STUDY OF SUPERVISED AND UNSUPERVISED TEXT SUMMARIZATION TECHNIQUE

A Major II Project Report

Submitted in partial fulfillment of
the requirements for the award of the degree
of
Master of Technology
in
Computer Science and Engineering
by

Parul Sharma

under the guidance of

Dr. Rajni Jindal
Associate Professor and Head of Department
Computer Science and Engineering Department

DEPARMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY, NEW DELHI

DELHI-110 042 (INDIA)

JULY 2017

CERTIFICATE

This is to certify that Project Report entitled "Study Of Supervised and Unsupervised Text Summarization Technique" Submitted by Parul Sharma (roll no. 2K15/CSE/12) in partial fulfillment of the requirement for the award of degree Master of Technology (Computer Science and Engineering) is a record of the original work carried out by him under my supervision.

Dr. Rajni Jindal
Associate Professor and Head of Department
Department of Computer Science & Engineering
Delhi Technological University

DECLARATION

I hereby declare that the Major Project-II work entitled "Study Of Supervised and Unsupervised Text Summarization Technique" which is being submitted to Delhi Technological University, in partial fulfillment of requirements for the award of the degree of Master Of Technology (Computer Science and Engineering) is a bonafide report of Major Project-II carried out by me. I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

(Parul Sharma)

2K15/CSE/12

ACKNOWLEDGEMENT

I am highly indebted to Dr. Rajni Jindal and obliged for giving me the autonomy of working and exploring different avenues regarding thoughts. The sustaining and blooming of the present work is primarily because of her significant direction, proposals, insightful judgment, valuable feedback and an eye for flawlessness. My mentor always answered the myriad of my doubts with smiling graciousness and immense patience, never letting me feel that I am novices by always lending an ear to my views, appreciating and improving them and by giving me a free hand in my project. It’s only because of her overwhelming interest and helpful attitude; the present work has attained the stage it has.

I would like to express my sincere gratitude towards family for their blessings, encouragement and moral support. In the end, I would like to thank my peer group for providing me with a conductive and competitive environment. Without light and fun moments shared with them, this research work would have been difficult.

(Parul Sharma)

ABSTRACT

Summarization is the art of abstracting key content from one or more information sources. Summarization includes text summarization, image summarization, and video summarization. Text summarization is one of application of natural language processing and is becoming more popular for information condensation. Information is accessible in great quantity for every topic on internet assembly the key information in the form of summary would benefit a number of users. Automatic text summarization system generates a summary, i.e. it contains short length text which comprises all the key information of the document. Summary can be generated through extractive as well as abstractive methods. There are two approaches for summarization Supervised and Unsupervised which are further classified into many techniques.

INDEX

Contents

TABLE OF FIGURES

1. INTRODUCTION

1.1 TEXT SUMMERIZATION

The development of the web has unfathomably expanded the measure of accessible data. Separating helpful data from the expansive volume of information present on the web is dependably the most loved point of analysts. The Internet today has a big collection of reachable text documents and with the large volume of text documents, getting a summarised form of each document deeply facilitates the job of understanding information from the documents. With the help of summarization users can find crucial information captured in the documents.

Compressed version of a document created by an automatic method from a computer is known as Automatic Summarization. Without summaries it would be practically unthinkable for human beings to access the huge amount of growing data available online. It’s been over 50 years since the research in this field started but still a lot needs to be done. Tex summarization is an important field in IT sector. With the availability of automatic text summarizers we can save a lot of efforts in understanding a huge document. Text summarization can help users to quickly evaluate the significance of a document or steer through the large amount of data present. Instead of going through an entire document people can easily understand the document through summarization saving them a lot of time. The heading, keywords can give an idea of the document but they may not give the information which the user wants to know. Thus text summarizers are really important these days.

Types of Summaries

- **Extractive Summary**

This sort of synopsis is created by choosing few sentence(s) shape the report and scores are allotted to vital sentences in the records and after that exceptionally scored sentences are produced the rundown. It is performed by connecting a few sentences taken precisely as they show up in the information being outlined. Synopsis' length relies upon the pressure rate

- **Abstractive Summary**

An abstractive outline does exclude the words or expressions from the first report rather it re-translated thoughts or ideas taken from the first record and appeared in an alternate frame. It is composed to bunch the primary data in the information and may reuse expressions or provisos from it, yet the synopses are general communicated in the expressions of the synopsis creator. It needs broad characteristic dialect handling. In this way, it is a great deal more mind boggling than extractive synopsis

Different philosophies for recognizing basic substance for customized content summary have been made to date. Topic depiction approaches first deduce a widely appealing depiction of the substance that gets the subjects discussed in the data. In perspective of these depictions of subjects, sentences in the information record are scored for criticalness. On the other hand, in pointer depiction approaches, the substance is addressed by an alternate game plan of possible markers of criticalness which don't go for discovering topicality. These pointers are combined, all the time using machine learning methodologies, to score the essentialness of each sentence. Finally, a blueprint is conveyed by picking sentences in an enthusiastic approach, picking the sentences that will go in the summary one by one, or generally enhancing the decision, picking the best plan of sentences to outline an once-over.

1.2 UNSUPERVISED TECHNIQUES

Unsupervised summarization techniques expel the requirement for training information. They approach the issue from an alternate point. They create the summary by recovering just the objective records. In this manner, they are proper for recently observed information with no propelled alterations. In summarization using word graph, which is one of the many unsupervised summarization technique, a graph is used with an algorithm. The algorithm similar finds top-ranked sentences. It is similar to the Page-Rank algorithm of Google. The key intuition is the notion of centrality or prestige in social networks i.e. a sentence should be highly ranked if it is recommended by many other highly ranked sentences.

The unsupervised techniques are further divided into two categories, namely summarization using sentence rank and summarization using word graph.

Sum basic , Luhn , Edmundson , Latent semantic analysis , Lex rank , Text rank , KL sum are the various techniques under summarization using sentence rank.

All these techniques are implemented in this thesis on the Amazon Food Reviews dataset to summarise the review of a user.

1.3 SUPERVISED TECHNIQUES

Many techniques have been developed to do text summarization. Mainly the techniques of achieving text summarization are classified into 2 categories supervised method of summarization and unsupervised method. Summarization with supervised technique uses machine learning to do summarization of documents. In this thesis I will be using 8 unsupervised summarization techniques and the supervised summarization technique to summarize *Amazon Fine Food Reviews* which are food product review by *Amazon* users. I aim to sum-up the information provided by a reviewer about a product in a concise manner.

Supervised techniques use a collection of documents and human-generated summaries for them to train a classifier for the given text. the Neural machine summarization model used in this thesis uses a method based on the *encoder-decoder* idea (Sutskever et al., 2014; Cho et al., 2014) with an encoder to understand the input and decoder to provide the corresponding output.

In Supervised techniques we use machine learning. We train some classifier with manually created summaries from a group of documents for a given text. The classifier learns some features from the manual summaries sentences that make them good candidates for summary. Important features of sentences include number of words in the sentence, location of the sentence, etc. Original text in the training document can be categorized as <in summary or not in summary>. There are certain disadvantages with these techniques, one is that training is not easily available. It can be a bit costly to get the training data. Another disadvantage is that most readily available manually created summaries are abstractive in temperament.

1.4 MOTIVATION

Automatic Summarization is an exceptionally fascinating and valuable assignment that offers support to numerous different undertakings and in addition it exploits the methods produced for related NLP (“Natural Language Processing”) errands. The increasingly large amount of data requires processing, thus the task of effective summarization gains more and more importance. I thought that working with *Automatic Summarization* as the course project would serve the purpose of understanding NLP techniques well.

Perusing the whole article, dismembering it and isolating the essential thoughts from the crude content require some serious energy and exertion. Perusing an article of 500 words can take no less than 15 minutes. Automatic summarization enables us to peruse less information yet at the same time get the most imperative data and make strong conclusions.

Automatic summarization simulates the work of intelligence analyst. It judges if a document is relevant to the topic of interest without completely reading the document saving a lot of time for the user. Automatic summarization enables us to find relative information quickly with fewer efforts required.

Most data nowadays is put away in content as a result of its perpetual quality and capacity to be shared.

For instance, as indicated by Google, there are 130 million books in the whole world. Quite a bit of this data is out of reach in light of the fact that there's essentially a lot of it. No human can read all of the books or research papers . This is the reason books have outlines, why inquire about papers have abstract, and why Wikipedia exists. In any case, with a specific end goal to make a synopsis, a man needed to physically gather data and compose it, which is a period concentrated errand. We trust that building a framework that can naturally develop summaries for us would enable us to get to data in a more edible organization and spare endless human hours spent outlining archives.

1.5 GOALS OF THESIS

The goal of this thesis is to explore the various techniques of Text Summarization. Text Summarization is a topic of great interest to researchers for a long period of time. Constant efforts have been made to find different techniques for Automatic Summarization and still many people are involved in finding better techniques because of the advantages Automatic Summarization has.

This thesis brings different Automatic Summarization techniques developed. There are mainly two types of summarization techniques Supervised and Unsupervised text summarization. Various types of unsupervised techniques and supervised techniques are discussed in this thesis. In this thesis I will be using 8 unsupervised summarization techniques and the supervised summarization technique to summarize *Amazon Fine Food Reviews* which are food product review by *Amazon* users. I aim to sum-up the information provided by a reviewer about a product in a concise manner.

A brief discussion on all the summarization techniques is done along with the implementation of these techniques on Amazon Food Reviews. The choice of this dataset is based on the fact that now a days people tend to review the products they use or buy and summarizing those reviews can be really helpful. This thesis brings major techniques of text summarization together.

1.6 ORGANISATION OF THESIS

The thesis is covered under seven chapters.

Chapter 1 - Introduction

This chapter contains the introduction giving a brief about the thesis. The sub chapters motivation and goals of thesis, as indicated in title, describe what lead to this study and what all this thesis cover.

Chapter 2 - Unsupervised Learning Methods for Text Summarization

This chapter covers all the unsupervised summarization techniques. The sub chapters are all different types of unsupervised techniques describing which are further divided. A brief explanation of each technique is provided.

Chapter 3 - Supervised Learning Methods for Text Summarization

This chapter includes the supervised technique for text summarization giving a brief explanation about how this technique works.

Chapter 4 - Dataset Used

Various techniques for summarization mentioned above are implemented on a dataset. This chapter describes the dataset used in the thesis.

Chapter 5 - Experiment and Results

This chapter describes the experiments performed on the dataset used. The result of each technique is provided in this chapter. Qualitative result for unsupervised techniques and qualitative and quantitative both for supervised technique.

Chapter 6 - Conclusion

This chapter concludes the thesis work and the results obtained.

Chapter 7 - Future Work

This chapter what all future work is possible further in this work.

2. UNSUPERVISED LEARNING METHODS FOR TEXT SUMMARIZATION

2.1 SUMMARIZATION USING SENTENCE RANKS

2.1.1 SUM BASIC [7]

This is the baseline model that produces standard multi-document summary. It is based on the observation that words occurring repeatedly in the document occur with greater possibility in the human summaries as compared to words which occur less frequently. In this model each sentence S is given a score representing how many frequency words are there in the sentence.

$$\text{Score}(S) = \sum \frac{1}{|S|} P_D(w) \qquad w \in S$$

Where $P_D(\cdot)$ at the start reflects the observed unigram probabilities obtained from the document collection D. A summary S is gradually built by adding the highest scoring sentence according to the above equation.

One downside with this model is that it inalienably supports duplication of successive constant words in the outline. Ideally, a synopsis model ought to be more review situated, punishing rundowns which pass up a great opportunity genuinely normal record set words and immediately fizzling the reward for visit utilization of word.

One more inconspicuous weakness is the utilize of the natural exact unigram dispersion to speak to content noteworthiness. For instance, there is no distinction between a word which happens every now and again in a similar record or a similar number of times crosswise over many reports. The second word is more demonstrative of huge record set substance.

Sum Basic model for document summarization works according to the following algorithm :-

Step 1: Compute the probability distribution over the words w_i appearing in the input, $p(w_i)$ for every i ; $p(w_i) = \frac{n}{N}$ where n is the number of times the word appeared in the input, and N is the total number of content word tokens in the input .
Step 2: For each sentence S_j in the input, assign a weight equal to the average probability of the words in the sentence, i.e., $$\text{Weight}(S_j) = \sum_{wi \in Sj} \frac{p(wi)}{\lvert wi \rvert wi \in Sj}$$
Step 3: Pick the best scoring sentence that contains the highest probability word.
Step 4: For each word w_i in the sentence chosen at step 3, update their probability: $$p_{new}(w_i) = p_{old}(w_i) \cdot p_{old}(w_i)$$
Step 5: If the desired summary length has not been reached, go back to Step 2

Figure 2-1 Algorithm for Sum Basic

2.1.2 LUHN [2]

This method is proposed by Luhn in 1958. It is basically a heuristic way of extracting summaries from the text.

In order to determine if a sentence can be the part of the summary, we need to find the significance of the sentence as compared to other sentences. The significance of a sentence is the obtained by the analysis of the words present in that sentence. It is here recommended that the recurrence of word event in an article outfits a valuable estimation of word significance. It is further recommended that the relative position inside a sentence of words having given estimations of criticalness outfits a valuable estimation for deciding the significance of sentences. The significance element of a sentence will subsequently be founded on a blend of these two estimations. The defence of measuring word significance by use-frequency depends on the way that an author typically rehashes certain words as he advances or shifts his contentions also, as he explains on a part of a subject. The likelihood is additionally little that a writer will utilize distinctive words to mirror a similar idea. Regardless of the possibility that the creator tries to choose equivalent words for elaborate reasons, he soon comes up short on genuine choices also, falls into reiteration if the thought being communicated was possibly noteworthy in any case.

Another important point for determining the significance of a sentence is that, whatever the subject, the nearer certain words are related, the more particularly a part of the subject is being dealt with. In this way, wherever the best number of every now and again happening distinctive words are found in most noteworthy physical nearness to each other, the likelihood is high that the data being passed on is most illustrative of the article.

From these contemplations a "significance factor" can be inferred which mirrors the quantity of events of noteworthy words inside a sentence and the direct separation between them because of the mediation of non-noteworthy words.

This method extracts summary from the text using the following algorithm:-

Step 1: Ignore all the common words or stop-words.
Step 2: Count the most frequently occurring words in the document.
Step 3: Select a small number of the most frequent words for scoring.
Step 4: Score the sentence according to the number of top words they contain. Select the top K sentences for summary.

Figure 2-2 Algorithm for Luhn

2.1.3 EDMUNDSON [3]

This method is proposed by Edmundson in 1969 which is a sort of continuation of Luhn's method which considers other factors along with the frequency of words while extracting the summaries from reviews.

Key words, pragmatic/cue words, locations of words (like heading etc.), title words are the four major weighting factors. These factors are used to assign score to the sentences in the review.

Cue Method:

The Cue Method is based on the assumption that the sentence relevance is affected by the cue words like "significant", "hardly" and "impossible". This method uses Cue dictionary of the selected pragmatic words built using some big corpus. The cue dictionary consists of three sub-dictionaries each for Bonus words (which positively describe the content of the sentence), Stigma words (which negatively describe the content of the sentence), and Null words (which are irrelevant when describing the content of the sentence). The score for each sentence is the sum of the weights (from the cue dictionary) of its constituent words.

Key Method:

This method is based on the assumption that the words which occur frequently in review are more probable to occur in summary of the review. This method builds a key dictionary which lists all words which are absent in cue dictionary in the decreasing order of frequency of occurrence of the words. Non-Cue words with frequency greater than threshold are designated key words in the review. Weights to these words are assigned based on the frequency of the words in the review. The score of a sentence based on this key dictionary is sum of the key weights of its constituent non-cue words the sentence.

Title Method:

The words which are conceived in the title depict the subject content in a document, this model works on this supposition. This method builds a Title dictionary for each document which comprises of all non-null words in the title, subtitle and heading in the document. Words in the dictionary are assigned positive weights. The score of a sentence is sum of weight of each title word in sentence. This factor could not be used in our evaluation since the title and headings are not available in dataset.

Location Method:

This method is based on the hypothesis that

– The sentences occurring close to headings are relevant in extracting the summary

– Topic sentences which depict the content of the document occur to the start or to the end of the document and its paragraphs.

Since headings to the review were not available, we could use first assumption.

This method uses pre-stored dictionary that occur in headings of documents which is built using statistical inference on corpus. The score is not just by the weights of each word in sentence but also a score is added to whole sentence based on the location of sentence in the review.

2.1.4 LATENT SEMANTIC ANALYSIS [4]

Generic text summarization when applied Singular Value Decomposition (SVD) gives the Latent Semantic Analysis. The procedure of Latent Semantic Analysis starts with building a term by sentences matrix X = [*X*1, *X*2, …, *X*n] where the j^{th} column vector in the matrix represent the weighted term-frequency of the sentence j in the article being analysed. In the event that there are a sum of a terms and b sentences in the record, at that point we will have a a x b matrix X for the archive. The matrix X is a sparse matrix because each word does not typically show up in each sentence.

The SVD of a matrix X (a x b matrix) where without loss of generality a ≥ b is defined as:

$$X = U \sum V^T \text{ where}$$

U = [*uij*] is an $a \times b$ *column*-orthonormal matrix whose columns are called left singular vectors;
Σ = diag (σ_1, σ_2, …, σ_n) is an $b \times b$ diagonal matrix, whose diagonal elements are non-negative singular values sorted in descending order, and
V = [*vij*] is a $b \times b$ orthonormal matrix, whose columns are called right singular vectors (see figure 1). If rank(X) = *r*, then **Σ** satisfies: $\sigma_1 \geq \sigma_2 \geq ..... \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0$

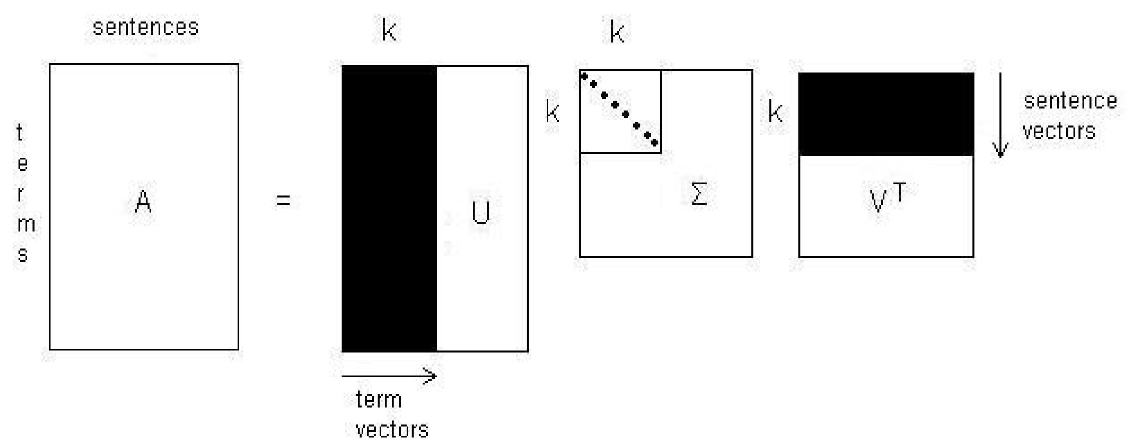


Figure 2-3 Singular Value Decomposition

SVD can be connected by sentences lattice X to the terms utilizing two particular points of view. As per the change viewpoint, the SVD surmises a mapping among the a-dimensional space delivered by the weighted term-recurrence vectors and the particular vector space of r-measurement. As per the semantic point of view, the SVD induces the dormant semantic structure from the article spoken to by network X. This operation reflects a breakdown of the first article into r directly autonomous base vectors or thoughts. Each term and sentence from the article is commonly recorded by these base vectors/thoughts. An exceptional SVD highlight is that can get and show interrelationships among terms so it can semantically bunch terms and sentences. Help more, if the blend example of a word is striking and rehashing in article, this case will be gotten and addressed by one of the solitary vectors. The significance of level of this example inside the article is demonstrated by the size of the particular esteem. Every one of the sentences which contain this word example will be anticipated along the solitary vector. On the off chance that a sentence finest speaks to this example, that sentence will have the common record an incentive with this vector. Any sentences containing this word blend illustration will be expected along this specific vector, and the sentence that best addresses this case will have the greatest document a motivating force with this vector. As each particular mix of word depicts a specific subject/thought in the article, the substances depicted above regularly provoke that hypothesis that each specific vector addresses a striking topic/thought of the article, and the significance of the topic/thought is spoken to by the greatness of its relating esteem. In perspective of the above talk, an outline technique was proposed which utilizes the network V^T. This lattice depicts a criticalness level of each point in each sentence. The outline procedure picks the most edifying sentence for each theme. For k-th sentence to be separated, we pick greatest record an incentive in k-th right particular vector in framework V^T. It infers that k-th sentence we pick for the rundown has the biggest file an incentive in the k-th vector in lattice V^T.

2.1.5 LEX-RANK [5]

This strategy for Text Summarization depends on the approach of surveying the centrality of each sentence in a bunch and extricating the most critical ones and incorporating them in the synopsis.

We select a subset of sentences from the first record during the time spent Extractive Summarization. It can be viewed as recognizing the rule subject of a group by recognizing the most focal sentences in the bunch (group here alludes to multi-archives) that give the fundamental and sufficient measure of information. Centrality of the words in a sentence is eluded as the centrality of the sentence. A run of the mill strategy for looking over word centrality is to look at the centroid of the report group in a vector space. The centroid of a bunch can be characterized as a pseudo-archive which has a tf-idf score over a predefined confine, here recurrence of word in a group is alluded as tf, and idf regards are usually figured over a significantly greater and relative order of informational index. In Centroid-based Summarization, a focal sentence is the one that contains more words from the centroid of the bunch.

A system of sentences that are indistinguishable to each other is known as group of reports. A couple of sentences are more similar to each other while some others may impart only a little information to whatever is left of the sentences. We estimate that the sentences that resemble expansive parts of exchange sentences in a group are more focal (or striking) to the subject. There are two concentrations to illustrate in this meaning of centrality. To begin with are the methods by which to characterize likeness between two sentences. Second is the best approach to ascertain the general centrality of a sentence given its comparability to various sentences.

We utilize the pack of-words model to speak to each sentence as a N-dimensional vector, to characterize similitude. Here N is the quantity of every single conceivable word in the objective dialect. The quantity of times a word happens in a sentence duplicated by the idf of that word gives the estimation of the consequent measurement in the vector rendition of the sentence. The cosines between two vectors characterize the likeness in the midst of the two consequent sentences:

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x} \text{tf}_{w,y} (\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (\text{tf}_{x_i,x} \text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (\text{tf}_{y_i,y} \text{idf}_{y_i})^2}}$$

Where tf_w = number of times the word w occurs in the sentence s.

The algorithm for Lex-Rank can be summarised as –

```
MInputAn array S of n sentences, cosine threshold t output: An array L of LexRank scores
Array CosineMatrix[n][n];
Array Degree[n];
Array L[n];
for i 1 to n do
for j 1 to n do
CosineMatrix[i][j] = idf-modified-cosine(S[i],S[j]);
if CosineMatrix[i][j] > t then
CosineMatrix[i][j] = 1;
Degree[i] + +;
end
else
CosineMatrix[i][j] = 0;
end
end
end
for i 1 to n do
for j 1 to n do
CosineMatrix[i][j] = CosineMatrix[i][j]=Degree[i];
end
end
L = PowerMethod(CosineMatrix,n,ϵ);
return L;
```

Figure 2-4 Algorithm for Lex-Rank

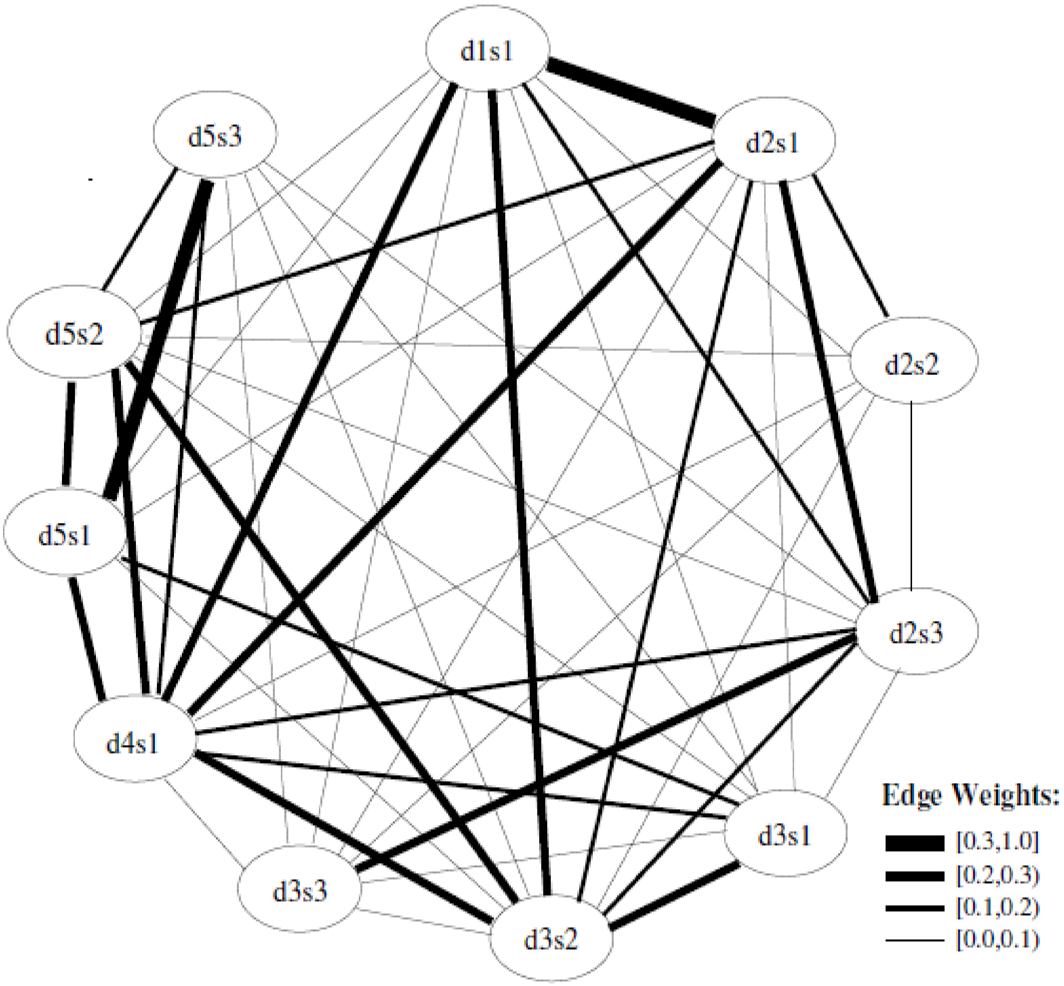


Figure 2-5 Cosine graph for a cluster

For extracting summaries Lex-Rank method uses cosine similarity of TF-IDF vector to estimate semantic similarity between sentences. A graph is build where cosine similarities between sentences become the weights between two sentences. Use Page-Rank algorithm to score the sentences in the graph of weights(cosine similarity) to select top k ranked sentences.

2.1.6 **TEXT-RANK** [6]

Text-Rank is a chart based positioning model for diagrams separated from normal dialect writings. Diagram based positioning calculations are utilized for finding the importance of a vertex inside a chart, in view of general data recursively drawn from the total chart. The major idea executed by a chart based positioning model is that of "voting" or "suggestion". When one vertex is associated with another, it is on a very basic level settling on a decision for that other vertex. The higher the amount of votes that are tossed for a vertex, the higher the importance of the vertex. Also, the importance of the vertex settling on the decision chooses how crucial the vote itself is, and this information is moreover considered by the situating model. Consequently, the score related with a vertex is settled in perspective of the votes that are tossed for it, and the score of the vertices tossing these votes.

Text-Rank for Text Summarisation

To apply Text-Rank, we at first need to collect a diagram related with the content, where the chart vertices are operator for the units to be positioned. For the errand of sentence extraction, the goal is to rank entire sentences; besides, along these lines a vertex is added to the chart for each sentence in the substance.

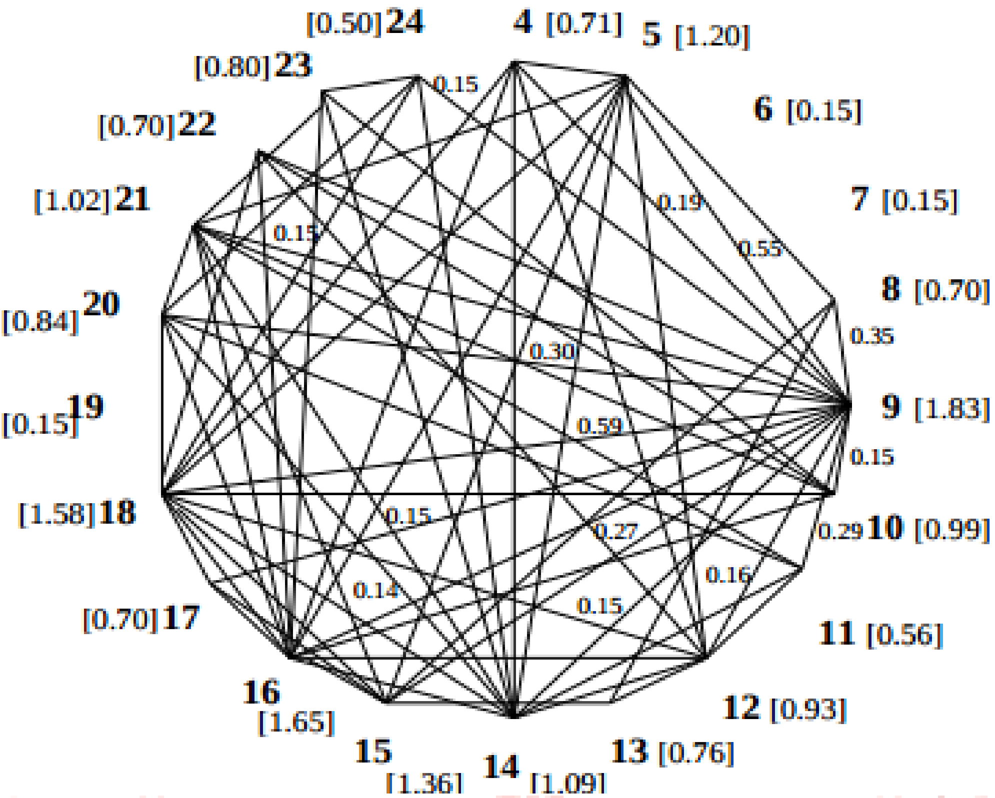
A "comparability" connection is characterized which decides an association between two sentences. "Likeness" is the assessment of the substance overlie of two sentences. The quantity of regular tokens among the lexical portrayals of the two sentences decides the cover between two sentences. The cover can likewise be figured by go through syntactic channels, which just number expressions of a specific syntactic classification, e.g. all open class words, things and verbs, and so on all in all terms let there be two sentences spoken to by S_i and S_j , with a sentence being spoken to by the arrangement of Ni words that show up in the sentence: $S_i = w^i_1, w^i_2, \ldots\ldots, w^i_{Ni}$, the likeness of S_i and S_j can be characterized as :

$$Similarity(S_i, S_j) = \frac{|\{wk \mid wk \in Si \,\&\, wk \in Sj\}|}{\log(|Si|) + \log(|Sj|)}$$

The subsequent chart is exceptionally associated, with a weight related with each edge, showing the quality of the associations set up between different sentence combines in the content. The content is thusly spoken to as a weighted chart.

Summarizing Text-Rank method, for extracting summary, it builds a graph which uses sentences in the review as vertices. Edges have weights which correspond to some measure for semantic similarity. The graph is subjected to page rank algorithm to score the sentences i.e. vertices with underlying assumption that the summary sentences are close or similar to other sentences. The top k vertices based on their score are returned as summary to the review.

3: BC−HurricaineGilbert, 09−11 339
4: BC−Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.



TextRank extractive summary
Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast.

Manual abstract I
Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high wind and seas. Tropical storm Gilbert formed in the eastern Carribean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and in the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.

Manual abstract II
Tropical storm Gilbert in the eastern Carribean strenghtened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15 mph with a broad area of cloudiness and heavy weather with sustained winds of 75 mph gusting to 92 mph. The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Figure 2-6 Sample graph build for sentence extraction from a newspaper article. Manually assigned summaries and TextRank extractive summary are also shown.

2.1.7 KL-SUM [8]

This method greedily adds sentences to a summary as long as it decreases the KL Divergence between the summary extracted so far and the actual review.

$$S^* = \min_{S:words\ (S)\leq L} KL(P_D \,||\, P_S)$$

Where D is the document collection.

P_S is the experimental unigram distribution of the applicant summary S.

KL(P ||Q) represents the Kullback-Lieber (KL) divergence given by

$$\sum_w P(w) log \frac{P(w)}{Q(w)}$$

This sum addresses the divergence between the honest to goodness dissemination P (here the record set unigram circulation) and the approximating conveyance Q (the rundown dispersion). This strategy gives synopsis a part as finding an arrangement of rundown sentences which eagerly organize the archive set unigram conveyance. As per this paradigm, the target of the summarizer is to locate an arrangement of sentences whose length is not as much as L words and whose unigram distribution is as comparative as conceivable to the source report set. The global optimisation of the method is exponential to the quantity of sentences in the record set D. As a guess, KL-Sum utilizes a greedy optimisation technique which uses the following algorithm:

1. Set S = {} and d = 0
2. While |S| <= L do:
3. For i in [1..N_D], d_i = KL(P_{S+Si} || P_D)
4. Set S = S + S_i with minimum d_i and d = d_i
5. Stop if there is no i such that d_i < d

Figure 2-7 Algorithm for KL-Sum

2.2 SUMMARIZATION USING WORD GRAPH [9]

I tried to convert multi-document summarization to single-document summarization using the ideas from [9]. The steps required in this technique are as per the following:

- Sentence Clustering
- Summary Sentence Generation

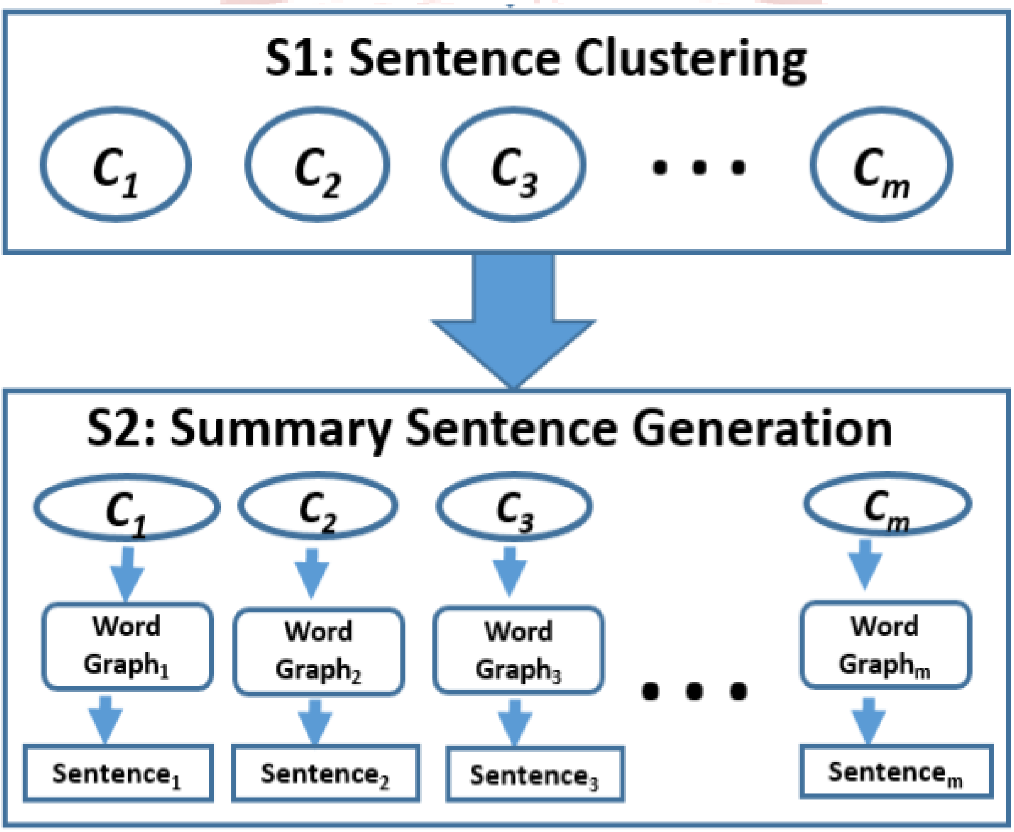


Figure 2-8 Overview of the approach

Sentence Clustering

I represented the sentences as vectors using **weighed BoV, tf-idf BoW**, and the rest of the approaches. Now in order to cluster, this space is very sparse, so clustering in this space is not a good idea. To tackle this problem, I used Principal component analysis to reduce the dimension of the current space and clustered in the new space using different clustering algorithms. We only keep those clusters whose size is greater than some threshold to avoid outliers.

Now we need to order the clusters which will determine how the finally summary will look like. We use **Average Position Ordering** to order the clusters. In this method, the sentences (s_1, s_2, …., s_m) in each clusters C_i are assigned a score, $S(s_i)$ = Position of the sentence in the document. Now for each cluster, we take the average of all the scores,

$$S_{avg}(C_i) = \frac{\sum_{i=0}^{m} S(si)}{m}$$

The clusters are ordered in the increasing order of their score and this is the order in which the sentences will be generated in the summary.

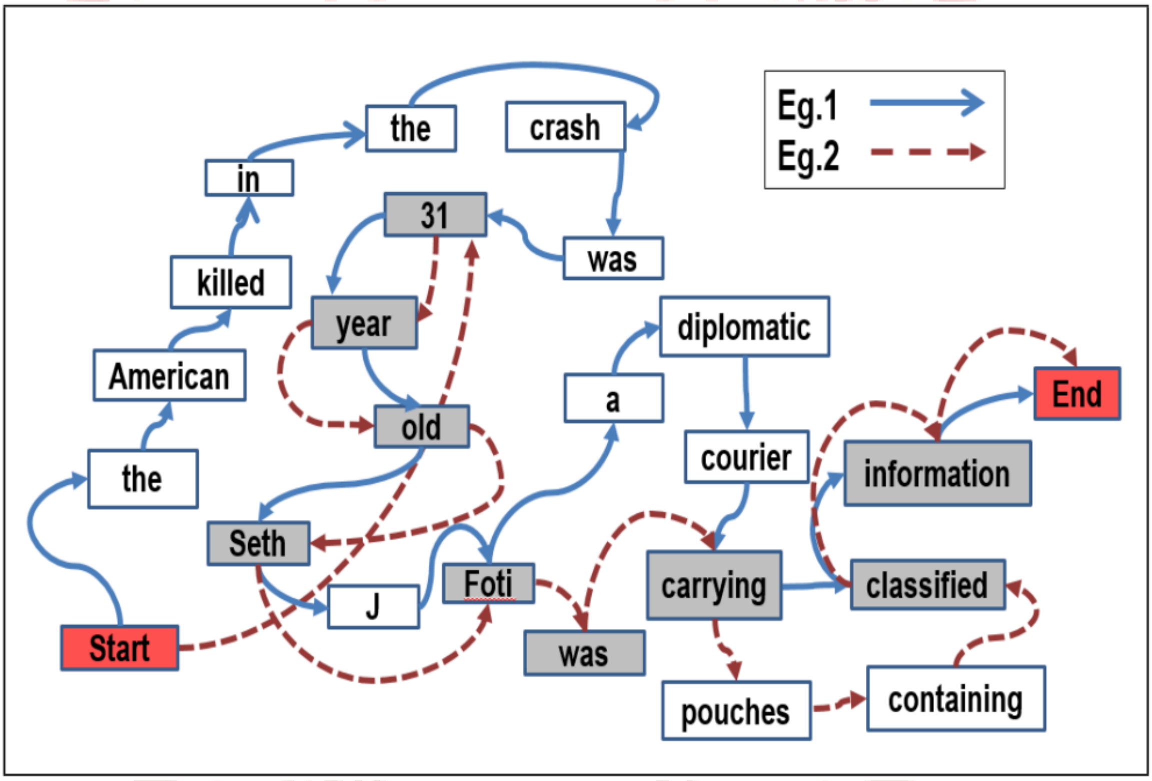


Figure 2-9 Word Graph

Eg.1 *American killed in the crash was 31-year-old Seth J. Foti, a diplomatic courier carrying classified information.*

Eg.2 *31-year-old Seth Foti was carrying pouches containing classified information.*

Summary Sentence Generation

We generate one sentence summary from each cluster. This is because the cluster represent the same type of sentences and basically represent the same/similar information. We speak to each group as a Word Graph where each sentence is associated with a begin and end node and every node speaks to the word alongside its POS label which we gain from a **LSTM POS tagger**. We connect nodes corresponding to the adjacent words in the sentence with a directed edge. Now we list all the possible paths from Start to End and calculate the following measures for them. If there are more than one possible weights for an edge, we assign it the least one.

- **Informativeness**

We use Sentence Ranking algorithms to get the score for each path (sentence) and for each edge in the path, we assign weight equal to this score.

- **Linguistic Quality**

A 3-gram Language Model which was trained on the Dataset (80%) to get the score for each path (sentence) was used. I constructed the Language Model using **CMU Sphinx** speech recognition toolkit and convert it to binary format using **KenLM** for faster computation. For each edge in the path, we assign weight equal to $\frac{1}{1-Language\ Model\ Score}$.

Now in each cluster, we choose the path with the least $\frac{Sum\ of\ weight\ of\ all\ edges\ along\ the\ path}{Number\ of\ edges\ in\ the\ path}$ and hence for each cluster, we have one sentence in the summary.

3. SUPERVISED LEARNING METHODS FOR TEXT SUMMARIZATION

3.1 ATTENTION BASED SEQ-2-SEQ MODEL [1]

A large portion of the Neural machine rundown models utilize techniques in view of the encoder-decoder thought (Sutskever et al., 2014; Cho et al., 2014a) with an encoder to comprehend the information and decoder to give the comparing yield. An issue with a fundamental encoder-decoder approach is that the system needs to collect all essential data of information record into a settled length vector. This makes it hard to manage reports having vast lengths. To address this issue we utilized a model as depicted in [1]. In this approach, we utilize an encoder-decoder display which figures out how to adjust and condense mutually. At each decoder yield, the model looks for an arrangement of positions in the info archive where significant data is totalled. Show at that point utilizes these setting vectors related with the information report positions and the already created words to foresee the following target word. Figure beneath demonstrates a realistic delineation of the model. We utilize a bidirectional RNN for the encoder. It comprises of forward and in reverse RNN's. The forward RNN peruses the info grouping as it is requested (from x_1 *to* x_T) and ascertains an arrangement of forward concealed states ($\vec{h}_1$ ..... $\vec{h}_T$). In a comparative way, the regressive RNN peruses the arrangement in the switch arrange (from x_T *to* x_1) , bringing about a grouping of in reverse shrouded states ($\overleftarrow{h}_1$ ..... $\overleftarrow{h}_T$). We link them to locate the joined shrouded state explanations, h_j = [$\vec{h}_j$, $\overleftarrow{h}_j$]. The decoder now utilizes these comments by utilizing an arrangement model to decide how well the contributions around position j (h_j) and the yield at position i coordinate. The decoder can see the comments for every one of the words in the info record while figuring every y_i and consequently with this approach data can be spread all through the information grouping of explanations and can be specifically recovered by the decoder in like manner.

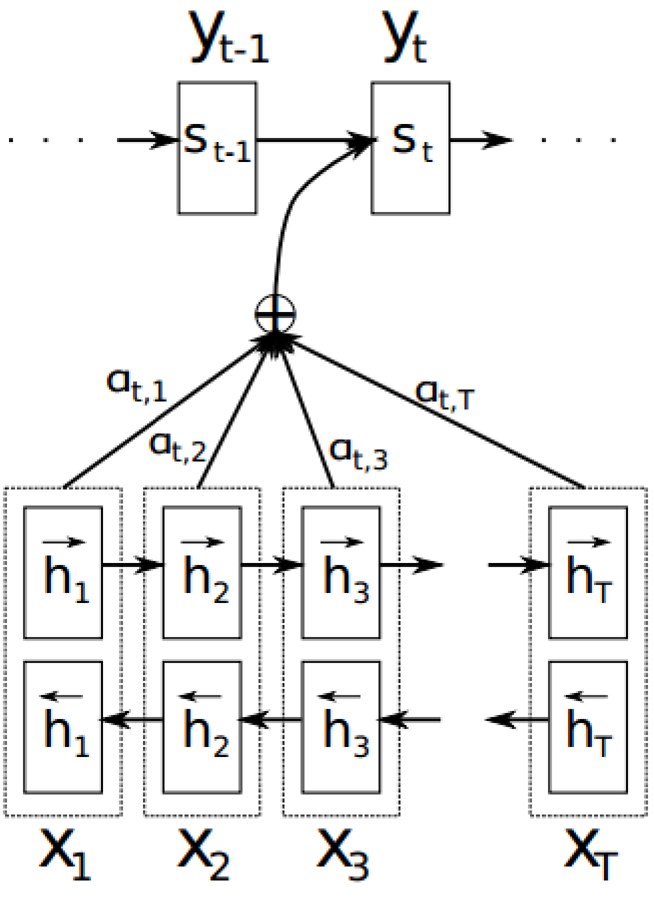


Figure 3-1 Generating y_t given source $\{x_1 \ldots x_T\}$

I used the above described model as an Abstractive Summarization model and it -

- Predicts an outline in view of the setting vectors and all the past produced target words.
- Does not endeavour to encode an entire information sentence into a solitary settled length vector

I used different bucket sizes such as (100, 20), (50, 10) representing (document size, summary size) for the same model to fit different types of training examples accordingly.

4. DATASET USED

The dataset used by me is the “**Amazon Fine Food Reviews”** dataset. It comprises of 568,454 sustenance surveys Amazon clients surrendered over to October 2012 alongside an outline for each audit. The figure below shows a *WordCloud* of the words occurring in the dataset.

Figure 4-1 Word graph of the words occurring in the dataset

www.kaggle.com/snap/amazon-fine-food-reviews

5. EXPERIMENT AND RESULTS

A major setback for me was the quality of the provided review summaries in the dataset. For example, here are the initial few review summaries,

Good Quality Dog Food, Not as Advertised, "Delight" says it all, Cough Medicine, Great taffy Nice Taffy.

As seen from examples, the summaries in the dataset were few word summaries rather than proper review summaries. Clearly, this was detrimental for the supervised learning approach as described in chapter 3 the nature of the dataset also meant that calculating similarity measures between the results from unsupervised learning approaches and the dataset summaries were futile as the results from the unsupervised learning approaches were multi-sentence text summaries in contrast to the few word comments in the dataset. Thus, for approaches described in chapter 2 I have provided **qualitative analysis only**.

For the supervised method, I proceeded with training a model on the available dataset and see how what types of summaries were generated by the model. I did **both quantitative and qualitative** analysis in this case.

5.1 EXPERIMENTS ON THE UNSUPERVISED LEARNING APPROACH

5.1.1 EXPERIMENTS WITH SUMMARIZATION USING SENTENCE RANKS

Since I didn't have good dataset of reviews and summaries, I couldn't evaluate the extracted summaries from unsupervised methods with the ground truth. So I present here some qualitative results of different methods summaries'.

Since the reviews are small i.e. generally 3-4 sentences, so I restricted the number of sentences in summary to be one in one evaluation. In other evaluation, I extracted summaries of size 20% of the size of the review. In both types of evaluations, 95.6% of all the summaries have one sentence.

- **Example 1**

Original: I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.

Summary from Dataset: Good Quality Dog Food

Luhn: I have bought several of the vitality canned dog food products and have found them all to be of good quality.

Edmundson: I have bought several of the vitality canned dog food products and have found them all to be of good quality.

Latent Semantic Analysis: I have bought several of the vitality canned dog food products and have found them all to be of good quality.

Lex-Rank: I have bought several of the vitality canned dog food products and have found them all to be of good quality.

Text-Rank: my labrador is finicky and she appreciates this product better than most.

KL-Sum: The product looks more like a stew than a processed meat and it smells better.

In Example 1, the summaries by Luhn, Edmundson, LSA, Lex-Rank are same. KLSum and Text-Rank extracted entirely different sentence which scored better in their algorithms.

- **Example 2**

Original: Product arrived labeled as Jumbo Salted Peanuts...the peanuts were actually small sized unsalted. Not sure if this was an error or if the vendor intended to represent the product as ”Jumbo”.

Summary from Dataset: Not as Advertised

Luhn: Product arrived labeled as jumbo salted peanuts...the peanuts were actually small sized unsalted.

Edmundson: Product arrived labeled as jumbo salted peanuts...the peanuts were actually small sized unsalted.

Latent Semantic Analysis: Not sure if this was an error or if the vendor intended to represent the product as ”jumbo”.

Lex-Rank: Product arrived labeled as jumbo salted peanuts...the peanuts were actually small sized unsalted.

Text-Rank: Product arrived labeled as jumbo salted peanuts...the peanuts were actually small sized unsalted.

KL-Sum: Not sure if this was an error or if the vendor intended to represent the product as ”jumbo”.

In Example 2, the summaries generated by Luhn, Edmundson, Lex-Rank, Text-Rank methods didn’t actually depict the negativeness of the review but giving the information about the product, LSA, KL-Sum could not capture the negativeness in the review and extracted more negative sentences in summary.

5.1.2 EXPERIMENTS ON SUMMARIZATION USING WORD GRAPH

- **Example 1**

Original: The order arrived as advertised. I have had only a few chances to use it, but it is awesome. The flavor is exactly what I hoped for! It may take me a while to finish the entire gallon, but I will definnantly order again! I like a lot of sesame oil and use it in salads regularly. Great quality, flavor and aroma. Can’t beat that its organic!

Summary: The order again I like a lot of sesame oil and use it is awesome. The flavor is exactly what I hoped for it may take me a while to use it in salads regularly.

- **Example 2**

Original: I have never been a huge coffee fan. However, my mother purchased this little machine and talked me into trying the Latte Macciato. No Coffee Shop has a better one and I like most of the other products, too (as a usually non-coffee drinker!). The little Dolche Guesto Machine is super easy to use and prepares a really good Coffee/Latte/Cappuccino/etc in less than a minute (if water is heated up). I would recommend the Dolce Gusto to anyone. Too good for the price and I’am getting one myself! This offer is a great price and a great taste, thanks Amazon for selling this product.

Summary: I like most of the Dolce Gusto to use and prepares a minute if water is heated up. However my mother purchased this little machine and talked me into trying the Latte Macciato. No Coffee Shop has a better one and I would recommend the Latte Macciato.

5.2 EXPERIMENTS ON THE SUPERVISED LEARNING APPROACH

Due to computation limitations, I trained our Attention based Seq-2-Seq model on a train-set of 10000 review-summary pairs from the dataset and sought test results on a set of 1000 reviews. Shown in the table below are model performances based on *Blue* and *Rouge* metrics.

Method	Bleu-1	Bleu-2	Bleu-3	Rouge-L
Seq-2-Seq with Attention	0.0379	0.0060	0.0003	0.05984

The results of the model are not very good which is as expected from the discussion above regarding the poor quality of training set summaries. I also present some examples for analysis. **Qualitative Results**

Shown below are a few examples of my model's results as compared to dataset summaries.

- **Example 1**

Original: this saltwater taffy had great flavors and was very soft and chewy. Each candy was individually wrapped well. None of the candies were stuck together, which did happen in the expensive version, fralinger's. Would highly recommend this candy! i served it at a beach-themed party and everyone loved it!

Summary from Dataset: great! just as good as the expensive brands!

Summary form Model: great select for food .

- **Example 2**

Original: i was looking for a natural energy shot that works, and i found it. steaz energy shot is my favorite! i recently had gastric bypass surgery, so tons of sugar is a no-no. and i'm also allergic to sucralose (splenda) so most other energy drinks are out. i tried bazi energy shots and they didn't work as well as steaz, and definitely didn't taste as well. these have a taste that's almost like coca-cola–not bad at all. if you're looking for natural energy, give these a try.

Summary from Dataset: fantastic, natural energy

Summary form Model: great energy find

On going through the dataset and model summaries I found that the model learnt to produce frequently occurring words such as *great, good* with high probability. This is also seen from the examples above. As the summaries in the dataset have very few words highlighting the reviewers quick opinion, the model also tried to predict such short opinions and generated common words *great, good*, etc very frequently.

6. CONCLUSION

I understood and implemented different approaches on top of libraries in supervised and unsupervised learning for the text-summarization problem. Since the dataset is not as good as to conduct evaluations on text-summarization, I could not comprehensively evaluate on the dataset. Though I could not provide quantitative results for all approaches, I added qualitative results by showing the summaries generated by various methods and giving potential explanation for the results generated. The results were quite promising and believe that I was able to explore many approaches. I was able to get nice summaries through the unsupervised methods. My goal of exploring the approaches of text-summarization was successfully achieved.

7. FUTURE WORK

One can build a small and good dataset of reviews and summaries and evaluate the unsupervised extractive methods mentioned in the methods section. This would help increase the credibility of the approaches and also fine tune the models.

Another application I thought was generation of *product based summaries* using all the reviews of a particular product. I did some work on this problem, was not able to fully implement the approaches [9] of generating summary for a product based on reviews on the product. So we could work on implementing them completely and qualitatively understand and evaluate the summaries generated for every product.

8. REFERENCES

[1.] Bahdanau, D., Cho, K. & Bengio, Y. “Neural machine translation by jointly learning to align and translate”. In *Proc. International Conference on Learning Representations* http://arxiv.org/abs/1409.0473 (2015).

[2.] ”The Automatic Creation of Literature Abstracts *Luhn*” IBM Journal of Research and Development (Volume: 2, Issue: 2, Apr. 1958)

[3.] H. P. Edmundson, “New Methods in Automatic Extracting”, Journal of the ACM (JACM), v.16 n.2, p.264-285, April 1969

[4.] J. Steinberger and K. Ježek, "Using latent semantic analysis in text summarization and summary evaluation," in Proceedings of ISIM 2004, Roznov pod Radhostem, Czech Republic, April 2004, pp. 93--100.

[5.] Günes Erkan , Dragomir R. Radev, “LexRank: graph-based lexical centrality as salience in text summarization”, Journal of Artificial Intelligence Research, v.22 n.1, p.457-479, July 2004

[6.] R. Mihalcea, P. Tarau, "TextRank: Bringing order into texts", *Proc. Conf Empirical Methods Natural Language Processing*, 2004.

[7.] Lucy Vanderwende , Hisami Suzuki , Chris Brockett , Ani Nenkova, “Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion”, Information Processing and Management: an International Journal, v.43 n.6, p.1606-1618, November, 2007

[8.] Aria Haghighi , Lucy Vanderwende, “Exploring content models for multi-document summarization”, Proceedings of Human Language Technologies: May 31-June 05, 2009, Boulder, Colorado

[9.] Siddhartha Banerjee , Prasenjit Mitra , Kazunari Sugiyama, “Multi-document abstractive summarization using ILP based multi-sentence compression”, Proceedings of the 24th International Conference on Artificial Intelligence, p.1208-1214, July 25-31, 2015, Buenos Aires, Argentina

[10.] A. Nenkova and L. Vanderwende. 2005. “The impact of frequency on summarization. Technical report, Microsoft Research”.

[11.] J. L. Neto, A. A. Freitas, C. A. A. Kaestner, "Automatic text summarization using a machine learning approach", *Proc. 16th Brazilian Symp. Artif. Intell.*, pp. 205-215, 2002-Nov.-11–14.

[12.] Axelrod, A., He, X., and Gao, J. (2011). “Domain adaptation via pseudo in-domain data selection”. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 355–362. Association for Computational Linguistics.

[13.] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I. J., Bergeron, A., Bouchard, N., and Bengio, Y. (2012). Theano: “new features and speed improvements. Deep Learning and Unsupervised Feature Learning”

[14.] [Berg-Kirkpatrick et al., 2011] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. “Jointly Learning to Extract and Compress”. In Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-HLT 2011), pages 481–490, 2011.

[15.] [Chen and Manning, 2014] Danqi Chen and Christopher D Manning.“A Fast and Accurate Dependency Parser using Neural Networks”. In Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 740–750, 2014.

[16.] Conroy, J.M., J. Schlesinger and J. Goldstein Stewart. 2005. CLASSY “Query-Based Multi-Document Summarization”. In *Proceedings of DUC 2005*.

[17.] Y. Gong, X. Liu: *“Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis”*. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, United States 2001, pp. 19-25

[18.] R. Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, H. Qi, A. Celebi, D. Liu, E. Drabek: *“Evaluation Challenges in Large-scale Document Summarization”*. Proceeding of the 41th annual meeting of the Association for Computational Linguistics, Sapporo, Japan 2003, pp. 375-382