LINGUISTIC ANALYSIS BASED TEXT SHRINKING

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Prerna (123215)

Under the supervision of

Mrs. Sanjana Singh

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **" Linguistic Analysis based Text Shrinking"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2016 to May 2016 under the supervision of **Mrs. Sanjana Singh** (Assistant Professor, Computer Science and Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Prerna
123215

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mrs. Sanjana Singh
Assistant Professor
Computer Science and Engineering

Dated:

ACKNOWLEDGEMENT

It is my proud privilege to epitomize my deepest sense of gratitude and indebtedness to our guide, Mrs. Sanjana Singh, for her valuable instructions, guidance and support throughout our project work. Her inspiring assistance and affectionate care enabled us to complete our work smoothly and successfully. This report is a dedicated contribution towards that greater goal.

Date:

Prerna
123215

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

S No.	Abbreviations	Term
1	Key words in sentence	KWDS
2	Key words in title	KWDT
3	Sentence Length	SL
4	Longest sentence length	LSL
5	Each sentence weight	SW
6	Maximum Sentence Weight in document	MSW
7	Total number of sentences in document	TNS
8	Total number of Thematic Words	TNTW
9	Thematic words in sentence	TW

LIST OF FIGURES

LIST OF TABLES

Abstract

Technology of automatic text summarization plays an important role in information retrieval and text classification, and may provide a solution to the information overload problem. Text summarization is a process of reducing the size of a text while preserving its information content. In this project, I implemented a text shrinking based on summarization of text which takes an input consisting of paragraphs and chooses the most relevant lines based on keyword frequency, title feature, sentence length and sentence position. Then summary is generated based on sorting of these sentences and shrinking percentage. After Summary has been generated certain words are further shrunk in understandable format. Finally a GUI outputs a shrunk stream of text having the most relevant lines.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The World Wide Web has brought us a vast amount of on-line information. Due to this fact, every time someone searches something on the Internet, the response obtained is lots of different Web pages with behemoth information, which is impossible for a person to read completely. The main idea is to find a representative subset of the data, which contains the information of the entire set. Text Shrinking tries to automatically create a representative summary or abstract of the entire document, by finding the most informative sentences.

With the growing amount of data in the world, interest in the field of automatic summarization generation has been widely increasing so as to reducing the manual effort of a person working on it.

Generally, there are two approaches to automatic summarization: *extraction* and *abstraction*. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. First remove the stop words. Then calculate the frequency of each word and select top words which have maximum frequency. This technique retrieves important sentence emphasize on high information richness in the sentence as well as high Information retrieval. These related maximum sentence generated scores are clustered to generate the summary of the document. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original.

1.2 Problem Statement

Produce an abridged version of text, retaining the significant portion of information in the original text by distilling out more important points from source.

1.3 Objectives

1) Propose a simple and effective approach for word ranking
2) An approach to rank sentences
3) Customizing it based on user requirement of shrinkage

1.4 Methodology

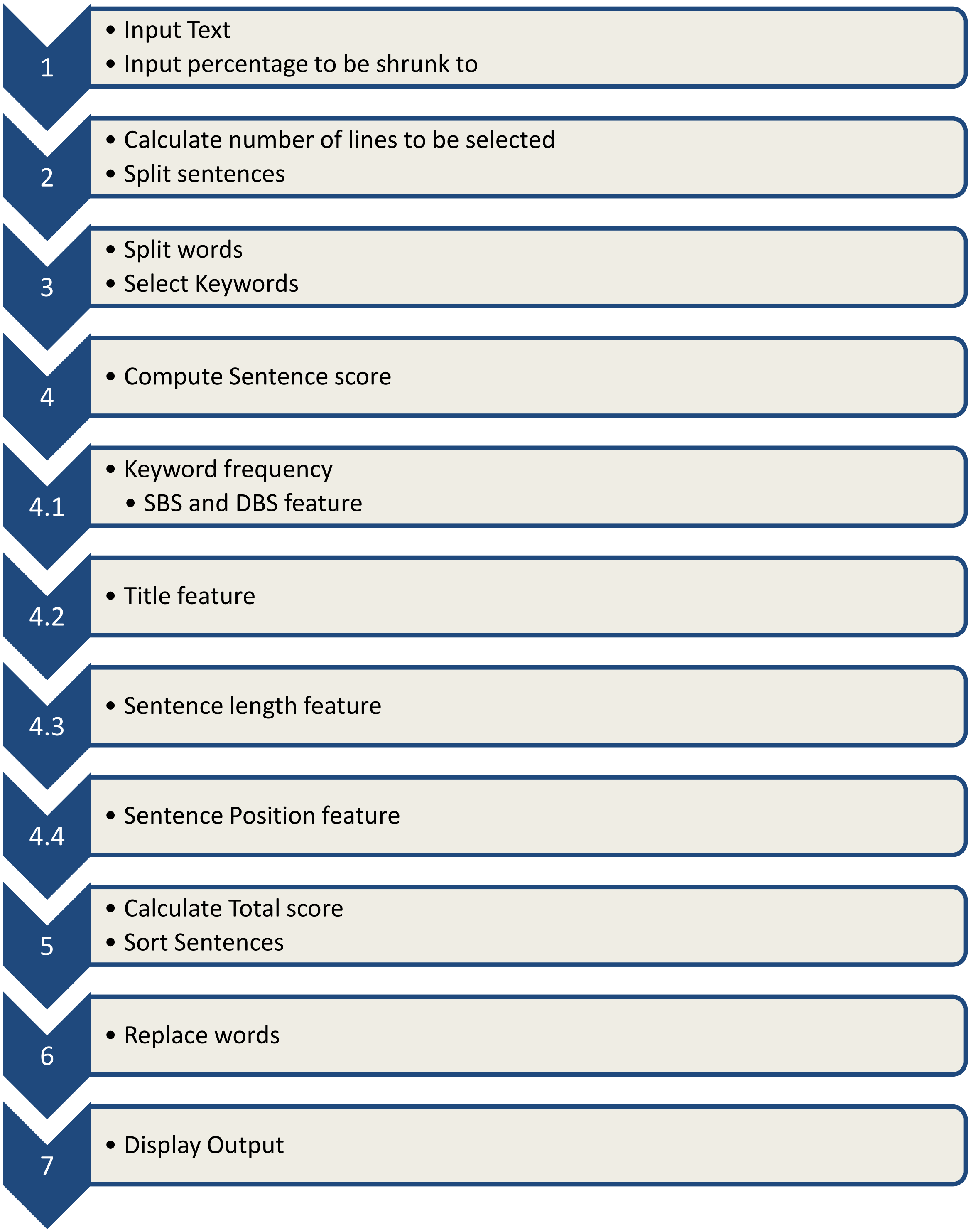


Fig 1 Flowchart

1.5 Modules

1.5.1 Preprocessing

The first step of the project is getting the text and getting percentage input the text needs to be shrunk to.

Then the number of lines to be selected is calculated.

We start the shrinking process by the sentence and word ranking module which takes the whole paragraph as the input and first splits the content on the basis of full stop and spaces. The punctuations are removed and further those sentences are split to words.

Getting Keywords

Stop words are removed

Stop words are the most general words that are used frequently in a sentence and they provide very less meaning to the content of the document. Stop words are maintained in a file for checking like ‘a’, ‘an’, ‘the’, ‘above’, etc. For example, consider the following sentence: The heart muscle requires a constant supply of oxygen-rich blood to nourish it. The stop words present in the above sentence are: the, a, of, to, it. After removing these words, we get the following sentence: Heart muscle requires constant supply oxygen-rich blood nourish.

In computing, stop words are words which are filtered out before or after processing of natural language data (text). Any group of words can be chosen as the stop words for a given purpose. Here we specifically delete articles and prepositions which usually do not represent any main part of the extract. To do the same, check each sentences for stop words, deleting the same and passes on to the next module.

Create List of Unique Keywords

A list of unique keywords left out after stop words removal is created. To do so set of these unique keywords is created and sorted in descending order of number of occurrences.

Get top Keywords

We have a list of keywords in order of their maximum occurrences out of which we select a group of top keywords example 10, 20 based on application domain. This is needed to rank sentences based on the inclusion of these keywords.

1.5.2 Feature Extraction

In this step, all the preprocessed sentences are made to go through test that checks the features related to it. We have laid emphasis on four important features, these features are:

Computing Sentence Score

Keyword Frequency

Summation Based Selection is just the normal count of keywords in the sentence. Summation-based selection (SBS), gives a higher representativeness score to a sentence if it contains more representative words.

Density Based Selection also considers the distance between each keyword in the sentence.

Based on these two methods we calculate the keyword frequency score of the sentence.

Title Feature

It is the similarity between the sentence & the document title.

Sentence Length

Since long sentences contain more number of words, they usually get more score. This factor needs to be considered while calculating the score of the sentence. In our system we normalize the sentence score by the number of words in that sentence, which is the score of the sentence per word.

Sentence Position

We assume that the first sentences of a paragraph are the most important. Therefore, paragraph sentences are ranked according to its position in the paragraph and considering the range between 0 to 1.

1.5.3 Generate Summary

Now sentences are scored based on weighted mean of these features.

After all the sentences have been passed and relative scores have been generated, the list is sorted based on scores and the top scorers reflect the main words of the passage which would be included in summarization.

Certain words are then replaced to an understandable yet short format. Example for can be represented as 4 or and can be represented as & and so on.

1.5.4 Displaying Output

The output is then displayed to the user as the shrunk form of input text.

1.6 Support for Novelty/ Significance of problem

This project would provide the user with customized approach of selecting the degree of text to be shrunk which can vary with various application domains.

The existing tools are not very popular and efficient.

In my solution I provide significance to various features mentioned above hence making it more efficient and trust worthy.

This defines the significance of my project and the integrated use of text features to summarize text accurately.

CHAPTER 2

LITERATURE SURVEY

2.1 Summary of papers

Title of Paper	Automatic Summarization
Author	Ani Nenkova and Kathleen McKeown
Year of Publication	2011
Summary	The vast majority of summarization systems today continue to rely on sentence extraction. other major advances within the extractive paradigm were also introduced. One was the introduction of more sophisticated natural language analysis for the extraction of features. The use of discourse to determine the importance of any individual sentence was a signature theme of a number of approaches in single document summarization. By measuring how connected a sentence is to the remainder of the article, whether through conferential or lexical chains, a system could determine its importance. As social networking grows, summaries may be helpful in navigating a network, determining who talks to who, or summarizing activities. Blogs or chat are a new form of media that, like email, share characteristics of both text and speech. There can be multiple responses to the same posting. They often involve informal language and in the case of chat, many abbreviations.
Web link	https://www.cis.upenn.edu/~nenkova/1500000015-Nenkova.pdf

Title of Paper	Text Summarization: An Overview
Author	Elena Lloret
Year of Publication	2008
Summary	This paper addresses the current state-of the-art of Text Summarization and gives an overview of the field Text Summarization and we present the factors related to it. It defines ‘Summary’ as a text that is produced from one or more texts that contains a significant portion of the information in the original text, and that is no longer than half of the original text. When this is done by means of a computer, i.e. automatically, we call this Automatic Text Summarization. This paper contains a large literature review in the research field of Text Summarization (TS) based on Human Language Technologies (HLT). TS helps users manage the vast amount of information available, by condensing documents' content and extracting the most relevant facts or topics included in them. The rapid development of emerging technologies poses new challenges to this research field, which still need to be solved. Therefore, it is essential to analyze its progress over the years, and provide an overview of the past, present and future directions, highlighting the main advances achieved and outlining remaining limitations. With this purpose, several important aspects are addressed within the scope of this survey. On the one hand, the paper aims at giving a general perspective on the state-of-the-art, describing the main concepts, as well as different summarization approaches, and relevant international forums. Furthermore, it is

	important to stress upon the fact that the birth of new requirements and scenarios has led to new types of summaries with specific purposes (e.g. sentiment-based summaries), and novel domains within which TS has proven to be also suitable for (e.g. blogs). In addition, TS is successfully combined with a number of intelligent systems based on HLT (e.g. information retrieval, question answering, and text classification). On the other hand, a deep study of the evaluation of summaries is also conducted in this paper, where the existing methodologies and systems are explained, as well as new research that has emerged concerning the automatic evaluation of summaries' quality. Finally, some thoughts about TS in general and its future will encourage the reader to think of novel approaches, applications and lines to conduct research in the next years. The analysis of these issues allows the reader to have a wide and useful background on the main important aspects of this research field.
Web link	http://www.dlsi.ua.es/~elloret/publications/TextSummarization.pdf

Title of Paper	Sentence Reduction for Automatic Text Summarization
Author	Hongyan Jing
Year of Publication	2000
Summary	The reduction algorithm we present assumes generic summarization; that is, we want to generate a summary that includes the most important information in an article. We can tailor the reduction system to queries-based summarization. In that case, the task of the reduction is not to remove phrases that are extraneous in terms of the main topic of an article, but phrases that are not very relevant to users' queries. We extended our sentence reduction program to query-based summarization by adding another step in the algorithm to measure the relevance of users' queries to phrases in the sentence. In the last step of reduction when the system makes the final decision, the relevance of a phrase to the query is taken into account, together with syntactic, context, and corpus information. Ideally, the sentence reduction module should interact with other modules in a summarization system. It should be able to send feedback to the extraction module if it finds that a sentence selected by the extraction module may be inappropriate (for example, having a very low context importance score). It should also be able to interact with the modules that run after it, such as the sentence combination module, so that it can revise reduction decisions according to the feedback from these modules. Some researchers suggested removing phrases or clauses from sentences for certain applications.
Web link	https://www.aclweb.org/anthology/A/A00/A00-1043.pdf

Title of Paper	Review of text reduction algorithms and text reduction using sentence vectorization
Author	Sneh garg and Sunil Chillar
Year of Publication	2000
Summary	The reduced text of a document is the collection of sentences that contains the important sentences containing keywords of the document. The authentic keywords extraction is the primary target for any text reduction algorithm. The presented survey shows the primary algorithm used for document summarization based on keywords. Also, the work presents a novel approach for keywords identification and in turn text reduction based on words histogram, the no. of sentences containing the words and knowledge corpus. The text summary is extracted using the sentence vectorization process. The sentence vectorization gives the sentences that have at least one of the key words in the sentence from the entire document. The algorithm works fine for the textual matter in the document in MS Notepad format. Factual information that is normally covered under double inverted comas is also given due attention in text summary. Most of the text summarization is done using the same information obtained from the same document rather than the concept analysis/exploring of the document. Most of summarization methods extract keywords for document only that are written in document method. However, the study shows that the importance of synonyms and relevant terms of keywords is ignored most of the time during text summarization. Therefore, comprehensive gain in summarization is required for a quality text

summary of the document considering the effects of synonyms as well. The primary contents of a summary include the principal information in least amount of words or sentence or factual statements. Identification of main content from rest information is the major challenging job in summarization. Basically, summary is the important information from original text and that is not more than half or one third of the original text. The target is to extract the useful information from a document in less space. All sentences in a document do not contribute in generating the text summary and are only language supportive. Therefore, the sentences may be given weightage and may be made part of the text summary depending upon the programmable or controlled weightage parameter. It is the key point for the reduction of text. As text and documents are growing day by day so it's a tedious task to extract useful information from such a large text and documents data base. Text reduction algorithm is an efficient way to get important information or brief summary of the whole document.

ALGORITHM:

The proposed scheme starts with the document reading for words. An array of each word is generated and modified for its uniqueness. This means that each word appear only once. Primarily, the text summary is based on important key words that occur many times in the document and any factual information. It is observed that the document contains only few keywords and most of the text material is language supporting words and phrases. Therefore, before

	exercising for keywords extraction, common words, generally referred as stop words are eliminated using the string comparison method. The filtered text array is exposed to keywords generation algorithm. The keywords are based on their frequency in the document and no. of sentences contacting the term. Further, the document title words are also considered in keyword category. Once a keyword vector set is derived, sentence vectorization process is performed. In sentence vectorization, the keyword vector set is compared with each of the sentence in the document. The document that contains at least one of the keyword vector set entry, the sentence is put into the text summary array. The entire document is scanned using the sentence vectorization algorithm. Finally, the text summary is compiled by concatenating all the sentences obtained during the sentence vectorization process.
Web link	http://research.ijcaonline.org/volume107/number12/pxc3900380.pdf

Title of Paper	Sentence Extraction Based Single Document Summarization
Author	Jagadeesh J, Prasad Pingali, Vasudeva Varma
Year of Publication	2005
Summary	A huge amount of on-line information is available on the web, and is still growing. While search engines were developed to deal with this huge volume of documents, even they output a large number of documents for a given user's query. Under these circumstances it became very difficult for the user to find the document he actually needs, because most of the naive users are reluctant to make the cumbersome effort of going through each of the documents. Therefore systems that can automatically summarize one or more documents are becoming increasingly desirable. In this paper we presented a sentence extraction based single document summarization system. We used shallow text processing approaches as opposed to semantic approaches related to natural language processing. We presented a detailed architecture and internal working of our system while discussing some of the challenges we came across in generating readable and coherent summaries. While the evaluation that we have presented here is subjective to the user, we would like to evaluate our system in the environments like DUC, where the evaluation is done using automated systems like ROUGE. In our system we have come up with arbitrary weights by trial and error method. We plan to implement machine learning techniques to learn these weights automatically from training data. We would like to use more NLP tools such as word sense disambiguation

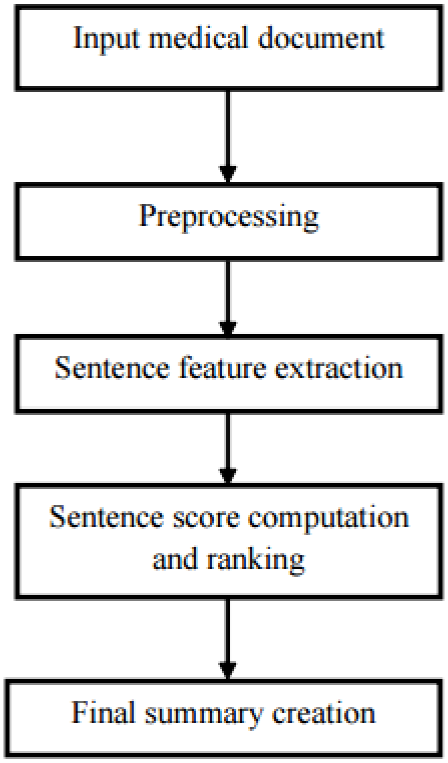
	and co-reference resolution modules to obtain precise weights for the sentences in the document. We also plan to extend this system to perform deeper semantic analyses of the text and add more features to our ranking function. We would like to extend this system to be able to generate multi-document summaries.
Web link	web2py.iiit.ac.in/.../inproceedings.pdf.60ed1ced-3d36-43f0-b4d3-a1f48519166f.pdf

Title of Paper	Comments-Oriented Blog Summarization by Sentence Extraction
Author	Meishan Hu, Aixin Sun and Ee-Peng Lim
Year of Publication	2007
Summary	Much existing research on blogs focused on posts only, ignoring their comments. Our user study conducted on summarizing blog posts, however, showed that reading comments does change one's understanding about blog posts. In this research, we aim to extract representative sentences from a blog post that best represent the topics discussed among its comments. Based on the findings in our user study that reading comments does affect one's understanding about a blog post (and probably other kind of Web objects), we define the problem of comments-oriented blog post summarization. Our proposed solution measures word representativeness using information hidden in comments, and then selects sentences based on the representativeness of the words contained in sentences. Using human labeled sentences, we evaluated two sentence selection methods with four word representativeness measures.
Web link	http://www.ntu.edu.sg/home/axsun/paper/sun_cikm07s.pdf

Title of Paper	An Efficient Medical Document Summarization using Sentence Feature Extraction and Ranking
Author	P. Gayathri and N. Jaisankar
Year of Publication	2015
Summary	Background/Objectives: As all documents related to medical domain do not come with author written summary, the objective is to introduce a summarizer that exploits medical domain-specific knowledge. Methods/Statistical Analysis: Sentence ranking technique has been used to produce high quality summary. The features such as sentence position, length, cue-words (domain-related terms) and acronyms are extracted to assign sentence score. Sentences are ranked and arranged in the decreasing order of their normalized score. The existing summarization approaches in the literature use few or more sentence features but we have opted for few best sentence features. Pre-existing summarizers are used for performance evaluation. Findings: The few best features to be considered in developing medical domain-specific summarizers are sentence position, sentence length, number of cue-words and number of acronyms. Summary produced by any summarizer can be highly informative if and only if it contains dissimilar sentences. Therefore, similarity between sentences is an important feature to be considered for creating highly informative summary. The proposed summarizer is compared with the preexisting summarizers. The evaluation is done by using traditional metrics such as precision and recall and ROUGE. Not all medical documents come with an author written abstract or summary. So,

	medical documents with author written abstracts are used to test the performance. Results reveals that the proposed summarizer performs better when compared with existing summarizers and attained ROUGE scores also reveals the same with respect to quality of summary produced. Thus, proposed summarizer provide highly acceptable summary to user. Application/Improvements: Summarization is one of the information retrieval tasks. It helps to determine whether the retrieved document is relevant for in-depth study or not. The summarizer proposed in this paper consists of following phases for generation of extractive informative single-document summary to exploit domain-specific knowledge. They are: 1. Pre-Processing. 2. Sentence feature extraction. 3. Sentence score computation and ranking. 4. Final summary creation by using highly dissimilar sentences. Pre-Processing includes two major activities: • Sentence Segmentation. • Stop Word Removal. The second phase in the process of summarization is sentence feature extraction. The following features have been used: • Position of the sentence. • Length of the sentence. • Number of medical related terms in the sentence. • Number of medical related acronyms in the sentence. The third phase is sentence score computation and ranking in which sentences are arranged in the decreasing order of their sentence score. The last phase is final summary creation by

using highly dissimilar sentences.


Input medical document
Preprocessing
Sentence feature extraction
Sentence score computation and ranking
Final summary creation

gure 1. Proposed summarization approach.

Fig 2.

The document summarization in medical domain on basis of sentence feature extraction has been discussed in this paper. The other summarizers in the literature used few or more sentence features for summary generation. We have opted for few best sentence features. Extractive informative single document summarization of medical document is produced by including highly dissimilar sentences. Thus, proposed summarizer provide high acceptance summary to user. By ROUGE scores, we can say that the proposed approach performs better than the pre-

	existing summarizers using human generated summaries as reference summaries. An informative summary is a summary that covers and provides all the important features in the document with some level of detailing. These summarization approaches helps to understand the insights of data in the document, incase if a document does not contain the author-written summary
Web link	www.indjst.org/index.php/indjst/article/download/71257/65590

2.2 Website Referred

http://www.igi-global.com/article/document-summarization-using-sentence-features/128277

Problem of exponential growth of information available electronically, there is an increasing demand for text summarization. Text summarization is the process of extracting the contents of the original text in a shorter form that provides useful information to the user. This paper presents a summarizer to produce summaries while reducing the redundant information and maximizing the summary relevancy. The proposed model takes several features into an account, including title feature, sentence weight, term weight, sentence position, inter sentence similarity, proper noun, thematic word and numerical data. The score of each feature for the model can be obtained from the document sets. However, the results of such models are evaluated to measure their performance based on F-score of extracted sentences at 20% compression rate on a C-50 data corpus. Experimental studies on C-50 data corpus, PSO summarizer show significantly better performance compared to other summarizer.

Overview of Summarization System

Figure illustrates the proposed automatic model for summarization. It includes three basic steps to generate summary which are preprocessing, feature extraction and summary generation.

Preprocessing

Initially the document is segmented into sentences and words for each sentence are extracted. Then the functional words or stop words like “a”, “the”, “of” (frequently occurring insignificant words) are removed from the word list. The words remaining in the sentences are stemmed.

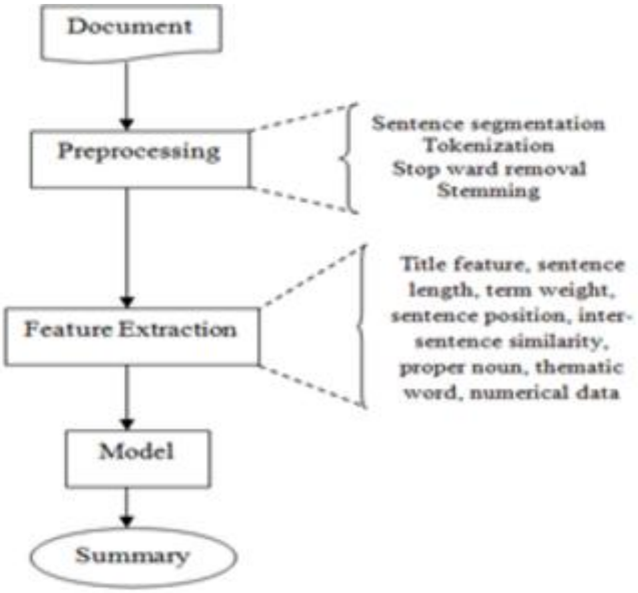

Document
Preprocessing
Sentence segmentation
Tokenization
Stop ward removal
Stemming
Feature Extraction
Title feature, sentence length, term weight, sentence position, inter-sentence similarity, proper noun, thematic word, numerical data
Model
Summary

Fig 3 Summarization System

Feature Extraction

Feature is one of the important aspects of any text mining. Therefore the following features for each sentence need to be prepared for input to the optimization model

1. **ft1= Title Feature:** It is the similarity between this sentence & the document title.

   The score of ft1 is calculated as follows: $Score_{ft1}(s) = \left|\frac{KWDS \cap KWDT}{KWDS \cup KWDT}\right|$

2. **ft2= Sentence Length:** This feature is employed to penalize sentences that are too short, since these sentences are not expected to belong to the summary. We use longest sentence length of the sentence for normalization. $Score_{ft2} = \frac{SL(i)}{LSL}$ Where i =1…TNS

3. **ft3= Average Sentence Weight:** This feature specifies the weight of each sentence by taking term frequency into an account. The score of ft3 is calculated as follows: $Score_{ft3} = \frac{SW(i)}{MSW}$ Where i=1…TNS

4. **ft4= Sentence Position:** We assume that the first sentences of a paragraph are the most important. Therefore, paragraph sentences are ranked according to its position in the paragraph and considering the range between 0 to1. For instance, the first sentence in a paragraph has a score value 1; the second has reduced with some value and so on. The score of ft4 is calculated as follows: $Score_{ft4}(s) = \frac{TNS - i + 1}{TNS}$ where i=1…TNS

5. **ft7= Thematic Word:** The most frequent words are defined as thematic words. Sentence scores are functions of the thematic words' frequencies. The score of ft7 is calculated as follows: $Score_{ft7} = \frac{TNTW}{TW}$

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 TECHNOLOGIES USED

3.1.1 PYTHON

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such asPy2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, allowing the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, as well as to various windowing systems. New built-in modules are easily written in C or C++ (or other languages, depending on the chosen implementation). Python is also usable as an extension language for applications written in other languages that need easy-to-use scripting or automation interfaces.

Why Python

Better code readability

It recognizes that you'll spend a lot more time reading code than writing it, and focuses on guiding developers to write readable code. It's possible to write obfuscated code in Python, but the easiest way to write the code (assuming you know Python) is almost always a way that is reasonable terse, and more importantly: code that clearly signals intent. If you know Python, you can work with almost any Python with little effort. Even libraries that add "magic" functionality can be written in perfectly readable Python.

Speed of development

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this - Loc may be at all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors**.**

Error Reporting on Runtime

When you're learning something for the first time, you're inevitably going to make mistakes. Python makes it easy to identify and fix these mistakes immediately. That's because Python displays errors at run time, instead of simply failing to compile the program.

Dynamic Typing

One of the biggest differences between Python and other languages is the way that each language handles variables. Languages like Java force you to define the type of a variable when you first declare it and will not allow you to change the type later in the program. This is known as static typing. In contrast, Python uses dynamic typing, which allows you to change the type of a variable which is easier for the novice programmer to get to grips with, because it means you can just use your variables as you want to without worrying too much about their types.

In-Built Natural Language Processing Support

While python having so many advantages over other programming languages, it also offers basic in- built natural language processing libraries which helps making works like speech synthesis, tagging, generation etc. so much easy and efficient. Moreover natural language processing requires dynamic data type for flexibility of input and output operations which is native in python.

3.1.2 NLTK

Natural **L**anguage **T**ool**K**it is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more.

We used NLTK for our software specifically because:-

Ease of Use

The primary purpose of the toolkit is to allow students to concentrate on building natural language processing (NLP) systems. The more time students must spend learning to use the toolkit, the less useful it is. Hence NLTK uses simple procedures which are easy to learn and easier to use.

Consistency

The NLTK toolkit uses consistent data structures and interfaces which are derived from python

Documentation

The toolkit, its data structures, and its implementation are carefully and thoroughly documented. All nomenclature is carefully chosen and consistently used. There's a chapter based book available too which gives all uses of NLTK along with examples.

Efficiency

The toolkit does not need to be highly optimized for runtime performance. However, it should be efficient enough that programmers can use their NLP systems to perform real tasks as in NLTK toolkit.

Cleverness

Clear designs and implementations are far preferable to ingenious yet indecipherable ones. NLTK has modules built in which helps check at each step what to do and what has been done.

3.1.3 TKINTER

The **Tkinter** module (“Tk interface”) is the standard Python interface to the Tk GUI toolkit from Scriptics. Tkinter is Python's de-facto standard GUI (Graphical User Interface) package.

Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Why Tkinter

Layered approach

The layered approach used in designing Tkinter gives Tkinter all of the advantages of the TK library. Therefore, at the time of creation, Tkinter inherited from the benefits of a GUI toolkit that had been given time to mature. This makes early versions of Tkinter a lot more stable and reliable than if it had been rewritten from scratch. Moreover, the conversion from Tcl/Tk to Tkinter is really trivial, so that Tk programmers can learn to use Tkinter very easily.

Accessibility

Learning Tkinter is very intuitive, and therefore quick and painless. The Tkinter implementation hides the detailed and complicated calls in simple, intuitive methods. This is a continuation of the Python way of thinking, since the language excels at quickly building prototypes. It is therefore expected that its preferred GUI library be implemented using the same approach. For example, here is the code for a typical “Hello world”-like application:

```
from Tkinter import *
root = Tk( )
```

```
root.title("A simple application")
root.mainloop( )
```

The first 2 lines allow to create a complete window. Compared to MFC programming, it makes no doubt that Tkinter is simple to use. The third line sets the caption of the window, and the fourth one makes it enter its event loop.

Portability

Python scripts that use Tkinter do not require modifications to be ported from one platform to the other. Tkinter is available for any platform that Python is implemented for, namely Microsoft Windows, X Windows, and Macintosh. This gives it a great advantage over most competing libraries, which are often restricted to one or two platforms. Moreover, Tkinter will provide the native look-and-feel of the specific platform it runs on.

Availability

Tkinter is now included in any Python distribution. Therefore, no supplementary modules are required in order to run scripts using Tkinter.

3.2 SYSTEM REQUIREMENTS SPECIFICATIONS

FUNCTIONAL

- The system should take text input from the user
- The system should take only relevant sentences to output stream
- The system should be scalable to take on long and lengthy paragraphs as input
- The output from system must be able to match human generated shrinked stream
- The interface of the application should be user friendly
- The system must undergo thorough testing for the output
- The system on which shrinker is to run must have python support

NON FUNCTIONAL

- The system must perform the shrinking task efficiently
- The system must be low on memory usage so that it can be implemented on portable devices like cell phones too
- The system must be reliable and robust
- The error rate of the system's output should be as minimum as possible

3.3 ALGORITHM LEVEL DESIGN

The algorithm to develop this model is composed of ten major steps or modules which are also shown in the activity diagram below. We pass the input inside the code as of now which is sent to various modules of the software. These modules process the input stream and pass the processed data to the following module which is passed back to the end user at last

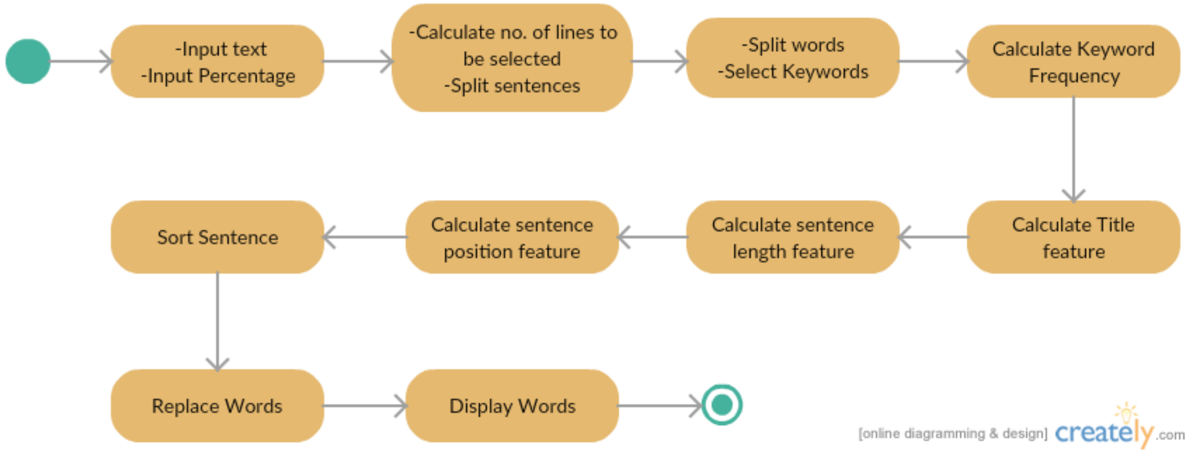
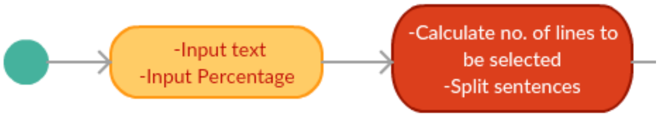

-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence
Replace Words
Display Words
[online diagramming & design] creately.com

Fig 4 Activity Diagram

Input Text and Input Percentage

The first step of the project is getting the text and getting percentage input the text needs to be shrunk to.

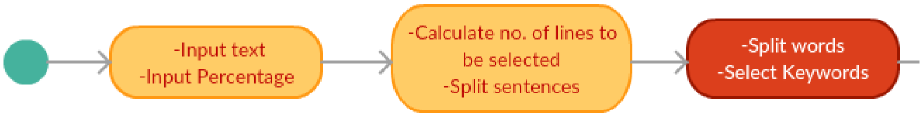
Calculate no. of lines to be selected and Split sentences



Then the number of lines to be selected is calculated.

Formula: count = (percentage input/100) *total lines

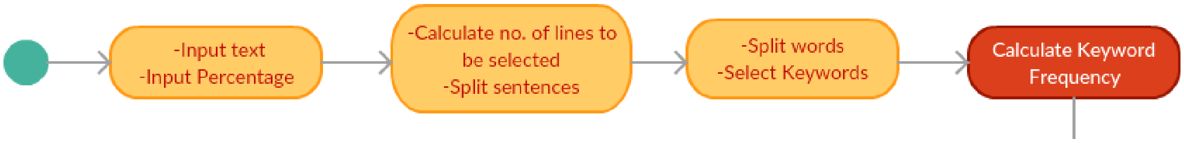
Split words and select keywords



We start the shrinking process by the sentence and word ranking module which takes the whole paragraph as the input and first splits the content on the basis of full stop and spaces. The punctuations are removed and further those sentences are split to words.
Stop Words are fetched from nltk toolkit and removed from the text.
Then a list of unique words is created and sorted in descending order of number of occurrences.

We have a list of keywords in order of their maximum occurrences out of which we select a group of top keywords example 10, 20 based on application domain. In this application we selected 10. This is needed to rank sentences based on the inclusion of these keywords.

Calculate keyword frequency



Summation Based Selection is just the normal count of keywords in the sentence. Summation-based selection (SBS), gives a higher representativeness score to a sentence if it contains more representative words.

Calculating SBS feature:

To calculate sbs feature we check if a keyword is present in the sentence and add its totalscore from sorted list to sentence score.

Density Based Selection also considers the distance between each keyword in the sentence.

Calculating DBS feature:

Check if a keyword is present. If yes, assign it to ‘first keyword’.

Check if another keyword is present. If yes, assign it to ‘second keyword’.

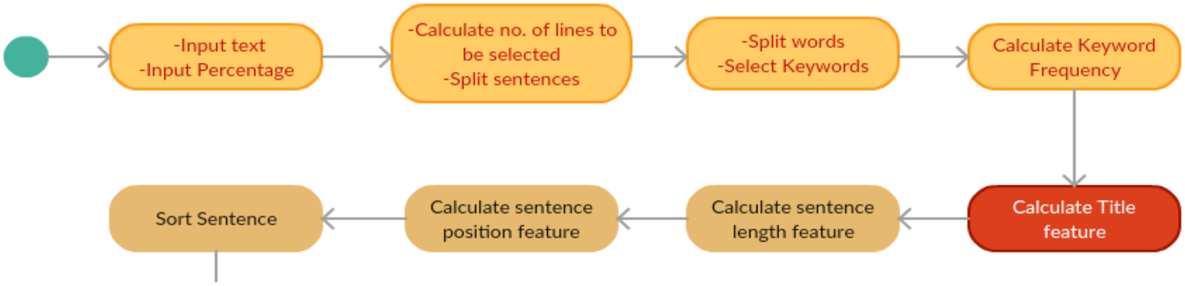
distance = firstWord['i'] - secondWord['i'] (i being index)

score += (firstWord['score'] * secondWord['score']) / (distance*2)

Based on these two keyword frequency is calculated as follows:

keywordFrequency = (sbsFeature + dbsFeature) / 2.0

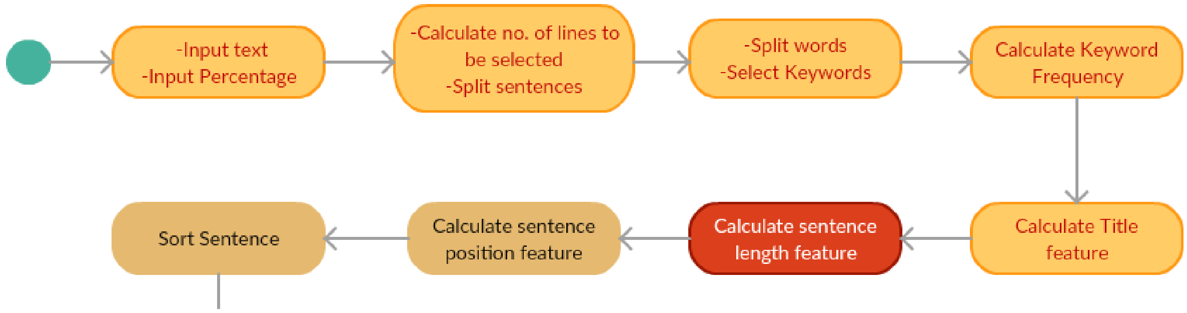
Calculate Title Feature


-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence

It is the similarity between the sentence & the document title.

This is done by removing stop words from title and sentences then number of matched words is calculated and **len(matchedWords)/len(title)** is returned.

Calculate Sentence Length Score

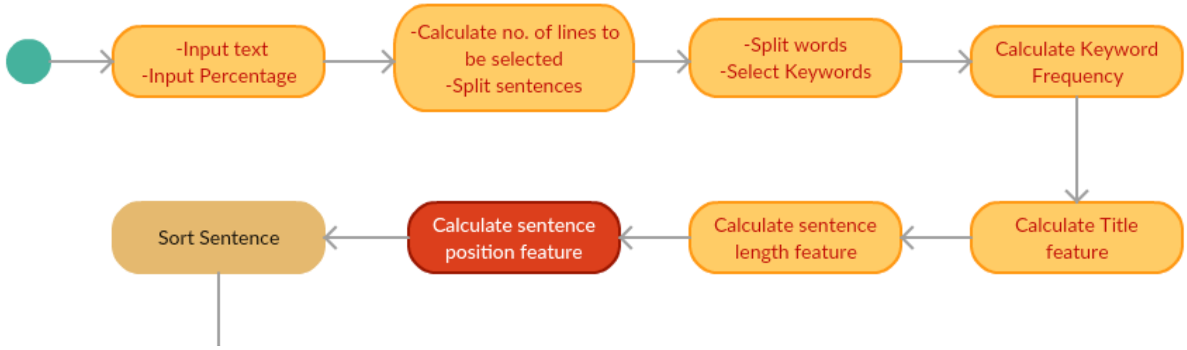

-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence

This feature is employed to penalize sentences that are too short, since these sentences are not expected to belong to the summary.

Ideal length is considered to be 20. So returned value is

(ideal - abs(ideal - len(sentence))) / ideal

Calculate sentence Position


-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence

We assume that the first sentences of a paragraph are the most important. Therefore, paragraph sentences are ranked according to its position in the paragraph and considering the range between 0 to1.

Normalization is done by : **normalized = i / (sentenceCount * 1.0)** where i is the index

And then

```
If normalized > 0 and normalized <= 0.1:
    return 0.17
elif normalized > 0.1 and normalized <= 0.2:
    return 0.23
elif normalized > 0.2 and normalized <= 0.3:
    return 0.14
elif normalized > 0.3 and normalized <= 0.4:
    return 0.08
elif normalized > 0.4 and normalized <= 0.5:
    return 0.05
elif normalized > 0.5 and normalized <= 0.6:
    return 0.04
elif normalized > 0.6 and normalized <= 0.7:
    return 0.06
elif normalized > 0.7 and normalized <= 0.8:
    return 0.04
elif normalized > 0.8 and normalized <= 0.9:
    return 0.04
elif normalized > 0.9 and normalized <= 1.0:
    return 0.15
else:
    return 0
```

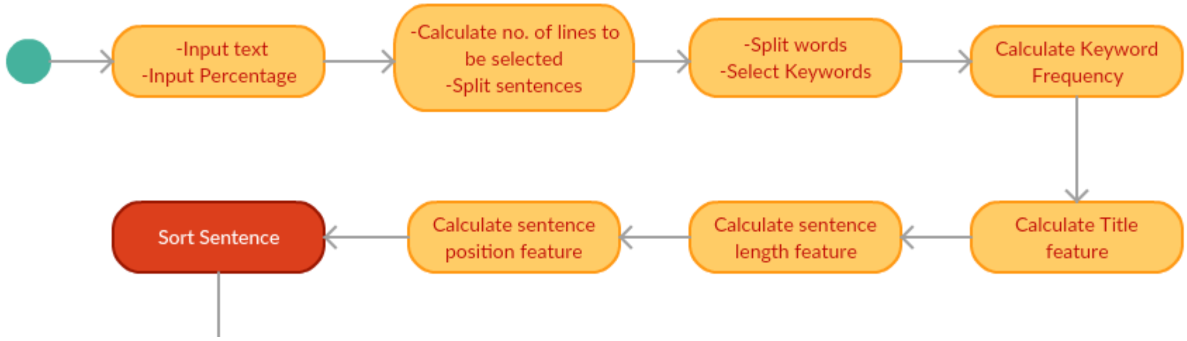
After features

After all these feature value calculations Total score of each sentence is generated by weighted mean.

TotalScore = (titleFeature * 1.5 + keywordFrequency * 2.0 + sentenceLength * 0.5 + sentencePosition * 1.0) / 4.0

Now we have **sentence**, its **score** and **order** in text.

Sort Sentences

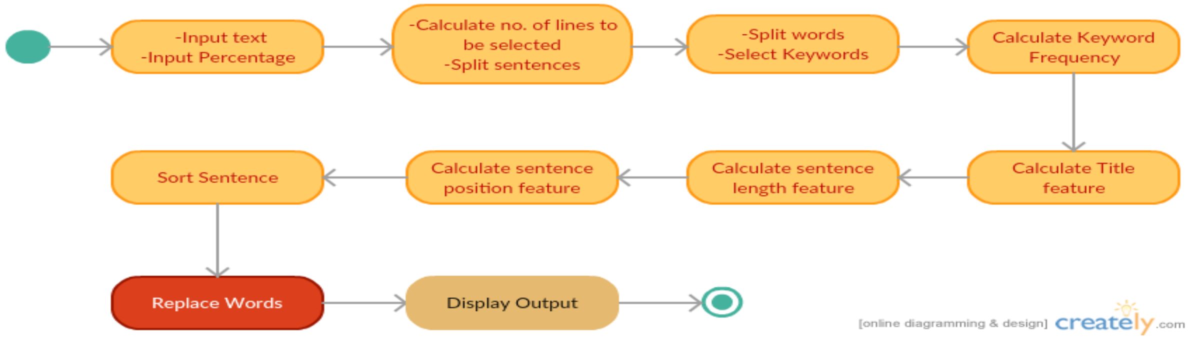


-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence

The sentences are then sorted in descending order of scores.

The number of sentences to be selected was calculated initially so that number of top sentences would be needed to generate summary.

Replace words



-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence
Replace Words
Display Output
[online diagramming & design] creately.com

Words are replaced to short yet understandable format. Example:

Great -> gr8

Are -> r

You -> u

For -> 4

To -> 2

Figure -> fig.

Number -> no.

Example -> eg.

And -> &

Before -> b4

Year ->yr

Fine -> f9

Between -> btwn

ing -> ‘g

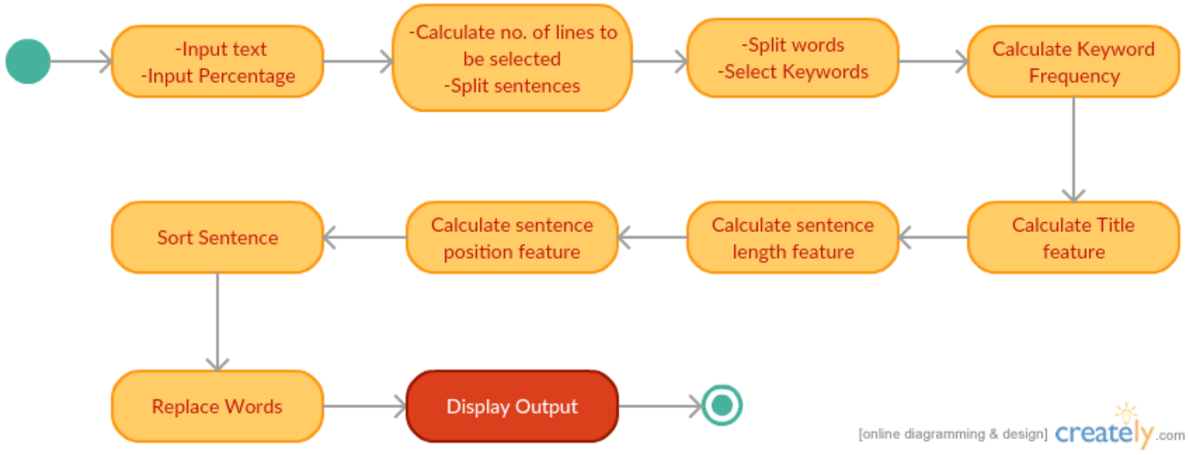
tion -> t‘n

one -> 1

two -> 2

And so on.

Display Output


-Input text
-Input Percentage
-Calculate no. of lines to be selected
-Split sentences
-Split words
-Select Keywords
Calculate Keyword Frequency
Calculate Title feature
Calculate sentence length feature
Calculate sentence position feature
Sort Sentence
Replace Words
Display Output
[online diagramming & design] creately.com

Output is then displayed on GUI presented using Tkinter.

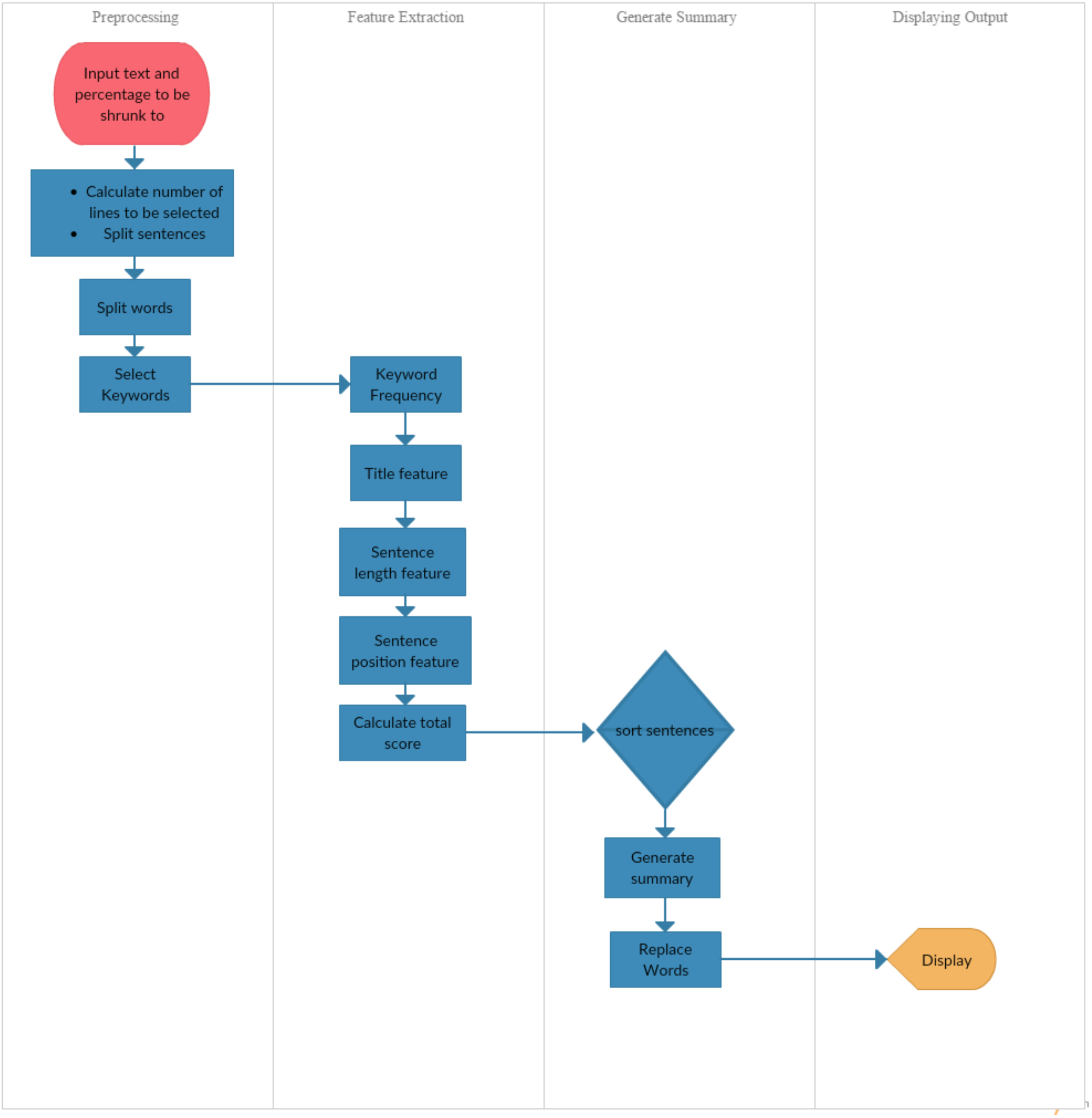


Fig 5. Swim lane diagram

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Input

Input source

http://www.monash.edu.au/lls/llonline/writing/general/essay/sample-essay/index.xml

```
title = "Divorce essay"
text = """A major change that has occurred in the Western family is an increased incidence in divorce. Whereas in the past, divorce was a relatively rare occurrence, in recent times it has become quite commonplace. This change is borne out clearly in census figures. For example thirty years ago in Australia, only one marriage in ten ended in divorce; nowadays the figure is more than one in three (Australian Bureau of Statistics, 1996: p.45). A consequence of this change has been a substantial increase in the number of single parent families and the attendant problems that this brings (Kilmartin, 1997).

An important issue for sociologists, and indeed for all of society, is why these changes in marital patterns have occurred. In this essay I will seek to critically examine a number of sociological explanations for the 'divorce phenomenon' and also consider the social policy implications that each explanation carries with it. It will be argued that the best explanations are to be found within a broad socio-economic framework.

One type of explanation for rising divorce has focused on changes in laws relating to marriage. For example, Bilton, Bonnett and Jones (1987) argue that increased rates of divorce do not necessarily indicate that families are now more unstable. It is possible, they claim, that there has always been a degree of marital instability. They suggest that changes in the law have been significant, because they have provided unhappily married couples with 'access to a legal solution to pre-existent marital problems' (p.301). Bilton et al. therefore
```

believe that changes in divorce rates can be best explained in terms of changes in the legal system. The problem with this type of explanation however, is that it does not consider why these laws have changed in the first place. It could be argued that reforms to family law, as well as the increased rate of divorce that has accompanied them, are the product of more fundamental changes in society.

Another type of explanation is one that focuses precisely on these broad societal changes. For example, Nicky Hart (cited in Haralambos, 1995) argues that increases in divorce and marital breakdown are the result of economic changes that have affected the family. One example of these changes is the raised material aspirations of families, which Hart suggests has put pressure on both spouses to become wage earners. Women as a result have been forced to become both homemakers and economic providers. According to Hart, the contradiction of these two roles has lead to conflict and this is the main cause of marital breakdown. It would appear that Hart's explanation cannot account for all cases of divorce - for example, marital breakdown is liable to occur in families where only the husband is working. Nevertheless, her approach, which is to relate changes in family relations to broader social forces, would seem to be more probing than one that looks only at legislative change.

The two explanations described above have very different implications for social policy, especially in relation to how the problem of increasing marital instability might be dealt with. Bilton et al. (1995) offer a legal explanation and hence would see the solutions also being determined in this domain. If rises in divorce are thought to be the consequence of liberal divorce laws, the obvious way to stem this rise is to make them less obtainable. This approach, one imagines, would lead to a reduction in divorce statistics; however, it cannot really be held up as a genuine solution to the problems of marital stress and breakdown in society. Indeed it would seem to be a solution directed more at symptoms than addressing fundamental causes. Furthermore, the experience of social workers, working in the area of family welfare suggests that restricting a couple's access to divorce would in some cases serve only to exacerbate existing marital problems (Johnson, 1981). In those cases where violence is involved, the consequences could be tragic. Apart from all this, returning to more restrictive divorce laws seems to be a solution little favoured by Australians. (Harrison, 1990).

Hart (cited in Haralambos, 1995), writing from a Marxist-feminist position, traces marital conflict to changes in the capitalist economic system and their resultant effect on the roles of men and women. It is difficult to know however, how such an analysis might be translated into practical social policies. This is because the Hart program would appear to require in the first place a radical restructuring of the economic system. Whilst this may be desirable for some, it is not achievable in the present political climate. Hart is right however, to suggest that much marital conflict can be linked in some way to the economic circumstances of families. This is borne out in many statistical surveys which show consistently that rates of divorce are higher among socially disadvantaged families (McDonald, 1993). This situation suggests then that social policies need to be geared to providing support and security for these types of families. It is little cause for optimism however, that in recent years governments of all persuasions have shown an increasing reluctance to fund social welfare programs of this kind.

It is difficult to offer a comprehensive explanation for the growing trend of marital breakdown; and it is even more difficult to find solutions that might ameliorate the problems created by it. Clearly though, as I have argued in this essay, the most useful answers are to be found not within a narrow legal framework, but within a broader socio-economic one.

Finally, it is worth pointing out that, whilst we may appear to be living in a time of increased family instability, research suggests that historically, instability may have been the norm rather than the exception. As Bell and Zajdow (1997) point out, in the past, single parent and step families were more common than is assumed - although the disruptive influence then

```
was not divorce, but the premature death of one or both parents. This situation suggests that
in studying the modern family, one needs to employ a historical perspective, including the
possibility of looking to the past in searching for ways of dealing with problems in the
present."""


Enter the percentage needed to be shrunk to: 30
```

4.2 Summarized Output

```
Title:
Divorce essay
Shrunk Text:
A major change that has occurred in the Western family is an increased incidence in divorce.
Whereas in the past, divorce was a relatively rare occurrence, in recent times it has become
quite commonplace.
For example thirty years ago in Australia, only one marriage in ten ended in divorce;
nowadays the figure is more than one in three (Australian Bureau of Statistics, 1996: p.45).
In this essay I will seek to critically examine a number of sociological explanations for the
'divorce phenomenon' and also consider the social policy implications that each explanation
carries with it.
One type of explanation for rising divorce has focused on changes in laws relating to
marriage.
For example, Bilton, Bonnett and Jones (1987) argue that increased rates of divorce do not
necessarily indicate that families are now more unstable.
therefore believe that changes in divorce rates can be best explained in terms of changes in
the legal system.
It could be argued that reforms to family law, as well as the increased rate of divorce that has
```

accompanied them, are the product of more fundamental changes in society.

For example, Nicky Hart (cited in Haralambos, 1995) argues that increases in divorce and marital breakdown are the result of economic changes that have affected the family.

It would appear that Hart's explanation cannot account for all cases of divorce - for example, marital breakdown is liable to occur in families where only the husband is working.

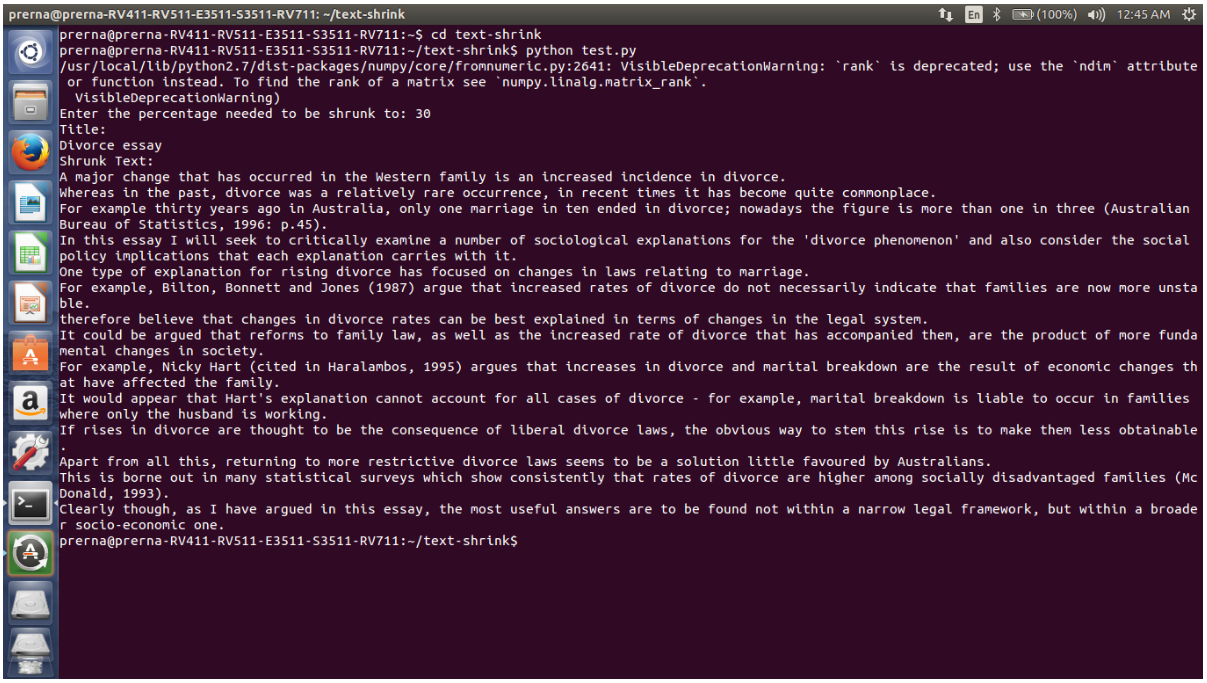
If rises in divorce are thought to be the consequence of liberal divorce laws, the obvious way to stem this rise is to make them less obtainable.

Apart from all this, returning to more restrictive divorce laws seems to be a solution little favoured by Australians.

This is borne out in many statistical surveys which show consistently that rates of divorce are higher among socially disadvantaged families (McDonald, 1993).

Clearly though, as I have argued in this essay, the most useful answers are to be found not within a narrow legal framework, but within a broader socio-economic one.

4.2.1 Screenshot



```
prerna@prerna-RV411-RV511-E3511-S3511-RV711:~$ cd text-shrink
prerna@prerna-RV411-RV511-E3511-S3511-RV711:~/text-shrink$ python test.py
/usr/local/lib/python2.7/dist-packages/numpy/core/fromnumeric.py:2641: VisibleDeprecationWarning: `rank` is deprecated; use the `ndim` attribute
 or function instead. To find the rank of a matrix see `numpy.linalg.matrix_rank`.
  VisibleDeprecationWarning)
Enter the percentage needed to be shrunk to: 30
Title:
Divorce essay
Shrunk Text:
A major change that has occurred in the Western family is an increased incidence in divorce.
Whereas in the past, divorce was a relatively rare occurrence, in recent times it has become quite commonplace.
For example thirty years ago in Australia, only one marriage in ten ended in divorce; nowadays the figure is more than one in three (Australian
Bureau of Statistics, 1996: p.45).
In this essay I will seek to critically examine a number of sociological explanations for the 'divorce phenomenon' and also consider the social
policy implications that each explanation carries with it.
One type of explanation for rising divorce has focused on changes in laws relating to marriage.
For example, Bilton, Bonnett and Jones (1987) argue that increased rates of divorce do not necessarily indicate that families are now more unsta
ble.
therefore believe that changes in divorce rates can be best explained in terms of changes in the legal system.
It could be argued that reforms to family law, as well as the increased rate of divorce that has accompanied them, are the product of more funda
mental changes in society.
For example, Nicky Hart (cited in Haralambos, 1995) argues that increases in divorce and marital breakdown are the result of economic changes th
at have affected the family.
It would appear that Hart's explanation cannot account for all cases of divorce - for example, marital breakdown is liable to occur in families
where only the husband is working.
If rises in divorce are thought to be the consequence of liberal divorce laws, the obvious way to stem this rise is to make them less obtainable
.
Apart from all this, returning to more restrictive divorce laws seems to be a solution little favoured by Australians.
This is borne out in many statistical surveys which show consistently that rates of divorce are higher among socially disadvantaged families (Mc
Donald, 1993).
Clearly though, as I have argued in this essay, the most useful answers are to be found not within a narrow legal framework, but within a broade
r socio-economic one.
prerna@prerna-RV411-RV511-E3511-S3511-RV711:~/text-shrink$
```

Fig 6: Summarized output screenshot

4.3 Summarized and Shrunk Output

Title :

Divorce essay

Shrunk text:

A major change that has occurred in the Western family is an increased incidence in divorce. Whereas in the past, divorce was a relatively rare occurrence, in recent times it has become quite commonplace.

For eg. thirty years ago in Australia, only 1 marriage in ten ended in divorce; nowadays the figure is more than 1 in 3 (Australian Bureau of Statistics, 1996: p.45).

In this essay I will seek 2 critically examine a no. of sociological explanat'ns 4 the 'divorce phenomenon' & also consider the social policy implicat'ns that each explanat'n carries with it.

One type of explanation 4 rising divorce has focused on changes in laws relat'g 2 marriage.

For eg., Bilton, Bonnett & Jones (1987) argue that increased rates of divorce do not necessarily indicate that families r now more unstable.

therefore believe that changes in divorce rates can be best explained in terms of changes in the legal system.

It could be argued that reforms 2 family law, as well as the increased rate of divorce that has accompanied them, r the product of more fundamental changes in society.

For eg., Nicky Hart (cited in Haralambos, 1995) argues that increases in divorce & marital breakdown r the result of economic changes that have affected the family.

It would appear that Hart's explanation cannot account 4 all cases of divorce - 4 eg., marital breakdown is liable 2 occur in families where only the husband is working.

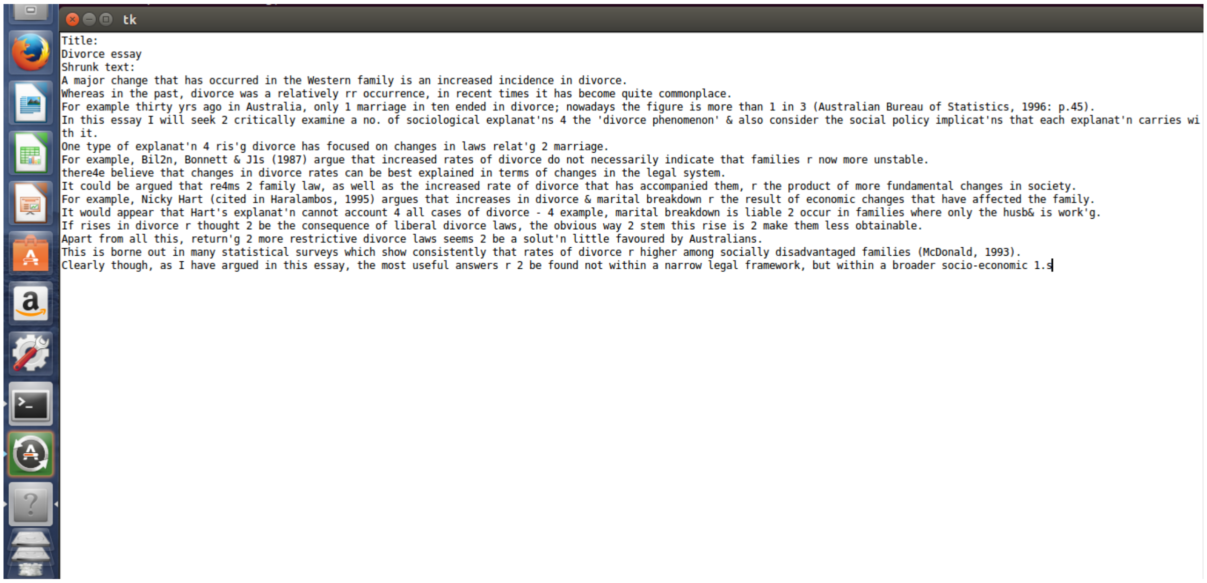
If rises in divorce r thought 2 be the consequence of liberal divorce laws, the obvious way 2 stem this rise is 2 make them less obtainable.

Apart from all this, returning 2 more restrictive divorce laws seems 2 be a solution little favoured by Australians.

This is borne out in many statistical surveys which show consistently that rates of divorce r higher among socially disadvantaged families (McDonald, 1993).

Clearly though, as I have argued in this essay, the most useful answers r 2 be found not within a narrow legal framework, but within a broader socio-economic 1.

4.3.1 Screenshot



```
Title:
Divorce essay
Shrunk text:
A major change that has occurred in the Western family is an increased incidence in divorce.
Whereas in the past, divorce was a relatively rr occurrence, in recent times it has become quite commonplace.
For example thirty yrs ago in Australia, only 1 marriage in ten ended in divorce; nowadays the figure is more than 1 in 3 (Australian Bureau of Statistics, 1996: p.45).
In this essay I will seek 2 critically examine a no. of sociological explanat'ns 4 the 'divorce phenomenon' & also consider the social policy implicat'ns that each explanat'n carries wi
th it.
One type of explanat'n 4 ris'g divorce has focused on changes in laws relat'g 2 marriage.
For example, Bil2n, Bonnett & J1s (1987) argue that increased rates of divorce do not necessarily indicate that families r now more unstable.
there4e believe that changes in divorce rates can be best explained in terms of changes in the legal system.
It could be argued that re4ms 2 family law, as well as the increased rate of divorce that has accompanied them, r the product of more fundamental changes in society.
For example, Nicky Hart (cited in Haralambos, 1995) argues that increases in divorce & marital breakdown r the result of economic changes that have affected the family.
It would appear that Hart's explanat'n cannot account 4 all cases of divorce - 4 example, marital breakdown is liable 2 occur in families where only the husb& is work'g.
If rises in divorce r thought 2 be the consequence of liberal divorce laws, the obvious way 2 stem this rise is 2 make them less obtainable.
Apart from all this, return'g 2 more restrictive divorce laws seems 2 be a solut'n little favoured by Australians.
This is borne out in many statistical surveys which show consistently that rates of divorce r higher among socially disadvantaged families (McDonald, 1993).
Clearly though, as I have argued in this essay, the most useful answers r 2 be found not within a narrow legal framework, but within a broader socio-economic 1.s
```

Fig 7: Summarized and shrunk output screenshot

4.4 Project Analysis

Analysis is based on shrunk percentage entered by user. In example it is 30.

Shrunk %=30	No. of Lines	No. of words	No. of characters
Input	80	1024	6282
Summarized Output	28	338	2055
Summarized and Shrunk Output	28	338	1986

Table 1: Project analysis

4.5 Limitations of Solution

- The output produced by the project might not be 100% accurate. This is because we are dealing with unstructured data.
- Result expected for a query may vary from user to user. A certain solution might be satisfactory for a user but might not satisfy the requirement of another user.
- Stop word elimination might eliminate a word which formed an important part of the query.
- Features might not be sufficient for a certain application.
- Python drawbacks
  - Python is interpreted language & is slow
  - Python is present on many server and desktop platforms, but it is weak in mobile computing; very few Smartphone applications are developed with Python
  - Design restrictions
- With the advent of more and more data big data analytics support would be needed.

CHAPTER 5

CONCLUSIONS

5.1 Findings

- Based on literature survey and books read we can conclude that there are various ways to go by this approach to shrink text.
- There various techniques to summarize based on extraction and abstraction of text.
- I also discovered the significance of various text features and presented and implementation plan.
- I learnt about NLTK and how it can be used to work with human language data.
- From the analysis we can deduce that the project is successfully summarizing according to percentage required.
- The shrink process further reduces the number of characters by a considerable amout although there is no change in number of words.
- This achieves the easily understandable yet shrunk form of text.

5.2 Conclusions

- We can conclude that this project provides the user with a context preserving summary and a shrunk form of text which is easy to interpret.
- The most relevant lines are selected based on keyword frequency, title feature, sentence length and sentence position.
- Summary is generated based on sorting of these sentences and shrinking percentage. After Summary has been generated certain words are further shrunk in understandable format.
- In the example output the input percentage to be shrunk to was 30% which was successfully achieved.

- We can see that the essence of input text is maintained and a novel approach is provided to achieve that.
- We could also conclude that summarization wasn't enough to shrink text. Some words could further be reduced shrinking the text more.

5.3 Future Work

- The project can be scaled to provide for multi-document shrinking. At present only the users looking for single document shrinking can make use of the tool.
- We can include support for other languages. By including other languages this tool could become the one place that be looked up for any language or document related summaries.
- A better GUI with more customized options like URL of input text as input can be provided.
- The present manner of displaying results i.e. as a text box is not very easy to go through. Therefore, various kinds of visualizations like scrolling and options for searching, saving can be applied to the results presented by the tool.
- Summarization can be improved by adding more features like sentences intersection or cosine distance.

5.4 Applications

- Summaries of email threads
- Action items from a meeting
- Simplifying text
- News Articles
- Chapter summary
- Shrinking text messages
- Shrinking twitter updates
- Summarization of information for government officials, businessmen, researches, etc.
- Summarization of web pages to be shown on the screen of a mobile device

REFERENCE

[1] Lloret, Elena. "Text summarization: an overview." *Paper supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01)* (2008).

[2] Jing, Hongyan. "Sentence reduction for automatic text summarization."*Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 2000.

[3] Lovins, Julie B. *Development of a stemming algorithm*. Cambridge: MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[4] Garg, Sneh, and Sunil Chhillar. "Review of text reduction algorithms and text reduction using sentence vectorization." *International Journal of Computer Applications* 107.12 (2014).

[5] Nenkova, Ani, Sameer Maskey, and Yang Liu. "Automatic summarization."*Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*. Association for Computational Linguistics, 2011.

[6] Jagadeesh, J., Prasad Pingali, and Vasudeva Varma. "Sentence Extraction Based Single Document Summarization." *International Institute of Information Technology, Hyderabad, India* 5 (2005).

[7] Hu, Meishan, Aixin Sun, and Ee-Peng Lim. "Comments-oriented blog summarization by sentence extraction." *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 2007.

[8] Gayathri, P., and N. Jaisankar. "A Framework for Efficient Medical Document Summarization using Sentence Feature Extraction and Ranking."*Indian Journal of Science and Technology* 8.33 (2015).

[9] "Text Summarization". *Summarization.com*. N.p., 2015. Web. 28 Dec 2015.

[10] “NLTK Book”. Nltk.org. N.p. , 2015. Web, 15 Nov 2015.

[11] "Tkdocs - Tk Tutorial". Tkdocs.com. N.p., 2016. Web. 02 May 2016

[12] Moore, Tim. "Sample Essay". Monash.edu.au. N.p., 2016. Web. 28 May 2016.

[13] Rautray, Rasmita, Rakesh Chandra Balabantaray, and Anisha Bhardwaj. "Document Summarization Using Sentence Features". International Journal of Information Retrieval Research 5.1 (2015): 36-47. Web. 28 May 2016.

[14] "Python Tutorial". www.tutorialspoint.com. N.p., 2015. Web. 13 oct 2015.