Heriot-Watt University

Research Report

Multi-document Summarization of online news articles

Author:
Keziya Mariam Thomas

Supervisor:
Dr. Hani Ragab

A thesis submitted in fulfilment of the requirements for the degree of MSc.Data Science

in the

School of Mathematical and Computer Sciences

April 2019

Declaration of Authorship

I, Keziya Mariam Thomas, declare that this thesis titled, 'Multi-document Summarization of online news articles' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: *Keziya Mariam Thomas*

Date: *April 3, 2019*

“Success is not final; failure is not fatal; it is the courage to continue that counts.”

Winston S. Churchill

Abstract

The increasing human-computer interaction along with overload of online data exposes users to a lot of information, all of which might not be interesting to them. Online readers should be able to control the amount of text they read before finding something that interests them. To a small extent, this can achieved by the use of headlines, title, etc., but a summary would be more helpful.

Text summarization is an application of Natural Language Processing (NLP) that is used to generate summaries from textual information. Summarization is the technique of condensing or shortening a text while preserving all the essential contents in the original version. With the increasing amount of information on the Internet, NLP's scope has widened and is gaining more importance. NLP has found application in numerous other areas like topic identification, text categorization, text normalization, etc. Many new approaches to automatic text summarization have also evolved over the past years.

In this project, we focus primarily on multi-document news summarization where a summary is derived from more than one news article, all of which report the same event. We explore several techniques that can be used to group articles by event/topic and to generate a summary for each group.

Keywords: Summarization, extractive, similarity, ranking, distance

Acknowledgements

I would like to thank Dr. Hani Ragab and Kayvan Karim for their valuable time and suggestions and for motivating me to successfully complete my research report.

I would also like to thank my friends and family who were supportive and encouraged me throughout this journey.

Contents

List of Figures

List of Tables

Abbreviations

ANES	**A**utomatic **N**ews **E**xtraction **S**ystem
ANSES	**A**utomatic **N**ews **S**ummarization and **E**xtraction **S**ystem
BoW	**B**ag of **W**ords
CBoW	**C**ontinuous **B**ag of **W**ords
CNN	**C**onvolution **N**eural **N**etwork
LCS	**L**ongest **C**ommon **S**ubsequence
LDA	**L**atent **D**irichlet **A**llocation
LSA	**L**atent **S**emantic **A**nalysis
MRW	**M**arkow **R**andom **W**alk
MUC	**M**essage **U**nderstanding **C**onference
NLP	**N**atural **L**anguage **P**rocessing
RIPTIDES	**R**ap**I**dly **P**ortable **T**ranslingual **I**nformation extraction and interactive multi-**D**ocum**E**nt **S**ummarization
RNN	**R**ecurrent **N**eural **N**etwork
ROUGE	**R**ecall **O**riented **U**nderstudy for **G**isting **E**valuation
RSS	**R**ich **S**ite **S**ummary
SCU	**S**ummarization **C**ontent **U**nits
SUMMONS	**SUMM**arizing **O**nline **N**ew**S** Articles
SVD	**S**emantic **V**alue **D**ecomposition
WMD	**W**ord **M**overs **D**istance

Chapter 1

Introduction

Text summarization is the task of producing a clear and concise summary while preserving the key information and the overall context and meaning of the original text [Allahyari et al., 2017]. With overload of online information and presence of irrelevant documents in a search result, summarization allows the users to choose whether to access an article or jump to the next one [McKeown and Radev, 1999]. Generating summaries manually for every document on the Internet is impractical owing to the vastness of online data. Automatic text summarization solves this problem by extracting relevant details from the document and creating a summary. Automatic summarization can be a challenging task as computers do not have the logical and expressive abilities that humans possess [Allahyari et al., 2017]. Despite the challenges, a number of automatic text summarization methods have been developed till date.

1.1 Aim

This project aims at generating multi-document summaries of news articles collected from various online sources. The main focus is on the use of semantic aware technologies to identify similarity between articles and between sentences in the same article. This project explores a variety of similarity measures and distance metrics that can be used for grouping and summarization of documents.

On completion of this thesis, we hope to answer the following question: How do semantic aware Machine Learning approaches perform when compared to frequency based approaches in summarizing documents?

1.2 Objectives

The project proposes to build a model that produces summary of an event from a group of online news articles that report the same event. The input to the system will be articles on different topics from different online sources. All articles on the same topic are grouped and a summary is generated for each group.

The objectives for this thesis are:

- Group articles that are related to the same topic.
- Evaluate the result of grouping.
- Generate one multi-document summary for each topic/group.
- Evaluate the quality of generated summary.
- Comparison of performance of this model with baseline methods.

1.3 Manuscript Organization

The next chapter in the report is a literature review of the technologies used in the domain of text summarization and grouping of documents. It also provides critical analysis and review of existing systems that make use of these technologies. Following that, we define the functional and non-functional requirements for this project. Then we have a chapter on the research methodology adopted for this project. Following that, we have a chapter that discusses the work done so far on this project. Towards the end of the report we have 2 chapters, one discussing the professional, legal, ethical and social issues and another laying out the entire plan for completion of this project.

Chapter 2

Literature review

In this chapter, we research, explore, critically analyse and compare the methodologies used in the domain of text summarization.

2.1 Types of summarization

Text summarization can be classified based on input type, output type and purpose Saggion and Poibeau [2013]. Based on input type, it could be single document or multi-document summarization. It could be extractive or abstractive based on the summary we derive from the text and indicative or informative depending on whether the purpose of the text is to alert or inform.

Extractive summarization is the process of generating summary by extracting sentences from the text while **abstractive summarization** generates a new brief text from the original text [Protim Ghosh et al., 2019]. Extractive summarization chooses important sections from the original text and frames a summary [Allahyari et al., 2017]. These selected sentences are not modified but concatenated to form the summary [Templeton and Kalita, 2018]. Abstractive summarization, on the other hand, analyses the text based on various Natural Language Processing (NLP) techniques and generates a more concise text that conveys the same information [Allahyari et al., 2017]. In abstractive methods, the original text is rephrased to obtain an abstract [Erkan and Radev, 2004].

A **Single-document** summarization summarises the contents of a single document while **multi-document** summarization generates summary from multiple documents, often related to the same topic [Choon et al., 2016].

An **indicative** summary alerts the user about the contents of the text while an **informative** summary gives more information about the concepts covered in that text or document [Wong, 2018].

Context based summarization

The relevance of a sentence in a text also depends on the context in which it is written. For example, web documents, scientific articles, medical documents, email conversations, etc. have different structure and characteristics that we have to consider in order to generate an efficient summary [Choon et al., 2016]. Domain-specific summarizations make use of additional information that help identify the important topics eg. comments on blogs, citations in a scientific article, etc [Allahyari et al., 2017]. Unlike domain specific summarization, a generic summarization generates summary regardless of the domain and considers all texts in the document alike [Choon et al., 2016].

2.2 History of news summarization

The concept of literary abstract was first introduced by Luhn [1958]. In this approach, a sentence is relatively significant if the presence of frequently occurring words are high and they are in close proximity in the sentence. Further down the lane, several other sentence scoring and ranking methods were developed following extensive research. In early 2000's, several graph based sentence ranking algorithms were developed such as TextRank [Mihalcea and Tarau, 2004] and LexRank [Erkan and Radev, 2004], both based on Google's PageRank algorithm [Brin and Page, 1998] which was originally designed to rank web pages.

2.3 Data Pre-processing

Data pre-processing helps us to convert raw data into a form that is appropriate for our tasks. The type of pre-processing differs with the task. Following are few pre-processing steps that could be applied to data before passing it as input to a summarizer.

- Tokenization: Tokenization is the task of splitting a document into sentences or a sentence into words using delimiters [Raju and Allarpu, 2017].
- Removal of stop words: Stop words are those words that do not add to the meaning of the text like articles, prepositions, etc [Ferreira et al., 2013]. Some text summarization methods can identify stop words based on frequency across a set of documents, in which case this step is not necessary.
- Stemming or lemmatization: Stemming maps variants of a word to the same stem word while lemmatization removes inflectional word endings after careful morphological study of the word using a dictionary like WordNet, etc [Balakrishnan and Lloyd-Yemoh, 2014]. Lemmatization also considers if the word is used as a verb or noun before modifying it.

2.4 Sentence scoring/ranking for extractive summarization

For extractive summarization, one of the main tasks is sentence scoring/ranking which decides the sentences that would appear in the final summary. Sentence extraction forms an integral part of extractive summarization. Hence, the choice of the ranking algorithm is important except for systems like 'Searchable LEAD' by MEAD Data Central which assumes that the leading portion of the body of the text can summarize the concept covered in the entire text [Wasson, 1998]. The most common and widely used approach to sentence scoring is by assigning weights to its constituent words based on factors like frequency, semantics, position of the word, similarity with the topic, etc.

The method proposed by Luhn calculated the weight of a sentence as the sum of weights of certain signature(or rare) words contained in the sentence. Later on, other methods based on frequency and probability of terms in the sentence were developed to generate more meaningful sentence ranking.

2.4.1 Feature-based sentence scoring

The importance of a sentence can also be determined by properties like its position in a paragraph or the entire document, its length, the number of title words it contains, the signature words in the sentence, etc. Edmundson [1969] proposed four feature based methods to weigh a sentence. These are:

1. Cue method: A Cue dictionary is maintained with three sub-dictionaries containing words with positive weight, negative weight and zero weight. The words and their weight are populated in each based on certain statistics. The final weight of the sentence is the sum of cue weights of its words.
2. Key method: In this method, a certain percentage of words in the text are identified as key words and their weights are calculated as its frequency over all the words in the document. The sentence weight is the sum of weights of its key words.
3. Title method: A Title glossary is created with words from title, headings, subheadings, etc and a weight is assigned to each word. The weight of a sentence is the sum of weights of words in that sentence that are in the title glossary.
4. Location method: In this method, each sentence is assigned a weight based on its position. It assumes that sentences at the beginning and end of each paragraph and those at the beginning and end of the document are more relevant when compared to others.

2.4.2 Similarity measures for sentence scoring

Most summarization methods make use of similarity measures to find the sentences that are highly correlated with the topic.

Text similarity measures

Some commonly used word-based similarity measures are:

1. Cosine similarity measures correlation between two sentences. It is the cosine angle between the vector representations of the sentences [Vijaymeena and Kavitha, 2016].

2. Euclidean distance is another common distance metrics which can also be used to determine similarity. This distance is the square root of the sum of squares of difference in elements from both the sentence vectors.

   Cosine similarity and Euclidean distance can only be applied to numerical (mostly vectoral) representation of sentences.

3. Jaccard coefficient is calculated as the number of common words in both sentences divided by the number of unique words in the sentences combined [Ferdous et al., 2009].

$$Jaccard\ coefficient = \frac{|A \cap B|}{|A \cup B|} \tag{2.1}$$

4. Dice's similarity coefficent measures similarity by dividing twice the number of common terms in the sentences by the total number of words in both the sentences [Vijaymeena and Kavitha, 2016].

The metrics mentioned above measures only the lexical similarity between the sentences i.e, they find sentences to be similar if same character sequences or word overlaps are present [Vijaymeena and Kavitha, 2016].

Frequency based representation of documents

The main concept behind frequency based measures is that words appearing more often in the text has a higher score or weight when compared to other words.

Word Probability is the frequency of the word in the document divided by the total number of words in the document [Allahyari et al., 2017].

$$P(word) = \frac{Frequency\ of\ the\ word}{Total\ number\ of\ words\ in\ the\ document} \tag{2.2}$$

Bag of words (Bow) is one of the simplest vector representations of a text. All the unique words in the document are combined to form a bag of words. The vector representation of a sentence is the list of these words with their frequency in the given sentence.

The disadvantage of traditional BoW is that it cannot identify unique words or words that define the text because it also counts the frequency of the most trivial words like and, the etc. in English language. Also, the order of words in the sentence is ignored in BoW.

Term Frequency - Inverse Document Frequency (TF-IDF) is another way of weighing words [Allahyari et al., 2017]. TF-IDF is calculated in two parts, term frequency and inverse document frequency. Term frequency is the number of times the term appeared in the document and Inverse document frequency is total number of documents divided by the number of documents that contain this term. This calculation allows lower weights to be assigned to common words that appear in most documents [Choon et al., 2016]. If a word appears with high frequency in all the documents, then these are identified as common words in the language and they are assigned a lower score. The tf-idf score of a word 'w' in a document is

$$tf-idf(w) = Frequency\ of\ w\ in\ document * \log \frac{Total\ no.\ of\ documents}{No.\ of\ documents\ containing\ w} \tag{2.3}$$

The first part of this equation represents term frequency and the second part represents inverse document frequency.

Different variants of the frequency based methods are explained in Salton and Buckley [1988].

Co-occurrence based similarity using N-grams

N-gram is the subset, of size N, of a string. This method calculates similarity based on the concept of co-occurrence [Cavnar et al., 1994]. This concept can be applied to sentences to find similarity among them based on co-occurrence of words. It is called unigram, bi-gram, tri-gram, etc. based on the value of N. The basic idea of n-gram is to assign a higher weight or score to sentences with frequent co-occurrence of words

[Ferreira et al., 2013]. The distance between two sentences can be calculated using n-grams by dividing the number of common n-grams in both by the total number of all possible n-grams [Vijaymeena and Kavitha, 2016].

Semantic similarity measures

One main disadvantage of the lexical or string based similarity measures is that it does not consider word synonyms. Thus even if two sentences mean the same thing, the distance between them will be huge and they will not be considered similar. There are two types of measures that consider the semantics while computing the similarity, namely corpus-based and knowledge-based measures [Mihalcea et al., 2006]. The corpus-based measure determines similarity between words based on information gained from the corpus while a knowledge-based measure decides the similarity based on information from semantic networks like WordNet.

Some common similarity measures that include semantics are:

1. Latent Semantic Analysis: The LSA is a corpus-based similarity measure first proposed by Landauer and Dumais [1997]. This method assumes that the context in which a word is used gives an overview of how similar it is to another word. It uses mathematical and statistical calculations for this purpose. A matrix is generated from the available text where each row is a word and each column is a paragraph in the text. The entries in the matrix is the number of times that the word appears in that paragraph or context. These entries are transformed by a function to include details about the importance of the word and the information that it carries. Following this, the dimensionlity of matrix is reduced by Single Value Decomposition (SVD). Finally, cosine similarity measure is applied to the final vector space to calculate similarity between words. The results of LSA are found to resemble human understanding of words and concepts [Landauer et al., 1998]. LSA can be used to overcome high dimensionality problems in other vector space similarity measures [Mihalcea et al., 2006].

2. Latent Dirichlet Allocation: LDA [Blei et al., 2003] is a probabilistic model that allocates each word in the document to a topic. Each document is modeled over a set of topics and each topic over a distribution of words. Identifying topic

probabilities in each sentence will help us determine similarity and therefore can be used in summarization.

3. Word Movers Distance (WMD) [Kusner et al., 2015] is a dissimilarity metric originally proposed to compute distance between text documents. This is built on top of word2vec, a word embedding in vector space [Mikolov et al., 2013]. The two word embedding models in word2vec are Continuous bag-of-words and skip-gram as described in Section 2.6.

   WMD, proposed by Kusner et al. [2015], utilises these embedding methods to represent each document as a set of embedded words. The semantic dissimilarity measure is obtained by calculating the distance between words in their embedded space where semantically similar words lie close to each other. The distance between two documents or their dissimilarity score is therefore the minimum distance required for each word in one document to travel to its matching word in the cloud of embedded words for other document. The distance is measured using Euclidean distance. Use of word embedding methods allows WMD to detect similarity even when there are no words common to both sentences and hence solves the synonym problem.

2.4.3 Graph-based sentence ranking algorithms

A graph-based sentence ranking algorithm decides the importance of each vertex (in our case, each vertex is a sentence) based on the entire graph structure. This involves recursive calculations till it converges. One such pioneer algorithms is PageRank [Brin and Page, 1998]. PageRank is an algorithm for ranking web pages but the concept was used by others to develop graph-based sentence ranking algorithms. TextRank [Mihalcea and Tarau, 2004] and LexRank [Erkan and Radev, 2004], for sentence ranking, are based on PageRank's concept.

TextRank [Mihalcea and Tarau, 2004] is an unsupervised ranking algorithm. The sentences in the text make the vertex of the graph. Each vertex is initially given an arbitrary score which is revised each time an edge is added to it. The main concept behind this model is "recommendation". This means that the importance of a vertex depends on the links with other vertices and the importance of those vertices. The computation of vertex scores is iterated until the scores converge. An advantage of this model is that it ranks all sentences in the text and therefore, can be used for generating summaries of any length [Mihalcea, 2004].

LexRank [Erkan and Radev, 2004] is a similar graph-based algorithm. The basic concept of this model is sentence centrality which is a measure of closeness of sentences to its centroid. Like TextRank, the vertices represent sentences and the edges represent similarities with other sentences. An idf-modified-cosine equation to find similarity between two sentences. This result is used to build a stochastic matrix that represents the graph. Sentence ranking is done by calculating centrality score for each sentence based on its links to other sentences. To avoid unusually high centrality score for sentences based on local information, the centrality of other sentences that are linked to this are also considered.

Markow Random Walk (MRW) is another graph-based method also inspired from PageRank algorithm [Liu et al., 2016]. Similar to the above two models, a graph is constructed with nodes representing sentences and edges representing their relationship. The weight of the edge between sentences in the graph is their cosine similarity measure. The sentence score is obtained by recursively computing the MRW saliency score which is based on the transition probability of this sentence to each of the other sentences in the graph.

2.5 Abstractive summarization

Abstractive summarization is different from extractive summarization in the way that it generates summary. Abstractive summarization techniques analyse the text using NLP methods and generate a concise text that contains the most relevant information [Allahyari et al., 2017]. Unlike extractive summarization, abstractive methods do not weigh or rank entire sentences but identify important sequences, patterns or features in the text using different similarity measures. These similar features or patterns can be used to generate templates that forms the base of the abstractive summary. Natural Language Generation (NLG) can be used to generate meaningful summaries from these templates [Dong, 2018].

2.6 Machine learning in Summarization

Machine Learning approaches require training the model to pick the right sentences to form the summary.

Neural networks

Neural network based summarization mostly proceeds in 3 steps: converting words to word embeddings using a look up table like word2vec, representing sentences/documents as continuous vectors and feeding the vector representation into a model (mostly CNN or RNN) for summarization [Dong, 2018]. Such neural based models have the potential for high performance if they are adequately trained.

- **Convolution Neural Network (CNN)**

  CNN is a good model for considering the global features of an input [Yin and Pei, 2015]. A basic CNN model consists of a convolution layer with sliding filters that perform local convolution on the input and a max-pooling layer that selects a single feature representation from the set of features generated by the convolution layer [Dong, 2018].

- **Recurrent Neural Network (RNN)**

  RNN is a time-dependent and bi-directional neural network unlike CNN [Dong, 2018].RNN connsists of input layer, hidden layers and output layer. The hidden layer gets updated based on current input and the previous hidden layer state. There are two basic RNNs, Gated Recurrent Unit (GRU) and Long Short-term Memory (LSTM). SummaRuNNer [Nallapati et al., 2017] uses a GRU based RNN model for summarization.

Word2Vec

Word2Vec [Mikolov et al., 2013] is a distributed representation (or word embeddings) using neural networks. It implements two context-based models namely Continuous Bag-of-Words (CBOW) and Skip-Gram.

CBOW is a feed forward neural network model that predicts the current word when a set of words from history and future are given as input. The order of the words do not influence the final result.

Skip-Gram is a recurrent neural network model that is used to learn distributed representation of words for predicting the nearby words. To train the model, we select a random number N and use N words from the past and N words from future of the input word. The distant words will be assigned lower weights than those that are closer.

2.7 Grouping of documents

One common technique for organizing large volumes of text documents into meaningful groups is clustering [Huang, 2008]. There are two types of clustering algorithms, Partitional clustering and Hierarchical clustering.

Hierarchical clustering is implemented in stages. It is either agglomerative or divisive [Fraley and Raftery, 1998]. In agglomerative, each object is initially an individual cluster and the closest objects are merged as clustering proceeds. Divisive is the inverse of agglomerative. All objects are in the same cluster initially and at each stage, they split.

Partitional algorithms are iterative and the number of output clusters should be specified [Fraley and Raftery, 1998]. Partitional methods were found to be more efficient while dealing with large datasets [Huang, 2008]. K-means clustering is an example of partitional algorithm.

K-means is an unsupervised clustering algorithm, where K refers to the number of clusters [Ferdous et al., 2009]. This algorithm is iterative and based on centroids. K centroids are chosen randomly in the beginning and all the closest points are assigned to this cluster [Ferdous et al., 2009]. This process is repeated till the algorithm converges. The final output depends on the initial assignment of centroids. The distance between a data point and centroid can be calculated using any distance metrics like cosine similarity, Euclidean distance, etc.

[Huang, 2008] compares the performance of similarity measures Euclidean distance, cosine similarity, Jaccard Ratio, Pearson coefficient and KLD for document clustering using K-means and found that all measures except Euclidean gave similar results. The selection of a similarity measure or distance metric is very important as this decides the accuracy of clustering [Huang, 2008].

2.8 Related work

2.8.1 News Summarization

One of the earliest works in news summarization is a system called Automatic News Extraction System (ANES) [Brandow et al., 1995].This method performs extractive summarization by building a list of signature words from the document text. This list is chosen by weighing words using TF-IDF and selecting those words weights between certain values. The words in the headline that contain topic related information but are not present in the list are added. The sentence score is then calculated by adding the weights of signature words in it. Sentences to be displayed in the summary are finally chosen based on the sentence score, length of summary and location of sentence in the document. Tf-idf and feature based scoring is a good combination for evaluating sentence relevance.

Searchable LEAD is a system that was implemented for customers using LexisNexis. This system assumes that the topics discussed in the document are introduced in the leading portion of its text [Wasson, 1998]. Wasson used Searchable LEAD on news articles and assigned the leading portion of the text to a LEAD field. Expert news analysts evaluated this field to find its effectiveness as a summary for the entire article. Brandow et al. [1995] compared LEAD system with ANES and found that an average of 92% of leading-text summaries were acceptable while only 74% of ANES summaries were acceptable. This was confirmed by Wasson [1998] who said that the results were consistent with Brandow et al. [1995] for general news articles but the acceptability decreases when the article contains lists, transcripts, etc. This is one of the simplest approaches to summarization that also gives an appreciable result.

Summarizing online news articles (SUMMONS) [McKeown and Radev, 1999] is another abstractive, multi-document summarization system based on traditional language generator. The input to the system are a set of documents that are manually clustered based on the topic. The final abstract is generated by utilizing the templates by Message Understanding Conference (MUC-4) [Sundheim, 1992]. This system consists of two main components, the content planner that decides what information should be included in the summary and the linguistic component that gives a syntactic structure to the summary. The input to the system is the templates carrying information from the articles and the output is a set of sentences generated using this information.

Rapidly Portable Translingual Information extraction and interactive multi-document Summarization (RIPTIDES) [White et al., 2001] is a system based on Information Extraction(IE) for multi-document summarization. There are two main components in this system, IE and Summarizer. The information extraction system takes the text as input and extracts patterns from this information which in turn is used to create templates. Summarizer, the second component takes these templates as input, merges them to get an event-oriented structure and assigns scores to each slots based on a combination of features. This is passed through a language generator for generating summary.

McKeown et al. [1999] proposed an abstractive, multi-document summarization system that integrates machine learning, statistical methods and language generation. The main idea behind the system is to generate a brief summary by identifying similarity across multiple documents on the same topic. In this method, the authors identify similar paragraphs or set of text from different documents based on similarity, positional information, etc. and assigns them to a theme. This is done using a machine learning algorithm which takes as input a vector representing the features that judges similarity. The classifier identifies similarity between paragraphs and puts them under same or different theme. The text under each theme is analysed to identify phrases that repeatedly occur within them. These phrases are given as input to the language generator for generating an abstract of contents from multiple documents. Use of machine learning for thematic classification and generation of templates ensures that no relevant information is missed out from the summary.

Newsblaster [McKeown et al., 2002] is a news summarization system developed in Columbia that provides updates on a daily basis by exploring news websites, extracting news from them, grouping articles on the same event and generates a summary. Newsblaster collects news from about 17 sources and groups them by event using hierarchical clustering. Summarization is done based on the type of events in these clusters. Each type of cluster is summarized by a separate subcomponent i.e, the clusters are routed to different summarizers based on the event.

Kaikhah [2004] proposed a method based on neural networks for news summarization. In this method, the document is initially tokenized to extract the sentences. Following this, a vector representation of each sentence is generated based on certain features. These features are information regarding the location, length and the words that are contained in it. The neural network is trained to identify sentences that are more likely to appear in summaries. This trained neural network is fused with a layer of neurons each representing a feature. After fusion, the connections in the network with smaller weights are pruned followed by clustering the neurons with common features. This neural network can now be used as a filter to select the sentences that are summary-worthy. This supervised model allows readers to personalize the system to include the type of information that they want in the summary. Taking location and signature words of a sentence into consideration makes this a better model.

Another technique for multi-document summarization based on sub-events is proposed by Daniel et al. [2003]. The basic idea is to capture unique information in articles from different sources but on the same topic. In this method, the sub-events in each document were identified and the judges were asked to manually score the relevance of each sentence with respect to each sub-event. So if we have N sub-events, N scores will be assigned to a sentence each corresponding to a sub-event. This data was then used to generate three summaries based on three different algorithms. The first algorithm chose the highest scoring sentences in all of the sub-events to be included in the summary. The second algorithm calculated a sum of scores for each sentence and picked the ones with the highest overall score to be in the summary. The next algorithm is a round robin, which picks the highest scoring sentence from each of the sub-events in round 1, second highest in round 2 and so on, till summary reaches the maximum length.

Liu et al. [2016] proposed a summarization system based on WMD. This method integrates WMD measure into Markow Random Walk (MRW) model. The WMD measure (i.e., the dissimilarity score) is inversely proportional to the similarity between sentences. This methodology uses a sigmoid function to convert WMD score to the similarity measure. This similarity score is used to create an affinity matrix to replace the original affinity matrix in MRW model that uses cosine similarity. The use of WMD instead of cosine similarity in MRW introduces semantic similarity and we could consider such integrations for our project.

Automatic news summarization and extraction system (ANSES) uses Lexical chains for summarization of broadcast news [Wong, 2018]. The thesaural relationship between different words is identified to build lexical chains, a sequence of related words. Whenever a new word is encountered, it checks for existing chains that are related to this word. If found, the word is added to the chain otherwise, a new chain is created with this word. The strength of the relationship is determined based on distance between the words. In framing the summary, chains with higher strength are preferred. The output summary was evaluated by comparing it against an ideal summary created by merging multiple human summaries.

Gong and Liu [2001] proposes two methods, one based on Information Retrieval and another on LSA, for extractive, generic summarization. The former calculates the relevance of each sentence with respect to the whole document and adds the highest scoring sentence to the summary. Once it is added to the summary, this sentence is removed from the document along with the words that appear in this sentence. Relevance score is recalculated with the modified document. The process is repeated till the summary reaches the desired length. The latter, on the other hand, uses LSA to identify the most important sentence that has highest index value in each of the singular vectors after SVD till we get the desired length of the summary. Different local and global weighting schemes were evaluated in the process. Output of both the methods were evaluated against manual summaries and the evaluation results were quite similar despite the use of two completely different approaches. In the first method, removal of words that have already gone into the summary, reduces similar sentences in the summary.

Summary of related work

Name/Paper	Abstractive or Extractive	Multi-document or single document	Approach
ANES [Brandow et al., 1995]	Extractive	Single document	Tf-idf + Sentence location
Searchable LEAD [Wasson, 1998]	Extractive	Single document	Leading portion of text
SUMMONS [McKeown and Radev, 1999]	Abstractive	Multi-document	Content Planner + Linguistic component
RIPTIDES [White et al., 2001]	Abstractive	Multi-document	Information Extraction
McKeown et al. [1999]	Abstractive	Multi-document	Machine Learning + Statistical techniques
Newsblaster [McKeown et al., 2002]	Abstractive	Multi-document	Machine Learning + Statistical techniques
Kaikhah [2004]	Extractive	Single document	Feature based vector representation + Neural Network
Daniel et al. [2003]	Extractive	Multi-document	Sub-event based
Liu et al. [2016]	Extractive	Broadcast news	MRW + WMD
ANSES [Wong, 2018]	Extractive	Broadcast news	Lexical chains based on thesaural relationship of words

Table 2.1: Summary of related work

Evaluation of generated summary

In evaluation, the sentence or summary that is evaluated is called the candidate and the ones against which a candidate is evaluated is known as reference.

1. **BLEU**

   BLEU [Papineni et al., 2002] is a precision-based evaluation measure that correlates highly with human evaluations. Precision can be calculated by dividing the number of words in candidate sentence that appear in any of the reference sentence by the total number of words in candidate. This calculation has a flaw. Consider the following example:

   Candidate : so so so so

Reference 1 : so are you

Reference 2 : This is it

The precision calculated here will be 4/4 equal to 1, which will give us a faulty evaluation. In order to avoid this, we can clip the numerator by the maximum count of the word in reference and divide by the total number of candidate words. This calculation is called the *modified unigram precision*. The modified precision for the above example is 1/4.

2. **Recall-Oriented Understudy for Gisting Evaluation (ROUGE)**

   ROUGE [Lin, 2004] is a set of methods that evaluate the quality of a summary against ideal manual summaries. There are four ROUGE measures:

   - ROUGE-N - This is a recall-based measure that uses n-grams as a measure of similarity. It is the percentage of matching n-grams in candidate and reference summaries, where n is a fixed value. For example, ROUGE-1 is based on unigrams, ROUGE-2 on bigrams and so on.
   - ROUGE-L - This measure is the sum of longest sequence of words, which need not be consecutive, that is common between candidate summary and each of the reference summaries.
   - ROUGE-W - This similar to ROUGE-L but each sequence is weighted. The sequence which is consecutive has a higher weight than those that are not.
   - ROUGE-S - This method is based on Skip-bigrams and is the measure of common skip-bigrams in candidate and reference summaries. ROUGE-SU is an extension of ROUGE-S that puts a constraint on the maximum distance between the words.

3. **Pyramid**

   Pyramid [Nenkova and Passonneau, 2004] is an evaluation method that is based on the concept that there is no ideal summary for a set of documents. It, therefore, uses a number of reference summaries for this evaluation. From the reference summaries, important clauses or sub-sections are identified and are called Summarization Content Units (SCU). Depending on the number of summaries in which they appear, a weight is assigned to each SCU. Finally, the SCU's are represented in the form of a pyramid divided into tiers, each tier containing SCU's of same weight. A good summary contains SCU's from the upper tiers with higher weights. A Pyramid score is between 0 and 1 and is calculated as the sum of weights of SCU's it contains divided by the sum of weights in an optimal summary having same number of SCU's.

Name of measure	Type
BLEU	Precision-based
ROUGE-N	Recall-based
ROUGE-L	Longest Common Subsequence(LCS)
ROUGE-W	Weighted LCS
ROUGE-S	Skip-bigram based
Pyramid	Summarization Content Units (SCUs) arranged as tiers in a pyramid

TABLE 2.2: Table of Evaluation

2.8.2 Grouping of documents

Ferdous et al. [2009] used Jaccard distance with K-means algorithm for clustering documents. Jaccard distance is the complement of Jaccard coefficient. In this paper, the authors found a way to get around the problem caused by improper assignment of initial clusters. They use Jaccard distance to find the 'K' most dissimilar documents among the set of input and assigns them as centroids. After this assignment, K-means algorithm is performed on the documents. Cosine similarity is used to measure the distance between documents and the centroid. This method was evaluated against the original K-means algorithm on the same dataset.

Evaluation of clustering results

Some measures for evaluating the performance of document clustering are [Ferdous et al., 2009]:

1. Sum of square error: This measure is the squared error between the cluster centroid and each of its documents.

2. Recall: Recall can be calculated as the number of relevant documents in the output cluster divided by the total number of documents that should be in the cluster.

3. Precision: Precision is the number of relevant documents in the output cluster divided by the total number of documents in the cluster.

Chapter 3

Requirement Analysis

This chapter is further divided into two sections, one for functional requirements specifying mandatory and optional requirements and another for non-functional requirements. All the requirements, to be satisfied to successfully complete this thesis, are mentioned in this section.

3.1 Functional requirements

3.1.1 Mandatory requirements

The mandatory functional requirements of our project are:

- **Group documents by topic**

  In order to perform multi-document summarization, we need to identify the documents that are on the same topic. The aim of this section of the project is to achieve the same. This grouping can be done by Clustering, an unsupervised method for organizing documents into disjoint clusters. The clustering result greatly impacts the subsequent steps of this project. Therefore, it is necessary that we get a good result for clustering.

  Evaluation

  The result of clustering can be evaluated using precision and recall (as described in Section 2.8.2).

- **Multi-document summarization**

  Once the grouping is done, we will have clusters containing multiple articles. One summary can be generated per cluster by processing all the documents in the cluster. The performance of summarization is influenced by the output of clustering. If a cluster contains unrelated articles, the summary might not be appropriate.

  Evaluation

  The generated summary can be evaluated using measures that are correlated with human evaluations like ROUGE or Pyramid (as described in Section 2.8.1).

- **Comparative study**

  The performance of the summarization technique that was used in our project can be compared with the performance of other frequency based baseline techniques. The same data should be used for summarization in all these methods.

  Evaluation

  All summaries should be evaluated using the same measure.

3.1.2 Optional requirements

As an optional requirement, we can consider extending this project to do a real-time summarization of online news articles. The model can be modified to generate summary from a continuous stream of news articles. The summary can be evaluated against the same criteria we used for the other summary.

3.2 Non-functional requirements

- **Cross-platform**

  The Python code developed for implementation should be compatible with the common operating systems like Mac OS, Linux and Windows. The compatibility can be tested by running the code on different systems.

- **Documentation**

  The source code for implementing the system should be documented. The details on how the code operates should be clearly mentioned.

- **Open-source**

  The source code should be made available to others for access and modification without requiring additional permission. The source code and documentation will be made available on GitHub.

- **Version Control**

  All changes to the source code and the report should be organized using version control. Dropbox can be used to maintain these versions online.

Chapter 4

Research Methodology

This chapter gives an idea of how we plan to implement the project including details of datasets, development model, softwares used, approaches adopted, evaluation and all other information relevant to the project.

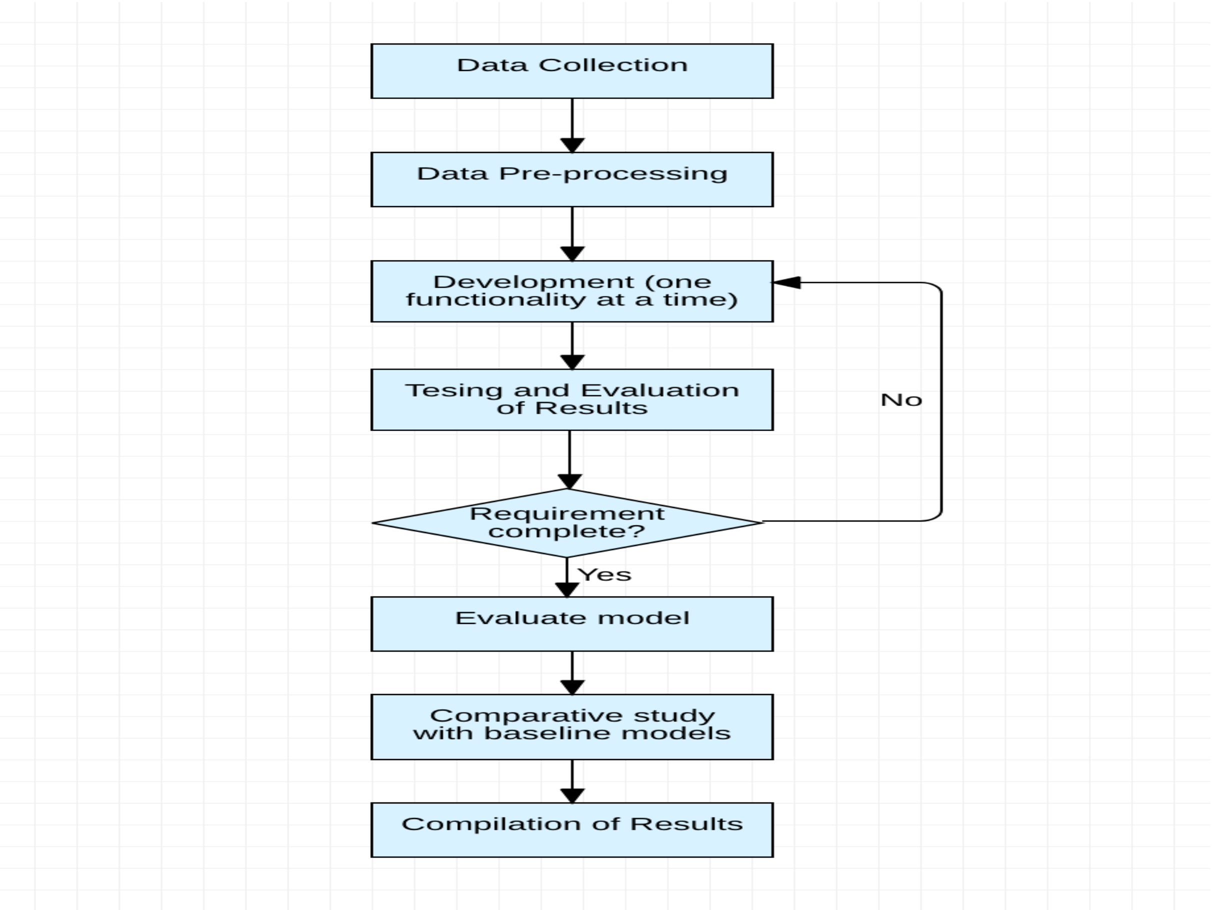

Data Collection
Data Pre-processing
Development (one functionality at a time)
Tesing and Evaluation of Results
No
Requirement complete?
Yes
Evaluate model
Comparative study with baseline models
Compilation of Results

Figure 4.1: Flow chart of project plan

4.1 Data Collection

The dataset to be used in this project can be generated by collecting online news articles from various sources like CNN, Fox News, New York Times etc. The requirement for this project is articles on related topic from multiple sources. The news articles can be collected over a period of time. We should be able to extract the following information from the article; headline, text content, author and timestamp. The data could be stored as RSS feeds from which the details can be retrieved or we can extract these details from the feed and save them separately for further processing.

4.2 Development

The plan for this project is to follow the Agile development model. There will be weekly meetings with the supervisors to analyse the progress of work, discuss the week's work, plan for the week ahead and set up project milestones.

We also plan to follow the Iterative and Incremental model, a feature-driven approach to development. The project will proceed in iterations and in each iteration, new functionalities will be added and/or existing design will be modified depending on the requirement. At the end of each iteration, we will have a working model implementing a subset of the functional requirements. The final iteration will produce a model implementing all the functionalities. Each increment will be tested and evaluated and any shortcomings will be eliminated in the subsequent increments.

The implementation will be carried out in Python. Python has a collection of libraries for Natural Language Processing (NLP) that will help with different tasks during development. Our implementation can be further divided into two sections; grouping articles by topic and summarization.

Grouping

The task of grouping by topic can be achieved by clustering based on timestamp and document distances. We can begin by filtering the documents whose timestamp is within a specified time frame. We can then convert the resulting set of documents into its vector representations and calculate the document distance using distance metrics like euclidean distance, cosine similarity, etc. This distance can be used in a clustering algorithm to get different clusters each containing articles of the same topic.

Summarization

After grouping documents by topic, summarization would be done by identifying relevant contents from the document that should appear in the summary. The plan, as of now, is to summarize in two steps. In the first step, we generate an extractive, single document summary for each document in the cluster and in the next step we generate a multi-document summary representing all documents in the cluster. The semantic aware technique to be used for measuring similarity will be finalized after careful study and evaluation.

4.3 Evaluation

After each iteration, the resulting model from the iteration will be evaluated based on the functionality added in that iteration and optionally, on the ones added in previous iterations.

After grouping the articles using clustering, the performance can be evaluated using precision and recall. The exactness and completeness of the resulting clusters can evaluated using these two measures. The summary can be evaluated similarly using measures like BLEU, ROUGE, etc.

Once the implementation is complete, the project can be evaluated as a whole and the results can be compiled. The project can be evaluated against the requirements mentioned in Chapter 3 of this report. Optionally, and if time permits, we plan on having a questionnaire for others to evaluate the quality of our summary.

4.4 Comparative Study

When the project implementation is complete, we can do a comparative study of the result obtained in our project and the result when using baseline methods. The findings can be recorded and conclusions can be drawn from this comparison.

Chapter 5

Initial Analysis

We have done some initial analysis, data pre-processing and implementation of an extractive, single document summary using a baseline frequency-based method.

5.1 Data

The RSS Url of an online news article was given as input. The RSS feed contains a variety of information including the title, author, timestamp, etc. The actual content of the article was identified, extracted and stored in a separate variable for further processing.

5.2 Data pre-processing

In this step, we pre-process the data and make it suitable for the tasks in the subsequent steps. The pre-processing done on our input are:

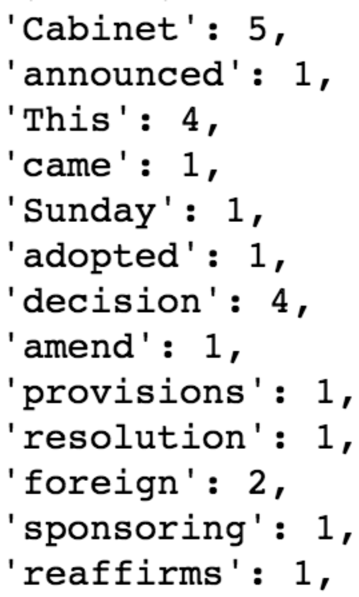
- *Conversion to plain text*: The textual content, identified and separated from the RSS feed, contained HTML tags. In this step, we remove the tags to obtain plain text.
- *Removal of formatting*: We remove all the formatting references from the text. This includes \n, \r, etc.

- *Tokenization*: Tokenization is done to obtain the list of sentences and the list of words from the text.

- *Identifying stop words*: The words that frequently appear in English language and do not contribute to the meaning like prepositions, articles, etc are considered as stop words [Ferreira et al., 2013]. We used the list of stopwords from nltk Python library for identifying the ones in our text.

5.3 Sentence scoring

The frequency of each word, except the stop words, were calculated and stored in a dictionary as key-value pairs. The key is the word and the value is the corresponding frequency as shown in the figure below.

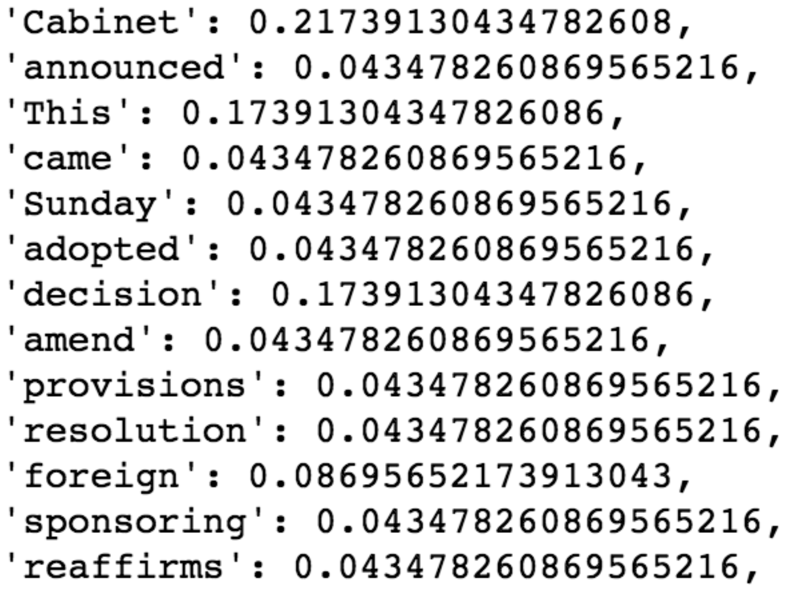


```
'Cabinet': 5,
'announced': 1,
'This': 4,
'came': 1,
'Sunday': 1,
'adopted': 1,
'decision': 4,
'amend': 1,
'provisions': 1,
'resolution': 1,
'foreign': 2,
'sponsoring': 1,
'reaffirms': 1,
```

Figure 5.1: Words and its frequencies

The weighted frequency of a word is obtained by dividing each frequency by the maximum frequency in the list. The weighted frequency of the words in our text are shown in the figure below. Now we have a value between 0 and 1 as a weight for each word in our text.

The sentence score is the sum of weight of each word that appears in the sentence with the exception of stopwords.

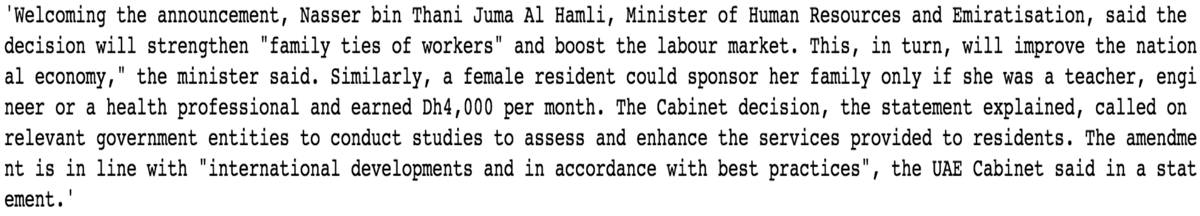


```
'Cabinet': 0.21739130434782608,
'announced': 0.043478260869565216,
'This': 0.17391304347826086,
'came': 0.043478260869565216,
'Sunday': 0.043478260869565216,
'adopted': 0.043478260869565216,
'decision': 0.17391304347826086,
'amend': 0.043478260869565216,
'provisions': 0.043478260869565216,
'resolution': 0.043478260869565216,
'foreign': 0.08695652173913043,
'sponsoring': 0.043478260869565216,
'reaffirms': 0.043478260869565216,
```

FIGURE 5.2: Words and its weighted frequencies

5.4 Generating summary

Once all the sentences are scored, the sentences with higher scores are extracted and a summary is generated. If the summary is limited to n sentences, those with first n highest scores are included in the summary in the order of its score. The summary with 5 sentences generated for our input is shown in the figure below.



```
'Welcoming the announcement, Nasser bin Thani Juma Al Hamli, Minister of Human Resources and Emiratisation, said the
decision will strengthen "family ties of workers" and boost the labour market. This, in turn, will improve the nation
al economy," the minister said. Similarly, a female resident could sponsor her family only if she was a teacher, engi
neer or a health professional and earned Dh4,000 per month. The Cabinet decision, the statement explained, called on
relevant government entities to conduct studies to assess and enhance the services provided to residents. The amendme
nt is in line with "international developments and in accordance with best practices", the UAE Cabinet said in a stat
ement.'
```

FIGURE 5.3: Generated Summary

Chapter 6

Professional, Legal, Ethical and Social issues

6.1 Professional and Legal Issues

Any information taken from research papers, books, articles, etc will be referenced and properly cited. Code snippets from other sources, if used, will also be acknowledged. All resources used in this project, including datasets, softwares and libraries will be open source and their policies and terms of use will be abided by.

The source code will be extensively commented and properly tested before making it available to the public. The BCS code of conduct will be adhered to.

The source code developed for this project will be open source. The source code will be published under a GNU General Public License [Free Software Foundation, 2007] which gives permission to use, modify and distribute the code. Any addition or modification to this code must be published under the same license.

6.2 Ethical issues

This project does not involve the participation of any human subjects. All the data used in building the dataset are publicly available and does not require additional permission.

If time permits, we plan to have a few people evaluate the summary generated by our model. This evaluation would be done with the help of a questionnaire. The participation is completely voluntary and no one will be forced against their will. No sensitive, personal or irrelevant information will be collected.

6.3 Social Issues

Any use of public data is acknowledged and data privacy is respected. There is no risk of information theft as all the data needed for completing the project are taken from public sources.

Safety

The project uses no wearable or hazardous equipments. The project stakeholders will be exposed to no more hazards than that at a standard office setting.

Chapter 7

Project plan

7.1 Risk Assessment

As with any other project, this research oriented project also has certain risks associated with it. This could be related to technology, resources, computation, etc.

Risk	Likelihood	Impact	Mitigation
Loss of source code/ Thesis report	Low	High; Will have to start over	Maintain an online version control, take regular back-ups
Computational challenges	Medium	Medium; Progress will be hindered	Use HW University's resources
Shortage of resources	Medium	Medium; Affects performance	Plan ahead, make sure all resources are available before starting the project
Falling behind on project schedule	High	Medium; Rushing other tasks to complete on time	Start new task as soon as one is over without waiting for the scheduled start date
Natural Calamities	Low	High; Possible loss of resources	Unavoidable
Health problems	Medium	Medium; Falling behind schedule	Following a healthy lifestyle and proper food habits

Table 7.1: Table of Risks

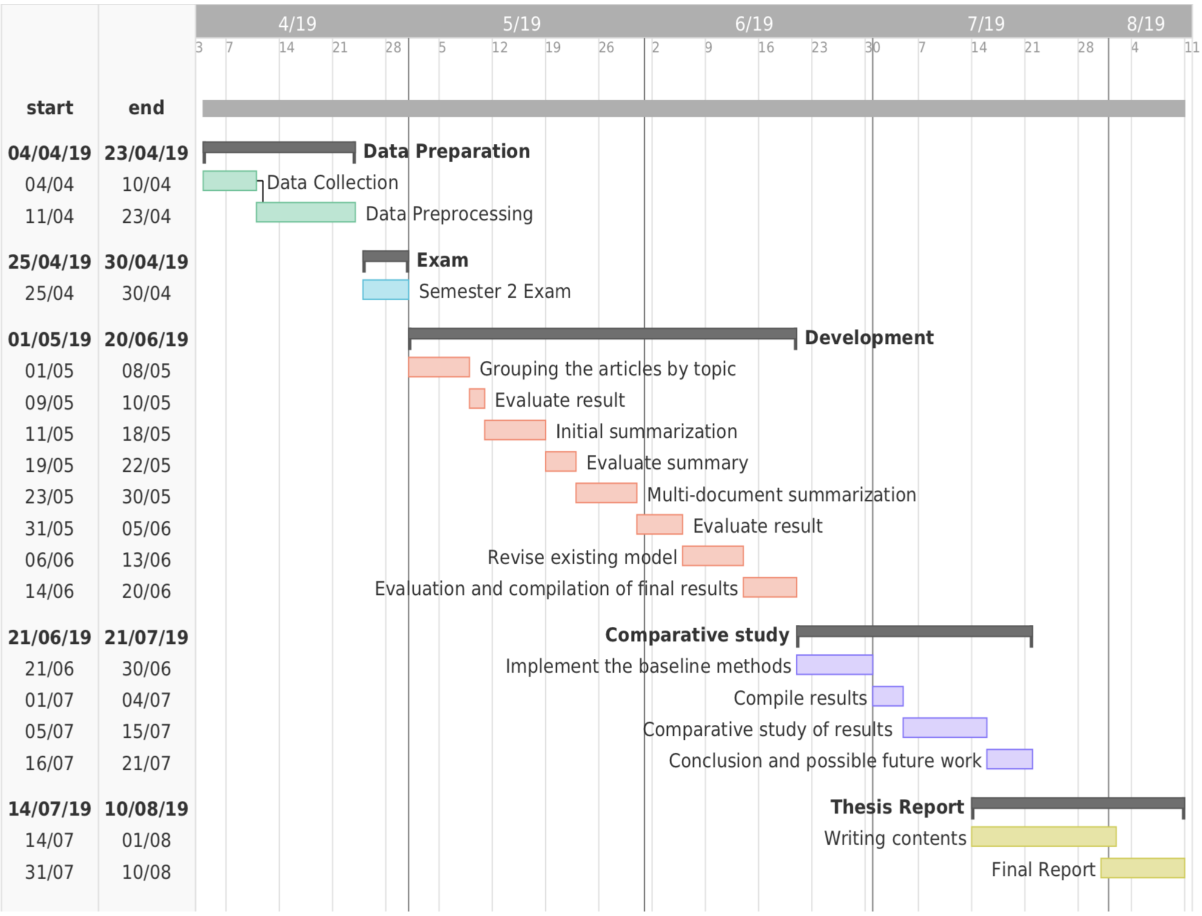
Table 7.1 is a list of possible risks that might arise during the span of this project. The likelihood of such an event, the severity of impact in case it happens and ways to mitigate such circumstances are also included in the table.

7.2 Tasks and Timelines

Tasks	Duration (in days)	Start Date	Completion Date
Data Preparation	**20**	**04/04/19**	**23/04/19**
Data Collection	7	04/04/19	10/04/19
Data Pre-processing	13	11/04/19	23/04/19
Exam	**6**	**25/04/19**	**30/04/19**
Development	**51**	**01/05/19**	**20/06/19**
Grouping articles (First iteration)	8	01/05/19	08/05/19
Evaluate Result	2	09/05/19	10/05/19
Initial summarization (Second iteration)	8	11/05/19	18/05/19
Evaluate summary	4	19/05/19	22/05/19
Multi-document summarization (Third iteration)	8	23/05/19	30/05/19
Evaluate result	6	31/05/19	05/06/19
Revise existing model	8	06/06/19	13/06/19
Evaluation and compilation of final results	7	14/06/19	20/06/19
Comparative study	**31**	**21/06/19**	**21/07/19**
Implement baseline models	10	21/06/19	30/06/19
Compile Results	4	01/07/19	04/07/19
Comparative study	11	05/07/19	15/07/19
Conclusion	6	16/07/19	21/07/19
Thesis Report	**28**	**14/07/19**	**10/08/19**
Write contents	18	14/07/19	01/08/19
Final report	11	31/07/19	10/08/19

Table 7.2: Table of project tasks and timelines

7.3 Gantt chart



start	end	
04/04/19	**23/04/19**	**Data Preparation**
04/04	10/04	Data Collection
11/04	23/04	Data Preprocessing
25/04/19	**30/04/19**	**Exam**
25/04	30/04	Semester 2 Exam
01/05/19	**20/06/19**	**Development**
01/05	08/05	Grouping the articles by topic
09/05	10/05	Evaluate result
11/05	18/05	Initial summarization
19/05	22/05	Evaluate summary
23/05	30/05	Multi-document summarization
31/05	05/06	Evaluate result
06/06	13/06	Revise existing model
14/06	20/06	Evaluation and compilation of final results
21/06/19	**21/07/19**	**Comparative study**
21/06	30/06	Implement the baseline methods
01/07	04/07	Compile results
05/07	15/07	Comparative study of results
16/07	21/07	Conclusion and possible future work
14/07/19	**10/08/19**	**Thesis Report**
14/07	01/08	Writing contents
31/07	10/08	Final Report

Figure 7.1: Gantt chart of project plan

Bibliography

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). Text Summarization Techniques: A Brief Survey. (1).

Balakrishnan, V. and Lloyd-Yemoh, E. (2014). Stemming and lemmatization: a comparison of retrieval performances.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Brandow, R., Mitze, K., and Rau, L. F. (1995). Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675–685.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Cavnar, W. B., Trenkle, J. M., et al. (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer.

Choon, N. H., Kumar, Y. J., Suppiah, P. C., Goh, O. S., and Basiron, H. (2016). A Review on Automatic Text Summarization Approaches. *Journal of Computer Science*, 12(4):178–190.

Daniel, N., Radev, D., and Allison, T. (2003). Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 9–16. Association for Computational Linguistics.

Dong, Y. (2018). A survey on neural network-based summarization methods. *arXiv preprint arXiv:1804.04589.*

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Ferdous, R. et al. (2009). An efficient k-means algorithm integrated with jaccard distance measure for document clustering. In *2009 First Asian Himalayas International Conference on Internet*, pages 1–6. IEEE.

Ferreira, R., de Souza Cabral, L., Lins, R. D., e Silva, G. P., Freitas, F., Cavalcanti, G. D., Lima, R., Simske, S. J., and Favaro, L. (2013). Assessing sentence scoring techniques for extractive text summarization. *Expert systems with applications*, 40(14):5755–5764.

Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.

Free Software Foundation (2007). Gnu general public license, version 3. http://www.gnu.org/licenses/gpl.html. Accessed 2 April 2019.

Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM.

Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56.

Kaikhah, K. (2004). Text summarization using neural networks.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Liu, S.-H., Chen, K.-Y., Hsieh, Y.-L., Chen, B., Wang, H.-M., Yen, H.-C., and Hsu, W.-L. (2016). Exploring word mover's distance and semantic-aware embedding techniques for extractive broadcast news summarization. In *INTERSPEECH*, pages 670–674.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

McKeown, K., Klavans, J. L., Hatzivassiloglou, V., Barzilay, R., and Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects.

McKeown, K. and Radev, D. R. (1999). Generating summaries of multiple news articles. *Advances in automatic text summarization*, pages 381–389.

McKeown, K. R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B., and Sigelman, S. (2002). Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann Publishers Inc.

Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*.

Mihalcea, R., Corley, C., Strapparava, C., et al. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nallapati, R., Zhai, F., and Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Protim Ghosh, P., Shahariar, R., and Hossain Khan, M. A. (2019). A Rule Based Extractive Text Summarization Technique for Bangla News Documents. *International Journal of Modern Education and Computer Science*, 10(12):44–53.

Raju, T. R. and Allarpu, B. (2017). Text Summarization using Sentence Scoring Method. *International Research Journal of Engineering and Technology*, pages 2395–56.

Saggion, H. and Poibeau, T. (2013). Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 3–21. Springer.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sundheim, B. M. (1992). Overview of the fourth message understanding evaluation and conference. In *Proceedings of the 4th conference on Message understanding*, pages 3–21. Association for Computational Linguistics.

Templeton, A. and Kalita, J. (2018). Exploring Sentence Vector Spaces through Automatic Summarization. (1359275).

Vijaymeena, M. and Kavitha, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3(2):19–28.

Wasson, M. (1998). Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*, volume 2.

White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., and Wagstaff, K. (2001). Multidocument summarization via information extraction. In *Proceedings of the first international conference on Human language technology research*.

Wong, L. (2018). Automatic news summarization and extraction system. *MEng Computing. Imperial College Dept. of computing.*, 2.

Yin, W. and Pei, Y. (2015). Optimizing sentence modeling and selection for document summarization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.