Automatic Text Summarization Using Importance of Sentences for Email Corpus

by

Sravan Nadella

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved June 2015 by the
Graduate Supervisory Committee:

Hasan Davulcu, Chair
Baoxin Li
Arunabha Sen

ARIZONA STATE UNIVERSITY

August 2015

ProQuest Number: 1597382

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 1597382

Published by ProQuest LLC (2015). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

With the advent of Internet, the data being added online is increasing at enormous rate. Though search engines are using IR techniques to facilitate the search requests from users, the results are not effective towards the search query of the user. The search engine user has to go through certain webpages before getting at the webpage he/she wanted. This problem of Information Overload can be solved using Automatic Text Summarization. Summarization is a process of obtaining at abridged version of documents so that user can have a quick view to understand what exactly the document is about. Email threads from W3C are used in this system. Apart from common IR features like Term Frequency, Inverse Document Frequency, Term Rank, a variation of page rank based on graph model, which can cluster the words with respective to word ambiguity, is implemented. Term Rank also considers the possibility of co-occurrence of words with the corpus and evaluates the rank of the word accordingly. Sentences of email threads are ranked as per features and summaries are generated. System implemented the concept of pyramid evaluation in content selection. The system can be considered as a framework for Unsupervised Learning in text summarization.

To all who supported me

ACKNOWLEDGEMENT

Firstly, I would like to thank my Professor Dr. Hasan Davulcu, for his esteemed guidance, without which my thesis would not have been possible. His expertise and advice has been very helpful in completing my thesis. I would also like to thank Professor Dr. Baoxin Li and Professor Dr. Arunabha Sen for their support. I would like to thank my roommates and friends for their constant support. Finally, I would like to thank my family for having faith in me and always guiding me in pursuing dreams in my life.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

Chapter 1

INTRODUCTION

1.1 Motivation

In the present age of Internet and due to rapid growth of broadcast systems, there is massive amount of information being available online. Search Engines employs the method of constant indexing in order to accumulate the growing information in World Wide Web. Once the user enters the search request, documents are retrieved. The classic problem of Information Overload comes into play as the search engine retrieves hundreds of documents as search results. Although the retrieval time for the search is very less, the user has to go through the documents in order to attain at document he/she is searching for, because most of the naïve users are reluctant to make cumbersome effort of going through each of the documents [1]. As of 2009, the entire world wide web was estimated to contain close to 500 exabytes which one half zettabyte [2]. But by 2013, it is estimated to have reached 4 zettabytes [3]. With these statistics, one can understand the enormous amount of information available and need for summarization not only for saving the search time but also for having a cut short understanding of information available. Automatic Text Summarization is one such area of Data Mining, which deals with the classic problem of Information Overload as mentioned above. Summarization has been interest of study in the field of Computer Science from so long. But with growing large data sets, the research on Automatic Text Summarization has become the study of hour.

1.2 Document Outline

The rest of the document is organized as explained below.

Chapter 2 background Information

Chapter 3 is about the related work done in Summarization

Chapter 4 tells about the corpus used in this thesis.

Chapter 5 gives the detailed study of proposed method.

Chapter 6 discusses the results and evaluation methods

Chapter 7 gives the Conclusion

Chapter 2

BACKGROUND

2.1 Data Mining

Data Mining is the field of Computer Science, which is a combination of Artificial Intelligence, Machine Learning, Statistics and Data Base Systems. Data Mining is the process of obtaining at pattern from very large data sets. The goal is to process the data and obtains at human understandable patterns, which are unknown when seen unprocessed from a very large data sets. Anomaly detection, Association rule learning, clustering, classification, regression and summarization are the most common task in data mining. Data Mining is considered as synonym for Knowledge Discovery from Data (KDD)[4]. Data pre-processing, data transformation, data mining, pattern evaluation and presentation are the main steps involved in this process.

2.2 Natural Language Processing

Natural Language Processing is the unique combination of Artificial Intelligence and Computational Linguistics. The main idea in NLP is the make computers understand human languages. NLP tries to achieve the classic case of human – computer interaction as such of human – human interaction. This has been a very challenging problem of computer science as it requires handling various concepts of natural language understanding [5]. NLP systems used to be hard-coded because of complexity and numerous rules of human language. This hindered the growth in research of NLP. But with the increased advancements in Machine Learning techniques, the NLP systems are trained to learn the essentials of human language and thus making towards perfection. Parts of speech tagging process are one example of NLP system, which assigns the specific POS to the word. Decision tree, Hidden Markov Model, Statistical model is some machine learning techniques that are fruitful in training NLP systems.

2.3 Text Mining

Text Data mining is the process of getting patterns from text data. Bag of words and NLP based techniques are two prominent text-mining approaches. Bag of words won't deal with the morphological or semantic structure of input text, whereas NLP based techniques deal with some of morphological structure of text. The basic idea behind text mining is similar to that of data mining. Structuring the data, deriving patterns, evaluating and presenting the patterns are the common steps of text mining.

2.4 Information Extraction

Information Extraction (IE) concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from unstructured text.[6] Although output of IE varies from case to case, the underlying goal of IE is to represent the structured information we get into a database.

The primary focus on IE is because of its un-structured or semi-structure input. This helps in evaluating, and comparing, different Natural Language Processing technologies. Unlike other NLP technologies, ML for example, the evaluation process is concrete and can be performed automatically. The fact that a successful extraction system has immediate applications has encouraged research funders to support both evaluations of and research into IE [5].

2.5 Information Retrieval

Information Retrieval is the field of Computer Science, which is majorly used in web search engines today. It is the process of obtaining information resources relevant to a query from large collection information retrieval. Unlike traditional database retrieval, the large information collection is mostly semi-structure data like text. Information retrieval processes the data with syntactic methods leaving behind the semantic understanding of data (NLP). Information retrieval systems are one step towards solving the information overload problem. User enters a query, IR system process the query on its available data and returns a set of documents as results.

2.6 Machine Learning

Machine Learning is a sub field of Computer Science, which is a combination of Pattern recognition, and Artificial Intelligence. The idea of machine learning is to make computer program learn and predict the data. It can be viewed as a scenario where results are obtained without being explicitly programmed. The main application of Machine Learning is in problems where explicit programming is infeasible. Spam detection is one such area where there are no rules in identifying spam. Instead it is done through examples.

2.6.1 Supervised Learning

Supervised learning is a two-step Machine Learning process. The input for the system would be training data set from which the system learns and a testing data set on which the system would predict the hidden patterns or result that need to be obtained.

2.6.2 Unsupervised Learning

Unsupervised learning is a tough Machine Learning process. In this, there won’t be any labeled data as in Supervised Learning. The labeled data is the training data set that is provided to the system in Supervised Learning. In Unsupervised Learning, only test data is given and is expected to find patterns.

2.7 Automatic Text Summarization

Automatic Text Summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user and task [7]. The goal of this process is to obtain at a summary, which is very short in length without loosing the ideology of the document

With the increasing problem of Information Overload and data availability on Internet, the quality of results of search query of user is reducing. This made the need of research in the area of summarization very prominent. As NLP is still in evolving process, Information Retrieval based Automatic Text Summarization gained prominence. IR based summarization techniques mostly rely and work at word level. With the advancements in Machine Learning and graph theory techniques, research in summarization has escalated to sentence level. Broadly, Text Summarization can be classified as two approaches, Extractive and Abstractive.

2.6.1 Extraction based Summarization

The idea behind extraction-based summarization is to select a subset of sentences from the document and represent as summary. The sentences to be selected are based on word level features like frequency, key phrases etc. In some cases, Extractive summaries can be just a set of key phrases. Those types of summaries are predominant in news domains.

2.6.2 Abstractive based Summarization

In Abstractive based summarization, instead of selecting sentences or key phrases, the document is re-phrased to create a summary. This type of summarization builds internal semantic structures and uses NLP techniques to re-phrase the document. Although the ultimate goal of Automatic Text Summarization is to obtain abstractive summary that is very similar to human produced summary, lack of advancements in Natural Language processing and Understanding has hindered its research. Present day research is based on extractive based summarization systems to get at greater recall for summary with respect to original document.

2.6.3 Maximum Entropy based Summarization

A multi-document Text Summarization System was developed as a hybrid system using naïve bayes classifier and statistical language models. This was developed during DUC 2001 and DUC 2002. The researchers wanted to explore the effectiveness of a maximum entropy classifier. Maximum Entropy summarization is majorly developed for meeting summarizing task. Maximum Entropy is quite based on robust features instead of normal feature dependencies. Now, this type of summary is very useful in news domains.

2.6.4. Aided Summarization

Although automatic abstractive summarization is the ultimate goal in this field, extractive summarization is the main area researchers are working on. With implementation of machine learning techniques and information retrieval techniques, the extraction based summarization proved good to certain level. Aided Summarization is one kind of summarization where Human judged summary plays part in the overall summarization system. In Machine Aided Human Summarization (MAHS), users are aided in task of summarization, whereas, in Human Aided Machine Summarization (HAMS), humans check after, the generated summary from computer program.

2.7 Summary Evaluation

Unlike other problems in computer science, there is no gold-standard evaluation method for summarization. The goal is to be obtaining at a summary, which is very similar to human generated summary. As different authors has their unique way of writing and understanding, reaching at a gold standard summary standard has been difficult. Within extractive and abstractive methods of summarization, evaluation of extractive based summarization is kind of fruitful as the process involves selection of sentences within the document, whereas, in abstractive-based summarization, generated summary should be compared with human written summary through re-phrasing the sentences in the document. There are two types of evaluation approaches in automatic text summarization. ROGUE (Recall-Oriented Understudy for Gisting Evaluation) is most commonly used evaluation technique. This method relies on recall-based evaluation. The generated method will be compared with human generated summary for recall measure.

One other method of evaluation that is predominant is pyramid evaluation. The idea is to generate summaries using pyramid method and also generate pyramids for human generated summaries and evaluate basing on pyramid scores.

2.7.1 Intrinsic Summary Evaluation

In this type of summary evaluation, the system is tested within itself. The evaluation is pretty much confined to the quality of summary and quotient of information it can convey.

2.7.2 Extrinsic Summary Evaluation

In this type of summary evaluation, the system is tested in comparison with a real world task. The main aim in this type of evaluation is to assess the summary to the relevance measure or how affective this summary can be used for reading comprehension

Chapter 3

RELATED WORK

Waibel et al(1998) worked on meeting summarization where they employed Maximal marginal Relevance algorithm to select best sentences. They majorly worked on the effects of spoken communication vs written communication.

Murray et al(2005,2006,2007) compared text summarization approaches with feature based approach and also worked on speech-specific characteristics. Features such as speaker status, discourse markers and high level meta comments in meetings.

Rambow et al(2004) considered the problem of email summarization as binary classification problem. Feature set is applied on every sentence and a question whether the sentence could be selected for summary or not is answered. These features acted as attributes in classification problem, thereby letting the system to take a decision to include the sentence to summary or not. In this paper a set of 14 features were considered as feature set.

Muresan et al(2001) summarized individual email messages separately. Noun phase extraction rules are learned from machine learning approaches. The approach is much similar to single document classification.

Chapter 4

CORPUS

Though the research in automatic summarization both extractive and abstractive way is intensive, there isn't any perfect corpus to train and test the system because of the lack of perfect evaluation method. This made the necessity of having corpus particularly dedicated for summarization research more. Researchers and research group have started working on developing corpus to experiment, train and test the system. Annotated at different situations and scenarios, Enron corpus, TREC corpus, w3c corpuses are famous and are often used for research. Enron corpus is the email thread from Enron corporation that was made available as part of legal investigation on it. TREC corpus is the email conversation of enterprise which of highly technical. W3c corpus is that is crawled from web consisting of pdf, doc, docs, emails etc. The corpus being considered here is the 3 annotations of 40 emails thread from w3c corpus

40 email threads from w3c corpus such that the average number of emails per thread is 11 are taken[8]. They employed 10 students from University of British Columbia's Department of Computer Science and

Psychologists as annotators, Each of these annotators was asked to annotate 9 3 different threads. Each annotator will write 250 words abstract summary along with extractive summary where importance sentences are picked. Along with writing abstractive summary, each sentence of abstractive summary has to be linked with the sentences of original email thread. Email speech acts were also incorporated into the annotation. Each sentence is identified as one of propose, request, Commit, Agreement/Disagreement, Meeting. Meta sentences are made identified as sentences, which refer to the discussion in, email itself.

Chapter 5

Proposed Method

Automatic Text Summarization is a data mining process, which work towards obtaining a short version of a document, which conveys most content as that of the original document. The steps involved in text summarization are similar to that of any data mining procedure. As discussed in previous chapter, there are 40 emails threads in corpus with an average of 11 emails per thread. The initial step of this thesis is to parse the xml document in which 40 email threads are embed. Java API for XML processing (JAXP) is used to get the data into the program. The tasks in this process can be categorized broadly as three phases. They are

- Pre-Processing,
- Feature Selection and Sentence Ranking
- Evaluation.

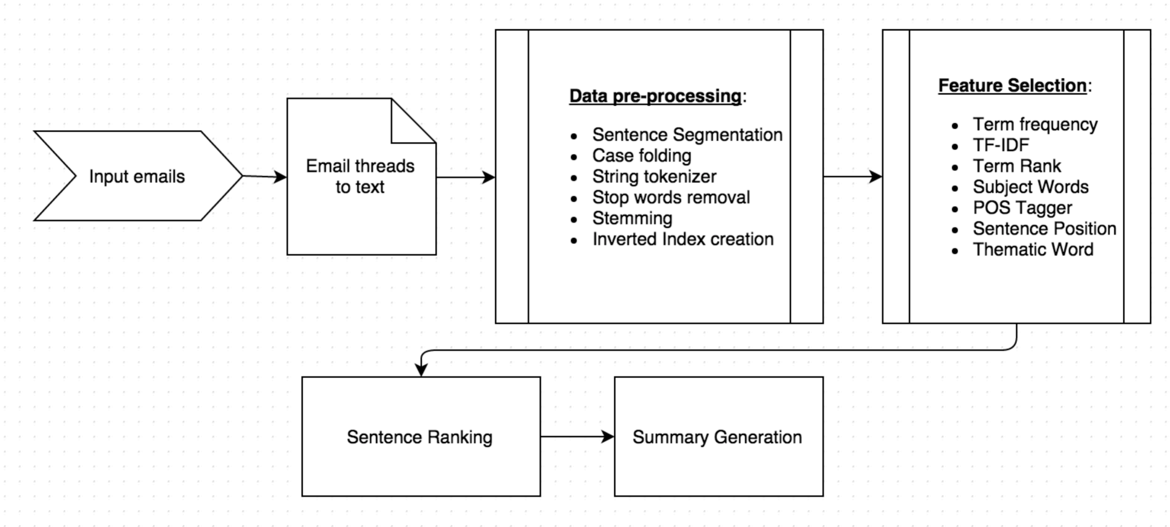


Figure 5.1: Architecture diagram of the summarization System.

5.1 Pre-Processing

Pre-Processing is the first step in a data mining process. The idea of this step to take the unstructured data and make it structured so that it can be processed. Unstructured data can have missing values, bad values or anything that can lead to unwanted results. Data Pre-Processing helps in getting data clean and ready for processing. In text summarization, pre-processing is a step where the document is parsed and important words are identified leaving behind stop-words and other unrelated stuff from documents. The main steps of pre-processing are

- Sentence Segmentation
- Case Folding
- String Tokenization
- Stop-Word Removal
- Stemming
- Creating Inverted Index.

5.1.1 Sentence Segmentation

Input document is divided into sentences. The end of sentence is identified by punctuation marks. An Regular expression is used to split the document into sentences. This regular expression is constructed in a way to identify all punctuation marks that could end a sentence.

5.1.2 Case Folding

The sentences, which are divided from text documents, are taken as the input for case folding. Case folding is a process of converting text into either Upper case or to lower case. Though this step might be trivial, it ensures there won't be any wrong calculations in ranking words due to different case issues.

5.1.3 String Tokenization

Once the case folding is done, sentences undergo string tokenization. It is the process of breaking a sentence into words, phrases or any meaningful elements. These meaningful elements are called tokens. These tokens are then processed through text mining or feature selection step.

5.1.4 Stop – Words Removal.

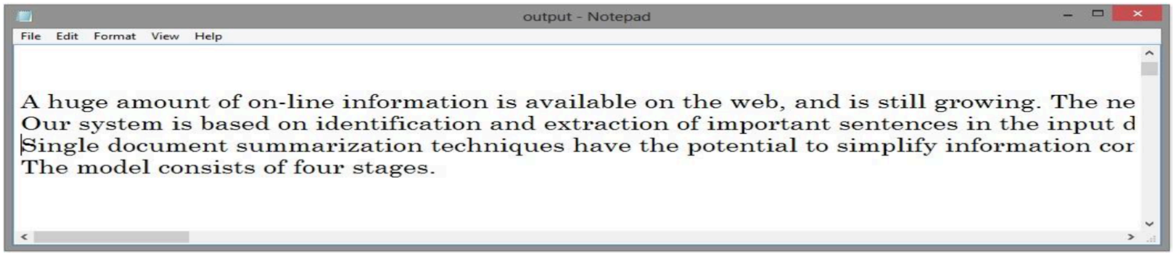
Stop – Words are very common in documents and occur in majority of documents. Stop – words don't provide any semantic information. Most stop-words are related to language such as article, auxiliary verbs etc. If dealing with a small document, stop –words won't be a problem but when dealing with a large sized document or multi- document summarization, removing stop – words would help in reducing space and time complexities. As stop – words really don't provide any information at word level, removing stop – words tokens would help in normalizing the word ranks.

5.1.5 Stemming

Stemming is a process identifying the root word from its other from like past tense word, future tense word etc. For example, work, works, working, worked come from the root word of word. While ranking the words, assigning different rank to each of above words will diverse the importance of word work in the whole document. Instead if stemming is applied on above words and rank is assigned to the root word work will help in increasing the weightage of word work in document irrespective of its grammar. There are two types of stemming, One is derivational stemming where a new word is created like musical to music, whereas in second type of stemming inflectional stemming, word is obtained by removing ending characters like worked to work. Porter – Stemmer algorithm is used for stemming process in this thesis.

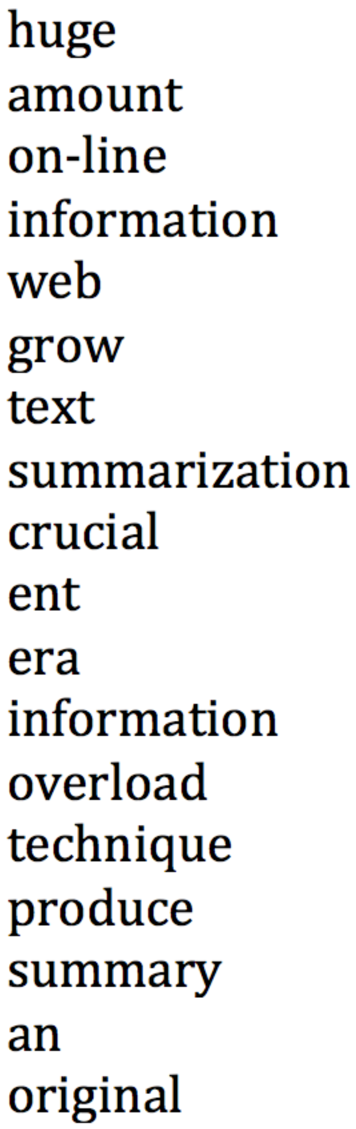
5.1.6 Creating Inverted Index

Once all the above-mentioned steps are done, we obtain at structured data from unstructured text. The result would be a collection of words/tokens from document, which identifies the significance of words. Following diagrams depicts how pre-processing is done and the respective output.



```
A huge amount of on-line information is available on the web, and is still growing. The ne
Our system is based on identification and extraction of important sentences in the input d
Single document summarization techniques have the potential to simplify information cor
The model consists of four stages.
```

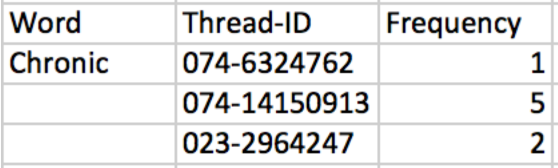
Figure 5.2: Input Document



```
huge
amount
on-line
information
web
grow
text
summarization
crucial
ent
era
information
overload
technique
produce
summary
an
original
```

Figure 5.3 Structured data after Pre – Processing

These words obtained after pre-processing are stored in data structures. Inverted index, the process of generating a data structure, which stores the words and its association with emails in threads, is used for this purpose. The main purpose of implementing inverted index is to allow fast text searches at the cost of processing while adding documents. Once the inverted index is created, the search for specific words during feature implementation or sentence ranking would be fast. Following diagram depicts how a word is stored in inverted index. This concludes the pre-processing phase and the structured data is available for Feature implementation and sentence ranking.



Word	Thread-ID	Frequency
Chronic	074-6324762	1
	074-14150913	5
	023-2964247	2

Figure 5.4 Inverted index representation of a word

Here in Inverted index, each word has two major attributes, a list of email threads in which they occur and the frequency with which they occur in each document. By this representation it would be faster computation in word ranking or sentence ranking in case we want to retrieve the email threads or frequencies.

5.2 Feature Selection and Sentence Ranking

Once the pre-processing is done, we obtain at list of keywords that are significant to the documents. Features that help in identifying importance sentences are applied on these words and subsequently words and sentences are ranked. Before discussing on how sentences are ranked using these feature metrics, following sections provide a brief introduction and implementation part of these features.

5.2.1 Term Frequency

As the word indicates, Term Frequency gives a picture of presence of a word in the document. When working on a corpus, Term frequency can indicate either frequency of occurrence in whole corpus or frequency of its occurrence in the document under consideration. This work perceived Term Frequency confining to a particular document not corpus. Also there are different weighting schemes for Term Frequency.

Binary weighing scheme would be having {0,1} value set. If the word is present in the document, it indicates 1 else it would indicate 0. Whereas raw frequency weighing scheme would represent the actual count on how many times the word occurred in the document. There are other weighing schemes like log normalization, double normalization.

Binary Weighing Scheme: $TF\ (t, d) = \{0,1\}$

Raw Frequency: $TF\ (t, d) = F\ (t, d)$

Double Normalization:

$Tf(t,d) = 0.5 + ((0.5 * f(t,d))/ \max\{f(w,d): w \text{ belongs to } d\})$

Figure 5.4: Weighing Schemes of Term Frequency

Where TF (t, d) indicates Term Frequency of Term t in Document d.

F (t, d) is number of times Term t occurred in Document d

5.2.2 Term Frequency – Inverse Document Frequency

Inverse – Document Frequency indicates the importance of word in the whole corpus. In other terms, it provides the specificity of the term. Though some words occur across all documents very frequently, they don’t carry any significance to the documents. With only

Term Frequency in consideration, these words can falsely increase the weightage of the word, thus increasing the weightage of sentence. Inverse – Document Frequency would normalization this affect.

Idf(t, D) = log(N/ |{d belongs to D : t belongs to d }|)

Figure 5.5: Inverse – Document Frequency calculation

Where idf(t, D) denotes Inverse – Document Frequency of Term t in Document d. |{d belongs to D : t belongs to d }| denotes Number of Documents in which Term t occurred

Term-Frequency - Inverse Document Frequency is the combination of term frequency and inverse document frequency. Term Frequency pertaining to whole corpus is considered in this feature. In a way Term Frequency – Inverse Document Frequency would give a better picture on how important the word is to the document.

Tfidf(t, d, D) = f(t,d) * log (N/n_t)

Figure 5.6: Term Frequency – Inverse Document Frequency

5.2.3 Term Rank

Term Rank proposed by Davulcu, Gelgi, Vadrevu is used [9]. This term rank is a solution for the ambiguous term searching. For example the query apple can be referred as either fruit or computer [9]. The main idea of this concept is to extract a relational graph among the key words such that the co-occurrence of words can be determined. Nodes of graph being the collection of terms and edges represent the association strength between the terms.

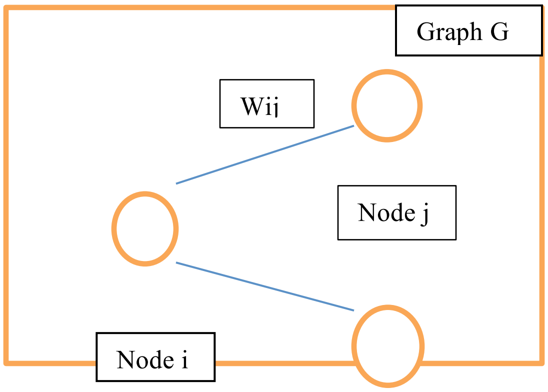


Figure 5.7: Relational Graph

The Relation Graph is a weighted undirected graph where nodes like i,j,k are the terms of corpus while Wij is count on how many times the word i and word j appeared together in corpus. This formula of Term Rank is a variation of Page rank algorithm [9]. The Term Rank of a word is called TR(i).

$$TR(i) = \sum_{j \in \mathcal{N}(i)} \frac{TR(j).w_{ij}}{\sum_{k \in \mathcal{N}(j)} w_{jk}}$$

Figure 5.8 Basic Term Rank Formula

Term rank of a node in the graph is determined through the cluster of words that occurred along with it. It is weighted summation over the nodes it is connected to. Also this one iteration doesn’t give the term rank of the word. Like the page rank it is made converged through multiple iterations.

$$TR^{(0)}(i) = \frac{w_i}{\sum_{j\in\mathcal{V}(\mathcal{G})} w_j} = TF(i)$$

$$TR^{(t+1)}(i) = \sum_{j\in\mathcal{N}(i)} \frac{TR^{(t)}(j).w_{ij}}{\sum_{k\in\mathcal{N}(j)} w_{jk}}$$

Figure 5.9: Term Rank in Successive Iterations

Initial value of Term Rank of a word is generally considered to be Term frequency of the word. In a way, it can be said that the iteration process converges the rank of term with respect to the co-occurrence of it with other terms thus depicting the real importance of it in the corpus. In this work a convergence factor of 0.000001 is used.

5.2.4 Subject Words

This feature identifies the words that have direct connection to the topic of document. Words in headings in a document or Subject field of an email are examples of this kind. The concept behind this feature is that the sentence that has words of headings in it are directly related to the topic of document, thus hold significance to the summary. In this work, subject field from each of the emails are collected and stored as words. While ranking the sentences using word features, these subject words list is referenced and the sentence is weighed accordingly.

5.2.5 Parts of Speech (POS) Tagger

POS tagger also called grammatical tagging is a main Natural Language Processing technique. As the name indicates, POS tagger assigns a word with its respective parts of speech. During the process of tagging, the program not only considers the definition of word but also the context of word in the sentence and assigns a tag. The tags are of numerous combinations like Noun, Cardinal Number, Noun Plural, Determiner, Verb, Adjective and Adverb etc. POS tagging technique is a supervised learning algorithm. The input to the system would be of two files, one being the training model from which the system learns and second being the file that needs to be POS tagged. Though there are some Unsupervised POS taggers, Supervised Taggers are most common. There are many POS tagger available as part of Open Source, this work considered Mark Watson's FastTag POS tagger. This POS tagger used Eric Bill's training model of lexicon and association rules for POS tagging.

5.2.6 Sentence Position

This feature is about the sentences that are prominent to extract. The idea is that the sentences that are significant to be a summary sentence occurs either at the start of document/paragraph or the end of document/paragraph. This features gives weightage to the those kinds of sentences.

5.2.7 Thematic Words

Thematic words are the words that occur with high frequency with in documents. Separating those kinds of words and giving more weightage is essentially the idea of this feature. In this words with higher frequency are made accountable towards summary.

5.3 Sentence Ranking and Summary Generation

Once all the features are applied on structured data obtained from data pre – processing step, Sentences of the document/email are ranked. Some of the word features depends on context and the place where they occur. POS tagging is one example for this. So the ranking process is divided into two parts. First part of ranking is done at word level, features of a word from the sentence are weighed. In the second part of ranking, sentence rank is summed over all the words of the sentence. Following two equations depicts the process

$\text{Score}(l, w) = \Pi\, i(w)$

$\text{Score}(l) = \Sigma\, \text{Score}(l, wi)$

Where l, denotes the sentence number and w denotes the word that occurred in the sentence

Chapter 6

RESULTS AND EVALUATION

6.1 Precision Results

BC3 Corpus is provided with two XML files, one being the corpus itself and second file being the annotated summary from students of British Columbia University. Each Email thread has 3 summaries extracted by three different annotators. To evaluate the results, same number of sentences as that of the annotated summary is extracted. In this scenario, there won't be any difference between Precision and Recall as the number of sentences is same. Following table represents the precision values of some of the threads with each of the annotated summaries.

Thread – ID	Annotated Summary I	Annotated Summary II	Annotated Summary II
074-6324762	40	59.09	37.5
059-11070771	12.5	16.67	16.67
007-7484738	58.82	37.5	62.5
079-4736087	16.67	40	61.5
067-11978590	35	54.55	33.33
074-14150913	50	50	41.67

Table 6.1: Precision Results for email threads with annotated summaries

6.2 Precision Results of Individual features

Above table is the precision result for all features combined. Following table gives detail account on how individual feature affect the quality of summary. The precision value will be the average of three annotated summaries over all threads.

Feature	Average Precision over all threads
Term Frequency	26.38
Term Frequency – Inverse Document Frequency	25.35
Term Rank	24
Subject Words	9
POS Tagger	25
Combination of all Features	28

Table 6.2: Average Precision over different features

6.3 Pyramid Evaluation

There isn't any best summary evaluation model or mechanism. The major employed options are calculation of precision, recall and f-measure. Summaries are generated from humans. The mechanism being employed in this work is of pyramid method of Nenkova and Passonneau in 2007 [10]. In pyramid method, summary content units are defined and each sentence is weighted with respect to the summary content units. After weighing each summary, the pyramid is formed. Level n of pyramid consists of sentences which are weighted n.[10].

In this work, annotators from corpus data select the summary sentence. Pyramid is constructed based on the concept that weight of sentence is equal to the number of annotators selecting the sentence for summary. Optimal summary of particular length is calculated from the pyramid [10]. The program generated summary and the human annotated summary will be compared with the optimal summary score and the weighted Recall is calculated.

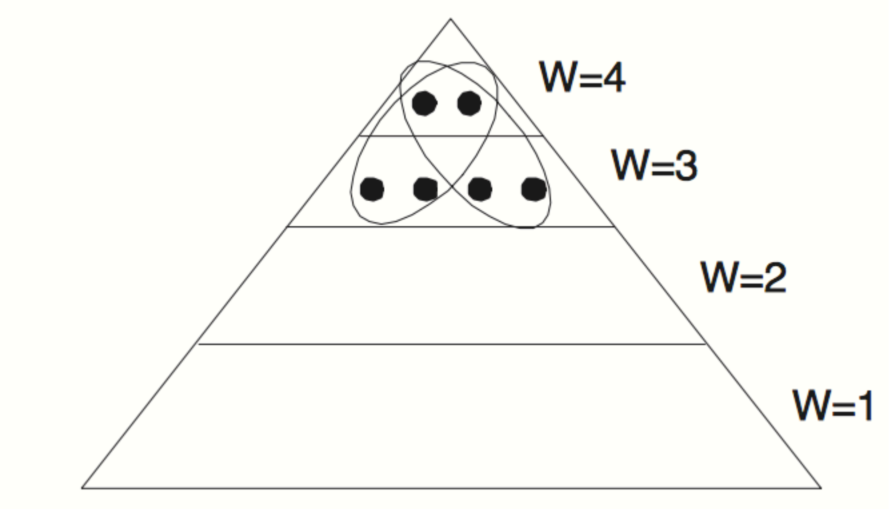


Figure 6.1: Pyramid for sentences with maximum weight 4

In this work, W denotes the number of annotated summaries in which a sentence is selected. W =3 would be the maximum pyramid with sentences selected by all 3 annotators are present.

6.3.1 Pyramid Evaluation of different features.

As the number of sentences that are extracted from email threads is same as that of annotated summaries, Pyramids are constructed on both annotated summary and system generated summary. From the pyramids, weighted re-call is calculated between the program generated summary and annotated summary from corpus.

Weighted Recall = $\sum i \in \text{Sent}_{Sum} \text{ Nscore}_i / \sum i \in \text{Sent}_{GS} \text{ Nscore}_i$

Figure 6.2 Weighted Recall Calculation

Where Nscore denotes the normalized value of corpus dependent sentence score obtained from pyramid. In other words, NScore is proportional to number of annotates summaries, the sentence occurred. In pyramid evaluation terms, the above formula is ratio of pyramid score of System generated summary to annotated summary. Jan Ulrich and Giuseppe Carenini and Gabriel Murray and Raymond Ng work on Regression-Based Summarization of Email Conversations in basis for this work's evaluation. Above weighted recall scheme is implemented by them. Following table gives the weighted recall values of different combination of features.

Feature	Weighted Recall
Term Frequency	33.5
Term Frequency – Inverse Document Frequency	35.45
Term Rank	36
Combination of all features	38.73

Table 6.3: Weighted Recall for different features

6.3.2 Precision and Recall results at different summary lengths.

The whole idea of Automatic Text Summarization is to condense the document/email to obtain at short version. But there are some standards during evaluation process in selecting percentage of sentences for summary. 5 % - 35 % of sentences as summary would be preferred. Following table gives the precision and weighted recall results for email threads at different lengths of summaries. POS tagger is implemented in this evaluation.

% of Summary	Weighted Recall
10	16.5
20	32.9
30	46.2
35	52

Table 6.4: Recall Results at different summary lengths

6.4 Weighted Recall Comparison

The proposed system is evaluated with a set of Supervised and Unsupervised learning machine-learning techniques. Jan Ulrich and Giuseppe Carenini and Gabriel Murray and Raymond Ng in their of Regression- Based Summarization of email conversations, used the same BC3 Corpus. The evaluation part of the system compares the weighted Recall scores of their output to the system-generated output. Following table gives the weighted recall values for methods they implemented.

Method	Weighted Recall
SVM	51
CWS	51
MEAD	43

Table 6.5: Weighted Recall values for evaluation

Out of three method in the above table, SVM is Supervised Learning techniques, whereas CWS and MEAD are Unsupervised Learning Techniques. The weighted recall for these methods are calculated at 35 % of summary length. From Table 6.4, the system generated a weighted recall of 52 % at 35 % of summary, which is greater than the 3 of the methods.

Though the system discussed in this thesis is Unsupervised Learning, with the implementation of Term Rank, it got more weighted recall than a Supervised learning method, SVM.

Chapter 7

CONCLUSION

As Natural Language Processing is in developing stage, Information Retrieval and Information Extraction is the only key for searching over the web. Though the IR techniques have been successful in extracting data, the amount of pages given as output for a query is more. This makes user to go through sometimes-irrelevant documents also. As the data is increasing at enormous rate, search engines and researchers can look forward for good summarization models to make the search engines more optimal. Pertaining to Email summarization, over the scale of an organization, summarization can be used as corporate memory, which helps the employees to have a short hand, notes on the proceedings and also serves as to-do list based on emergency of the situation. Research is going on to effectively implement Supervised Learning techniques as well as Unsupervised Learning techniques.

Summarization process is totally depended on the perspective of the human judge/ summarizer. The precision values of summary with different features when compared to human annotated summaries have an average of 28 %. The weighted recall values with different set of features have an average of 39%. But when checked at different lengths of summary, the system has some good results, 52 % weighted recall at 35 % of summary and 47 % weighted recall at 30 % of summary. Change of perspective from one human to another can be considered as the reason for this variation in weighted recall values. With a weighted recall of 52 % on Unsupervised learning system, it could be a good platform to do some future work and enhance the recall value and obtaining at summary most similar to a human annotated summary.

REFERENCES

[1] Jagadeesh J, Prasad Pingali, Vasudeva Varma, 2005. Sentence Extraction Based Single Document Summarization, in Workshop on Document Summarization, 19th and 20th March, 2005, IIIT Allahabad.

[2] Richard Wray(2009-05-18). “Internet data heads for 500 bngigabytes”

[3] Martin Hilbert and Priscila Lopez (2011-02-10). “The World’s Technological capacity to store, Communicate and Compute Information” University of Vermont. Vol. 332 no 6025 pp . 60-65.

[4] Lei Xu, Chunxiao Jiang, Jian Wang, Jian Yuan and Yong Ren Information Security in Big Data: Privacy and Data Mining, Proceedings of IEEE, published on October 9 2014

[5] Hamid Mousavi, Deirdre Kerr, Deirdre Kerr, Carlo Zaniolo, Mining Semantic Structures from Syntactic Structures in Free Text Documents, 2014 IEEE International Conference on Semantic Computing

[6] Raymond J. Mooney and Razvan Bunescu, Mining Knowledge from Text Using Information Extraction, Proceedings of SIGKDD Explorations, Volume 7, Issue 1 - Page 3

[7] Advances in Automatic Text Summarization. MIT Press.

[8] Ulrich J., Murray G., Carenini G., A Publicly Available Annotated Corpus for Supervised Email Summarization AAAI08 EMAIL Workshop, Chicago, USA, 2008.

[9] Fatih Gelgi, Hasan Davulcu, Srinivas Vadrevu. Term Ranking for Clustering Web Search Results. In Proc. of WebDB ’07, Beijing, China.

[10] Ani Nenkova, Rebecca Passonneau and Kathleen Mckeown. The Pyramid Method: Incorporating Human Content Selection Variation in Summarization Evaluation. ACM-Transaction April 19, 2007.

[11] Wikipedia Term-Frequency https://en.wikipedia.org/wiki/Tf–idf

[12] Jan Ulrich, Giuseppe Carenini, Gabriel, Murray,and Raymond Ng, Regression based Summarization of Email Conversations, Proceedings of the Third International ICWSM Conference (2009).

[13] Klaus Zechner and Alex Waibel, Flexible Summarization of Spontaneous Dialogues in Unrestricted Domains, Proceeding of Association of Computational Linguistics.

[14] Owen Rambow, Lokesh Shrestha, John Chen and Chirsty Lauridsen, Summarizing Email Threads. 2004. Proceeding of HLT-NAACL-Short ’04 Proceedings of HLT-NAA 2004