# Automatic Text Summarization

Rohit Shukla (B11028)
Saurabh Jain (B11033)
Shubham Ajmera (B11035)

# Motivation

- With tons of information pouring in every day text summaries have become essential.

- Instead of going through an entire text a concise summary helps in understanding a text quickly and easily.

# Problem Statement

- This approach of automatic text summarization uses machine learning algorithms.
- In this approach the summary generated will not be an abstract one rather it will be an extract summary(generated by extracting key segments).
- It will be good enough for the reader to get the main idea of the document.

# Introduction

- Sentences of each document are modeled as vectors of features
- Summarization task: two classification problem
- Sentence labeled as “correct” or “incorrect” if it belongs to the extractive reference summary or not.

# Cont..

- The trainable summarizer “learn” the patterns which lead to the summaries
- Identifying relevant feature values which are most correlated with the classes “correct” or “incorrect”.
- Learned patterns are used to classify each sentence of a new document

# Text Features

- Sentence Position
- Positive keyword in sentence
- Negative keyword in sentence
- Sentence centrality
- Sentence resemblance to title
- Sentence inclusion of name identity
- Sentence inclusion of numerical data
- Sentence relative length
- Bushy path of the node
- Summation of similarity of each node

# Sentence Position

- Sentences at the beginning and at the end of the text are more important

- Rank a paragraph sentence according to their position
- For instance, the first sentence in a paragraph has a score value of 5/5, the second sentence has a score 4/5

# Bushy Path of Node (Sentence)

- Bushiness of a node: Number of links connecting it to other nodes on the map
- A highly bushy node, linked to a number of other nodes, has an overlapping vocabulary with several sentences
- Likely to discuss topics covered in many other sentences

$$Score_{f_p}(s) = \#(branches\ connected\ to\ the\ node)$$

# Summarization of Similarities for Each Node

- Aggregate similarity measures the importance of a sentence.
- Instead of counting the number of links connecting a node (sentence) to other nodes (Bushy path), aggregate similarity sums the weights on the links.

$$Score_{f_{10}}(s) = \sum Node\ branch\ similarities$$

# Positive Keyword in Sentence

- Positive keyword is the keyword frequently included in the summary

$$Score_{f_2}(s) = \frac{1}{Length(s)} \sum_{i=1}^{n} tf_i * P(s \in S \mid keyword_i)$$

$$P(s \in S \mid keyword_i) = \frac{P(keyword_i \mid s \in S) P(s \in S)}{P(keyword_i)}$$

$$P(keyword_i \mid s \in S) = \frac{\#(sentence\ in\ summary\ ,and\ contains\ keyword_i)}{\#(sentence\ in\ summary)}$$

$$P(s \in S) = \frac{\#(sentence\ in\ training\ corpus\ ,and\ also\ in\ summary)}{\#(sentence\ in\ training\ corpus)}$$

$$P(keyword_i) = \frac{\#(sentence\ in\ training\ corpus\ ,and\ contains\ keyword_i)}{\#(sentence\ in\ training\ corpus)}$$

# Negative Keywords in Sentence

- Negative keywords are the keywords that are unlikely to occur in the summary

$$Score_{f_3}(s) = \frac{1}{Length(s)} \sum_{i=1}^{n} tf_i * P(s \notin S \mid keyword_i)$$

# Sentence Centrality

- Sentence centrality is the vocabulary overlap between this sentence and other sentences in the document.
- It is calculated as follows:

$$Score_{f_4}(s) = \left| \frac{Keywords\ in\ s \cap Keywords\ in\ other\ sentences}{Keywords\ in\ s \cup Keywords\ in\ other\ sentences} \right|$$

# Sentence Resemblance to the Title

- Sentence resemblance to the title is the vocabulary overlap between this sentence and the document title.
- It is calculated as follows:

$$Score_{f_s}(s) = \left| \frac{Keywords\ in\ s \cap Keywords\ in\ title}{Keywords\ in\ s \cup Keywords\ title} \right|$$

# Proper Noun

- Usually the sentence that contains more proper nouns is an important one and it is most probably included in the document summary
- The score of f6 is calculated as follows:

$$Score_{f_6}(s) = \frac{\#(proper\ nouns\ in\ s)}{Length(s)}$$

# Numerical Data

- Usually the sentence that contains numerical data is an important one and it is most probably included in the document summary
- The score of f7 is calculated as follows:

$$Score_{f_7}(s) = \frac{\#(numerical\ data\ in\ s)}{Length(s)}$$

# Sentence Relative Length

- Penalize sentences that are too short; not expected to belong to the summary

$$Score_{f_s}(s) = \frac{Length(s) * \#(article\ sentences)}{Length(article)}$$

# Calculating Ranks

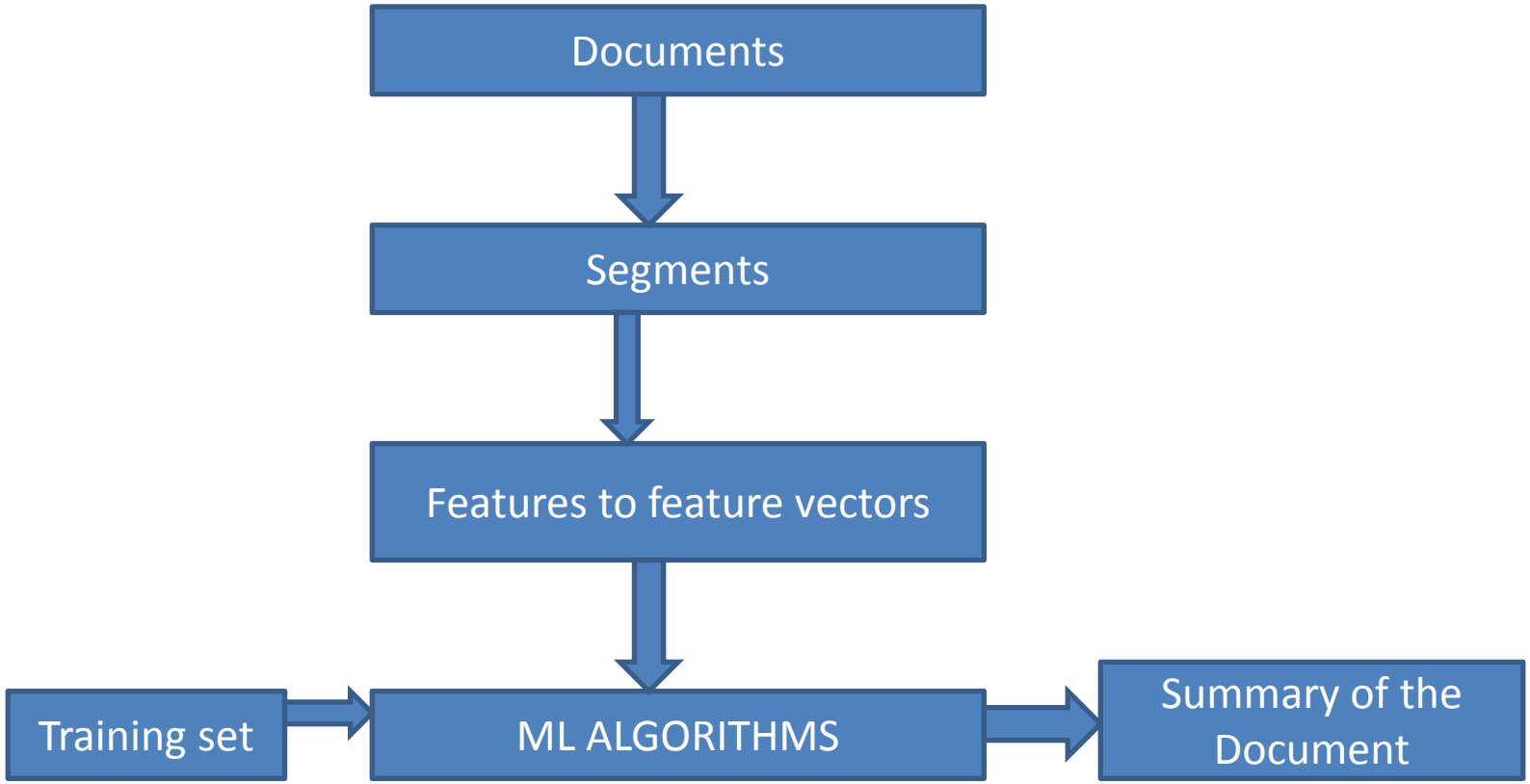
$$Score(s) = w_1.Score_{f_1}(s) + w_2.Score_{f_2}(s) + w_3.Score_{f_3}(s) +$$
$$w_4.Score_{f_4}(s) + w_5.Score_{f_5}(s) + w_6.Score_{f_6}(s) +$$
$$w_7.Score_{f_7}(s) + w_8.Score_{f_8}(s) + w_9.Score_{f_9}(s) +$$
$$w_{10}.Score_{f_{10}}(s)$$

# Mathematical Regression Model

- Mathematical Regression will be used to estimate text feature weights.
- The model will be trained by manually summarized training set.

$$\begin{bmatrix} Y0 \\ Y1 \\ : \\ : \\ Ym \end{bmatrix} = \begin{bmatrix} X01 & X02 & X03 & ..... & X010 \\ : & : & : & : & : \\ : & : & : & : & :: \\ : & : & : & : & : \\ Xm1 & Xm2 & Xm3 & ..... & Xm10 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ : \\ : \\ w_{10} \end{bmatrix}$$

# Schematic/Flow Diagram



# Methodology

## Sentence segmentation

- Shallow parsing by looking at special cue markers in order to determine sentence segments.

## Feature representation

- Set of features(average term frequency , rhetorical relations etc. ) are defined for each sentence segments.

## Training of the summarizer

- These features are then converted into vector representation and we apply machine learning algorithms in order to determine rules or conditions by which summary will be generated.

# Sentence Segmentation

- Separate out units that convey independent meanings.
- Sentence is segmented by a cue phrase.
- The purpose of segmentation is to use sentence segments as a basic unit of summarization.
- The complexity of segmentation process is O(n) where n is the number of sentences.

# Feature Representation

- The sentence segments are represented by a set of features.
- Two kinds of features:

Structured: Related to the structure of the text(e.g. rhetorical relations)

Non-structured: Not related to the structure.

e.g. title words

# Feature Vector

- F =< f1,f2,f3,f4,f5……..f22,f23>

f1,f2,f3,f4…..f23 are features for every segment in the segmentation process.

F is the feature vector.

- Features are divided into following three groups for analysis:

Group 1:paragraph number, offset in the paragraph, number of bonus words, number of title words

Group 2:antithesis,cause,circumstances,concession, condition

Group 3: weight of nucleus, weight of satellite, max level

# Continued….

- Group 1 contains non-structural attributes of the text.
- Group 2 contains distinct rhetorical relations.
- Group 3 contains collective descriptions of the rhetorical relations.

# Summarizer training

- A variety of machine learning algorithms are used for summarizer training.
- Two algorithms have been taken by us:

  Decision Trees

  Naïve Bayesian classifier

# Decision Trees

- Most widely used inductive learning methods.
- A decision tree is generated by finding a feature that yields the maximum information gain.
- A node is then created by set of rules corresponding to the feature and the process is repeated for other features until there is no information gain.
- In testing a pattern is compared with a node of a tree, following the branches of tree until a terminal node is reached.
- The pattern is then presumed to belong to the class in which the terminal node represents.

# Naïve Bayesian classifier

- Naïve Bayesian classifier is used as:

$$P(c \in C \mid F_1, F_2, \ldots F_k)$$

$$= \frac{\Pi_{j=1}^{k} P(F_j \mid c \in C) P(c \in C)}{\Pi_{j=1}^{k} P(F_j)}$$

It is the probability of finding a segment c in the target class C (i.e. in the summary or not in the summary) given the features f1,f2,f3…..fk.

- Values of most of the features are real numbers so Normal distribution is used for every feature:

$$P(F_j) = \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{F_j - \mu_j}{\sigma_j}\right)^2}$$

# Work done till now……

Month of September

- Training set for the Machine learning algorithms has been extracted from the internet.
- Algorithms/Model involved for summarizing the documents have been decided and implementation of them is the next step.
- Basic layout of the project is ready and division of work monthly has been decided.

# Work to be done

- Implementation of Decision Trees and Bayesian algorithms.
- Study and implementation of DistAI- neural network learning algorithm to ensure the results obtained from the above two algorithms are satisfactory.
- Comparitive study of the results obtained from these algorithms and MS Word summarizer.

# Timeline

Task to be completed at the end of this semester:

1. Extracting sentence segments from the document
2. Extracting features from the document based upon different methods as mentioned in Text Features