features for discriminating between temporal profiles.Berberich et al. [2] integrated temporal expressions into querylikelihood language modeling, which considers uncertainty inherent to temporal expressions in a query and documents, i.e., temporalexpressions can refer to the same time interval even if theyare not exactly equal. Metzler et al. [18]

considered implicit temporalinformation needs. They proposed mining query logs andanalyze query frequencies over time in order to identify stronglytime-related queries. Moreover, they presented a ranking concerningimplicit temporal needs, and the experimental results showedthe improvement of the retrieval effectiveness of temporal queriesfor web search.

Ranking Function In information retrieval, a rankingfunction is used to calculate relevance scores of matching
files to a given search request. The most widely usedstatistical measurement for evaluating relevance scorein the information retrieval community uses the TF_ IDF rule, where TF (term frequency) is simply thenumber of times a given term or keyword (we willuse them interchangeably hereafter) appears within a
file (to measure the importance of the term within theparticular file), and IDF (inverse document frequency) isobtained by dividing the number of files in the wholecollection by the number of files containing the term (tomeasure the overall importance of the term within the
whole collection). Among several hundred variations ofthe TF _ IDF weighting scheme, no single combinationof them outperforms any of the others universally [15].Thus, without loss of generality, we choose an exampleformula that is commonly used and widely seen in theliterature (see Chapter 4 in [7]) for the relevance scorecalculation in the following presentation. Its definition isas follows:

III.PROPOSED WORK

In this paper we are proposing an integrated approach of user search results with file relevance score with the basic factors term frequency(TF) that indicates the number of occurrences of the document and (IDF) i.e number of occurrences of the keyword with respect to the all the documents along with their time stamps and clustering. Initial phase involve file relevance score and second phase involves the clustering approach. We proposed a novel file relevance score measurement with number of terms in the file, number of occurrences of the term (term frequency) and number of files

relevance_Scores[j] = Convert.ToDecimal((1 / termsinfile[j]) * (1 + Math.Log(termfreqs[j])) * Math.Log(1 + (filecount / numberoffiles)));

Ranking function calculates the term frequency and inverse document frequency for finding the score of the query or keyword with respect to the files, and forwards the datasets according to the score to the user based on ranking.

Files can be retrieved based on the our novel file relevance scores

Clustering based on the measure time stamp, it groups the similar type of objects based on the time stamp, the following step by step procedure illustrates as follows

Step1: Read the all the documents which are retrieved based on file relevance score

Step2: Select a random centroid from the total documents (time stamp)

Step3: Compute the Euclidean distance between the centroid and the other documents

Step4: Compute until no changes made or upto user specified iterations

Step5: Return the optimal results

In the above architecture user enters the query and server process the query, Initailly it finds the term frequency(Number of occurrences of a keyword in a document ,Inverse document frequency that indicates the number of occurrences of the keywords in whole documents and Total number of documents, these final results forwarded to clustering process based on the time stamp

In clustering process initial ,it receives the number of clusters as input parameter then randomly select the k number of centroids from the retrieved results, now calculates the Euclidean distance between the centroid and all the documents with respect to the time stamps, continue the process until a maximum number of user specified iterations or until no changes made in the clusters.

IV.CONCLUSION

We are concluding our research work with efficient fiile relevance score and clustering mechanism for user interesting search results for given query, our approach gives the optimal solution with term frequency and with inverse document frequencies with file relevance score and to group the similar type objects with time stamps

REFERENCES

[1] R. Jones and F. Diaz, “Temporal Profiles of Queries,” ACM Trans.Information Systems, vol. 25, no. 3, article 14, 2007.

[2] X. Li and W.B. Croft, "Time-Based Language Models," Proc. 12th ACMConf. Information and Knowledge Management (CIKM '03), 2003.
[3] D. Metzler and W.B. Croft, "Combining the Language Model andInference Network Approaches to Retrieval," Information Processingand Management, vol. 40, no. 5, pp. 735-750, Sept. 2004.
[4] S.E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M.Lau, "Okapi at TREC," Proc. Fourth Text REtrieval Conf. (TREC-4),1994.
[5] S.E. Robertson, "Overview of the Okapi Projects," J. Documentation,vol. 53, no. 1, pp. 3-7, 1997.
[6] K.S. Jones, S. Walker, and S.E. Robertson, "A Probabilistic Modelof Information Retrieval: Development and Comparative Experiments- Part 1," Information Processing and Management, vol. 36,no. 6, pp. 779-808, 2000.
[7] I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes:Compressing and indexing documents and images," MorganKaufmann Publishing, San Francisco, May 1999.
[8] D. Song, D. Wagner, and A. Perrig, "Practical techniques forsearches on encrypted data," in Proc. of IEEE Symposium onSecurity and Privacy'00, 2000.
[9] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report2003/216, 2003, http://eprint.iacr.org/.
[10] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Publickey encryption with keyword search," in Proc. of EUROCRYP'04,volume 3027 of LNCS. Springer, 2004.
[11] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keywordsearches on remote encrypted data," in Proc. of ACNS'05, 2005.
[12] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchablesymmetric encryption: improved definitions and efficientconstructions," in Proc. of ACM CCS'06, 2006.
[13] A. Singhal, "Modern information retrieval: A brief overview,"IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.
[14] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Orderpreservingsymmetric encryption," in Proc. of Eurocrypt'09, volume5479 of LNCS. Springer, 2009.
[15] J. Zobel and A. Moffat, "Exploring the similarity space," SIGIRForum, vol. 32, no. 1, pp. 18–34, 1998.

BIOGRAPHIES

D. JYOTHIBHASKAR completed his MCA from Andhra University, and he is currently pursuing M.Tech in Department of CSE in Kaushik College of Engineering(Affiliated to JNTUK). His interested areas are data mining & data warehousing and Computer Networks.

Ch.Sunil is an Asst. Professor of the Department of CSE, Kaushik College of Engineering (Affiliated to JNTUK), Visakhapatnam, Andhra Pradesh, India. He obtained his M.Tech. in Computer Science & Engineering from AcharyaNagarjuna University. He is pursuing Ph.D. in Computer Science & Engineering from GITAM University, Visakhapatnam. His main research interests are Cryptography and Network Security.

