“Query Dependent Document Summarization Using Clustering Techniques”

Ganesh Jorvekar [1], Prof. Manish Rai [2], Dr. Mohit Gangwar[3]
[1, 2, 3] Dept of Computer Science and Engineering
[1, 2, 3] RKDF College of Engineering, Bhopal, India

Abstract- *World Wide Web is the largest source of information. Huge amount of data is present on the Web. There has been a great amount of work on query-independent summarization of documents. However, due to the success of Web search engines query-specific document summarization (query result snippets) has become an important problem. In this paper a method to create query specific summaries by identifying the most query-relevant fragments and combining them using the semantic associations within the document is discussed. Summarization is the process of automatically creating a compressed version of a given text that provides useful information for the user. An application of document summarization is the snippets generated by web search engines for each query result. In particular, first a structure is added to the documents in the preprocessing stage and converts them to document graphs..Document understanding techniques such as document summarization have been receiving much attention these years. Current document clustering methods usually represent documents as a term document matrix and perform clustering algorithm on it. Although these clustering methods can group the documents satisfactorily, it is still hard for people to capture the meanings of the documents since there is no satisfactory interpretation for each document cluster.*

The present research work focuses on analytical study of different document clustering and summarization techniques currently the most research is focused on Query-Independent summarization. The main aim of this research work is to combine the both approaches of document clustering and query dependent summarization. This mainly includes applying different clustering algorithms on a text document. Create a weighted document graph of the resulting graph based on the keywords. And obtain the document graph to get the summary of the document. The performance of the summary usingdifferent clustering techniques will be analyzed and the optimal approach will be suggested.

I. INTRODUCTION

Large amount of data is added to the web constantly and huge amount of data is present on the Web. Users always need to search for the required information by using particular keywords. As the number of documents available on users’ desktops and the Internet increases, so does the need to provide high-quality summaries in order to allow the user to quickly locate the desired information. Summarization is the process of condensing a source text into a shorter version preserving its information content.

With the coming of the information revolution, electronic documents are becoming a principle media of business and academic information. In order to fully utilize these on-line documents effectively, it is crucial to be able to extract the gist of these documents. Having a Text Summarization system would thus be immensely useful in serving this need. In order to generate a summary, we have to identify the most important pieces of information from the document, omitting irrelevant information and minimizing details, and assembling them into a compact Coherent report.

II. TEXT SUMMARIZATION

Although there are as many different descriptions of what summarization is (or should be) as people may wish to have, there is in fact not so much disagreement in them. For example, text summarization may be described as "to reduce (long) textual information to its most essential points", "to condense information down to critical bit", or "to distill the most important information from a source or sources to produce an abridged version for a particular user (or users) and task (or tasks)" (Endres- Niggemeyer, 1998; Mani and MaybUry, 1999; Sparck-Jones, 1999). These descriptions emphasize the purpose and goal of summarization.

Text summarization can also be understood from a process point of view. Humans read an entire text and understand it before summarizing it. They "take an original article, *I* understanding, and pack it neatly into a nutshell without loss of substance or clarity '- or at least ideally so. Thus, text summarization covers both ***text understanding*** and ***text generation*** .Text understanding again is not solely a process of language processing (or text processing) based on understanding of syntax and recognition of word meanings.
A more mechanical way or computational way to look at text summarization is to see it as a ***text transformation process***.

For example, Sparck-Jones (1999) modeled text summarization as a three-stage text transformation activity that includes *interpretation, transformation and generation.* Interpretation refers to source text interpretation that analyses source text and transforms it into appropriate text representation.

Transformation refers source representation mapped into summary text representation. This involves key content representation (as key words, key concepts, significant words and significant sentence) concept organization, synthesis of an appropriate summary output and summary text representation. Generation refers to the generation of summary text from summary representation. Such a model presents a more general concept of text summarization. Viewing text summarization as text transformation activities may or may not rely on text understanding.

What is a summary?

Summary definition

"An abbreviated, accurate representation of the content of a document preferably prepared by its author(s) for publication with it." Such abstracts are also useful in access publications and machine-readable databases (American National Standards Institute Inc., 1979).

A summary as the physical output of a summarization process is the Concise and condensed description of the most important information or the main ideas in a text, with extraneous, details and repeated information omitted. A summary can serve as surrogate for the complete, unabridged version of a document. A summary can also serve as only a rough indication of the topics and major substance in a document but not as a substitute of the original (Salton; Jones). News headlines, article outlines, meeting minutes, preview of a movie, or review of a book, chronologies of salient events, abridgements of a book are some examples of very different types of summaries.
The process of producing a summary from a source text consists of the

following steps:

1. The interpretation of the text;
2. The extraction of the relevant information which ideally includes the "topics" of the source;
3. The condensation of the extracted information and construction of a summary representation;
4. The presentation of the summary representation to the reader in natural language.

While some techniques exist for producing summaries for domain independent texts (Taeho JoK 2017) it seems that domain specific texts require domain specific techniques [1] (Taeho JoK 2017). In order to address the issue of topic identification, content selection and presentation, we have studied alignments (manually produced) of sentences from professional abstracts with sentences from source documents.
The term "clustering" is used in several research communities to describe methods for grouping of unlabeled data. These communities have different terminologies and assumptions for the components of the clustering process and the context in which clustering is used. Thus, we face a dilemma regarding the scope of this survey. The production of a truly comprehensive survey would be a monumental task given the sheer mass of literature in this area. The accessibility of the survey might also be questionable given the need to reconcile very different vocabularies and assumptions regarding clustering in the various communities.

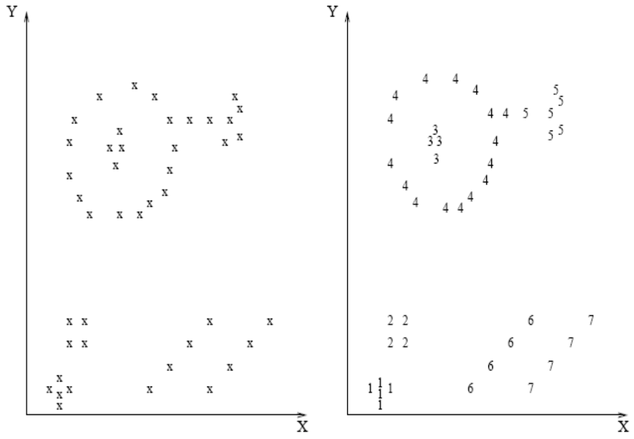


Figure 1 Data Clustering

III. CLUSTERING TECHNIQUES

Different approaches to clustering data can be described with the help of the hierarchy shown in Figure 2 (other taxonometric representations of clustering methodology are possible; ours is based on the discussion in Jain and Dubes [1988]). At the top level, there is a distinction between hierarchical and partitional approaches (hierarchical methods produce a nested series of partitions, while partitional methods produce only one).

The taxonomy shown in Figure 2 must be supplemented by a discussion of cross-cutting issues that may (in principle) affect all of the different approaches regardless of their placement in the taxonomy.

Agglomerative vs. divisive: This aspect relates to algorithmic structure and operation. An agglomerative approach begins with each pattern in a distinct (singleton) cluster, and successively merges clusters together until a stopping criterion is satisfied. A divisive method begins with all patterns in a single cluster and performs splitting until a stopping criterion is met.

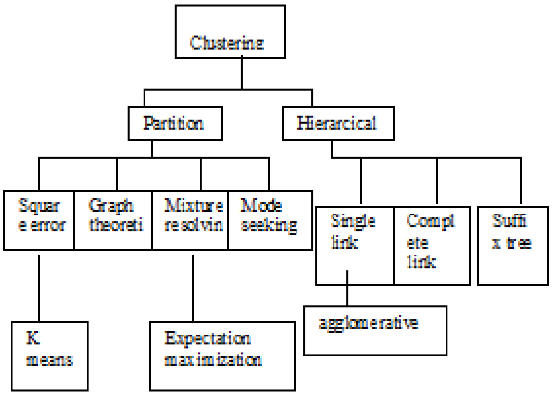


Figure 2 Clustering Techniques

Monothetic vs. polythetic: This aspect relates to the sequential or simultaneous use of features in the clustering process. Most algorithms are polythetic; that is, all features enter into the computation of distances between patterns, and decisions are based on those distances. A simple monothetic algorithm reported in Anderberg [1973] considers features sequentially to divide the given collection of patterns.

Hard vs. fuzzy: A hard clustering algorithm allocates each pattern to a single cluster during its operation and in its output. A fuzzy clustering method assigns degrees of membership in several clusters to each input pattern. A fuzzy clustering can be converted to a hard clustering by assigning each pattern to the cluster with the largest measure of membership.

Deterministic vs. stochastic: This issue is most relevant to partitional approaches designed to optimize a squared error function. This optimization can be accomplished using traditional techniques or through a random search of the state space consisting of all possible labelling.

Incremental vs. non-incremental: This issue arises when the pattern set to be clustered is large, and constraints on execution time or memory space affect the architecture of the algorithm. The early history of clustering methodology does not contain many examples of clustering algorithms designed to work with large data sets, but the advent of data mining has fostered the development of clustering algorithms that minimize the number of scans through the pattern set, reduce the number of patterns examined during execution, or reduce the size of data structures used in the algorithm's operations.

The approaches followed for document summarization is classified in many ways:

- ***Abstractive Vs Extractive :***

Abstraction involves a more in-depth analysis of the source document, condensation of its information content and generation of a summary which is cohesive and appears as if it was written by a human. At the same time, it should be able to satisfy the information need. This requires more sophisticated techniques and computational power.

Extraction techniques, on the other hand, focus on the most important in the document and perceive it in the form of words, clauses or sentences on the surface level. Complete understanding of the semantic and syntactic of the source document is not necessary. This requires lesser computational power and hence, it is more suitable for generating on-the-fly summaries.

- ***Multi-Document Vs Single-Document :***

Most of the early summarization systems were Single document summarizers. When used to summarize multiple documents which discuss about a similar topic (For example, several news articles pertaining to an event), they would process each of them individually and the resulting summaries would contain a considerable amount of repeated information as there is repetition in the source articles DUC 2005.

Whereas, a multi-document summarizer treats the whole set of documents as a single document representing a common topic. This way, the summaries will not contain repeated information.

- ***Query dependent Vs Query independent :***

Query dependent systems focus on summaries which are influenced by the query. The query is also analyzed semantically and provided as an input to the system. The summary generation process is guided by the information contained in the query.

Sumya Akter [2] in his PhD thesis dwells on the discourse structure of natural language texts citing from important work done by Ruchika Aggarwal, Latika Gupta [4] and Y.S.Deshmukh [5]. He argues that rhetorical relations between elementary textual units hold important keys to finding the relation between various units of a text. While relying on a tree like structure for finding the abstract structure of a text, Marcu explores the use of cue phrases in not only finding the structure of a text but also to evaluate the rhetorical relations between them. These rhetorical relations can be used to identify the importance of various textual units, hence their importance in text summarization. Using such rhetorical relations the most important parts in a discourse text can be found and some partial ordering can be enforced on them to select the vital parts of the text.

Anjali R. Deshpande [6] describe an approach to identify topic signatures present in text for automatic summary or for information retrieval purpose. Their approach uses a pre-classified corpus for training and thus has its own limitations. However their focused use of topic signatures for producing summaries outperforms the base lining method and tf-idf methods for extracting topic-relevant sentences for summary.

Deshmukh Yogesh S[7] incorporated a multi-document sentence trimmer into a feature based summarization system. They used trimming to pre-process documents and create multiple partially trimmed sentences as alternatives for the original sentence. The count of trimming operations done is then used as a feature in the sentence ranker. Ideally the trimmed sentences should be grammatically correct. Syntactic trimming offers three distinct advantages; firstly, a net increase in the average number of sentences per summary; second, removal on non-relevant constituents; and third, more space is created for adding relevant sentences. Error analysis has shown that while sentence compression is making space for additional sentences, more work is needed in the area of generating and selecting the right candidates.

D.M. Zajic [8] mention in their submission their modified approach for sentence splitting and sentence trimming. They remove the use of POS tagger for the sentence splitting and instead choose a conservative sentence trimming strategy which relies on a list of function words. The trimming is very conservative and the error analysis showed an error rate of less than 3% that is less than three percent of the input sentences were made ungrammatical by this trimming task.

The summarization approach discussed by Conroy, J., Schlesinger [9] is based on statistical methods. Initially the most important sentences are extracted from the source text. The extracted sentences are then joined together by analyzing their discourse structure and modifying them as required. The preprocessing step includes the following:

- Sentence reduction: Removal of extraneous phrases.
- Sentence combination: Combination of sentences based on the context.
- Syntactic transformation: Altering the grammatical structure of the sentences as required.
- Lexical paraphrasing: Replace phrases with their paraphrases.
- Reordering: The selected sentences are then reordered to make the overall summary comprehensible.

Summarization of document by clustering approach is another very used
concept.

Clustering of the Documents:
Clustering Algorithms:

K-means clustering algorithm

By using two different approaches of clustering K-means algorithm willbe used to form Cluster i.e. Hierarchical and Partitional algorithm.

Algorithms are as follows;

1. Agglomerative Hierarchical K-means
2. Square error K-means

K-Means algorithm will be used to form the related cluster.

Creating the document graph.

Each cluster becomes a node in the document graph. The *document graph G (V,E)* of a document *d* is defined as follows:

- *d* is split to a set of non-overlapping clusters *t(v)*, each corresponding to a node *v*□*V*.
- An edge *e(u,v)*□ □*E* is added between nodes *u, v*□*V* if there is an association between *t(u)* and *t(v)* in *d*.

Hence, we can view *G* as an equivalent representation of *d*, where the associations between text fragments of *d* are depicted.

A weighted edge is added to the document graph between two nodes if they either correspond to adjacent cluster node or if they are semantically related, and the weight of an edge denotes the degree of the relationship. Here two clusters are considered to be related if they share common words (not stop words) and the degree of relationship is calculated by “***Semantic parsing***”. Also notice that the edge weights are query-independent, so they can be pre-computed.

Let *D*={d1,d2,,,…,dn} be a set of documents d1,d2,,,…,dn. Also let size (di) be the length of di in number of words. Term frequency tf (d,w) of term (word) w in document d is the number of occurrences of w in d. Inverse document frequency idf (w) is the inverse of the number of documents containing term w in them.

A keyword query Q is a set of keywords Q={w1,…,wm}. A key component is the document graph G(V,E) of a document d.

Notice that *Q* is only used in ***assigning weights to the nodes*** of ***G and not for assigning weights to the edges,*** which is a desirable property since the rest of *G* can be computed before queries arrive.

ADDING WEIGHTED EDGES TO THE DOCUMENT GRAPH

(Note: Adding weighted edge is query independent)

The following input parameters are required at the pre computation stage to create the document graph:

1**. *Threshold for edge weights*.** Only edges with weight not below *threshold* will be created in the document graph. (A threshold is user configurable value that controls the formation of edges)

*2. **Minimum text fragment size***. This is used when a fragment is too long, which would lead to large nodes (text fragments) and hence large summaries. Users typically desire concise and short summaries.

Adding weighted edge is the next step after generating document graph. Here for each pair of nodes *u*,*v* we compute the association degree between them, that is, the score (weight) *EScore(e)* of the edge *e(u,v).* If *Score(e)≥threshold*, then *e* is added to *E*. The score of edge *e(u,v)* where nodes *u*, *v* have text fragments *t(u), t(v)* respectively is:

$$\text{EScore} = \frac{\sum_{w \in (t(u) \cap t(v))} ((\, tf(t(u),w) + tf(t(v),w))\ .idf(w)\ ))}{size(t(u)) + size(t(u))}$$

where *tf(d,w)* is the number of occurrences of *w* in *d*, *idf(w)* is the inverse of the number of documents containing *w*, and *size(d)* is the size of the document (in words).That is, for every word *w* appearing in both text fragments we add a quantity equal to the *tf*□ *idf* score of *w.* Notice that stop words are ignored.

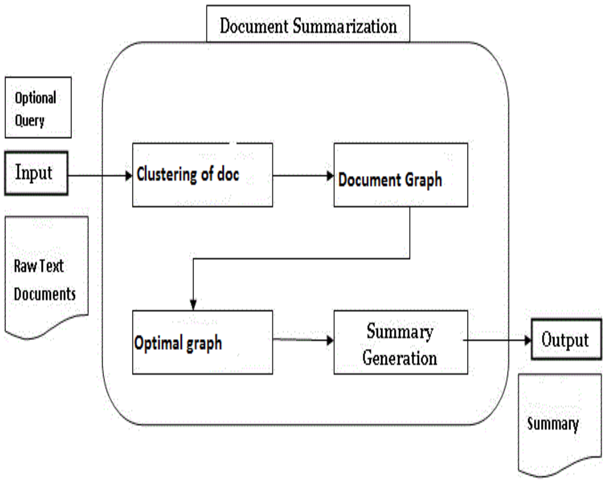
ADDING WEIGHT TO NODES IN DOCUMENT GRAPH

When a query *Q* arrives, the nodes in *V* are assigned query-dependent weights according to their relevance to *Q*. In particular, we assign to each node *v c*orresponding to a text fragment *t(v)* node score *NScore(v)* defined by the Okapi formula as given below (Equation 2).

$$\sum_{t \in Q,d} \ln \frac{N - df + 0.5}{df + 0.5} \; \frac{(k1 + 1)tf}{(k1(1-b)+b \frac{dl}{avdl}) + tf} \; \frac{(k3+1)\, qtf}{k3 + qtf}$$

tf is the term’s frequency in document,
qtf is the term’s frequency in query,
N is the total number of documents in the collection,
df is the number of documents that contain the term,
dl is the document length (in words),
avdl is the average document length and
k1 (between 1.0–2.0), b (usually 0.75), and k3 (between 0–1000) are constants.

IV. SYSTEM ARCHITECTURE



Application or Benefits

a. Search Engines: summarize the information in hit lists retrieved by search engines.

b. Meeting Summarization: find out what happened at the conference I missed.
c. Hand-held devices: create a screen-sized summary of a book.
d. Aids for the Handicapped: compact the text and read it out for a blind.

V. CONCLUSION

The main aim of this research work is to combine the both approaches of document clustering and query dependent summarization. The main constraints considered in this work can be outlined as shown below-

- The proposed work will be mainly focused on summarization of text files (i.e. .txt).
- The proposed work will be limited to clustering of text files of Standard files related to the topic popular amongst researchers will be used.
- Standard performance evaluation metrics will be used to validate performance.

REFERENCES

[1] Taeho JoK ,Nearest Neighbor for Text Summarization using Feature Similarity, 2017 International Conference on Communication, Control,
Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan, IEEE 2017.

[2] Sumya Akter , Aysa Siddika Asa , Md. Palash Uddin3 , Md. Delowar Hossain , Shikhor Kumer Roy , and Masud Ibn Afjal , An Extractive Text Summarization Technique for Bengali Document(s) using K-means Clustering Algorithm, IEEE 2017.

[3] Ruchika Aggarwal, Latika Gupta, AUTOMATIC TEXT SUMMARIZATION, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 6, Issue. 6, June 2017, pg.158 – 167.

[4] JUNG SONG LEE , HAN HEE HAHM , SOON CHEOL PARK, Less-redundant Text Summarization using Ensemble Clustering Algorithm based on GA and PSO, WSEAS TRANSACTIONS on COMPUTERS, 2017.

[5] Y.S.Deshmukh, R.D.Chintamani, S.T.Kolhe, S.S.Jore, “Query Dependent Multi-Document Summarization using Feature based and Cluster based Method”, International Journal of Electrical, Electronics and Computer Systems (IJEECS), Volume -2, Issue-10 2014.

[6] Anjali R. Deshpande , Lobo L. M. R. J. , Text Summarization using Clustering Technique, International Journal of Engineering Trends and Technology (IJETT) - Volume4 Issue8- August 2013

[7] Deshmukh Yogesh S., Waghmare Arun I. , Analysis of Clustering Techniques for Query Dependent Text Document Summarization. International Journal of Engineering science and Innovation Technology.
Volume 2, Issue 2 , March 2013.

[8] D.M. Zajic, B. Dorr, J. Lin, R. Schwartz, Sentence Compression as a Component of a Multi-Document Summarization System , Proceedings of the Document Understanding Conference, 2006.

[9] Conroy, J., Schlesinger, J., O'Leary, D., & Goldstein, J. Back to basics: CLASSY 2006. In Proceedings of the 2006 document understanding conference (DUC 2006) at HLT/NAACL 2006, New York, NY, 2006.

[10] M. Brunn, Y. Chali, C.J. Pinchak. Text Summarization Using Lexical Chains. In Workshop on Text Summarization, ACM SIGIR Conference. September 13-14, 2001, New Orleans, Louisiana USA

[11] M. Brunn, Y. Chali, C.J. Pinchak. Text Summarization Using Lexical Chains. In Workshop on Text Summarization, ACM SIGIR Conference. September 13-14, 2001, New Orleans, Louisiana USA

[12] T. Hofmann. Probabilistic latent semantic analysis. In Proceedings of the 15th Conference on Uncertainty in AI, pages 289–296, 1999