RESEARCH ARTICLE OPEN ACCESS

Single or Multi-document Summarization Techniques

Maithili Bhide
Department of Computer Science and Engineering
Pune Institute of Computer Technology
Pune - India

ABSTRACT

The current age of information overload demands a need for text summarization. The aim of automatic summarization is to create a compressed version of the source text while preserving salient information. Different approaches of summarization include extractive and abstractive summarization, semantic and syntactic techniques of summarization which may utilize supervised or unsupervised learning algorithms. Comparison and contrast of the results obtained by applying different algorithms like Kmeans /Clustering based Semantic Summarization (CSS), Latent Semantic Analysis (LSA) and graph based summarization algorithms like TextRank or LexRank applied over a particular dataset in terms of their efficiency and accuracy is demonstrated. The issues related to the automatic summarization of text documents and the use of summary generating agents in contemporary companies to analyze unstructured data can be further explored. The comparison of an automatic text document summary with a human generated summary to assess its intelligibility and information loss is also demonstrated.

Keywords:- Natural Language Processing, Text Mining, Machine Learning

I. INTRODUCTION

Information Retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

In contemporary companies the unstructured knowledge is essential, mainly due to the possibility of obtaining better flexibility and competitiveness of the organization. The analysis of the text documents is mainly based on document retrieval, information extraction, text mining and natural language processing. Summarization can include the contents of a document or set of documents.

The basic idea of summarization is to get a summary that contains the most important information from the source document. One of the parameters of this process is the text volume. A good document summary frees the system user (investor, manager) from the need to read and analyze all of the text documents, and give the opportunity to focus his attention on aspects of the rapid and effective decision. The key benefits of summarization include creation of reports that are both concise and comprehensive which simplify information search and cut the time by pointing to the most relevant information pertaining to a certain topic.

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. The main idea is to find a representative subset of data which contains information representative of the entire data set. [2]

Text summarization methods can be classified into extractive and abstractive summarization.

An extractive summarization method consists of selecting important sentences, paragraphs from the original document and concatenating them into shorter form. Abstractive summarization aims to interpret and examine the source text and creates a concise summary that usually contain compressed sentences or may contain some novel sentences not present in the original source text.

The simplest method for creating summaries is based on the assumption that the weight of the sentence depends on the weight of its words, calculated on the basis of their frequency in the text. In addition to the counted weight of words, other factors are also taken into account, such as the position of sentences in the text and the occurrence of words in the title or header. Currently, ROUGE metric (Recall Oriented Understudy for Gisting Evaluation) is used to evaluate automatic summaries. It calculates overlaps between automatically generated summaries and previously written human summaries. A high level of overlap indicates high level of shared concepts but the coherence of summaries cannot be evaluated. [5]

The major challenge faced in evaluation of summaries is the impossibility to construct a gold-standard against which rest of the summaries are compared. People are subjective and this leads to inconsistencies in the selection of sentences. Paraphrasing, that is expression of same meaning in different words also leads to logistic difficulties while evaluating summaries qualitatively.

II. SURVEY OF MATHEMATICAL MODELS

Algorithms used-

A. K-means

Kmeans is one of the simplest unsupervised learning algorithms to solve clustering problems.Clustering is the process of partitioning a group of data points into a small number of clusters.

Steps for K-means algorithm-

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

If we have n sample feature vectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ all from the same class, and we know that they fall into k compact clusters, k < n. Let $\mathbf{m}_i$ be the mean of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that $\mathbf{x}$ is in cluster i if $\| \mathbf{x} - \mathbf{m}_i \|$ is the minimum of all the k distances. This suggests the following procedure for finding the k means:

1) Make initial guesses for the means $\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_k$
2) Until there are no changes in any mean
3) Use the estimated means to classify the samples into clusters
4) For i from 1 to k
5) Replace $\mathbf{m}_i$ with the mean of all of the samples for cluster i
6) end_for
7) end_until

K-means is robust and relatively efficient when k, t, d << n

Where,

k = number of clusters

t = iterations

d = dimensions of each object

n = number of objects

B. Latent Semantic Analysis

Latent Semantic Analysis(LSA) or Latent Semantic Indexing(LSI) is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition(SVD) is used to reduce the number of rows while preserving the similarity structure among columns. Words are then compared by taking the cosine of the angle between the two vectors (or the dot product between the normalizations of the two vectors) formed by any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words.

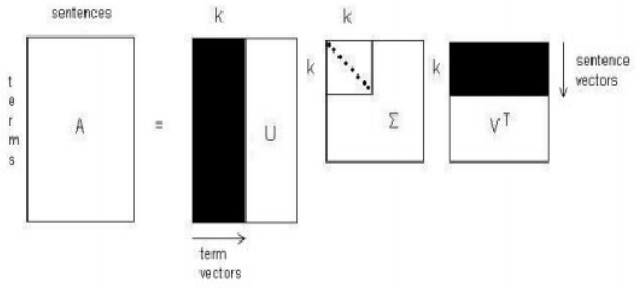


Fig. 1 Singular Value Decomposition

A = matrix of sentences

V^T = matrix indicating importance degree of each sentence in topic

U = matrix representing degree of importance of terms in salient topics/concepts

Σ = summation

C. Lex Rank

In LexRank a graph is constructed by creating a vertex for each sentence in the document.The edges between sentences are based on some form of semantic similarity or content overlap.

In the research, they measure similarity between sentences by considering every sentence as bag-of-words model. This means that the similarity measure between sentences is computed by frequency of word occurrence in a sentence. The basic measurement is using TF-IDF formulation, where term frequency (TF) contributes to the similarity strength as the number of word occurrences is higher. On the other hand, the inverse document frequency regards low frequency words inversely contributes to higher value to the measurement. This TF-IDF formulation is then used as a measurement for similarity between sentences by using it in this idf-modified-cosine formula:

idf-modified-cosine(x,y) =

$$\frac{\sum_{w \in x,y} \mathrm{tf}_{w,x}\mathrm{tf}_{w,y}(\mathrm{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\mathrm{tf}_{x_i,x}\mathrm{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\mathrm{tf}_{y_i,y}\mathrm{idf}_{y_i})^2}}$$

This formula measures the distance between 2 sentences x and y.

The more similar they are, the closer will be the representation. This similarity measure is then used to build a similarity matrix, which can be used as a similarity graph between sentences.

Algorithm for computing LexRank scores-

1) MInputAn array S of n sentences, cosine threshold t output: An array L of LexRank scores
2) Array CosineMatrix[n][n];
3) Array Degree[n];

```
4) Array L[n];
5) for i ← 1 to n do
6)         for j ← 1 to n do
7)                 CosineMatrix[i][j]=idf-modified-
                   cosine(S[i],S[j]);
8)                 if CosineMatrix[i][j]  > t then
9)                         CosineMatrix[i][j]  = 1;
10)                        Degree[i] + +;
11)                end
12)                else
13)                        CosineMatrix[i][j]  = 0;
14)                end
15)        end
16) end
17) for i ← 1 to n do
18)        for j ← 1 to n do
19)                CosineMatrix[i][j]
                   =CosineMatrix[i][j]/Degree[i];
20)        end
21) end
22) L = PowerMethod(CosineMatrix,n,Є);
23) return L;
```

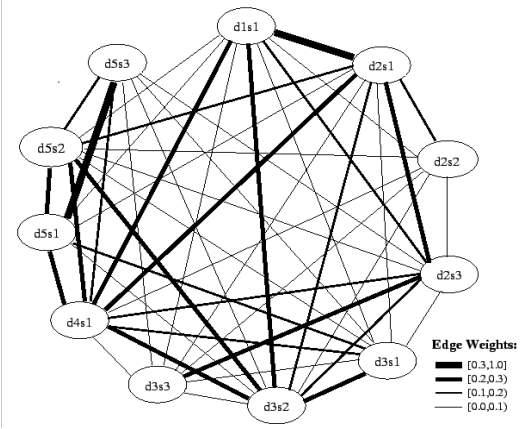


Fig. 2 Weighted cosine similarity graph example

III. PROPOSED MATHEMATICAL MODEL

Let 'S' be the solution perspective of the proposed system

$S = \{s, e, I, O, Fm, F, Su, Fa, DD, NDD \mid \phi\}$

Where,

s = start state
= installation of nltk, sumy

e = end state
= display of summary, time required for summarization

I = set of inputs
$= \{a_1, a_2, \ldots a_i \mid 0<i<10,\ i \varepsilon N\}$

Where a = text file to be summarized
N = set of natural numbers

O = set of outputs
$= \{(t_1,wc_1), (t_2,wc_2), \ldots (t_i,wc_i) \mid 0<i<10,\ i \varepsilon N\}$

Where t = time taken for summarization
wc = word count of text file

Function mapping-

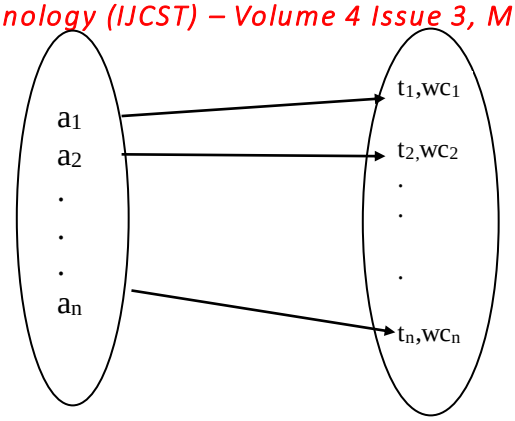


Fig. 3 One-to-One mapping between input and output sets

F_m = set of main functions i.e. functions required by summarizing tools

$F_m = \{f_{lexrank}(), f_{lsa}(), f_{kmeans}()\}$

$f_{lexrank}$ () = calculates summary using lexrank algorithm

f_{lsa} () = calculates summary using LSA

f_{kmeans} () = calculates summary using Kmeans

F = set of processing functions

F = {calc_time(), calc_wc(), plot_graph()}

calc_time() = calculates time of execution

$\{t_i = end\_time - start\_time \mid t_i \varepsilon Q, \forall i \varepsilon I\}$

where Q is a set of floating point real numbers

calc_wc() = calculates word count of input file

$\{wc_i \varepsilon Z^+ \mid Z^+ <= 65535, \forall i \varepsilon I\}$

Where Z is a set of all integers

plot_graph() = plots graph of word count against time of execution

DD = Deterministic Data
set of input files

NDD = Non-deterministic Data
size of input files
time required for summarization

Su = success case i.e. successful summarization

Fa = failure case i.e. unsuccessful summarization
Insensible summary generated
System failure

Φ = Constraints
size of input file < 65,535 bytes

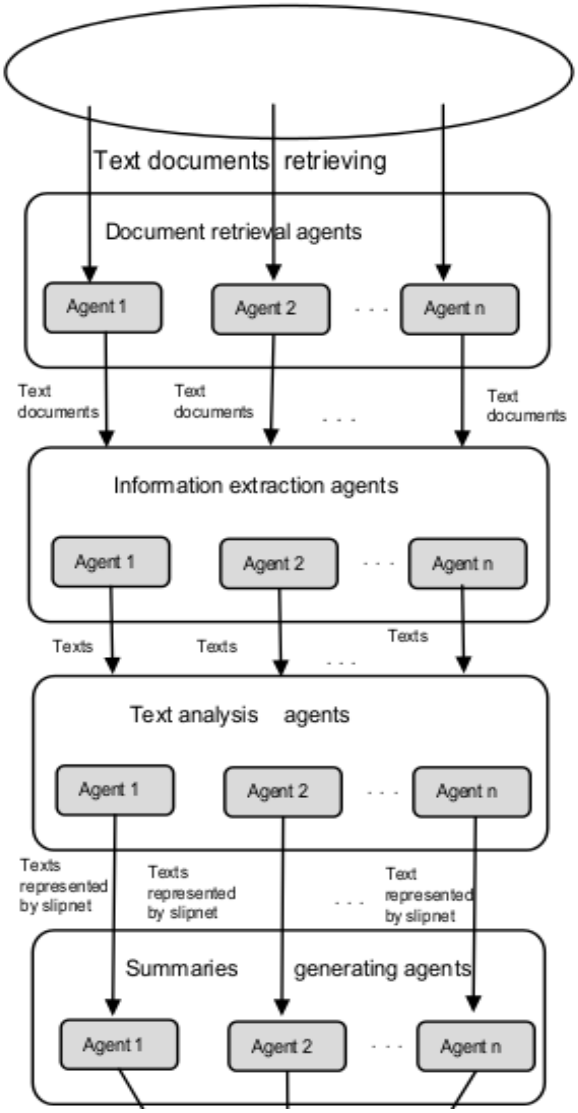
IV. DESIGN AND ANALYSIS OF SYSTEM

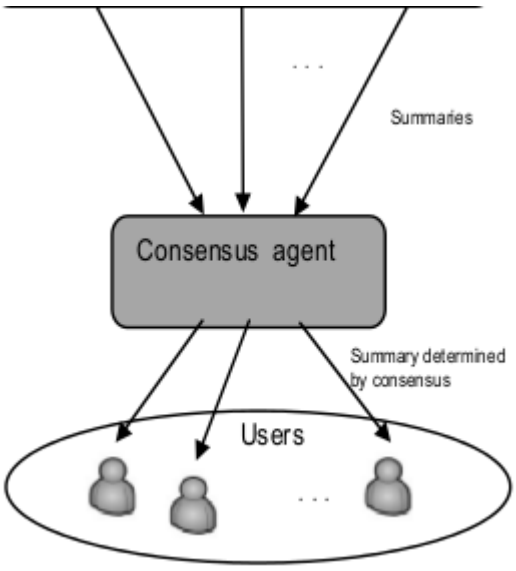
The Learning Intelligent Distribution Agent (LIDA) architecture is used in CIMIS(Cognitive Integrated Management Information System) construction.

Framework LIDA is a software underlying the implementation of the cognitive agents. The Framework contains the object class (implemented in JAVA).

The CIMIS consist of following sub-systems: fixed assets, logistics, manufacturing management, human resources management, financial and accounting, controlling, CRM, business intelligence. We focus on summarization module, which is a part of CRM subsystem. [2]

Fig. 4 Functional Architecture of Automatic Summarization Module





The module consists of the following groups of agents-

A. Document retrieval

Document retrieval agents search and retrieve, from the internet sources the documents according to users' needs.

B. Information extraction

Information extraction agents is to identify essential information in text documents.

C. Text analysis

Shallow and in-depth analysis of text is done by using a semantic network containing terms and connections between them.

D. Summaries generating

The most meaningful sentences are marked by nodes which have the highest total number of activation links in the semantic graph and are included in the summary.

V. DISCUSSION ON IMPLEMENTATION RESULTS

Thus Latent Semantic Analysis(LSA) and Lexrank text summarization algorithms have been applied to 10 text documents of varied word counts and their graph is plotted to compare and analyze the speed of summarization of the algorithms, i.e. the time required for their execution.

Tools used-

A. Natural Language Toolkit(NLTK)

A leading platform for building Python programs to work with human language data.

A suite of libraries and programs for symbolic and statistical natural language processing(NLP) for the Python programming language.

B. SUMY

Python library and command line utility version 0.4.1 used for extracting summary from html pages and plain text documents.

C. GNUPlot

Command line program that can generate two- and three-dimensional plots of functions, data and datafits.

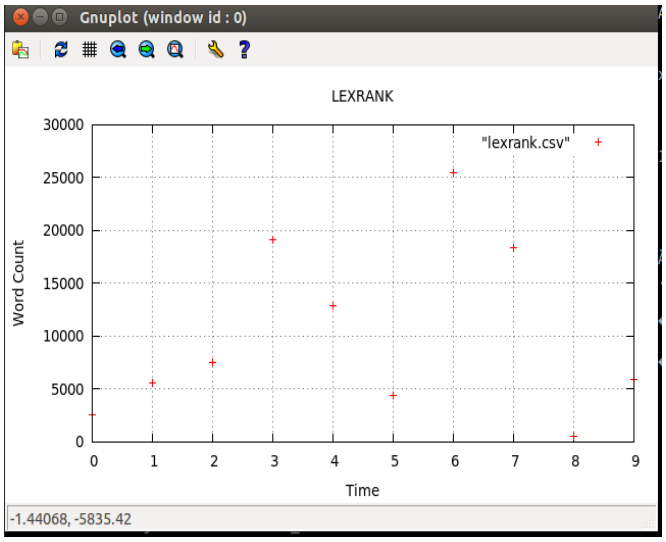


Fig. 5 Lex Rank

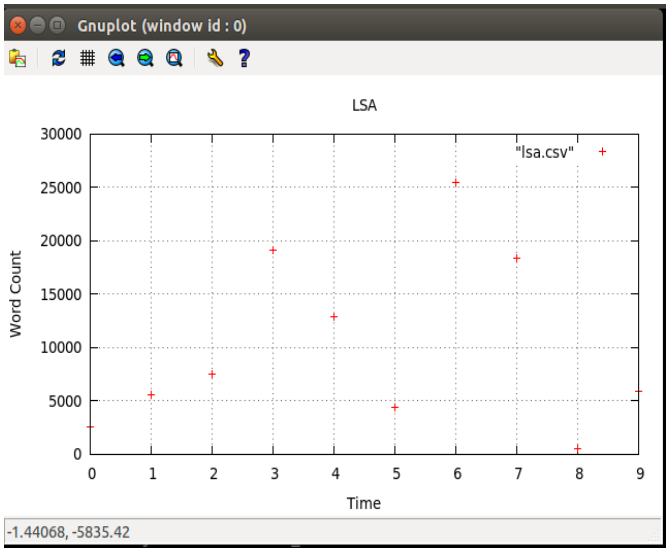


Fig. 6 LSA

TABLE I

EXECUTION TIME

WORD COUNT	LEX RANK	LSA
2559	0.0527679920197	0.142101049423
19138	0.934693098068	0.53851890564
25485	1.21983408928	0.708508968353
503	0.00763010978699	0.0138738155365

Thus we can see that for larger sized files, i.e. files with a greater word count, LSA is faster than Lexrank; however for smaller files, i.e. files with a lesser word count, Lexrank is faster.

VI. CONCLUSION AND FUTURE ENHANCEMENT

As amount of textual information available electronically grows rapidly, it becomes more difficult for a user to cope with all the text that is potentially of interest. Automatic document summarization methods are therefore becoming increasingly important. Document summarization is a problem of condensing a source document into a shorter version preserving its information content.

Various algorithms applied in summarization can be compared in terms of the quality of summary generated in addition to their efficiency in terms of speed for greater accuracy and ease of summarization

REFERENCES

[1] Ahmed, Mariwan, and Abdun Naser Mahmood. "Clustering based semantic data summarization technique: A new approach." *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*. IEEE, 2014.

[2] Hernes, Marcin, et al. "The automatic summarization of text documents in the Cognitive Integrated Management Information System." *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015.

[3] Khan, Atif, Naomie Salim, and Yogan Jaya Kumar. "Genetic semantic graph approach for multidocument abstractive summarization." *Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference on*. IEEE, 2015.

[4] Zhang, Yong, Meng Joo Er, and Rui Zhao. "Multi-Document Extractive Summarization Using Window-Based Sentence Representation."*Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, 2015.

[5] https://en.wikipedia.org/wiki/Automatic_summarization