INTRODUCTION

**Information retrieval** (**IR**) is the activity of obtaining [information system](https://en.wikipedia.org/wiki/Information_system) resources relevant to an information need from a collection. Searches can be based on [full-text](https://en.wikipedia.org/wiki/Full-text_search) or other content-based indexing. Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for [metadata](https://en.wikipedia.org/wiki/Metadata) that describe data, and for databases of texts, images or sounds.

Many Information retrieval algorithms exist such as set theoretic models such as Standard Boolean model,extended boolean model,Algebraic models such as Vector space model,Topic-based vector model,Latent semantic indexing and probabilistic models such as Binary independence model,Latent Dirichlet Allocation and okapi BM25. But their speed of retrieving documents decreases as the size of documents increases. Hence we proposed an approach where we use pre-computed summaries in the process of information retrieval instead of whole document.

According to our approach, whenever a document is added to the collection,its summary should be generated.Summarization techniques are of two types:

1) Extractive summarisation: The extractive text summarization technique involves pulling keyphrases from the source document and combining them to make a summary. The extraction is made according to the defined metric without making any changes to the texts.

2)Abstractive summarization: The abstraction technique entails paraphrasing and shortening parts of the source document. When abstraction is applied for text summarization in deep learning problems, it can overcome the grammar inconsistencies of the extractive method.The abstractive text summarization algorithms create new phrases and sentences that relay the most useful information from the original text — just like humans do.

Although there is text rank in Gensim for text summarisation, the main problem is if the document size is so big, then the graph constructed from each document can be so large and cannot be even fit into the Main Memory. The another disadvantage is, in gensim text rank there may be loss of information after summarisation because gensim text rank gives more weightage to frequently repeating sentences and words. Considering these challenges in gensim tect rank we first divide our large size documents into smaller partitions and parallel summarise these partitions so that the graph obtained in text rank can fit into the main memory and because of considering the sub-topics in the document there will be little loss of information after summarisation.

The work in this thesis only focuses on full text information retrieval. In [text retrieval](https://en.wikipedia.org/wiki/Text_retrieval), **full-text search** refers to techniques for searching a single [computer](https://en.wikipedia.org/wiki/Computer)-stored [document](https://en.wikipedia.org/wiki/Document) or a collection in a [full-text database](https://en.wikipedia.org/wiki/Full-text_database). Full-text search is distinguished from searches based on [metadata](https://en.wikipedia.org/wiki/Metadata)or on parts of the original texts represented in databases (such as titles, abstracts, selected sections, or bibliographical references).

In a full-text search, a [search engine](https://en.wikipedia.org/wiki/Search_engine) examines all of the words in every stored document as it tries to match search criteria (for example, text specified by a user). Full-text-searching techniques became common in online [bibliographic databases](https://en.wikipedia.org/wiki/Bibliographic_databases) in the 1990s. Many websites and application programs (such as [word processing](https://en.wikipedia.org/wiki/Word_processing) software) provide full-text-search capabilities. Some web search engines, such as [AltaVista](https://en.wikipedia.org/wiki/AltaVista), employ full-text-search techniques, while others index only a portion of the web pages examined by their indexing systems.[

CHALLENGES :

1. Handling large documents.
2. Summarising without information loss.
3. Maintaining the accuracy of the output(with summarisation) compared to the output of the original documents (without summarisation).
4. Improving the efficiency of information retrieval on large documents with respect to time.

ADDRESSING THE CHALLENGES:

1. To handle large documents we want to partition the documents into small chunks topic wise, for which we are using text-tiling.
2. Text Summarisation is done on the chunks of data (generated as above).As these chunks are partitioned according to their topics, we are ensuring less or no loss of information.
3. As described above there is very less information loss even after summarisation, this shows that our accuracy is nearly equal to that of information retrieval on original documents.
4. To improve the efficiency with respect to time we want to parallelise the text summarisation task which reduces the overhead for generating summaries with respect to time and this overhead in time for generating the summaries is compensated when using these summaries for information retrieval.

METHODOLOGY:

The detailed methodology for addressing the above mentioned challenges is as follows (flow is shown in fig 1.

Document partitioning:

To partition the large sized text documents into small chunks according to their topics, we are using text tiling. The detailed explanation about text tiling is given in the literature review.

The output from the text tiling algorithm will be the partition boundaries i.e the topic segmentations represented by “1”. So we separated the document accordingly (the lines between two consecutive ones as one document chunk). The parameters such as psuedosentence size (w) and block size are taken according to the results from the paper[] presented in the literature review.

Topic Modelling:

To reduce the loss of information in summarisation we are grouping the relevant chunks of documents as per their topics as one. Text summarisation is done on this group of relevant topics chunks. To find the relevance between the topic chunks, we are using LDA (Latent Dirichlet Allocation), a topic modelling algorithm (detailed description is given in the literature review).

The main challenge to be addressed in the LDA is to fix the optimal number of topics. To do this we are using coherence score.

Coherence score:

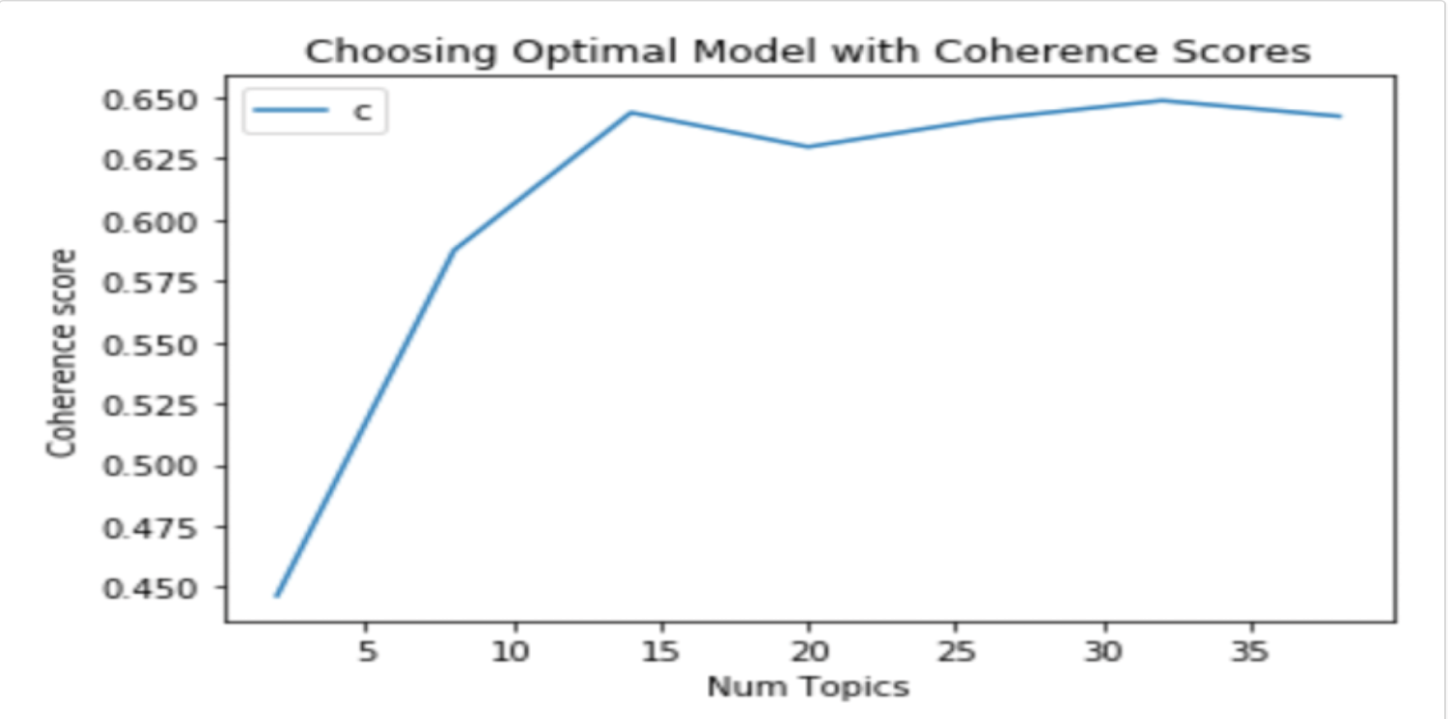
Topic coherence is a realistic measure of how good the topics produced by LDA really are. To find the optimal number of topics multiple models are constructed and then coherence score for these models are calculated. The number of topics with the best coherence score is considered as the optimal value.

Coherence =Σ scores.

Score = p(wi,wj)/p(wi).p(wj)

p(wi,wj) -probability of co-occurrence of wi,wj in a document.

p(wi) -probability of occurrence of wi in a document.



The graph above shows the relation between number of topics and coherence from which optimal number of topics is taken as 15 as it has highest coherence score.

So the input for LDA algorithm are the document chunks generated using text tiling. These chunks are considered as individual topics by the LDA algorithm. The outputs of the LDA algorithm are the document topic distribution.

Document grouping:

As described above, we are grouping the relevant document chunks. To do this we are using K-means clustering algorithm, which does position-based clustering on document chunks.We are using Jensen-Shannon distance metric as distance measure in K-means, as it is symmetric. So each document chunk is taken as a point in the K-means clustering which has dimensions as the document-topic distribution which we got from LDA topic modelling algorithm.

// Jensen-Shannon distance formula:

The optimal number of clusters is decided using elbow method.

Silhoutte score is used for evaluating the optimal number of clusters, as we do not have any training data to find distortion. We constructed a graph between number of clusters(k) and Silhoutte score, the corresponding value of the number of clusters at the elbow of the graph

Is taken as k value(optimal number of clusters). That is the number of clusters which gives highest value of Silhoutte score is considered as optimal.

//Silhoutte score formula:

Text Summarisation:

The group of relevant documents obtained from K-means clustering are given as input to the text summarisation algorithm and the summaries are generated parallely for each group.

The documents obtained from the above module will be now taken as vertices and the links between the documents are taken as weights representing the relevant words between those two vertices.We will get a highly connected graph and text rank is applied to that graph(details of the text-rank algorithm is given in the literature review).

