International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-2, Issue-10 E-ISSN: 2347-2693

Information Retrieval System Using Vector Space Model for Document Summarization

Vaibhav A. Chavan[1*] and Santosh R. Durugkar[2]

[1*2] *Department of Computer Engineering, Savitribai Phule Pune University, Maharashtra, India*

www.ijcaonline.org

Received: 11 Sep 2014 Revised: 02 Oct 2014 Accepted: 14 Oct 2014 Published: 31 Oct 2014

***Abstract*—** Document summarization is the process of reducing size of text document and that retains the most important content of the original document into the reduced document(Summary).In recent year there are huge work has been done in document summarization. There are various techniques available for document summarization but most of the techniques used similarity of sentences to extract sentence, in the document summarization a context of the document are important, so our current method used term indexing model to gives index to document as well as sentences in that document. In this proposed system we used context based document indexing based on vector space model. This document indexing model works with document frequency (DF) and term frequency (TF).DF and TF model gives document indexing weight which is used for document summarization. We compare our system with traditional term based indexing model and will prove that our system gives better result than this system.

Keywords — Vector space model, Document frequency, Term Frequency, Document context

I. INTRODUCTION

Text summarization is the process of automatically creating a compressed version of a given document pre-serving its information content. Automatic document summarization is an important research area in natural language processing (NLP). The technology of automatic document summarization is developing and may provide a solution to the information overload problem. Modern text retrieval systems principally rely on orthographic, semantic, and statistical analysis. The usual approach is to use white space to identify word boundaries, followed by stemming to conflate words with similar surface forms into a common term. A weight is then computed for each term in every document using the frequency of the term in the document, the selectivity of the term, and the length of the document. In vector space text retrieval, queries are represented in a manner similar to the documents, and the similarity of each document in the collection to the query is then computed as the normalized inner product of the document and query term weight vectors. In probabilistic text retrieval, a term weight is treated as the probability of relevance of a document to a query, conditioned on the presence of that term in the query. Probabilistic and vector space techniques are often combined with Boolean text retrieval, in which the presence or absence of a term or combination of terms can be explicitly required in the query specification. The principal advantage of vector space and probabilistic text retrieval over a purely Boolean approach is that lists of documents that are ranked in order of decreasing probability of relevance allow users to interactively decide how many documents are worth examining. Unranked Boolean techniques, on the other hand, might be preferred when no user interaction is possible before the next processing stage. In either case, when the document collection is relatively stable it is common to preprocess the collection to produce an index structure on the feature set that can be searched in sub-linear time. The utility of a text retrieval system depends strongly on how well the query is constructed, and that depends in turn on how well the user understands the collection and the way in which the indexed features can be used to select documents. It is usually fairly straightforward to find some relevant documents, but interactive inspection by the user is generally needed if the relevant documents must be more carefully separated from the irrelevant ones. An iterative query reformulation process such as Simulated Nucleation can be used to speed this process, leveraging inspection of a few documents to produce a query that better separates relevant and irrelevant documents.

Corresponding Author: *Vaibhav A*

II. LITERATURE SURVY

Text summarization can either be “abstractive” or “extractive.” The abstraction-based models mostly provide the summary by sentence compression and reformulation allowing summarizers to increase the overall information without increasing the summary length. However, these models require complex linguistic processing. Sentence extraction models, on the other hand, use various statistical features from the text to identify the most central sentences in a document/set of documents. Erkan and Radev proposed LexRank to compute sentence importance based on the concept of eigenvector centrality and degree centrality. They used the hypothesis that the sentences that are similar to many of the other sentences in a cluster are more salient to the document topic. Sentence similarity measures based on cosine similarity was exploited for computing the adjacency matrix. Once the document graph is constructed using the similarity values, the “degree centrality” of a sentence si are defined as the number of sentences similar to si, with similarity value above a threshold. Eigenvector centrality is computed using the

LexRank algorithm iteratively, which was an adaptation of the PageRank algorithm. Mihalcea and Tarau proposed TextRank, another iterative graph-based ranking framework for text summarization and showed that other graph-based algorithms can be derived from this model. None of the models, as described in this section, address the problem of "context insensitive document indexing." A propose system which uses the knowledge derived from the underlying corpus to give a context-sensitive indexing weight to the document terms. Sentence similarity will be calculated using the indexing weights thus obtained.

III. EXISTING SYSTEM

Existing methods for single document keyphrase extraction [3] usually make use of only the information contained in the specified document. This study proposes to construct an appropriate knowledge context for a specified document by leveraging a few neighbor documents close to the specified document. The neighborhood knowledge can be used in the keyphrase extraction process and help to extract salient keyphrases from the document. In particular, the graph-based ranking algorithm is employed for single document keyphrase extraction by making use of both the word relationships in the specified document and the word relationships in the neighbor documents, where the former relationships reflect the local information existing in the specified document and the latter relationships reflect the global information existing in the neighborhood. The framework for the system described in [3] is as follows:

i. Neighborhood Construction: Expand the specified document d0 to a small document set D= {d0, d1, d2…dk} by adding k neighbor documents. The neighbor documents d1, d2… dk can be obtained by using document similarity search techniques;

ii. Keyphrase Extraction: Given document d0 and the expanded document set D, perform the following steps to extract keyphrases for d0:

a) Neighborhood-level Word Evaluation: Build a global affinity graph G based on all candidate words restricted by syntactic filters in all the documents of the expanded document set D, and employ the graph-based ranking algorithm to compute the global saliency score for each word.

b) Document-level Keyphrase Extraction: For the specified document d0, evaluate the candidate phrases in the document based on the scores of the words contained in the phrases, and finally choose a few phrases with highest scores as the keyphrases of the document.

It is noteworthy that the proposed approach has higher computational complexity than the baseline approach because it involves more documents, and we can improve its efficiency by collaboratively conducting single document keyphrase extractions in a batch mode. But the focus on more test data was lacking compromising with the robustness of the system.

Xiaojun Wan [4], proposed a novel unified approach to simultaneous single-document and multi-document summarization by making using of the mutual influences between the two tasks. Experimental results on the benchmark DUC datasets show the effectiveness of the proposed approach. Given a document set, in which the whole document set and each single document in the set are required to be summarized, we use *local saliency* to indicate the importance of a sentence in a particular document, and use *global saliency* to indicate the importance of a sentence in the whole document set.

TextRank demonstrated [5] is a system for unsupervised extractive summarization that relies on the application of iterative graph based ranking algorithms to graphs encoding the cohesive structure of a text. The distinguishing characteristics of the proposed system is that it does not rely on any language-specific knowledge resources or any manually constructed training data, and thus it is highly portable to new languages or domains. It is shown by the author that iterative graph-based ranking algorithms work well on the task of extractive summarization since they do not only rely on the local context of a text unit (vertex), however it takes the information recursively drawn from the entire text (graph) into account. [6] proposes two enhancements to the above work investigated earlier by adding two more features to the existing one. Firstly, discounting approach was introduced to form a summary which ensures less redundancy among sentences. Secondly, position weight mechanism has been adopted to preserve importance based on the position they occupy. They investigated in depth, two graphical methods for multi document summarization namely SentenceRank (threshold) and SentenceRank (Continuous). In each case, discounting methods proposed by us are found to be superior as compared to their basic methods and the proposed SentenceRank methods which is a combination of discounting technique along and position weight is investigated to be the best.

Multiple document summarizations have been widely studied recently. The summary can be either generic or query specific. In a generic summary generation, the important sentences from the document are extracted and the sentences so extracted are arranged in the appropriate order. In a query specific summary generation, the sentences are scored based on the query given by the user. The highest scored sentences are extracted and presented to the user as a summary. Following are the two broad level classifications of text summarization techniques. Extractive summarization and abstractive summarization. Extractive summarization usually ranks the sentences in the documents according to their scores calculated by a set of predefined features, such as term frequency inverse sentence frequency (TF-ISF), sentence or term position, and number of keywords. Abstractive summarization involves information fusion, sentence compression and reformulation.

Early work in summarization dealt with single document summarization where systems produced a summary of one document, whether a news story, scientific article, broadcast show, or lecture. As research progressed, a new type of summarization task emerged: multi-document summarization. Multi-document summarization was motivated by use cases on the web. Given the large amount of redundancy on the web, summarization was often more useful if it could provide a brief digest of many documents on the same topic or the same event. In the first deployed online systems, multi-document summarization was applied to clusters of news

articles on the same event and used to produce online browsing pages of current events. A short one paragraph summary is produced for each cluster of documents pertaining to a given news event, and links in the summary allow the user to directly inspect the original document where a given piece of information appeared.

Approaches presented so far are examples of pure techniques to apply, in order to develop summarization systems. The predominant tendency in current systems is to adopt a hybrid approach and combine and integrate some of the techniques mentioned before (e.g. cue phrases method combined with position and word frequency based methods in [24], or position, length weight of sentences combined with similarity of these sentences with the headline. As we have given a general overview of the classical techniques used in summarization and there is a large number of different techniques and systems, we are going to describe in this section only few of them briery, considering systems as wholes.

There are two limitations with most of the existing multi-document summarization methods:

i. They work directly in the sentence space and many methods treat the sentences as independent of each other. Although few works tries to analyze the context or sequence information of the sentences, the document side knowledge, i.e. the topics embedded in the documents are ignored.
ii. Another limitation is that the sentence scores calculated from existing methods usually do not have very clear and rigorous probabilistic interpretations. Many if not all of the sentence scores are computed using various heuristics as few research efforts have been reported on using generative models for document summarization.

Recent work in multi-document summarization has leveraged information about the topics mentioned in a collection of documents in order to generate informative and coherent textual summaries. Traditionally, MDS systems have created informative summaries by selecting only the most relevant information for inclusion in a summary. In a similar fashion, coherent summaries have been created by ordering information extracted from texts in a manner that reflects the way it was originally expressed in a source document.

In recent years, graph-based ranking methods have been investigated for document summarization, such as TextRank (Mihalcea and Tarau, 2004; Mihalcea and Tarau, 2005) and LexPageRank (Erkan and Radev, 2004). Similar to PageRank (Page et al., 1998), these methods first build a graph based on the similarity relationships between the sentences in a document and then the saliency of a sentence is determined by making use of the global information on the graph recursively. The basic idea underlying the graph-based ranking algorithm is that of "voting" or "recommendation" between sentences.

.

IV. PRAPOSED SYSTEM

In the proposed system we used vector space model for document indexing. In the vector space model document is represented by Vector of terms as follows.

- Words (or word stems)
- Phrases (e.g. computer science)
- Removes words on "stop list"

Correlations between term vectors imply a similarity between documents. For efficiency, an inverted index of terms is often stored. In the vector space model we used term frequency which count of time terms occurs in the document. The more times a term *t* occurs in document *d* the more likely it is that *t* is relevant to the document. Document frequency the more a term *t* occurs throughout all documents, the more poorly *t* discriminates between documents. The term frequency and inverse document frequency High value indicates that the word occurs more often in this document than average.

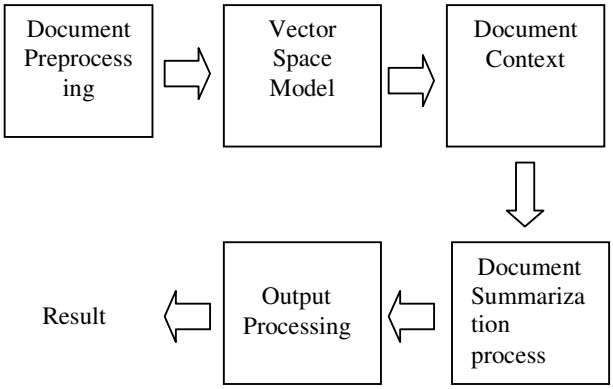

Fig 1: System Architecture

In the vector space model the document is presented as vector it having its magnitude and direction. Vector is a like as array of floating points each vector holds a place for each terms in the collection we used following mathematical model

$D_i = \mathcal{W}_{d_{i1}}, \mathcal{W}_{d_{i2}}, \ldots\ldots, \mathcal{W}_{d_{it}}$

$Q = \mathcal{W}_{q1}, \mathcal{W}_{q2}, \ldots\ldots, \mathcal{W}_{qt}$ $\mathcal{W}=0$ if a term is absent

Unormailized similarity: $sim(Q,D_i) = \sum_{j=1}^{t} \mathcal{W}_{qj} * \mathcal{W}_{d_{ij}}$

cosine: $sim(Q,D_2) = \frac{\sum_{j=1}^{t} w_{qj} * w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t}(w_{qj})^2 * \sum_{j=1}^{t}(w_{d_{ij}})^2}}$

(cosine is normalized inner product)

The tf-idf weighting scheme assigns to term t a weight in document d given by,

$$\text{tf}-\text{idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t$$

In other words, tf-idf$_{t,d}$ assigns to term *t* a weight in document *d* that is

i. Highest when *t* occurs many times within a small number of documents (thus lending high discriminating power to those documents);
ii. Lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
iii. Lowest when the term occurs in virtually all documents.

At this point, we may view each document as a vector with one component corresponding to each term in the dictionary, together with a weight for each component that is given by equation (1). For dictionary terms that do not occur in a document, this weight is zero. This vector form will prove to be crucial to scoring and ranking. As a first step, we introduce the overlap score measure: the score of a document *d* is the sum, over all query terms, of the number of times each of the query terms occurs in *d*. We can refine this idea so that we add up not the number of occurrences of each query term *t* in *d*, but instead the tf-idf weight of each term in *d*.

$$\text{Score (q, d)} = \sum_{t \epsilon q} \text{tf} - \text{idf}_{t,d}$$

A. Document Preprocessing

Document classification can be defined as the task of automatically categorizing collections of electronic documents into their annotated classes, based on their contents. In recent years this has become important due to the advent of large amounts of data in digital form. For several decades now, document classification in the form of text classification systems have been widely implemented in numerous applications such as spam filtering, e-mails categorizing, knowledge repositories, and ontology mapping, contributed by the extensive and active researches. An increasing number of statistical and computational approaches have been developed for document classification, including decision tree, rule induction, k-nearest-neighbor classification, naive Bayes classification, and support vector machines.

a) Conflation Algorithm

There are four automatic approaches. Affix removal algorithms remove suffixes and/or prefixes from terms leaving a stem. These algorithms sometimes also transform the resultant stem. The name stemmer derives from this method, which is the most common. Successor variety stemmers use the frequencies of letter sequences in a body of text as the basis of stemming. The n-gram method conflates terms based on the number of diagrams or n-grams they share. Terms and their corresponding stems can also be stored in a table. Stemming is then done via lookups in the table. There are several criteria for judging stemmers: correctness, retrieval effectiveness, and compression performance. There are two ways stemming can be incorrect–overstemming and understemming. When a term is overstemmed, too much of it is removed. Overstemming can cause unrelated terms to be conflated. The effect on IR performance is retrieval of nonrelevant documents. Understemming is the removal of too little of a term. Understemming will prevent related terms from being conflated. The effect of understemming on IR performance is that relevant documents will not be retrieved. Stemmers can also be judged on their retrieval effectiveness–usually measured with recall and precision and on their speed, size, and so on. Finally, they can be rated on their compression performance

Conflation Algorithm in simple steps:

i. Open and read each input file and create a single index file.
ii. Remove or filer out all stop words.
iii. Remove all suffixes/affixes from each word if present.
iv. Count frequencies of occurrences for each root word from 3.
v. Apply porter's rules/algorithm for each root word from 3 and store in index file.

B. Construction of Vector space model

The vector space model procedure can be divided in to three stages. The first stage is the document indexing where content bearing terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user. The last stage ranks the document with respect to the query according to a similarity measure. The vector space model has been criticized for being ad hoc.

C. Calculating Document Context

Represents a position in a source file. For languages where the source file may not be present, a document context identifies a position in a document typically generated by the run-time environment. For example, a scripting engine might generate a document from script. For more information, see document Position. Describes a position in a source document that corresponds to a code context. The symbol handler maps a code context to documentation context, using information generated by a compiler or interpreter.

D. Document Summarization Based On The Vector Space Model Results

In the sixties, a large amount of scientific papers and books have been digitally stored. However, the storage media to store such a large database was very expensive. Therefore the concept of automatic shortening of texts was introduced to store the information about papers and books in limited storage space. Now, due to advancement in technology, the storage media are no longer expensive and bulk of information can be fit into the large databases these days. But due to increased use of the Internet, and large amount of information available on the web, there is a need to represent each document by its summary to save time and effort for searching the correct information. Automatic document summarization is extremely helpful in tackling the information overload problems. It is the technique to identify the most important pieces of information from the document, omitting irrelevant information and minimizing details to generate a compact coherent summary document.

V. CONCLUSION

Thus we have investigated different methods for document summarization and have proposed a novel approach using vector space model for context- based document summarization. This document indexing model works with the document frequency and term frequency. The concept of using vector space model was used to modify the indexing weights of the document terms.

Analysis of some of the documents and the corresponding summary figured out the specific advantage offered by the proposed vector space model-based context sensitive indexing. Vector space model provide better summary based on context of sentences than other summarization method.

REFERENCES

[1] X. Wan and J. Xiao, "Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction," ACM Trans. Information Systems, vol. 28, pp. 8:1-8:34, http://doi.acm.org/10.1145/1740592.1740596, June 2010.

[2] K.S. Jones, "Automatic Summarising: Factors and Directions," Advances in Automatic Text Summarization, pp. 1-12, MIT Press, 1998.

[3] L.L. Bando, F. Scholer, and A. Turpin, "Constructing Query-Biased Summaries: A Comparison of Human and System Generated Snippets," Proc. Third Symp. Information Interaction in Context, pp. 195-204, http://doi.acm.org/10.1145/1840784. 1840813, 2010.

[4] X. Wan, "Towards a Unified Approach to Simultaneous Single- Document and Multi-Document Summarizations," Proc. 23rd Int'l Conf. Computational Linguistics, pp. 1137-1145, http://portal. acm.org/citation.cfm?id=1873781.1873909, 2010.

[5] X. Wan, "An Exploration of Document Impact on Graph-Based Multi-Document Summarization," Proc. Conf. Empirical Methods in Natural Language Processing, pp. 755-762, http://portal.acm.org/ citation.cfm?id=1613715.1613811, 2008.

[6] Q.L. Israel, H. Han, and I.-Y. Song, "Focused Multi-Document Summarization: Human Summarization Activity vs. Automated Systems Techniques," J. Computing Sciences in Colleges, vol. 25, pp. 10-20, http://portal.acm.org/citation.cfm?id=1747137. 1747140, May 2010.

[7] C. Shen and T. Li, "Multi-Document Summarization via the Minimum Dominating Set," Proc. 23rd Int'l Conf. Computational Linguistics, pp. 984-992, http://portal.acm.org/citation.cfm?id= 1873781.1873892, 2010.

[8] X. Wan and J. Yang, "Multi-Document Summarization Using Cluster-Based Link Analysis," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 299-306, http://doi.acm.org/10.1145/1390334.1390386, 2008.

[9] D. Wang, T. Li, S. Zhu, and C. Ding, "Multi-Document Summarization via Sentence-Level Semantic Analysis and Symmetric Matrix Factorization," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 307-314, http://doi.acm.org/10.1145/1390334.1390387, 2008.

[10] S. Harabagiu and F. Lacatusu, "Using Topic Themes for Multi- Document Summarization," ACM Trans. Information Systems, vol. 28, pp. 13:1-13:47, http://doi.acm.org/10.1145/1777432.1777436, July 2010.

[11] H. Daume´ III and D. Marcu, "Bayesian Query-Focused Summarization," Proc. 21st Int'l Conf. Computational Linguistics and the 44th Ann. meeting of the Assoc. for Computational Linguistics, pp. 305-312, http://dx.doi.org/10.3115/1220175.1220214, 2006.

[12] D.M. Dunlavy, D.P. O'Leary, J.M. Conroy, and J.D. Schlesinger, "QCS: A System for Querying, Clustering and Summarizing Documents," Information Processing and Management, vol.43, pp.1588-1605, http://portal.acm.org/citation.cfm?id=1284916. 1285163, Nov. 2007.

[13] R. Varadarajan, V. Hristidis, and T. Li, "Beyond Single-Page Web Search Results," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 3, pp. 411-424, Mar. 2008.

[14] L.-W. Ku, L.-Y. Lee, T.-H. Wu, and H.-H. Chen, "Major Topic Detection and Its Application to Opinion Summarization," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 627-628, http://doi.acm.org/10.1145/ 1076034.1076161, 2005.

[15] E. Lloret, A. Balahur, M. Palomar, and A. Montoyo, "Towards Building a Competitive Opinion Summarization System: Challenges and Keys," Proc. Human Language Technologies: The 2009 Ann. Conference of the North Am. Ch. Assoc. for Computational Linguistics, Companion Vol. : Student Research Workshop and Doctoral Consortium, pp. 72-77, http://portal.acm.org/citation.cfm?id= 1620932.1620945, 2009.

[16] J.G. Conrad, J.L. Leidner, F. Schilder, and R. Kondadadi, "Query- Based Opinion Summarization for Legal Blog Entries," Proc. 12th Int'l Conf. Artificial Intelligence and Law, pp. 167-176, http://doi.acm.org/10.1145/1568234.1568253, 2009.

AUTHORS PROFILE

NAME-MR.VAIBHAV ASHOK CHAVAN
CONTACT NO-9579030613
STATE-MAHARASTRA(INDIA)
HAS GOT BE IN INFORMATION TECHNOLOGY FROM DR.BABASAHEB AMBEDKAR MARATHWADA UNIVERSITY (AURANGABAD),
THE AUTHOR IS PRESENTLY WORKING ON HIS THESIS IN FOURTH SEMESTER OF HIS ME IN COMPUTER ENGINEERING IN DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, SAVITRIBAI PHULE PUNE UNIVERSITY, MAHARASTRA (INDIA)-

