LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

REFRAMING OF ENGLISH SENTENCES USING NLP

A Dissertation Proposal
submitted

By

Vikas Kumar
(11009570)

To

Department of Computer Science and Engineering

In partial fulfilment of the Requirement for the

Award of the Degree of

Master of Technology in Computer
Science and Engineering

Under the guidance of

Mr. Prateek Agrawal

(April 2015)

INDIA'S LARGEST UNIVERSITY

LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education, Transforming India

School of: Computer Science and Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Vikas Kumar	Registration No: 11009570
Batch: 2010	Roll No. RK2006B31
Session: 2014-2015	Parent Section: K2006
Details of Supervisor:	Designation: Asst. Professor
Name: Prateek Agrawal	Qualification: Mtech-CSE
U.ID : 13714	Research Experience: 5 yrs

SPECIALIZATION AREA: Intelligent System **(pick from list of provided specialization areas by DAA)**

PROPOSED TOPICS

1. Reframing of English sentences using NLP.
2. Deriving the title from a given paragraph.
3. Solving blood relation question using NLP.

13714

Signature of Supervisor

PAC Remarks:

Topic 1 is approved. Research paper is also attached.

18/9/14

APPROVAL OF PAC CHAIRPERSON: 11064 **Signature:** **Date:** 18/9/14

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

ABSTRACT

The term Reframing means expressing a sentence differently keeping the meaning of the sentence same as earlier one. Reframing of sentences can be used to change a complex sentence in simplified from. It can also be used to convert direct speech to indirect speech, active voice to passive and vice versa. The scope of reframing sentences is very vast. Reframing sentences concept can be used to write articles from example of earlier articles in new words keeping the meaning the meaning same as earlier one. Reframing of sentences is helpful in making a robot understand different forms of sentences. It can also help English scholars to get the idea about different form of sentences. It is also helpful in designing an intelligent system that can take decisions like humans.

Reframing requires NLP (Natural Language Processing) technique for parsing and morphological analysis of sentences. It also requires a large database for matching and mapping of given words of a sentence to new words stored in database. This research will facilitate in machine learning and processing to sentences by computers just like humans. Since Natural Language Processing can be done.

CERTIFICATE

This is to certify that Vikas Kumar has completed M.Tech dissertation proposal titled Reframing of English sentences using NLP under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma. The dissertation proposal is fit for the submission and the partial fulfilment of the conditions for the award of M.Tech Computer Science and Engineering.

Date:04/05/2015

Signature of Advisor
Name: Prateek Agrawal
UID: 13714

ACKNOWLEDGEMENTS

It is not until I undertake the topic like this one that I realize how massive the effort it really is, and how much I must depend on efforts and working help of others. There are many who helped me in finding this topic for further research, and I want to thank them all from the core of my Hearts

Most importantly I offer my sincerest appreciation to my mentor, Mr. Prateek Agrawal, who has underpinned me throughout my thesis with his tolerance and learning whilst permitting me the room to work in my own specific manner. He always helped to clear all doubts generated during different parts of this literation review and formulation of statement for my research work. His guidance is also a motivation for me to do work on time. His guidance was crucial for formulation of problem statement. I attribute the level of my Master's degree to his consolation and effort and without him this proposal, would not have been composed.

I am also very thankful to our Computer Department for all their valuable technical advices. I would like to thank my friends, who were always willing to help and give their best suggestions. My research would not have been possible without their help.

Finally, I would like to thank my parents. They were always there cheering me up and stood by me through the good and bad times.

DECLARATION

I hereby declare that dissertation proposal entitled, Reframing of English sentences using NLP, submitted for the M.Tech degree is entirely my original work and all ideas and references have been fully acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: 04/05/2014

Vikas Kumar
11009570

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Natural language processing (NLP) is a field of computer science and scientific study of language which deals with the interactions between computers and human languages.

There are many challenges for interaction between computers and humans. Computers understand only binary digits but humans can’t deal with binary digi4ts. So we requires huge database stored in our system for processing of human understandable words by computers.

Natural Language Processing (NLP) is one of the technique through which humans can interact with computers.

Since, NLP is interconnected to the field of human–machine interaction there are many difficulties involved in making a computer understand a human language.

First and the foremost challenge is the platform or the framework through which humans can interact with computer. Natural language processing (NLP) technique can be used to reframe a given sentence differently or in different form keeping the meaning of the sentence same as earlier one. It can also be used to convert direct speech to indirect speech, active voice to passive and vice versa. Reframing sentences concept can be used to write articles from example of earlier articles in new words keeping the meaning the meaning same as earlier one. One of the best suitable language for working of Natural Language Processing is Java.

The historical backdrop of NLP by and large begins in the 1950s. But despite of this fact the work of NLP can be found from very early time. In 1950, Alan Turing distributed an article titled Processing Machinery and Intelligence" which proposed what is currently called the Turing test as a rule of knowledge[1][4][6].

There are various tasks that can be performed using NLP. Some of these tasks are:-

i) Automatic Summarization

In this technique produce a meaningful outline of a summary of the content of a document. Frequently used to give synopses of content of a referred to sort.

For example, articles in the sports related segment of a daily newspaper.

ii) Coreference resolution

Coreference resolution the assignment of all the aspects that refers to same element in a text or paragraph. It is one of the most important task for any NLP operation. It helps in information extraction from any paragraph. Here in the proposed system Stanford CoreNLP is used to resolve coreferences in any given text or paragraph

For example: John is a boy. He live in Jalandhar. He is a student.

Table 1.1: Coreference resolution

Sentence Number	Word
1	John
2	He
3	He

iii) Discourse Analysis

This discourse analysis incorporates various inter-related assignments. One undertaking is distinguishing the structure of associated content that is the way of the connections are made between the sentences. Other conceivable assignment is perceiving and ordering the discourse demonstrations in a summary of content that is yes-no queries, substance questions, explanation, declaration, etc.

iv) Machine Translation

One of the most important task in NLP is automatic machine translations from one language to another language. It is one of the most difficult task and it requires different types of knowledge that a human possess such as grammar rules, semantic rules, and various other knowledge.

v) Morphological segmentation

Separating the words in different units and recognize the class of these units. The trouble of this assignment depends significantly on the intricacy of the word structure of the dialect being taken into consideration. English has genuinely straightforward word structure, particularly inflectional word structure, and hence it is frequently conceivable to overlook this undertaking totally and basically show every conceivable type of a word as partitioned words.

vi) Name Entity Recognition (NER)

When a paragraph or text is given, Natural Language Processing technique is used to identify names entities in the paragraph. These name entities can be name of a place, individual, association, area etc. Name entities enable a machine to work like humans and process sentences in the same way as humans.

vii) Natural language understanding

When a text has formal representations, in different way which a not defined in the databases, it is difficult for a computer to get the meaning of that paragraph. Here natural language processing plays an important role. NLP enables a computer get the paragraph meaning and apply different grammar rules accordingly.

viii) Optical character recognition (OCR)

One of the most important task of Natural Language Processing is Optical Character Recognition. NLP techniques scans an image and identifies the character in that image. This techniques is very helpful to read texts from a banner and posters where direct text is not available and everything is in the form of an image.

ix) Question answering

When question is presented to a machine it is very difficult for a machine to understand that question without any processing. Here comes the role of Natural Language Processing. NLP techniques process the questions and converts it in the form of an equation which is easily understandable of the machines.

x) Speech recognition

When a sound clip of a person is given to a computer, the computer finds the way in which that text is represented. It is very difficult for a machine to process speech without any NLP techniques. Many types of the languages that are spoken and represented in different ways. So the machine converts those analog signal into distinct characters and processes accordingly.

There is a certainly no pause between two words in speech to is it very difficult to segment a speech into subtask of speech. So natural languages processing plays an important role in speech processing[2][3][14].

1.1 Sentences

A sentence can incorporate words assembled definitively to express a statement, question, exclamation, demand, summon or suggestion. A sentence consists of words that on a basic level tells a complete thought, despite the fact that it may make little sense taken in disconnection outside of any relevant connection to the subject [15].

A sentence may consists of following components

i. Subject
ii. Predicate
iii. Clause
iv. Phrase
v. Modifier

Subject

The subject of a sentence refers to an individual, thing, or thought which is doing or being something. The subject of a sentence is either a noun or a noun phrase.
For example:
"Ram is running."
Here ‘Ram’ is the subject, because he is the actor in the sentence, who is doing something.

Predicate

There are mainly two parts of a sentence, one is subject and other is predicate. A predicate modifies the subject. The predicate consists of a verb, and the verb bounds that allows a sentence to complete in meaningful way.
For example:
“Ram is running.”
Here ‘is running ’, is the predicate because it contains a verb and it is providing the information about the subject.

Clause

A clause tells some additional information about the subject.

For example:

“Ram is running fast.”

Here ‘fast’ is clause because it is adding additional information to subject.

Phrase

A phrase is like a dependent clause. A phrase is a group of words which can’t be alone as a sentence, but it can add some information to a sentence.

For example:

‘In the river’

Here, ‘In the river’ is a phrase and it can be attached to a sentence.

Modifiers

A modifier is a part of a sentence which adds certain information to a sentence or change the actual meaning of a sentence [8].

1.2 Reframing of sentences

The term Reframing means expressing a sentence differently keeping the meaning of the sentence same as earlier one. Reframing of sentences can be used to change a complex sentence in simplified from. It can also be used to convert direct speech to indirect speech, active voice to passive and vice versa. The scope of reframing sentences is very vast. Reframing sentences concept can be used to write articles from example of earlier articles in new words keeping the meaning the meaning same as earlier one. Reframing of sentences is helpful in making a robot understand different forms of sentences. It can also help English scholars to get the idea about different form of sentences. It is also helpful in designing an intelligent system that can take decisions like humans.

Different ways to reframe a sentence

i. Conversion from active to Passive voice.

ii. Conversion from direct speech to indirect speech.

iii. Replacing words with its synonyms.

iv. Replacing negative words with antonyms.

v. Changing the ordering of sentences.

vi. Prioritizing the sentences in a paragraph.

1.3 Subject Verb Agreement

The subject verb agreement principle tell, singular subjects must have singular verb and plural subjects must have plural verbs and a sentence must satisfy this principle to be grammatically correct. This will form the basis of sentence reframing technique [16].

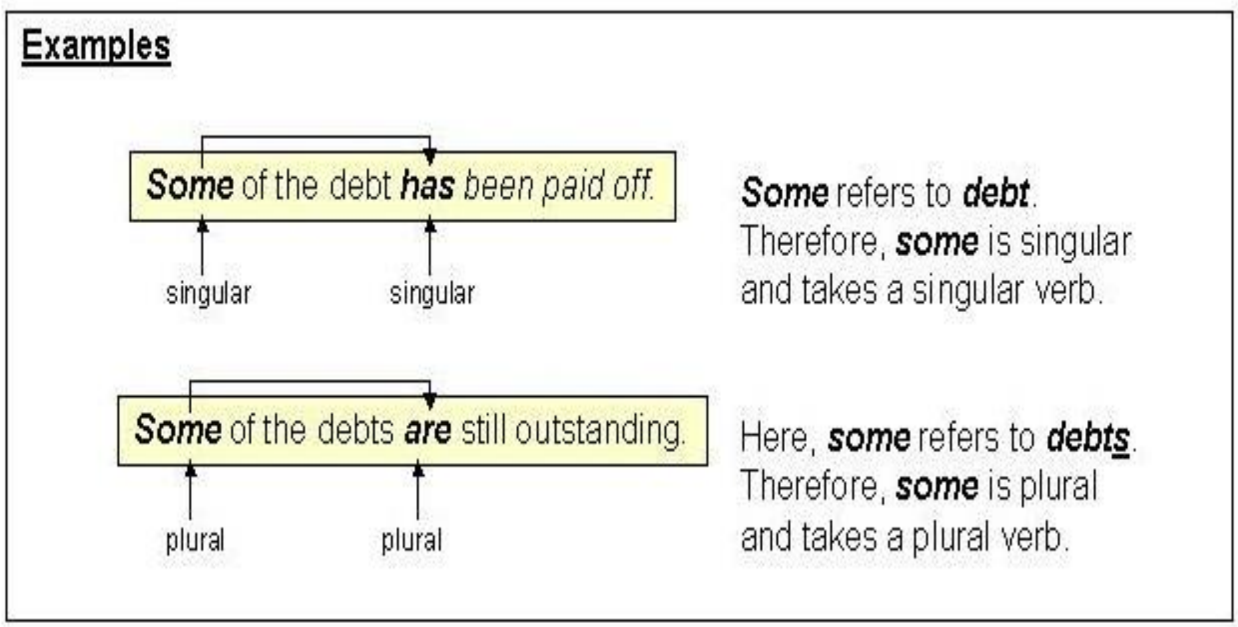
For example:

i. Mohan is playing cricket.

Here 'Mohan' is singular, thus 'is' verb is used.

ii. They are dancing.

Here 'They' is plural, thus 'are' verb is used.



Examples

Some of the debt **has** *been paid off.*

Some refers to **debt**. Therefore, **some** is singular and takes a singular verb.

Some of the debts **are** still outstanding.

Here, **some** refers to **debts**. Therefore, **some** is plural and takes a plural verb.

Figure 1.1: Subject Verb Agreement[16]

1.4 General architecture for using NLP for sentence reframing

1.4.1 Text Segmentation

Firstly the text is divided into segments or different sentences. Sentence division is the issue of isolating a string of composed dialect into its segment sentences. In English and some different dialects, utilizing accentuation, especially the full stop character is a sensible estimate. However even in English this issue is important because of the utilization of the full stop character for truncations, which might possibly additionally end a sentence. When preparing plain content, tables of truncations that contain periods can help forestall erroneous task of sentence limits.

Similarly as with word division, not every composed dialect contain accentuation characters which are helpful for approximating sentence limits.

Programmed division is the issue in characteristic dialect preparing of executing a PC methodology to section content. At the point when accentuation and comparable intimations are not reliably accessible, the division errand frequently requires reasonably non-minor methods, for example, factual choice making, vast word references, and thought of syntactic and semantic limitations. Compelling regular dialect handling frameworks and content division apparatuses typically work on content in particular spaces and sources. As a case, preparing content utilized as a part of restorative records is an altogether different issue than handling news articles or land promotions[14].

1.4.2 Parts of Speech Tagging

Parsing or syntactic examination is the strategy of separating a progression of pictures, either in trademark tongue or in scripts, acclimating to the fundamentals of a formal sentence structure. The term has possibly differing ramifications in particular branches of semantics and programming building. Customary sentence parsing is routinely executed as a framework for understanding the exact significance of a sentence, on occasion with the backing of devices, for instance, sentence diagrams. It usually underlines the criticalness of semantic divisions, for instance, subject and predicate. Inside computational phonetics the term is used to insinuate the formal examination by a PC of a sentence or different arrangement of words into its constituents, achieving a parse tree exhibiting their syntactic association with each other, which may in like manner contain semantic and other information.

Part-of-speech tagging (POS tagging)/POST is a process of giving identification to words of a sentence such as noun , pronoun , verb, adverb , adjective , determinants etc. After POS tagging a sentence and be further processed as per requirements.
For example: John is a boy. He live in Jalandhar. He is a student[17].

Sentence after POS tagging
John_NNP is_VBZ a_DT boy_NN ._. He_PRP live_VBP in_IN Jalandhar_NNP ._. He_PRP is_VBZ a_DT student_NN ._.

1.4.3 Syntax Analysis (Parsing)

The main scope of this process is parsing. It plays an essential role in understanding many language systems. In this step, a simple sentence is granted as an input which is converted into a hierarchical form that address to the units of meaning in the sentence. It uses the first components of the token to produce tree probably intermediate structure that deplicts the grammatical structure of the token stream[15].

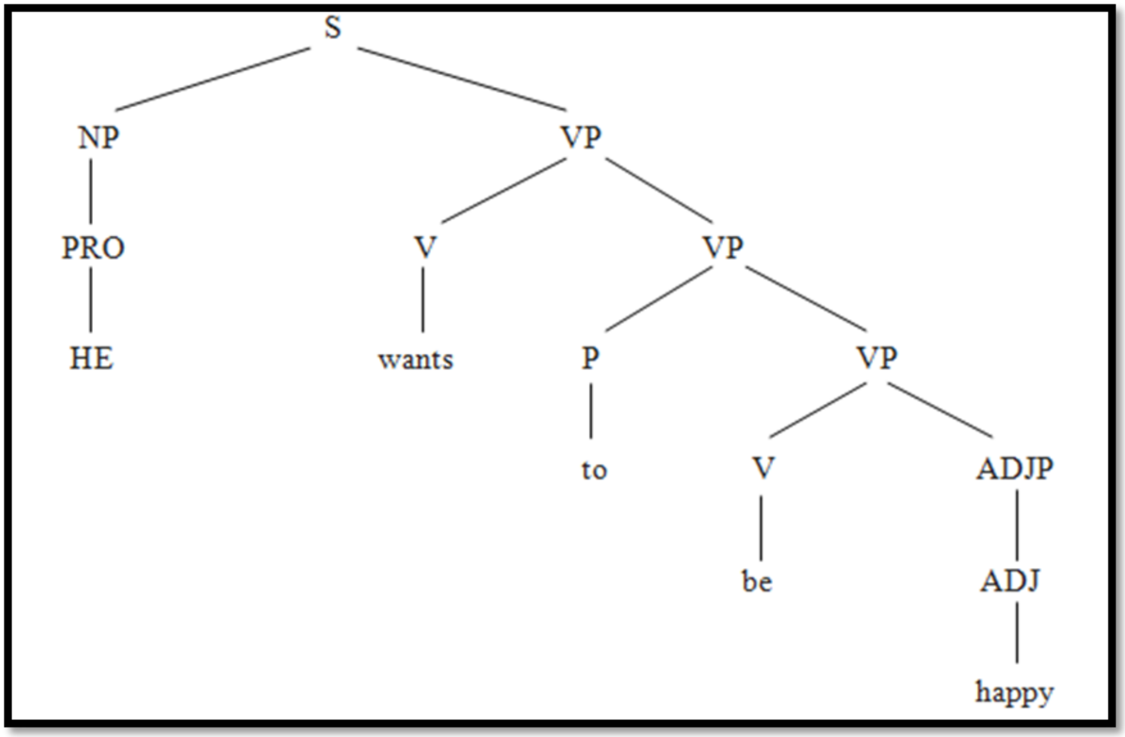


Figure 1.2: The result of syntactic analysis of "He wants to be happy"

There are distinct parsing formalisms and algorithms in which formalism has two leading components:

i) Grammar: A declaratory representation that portray the syntactic structure of sentences in the language.

ii) Parser Based on the morpho-syntax rules, it captivate the input and gives output as a syntax tree.

1.4.4 Semantic Analysis

Semantic analysis is used to check whether inserted sentence is accurate or not. Although the main intend of semantic analysis is the formation of the target language representation of the sentence's meaning which indicate assigning meanings to the structures created by syntactic analysis. Semantic can play an import role in opt among contend in syntactic analyses. For example: I need to complete the project – here project is a plan/ a proposal or to cause a shadow. So here we have to make up one's mind the formalisms which will be utility in the meaning representation[5].

1.4.5 Mapping

Mapping is the drawing conclusions from the given premises to solve problems and make decisions. It basically manipulates the given knowledge and generates new knowledge from the defined rules and facts that are stored in the database. It derives the new knowledge with the help of logics or by using inference. So, the relationships that are stored in database and will be used[13].

1.4.6 Reframing Rules

In the database the knowledge is represented in the forms of rules and facts. These facts and rules are applied to get the desired output. The rules can be certain patterns or symbols that matches with the input and for applying the processing.

The reframing restatement the paragraph without changing the meaning of the actual text. It explains the paragraph in simple words and enables to present the same paragraph in different ways.

Following example explains the meaning of reframing:

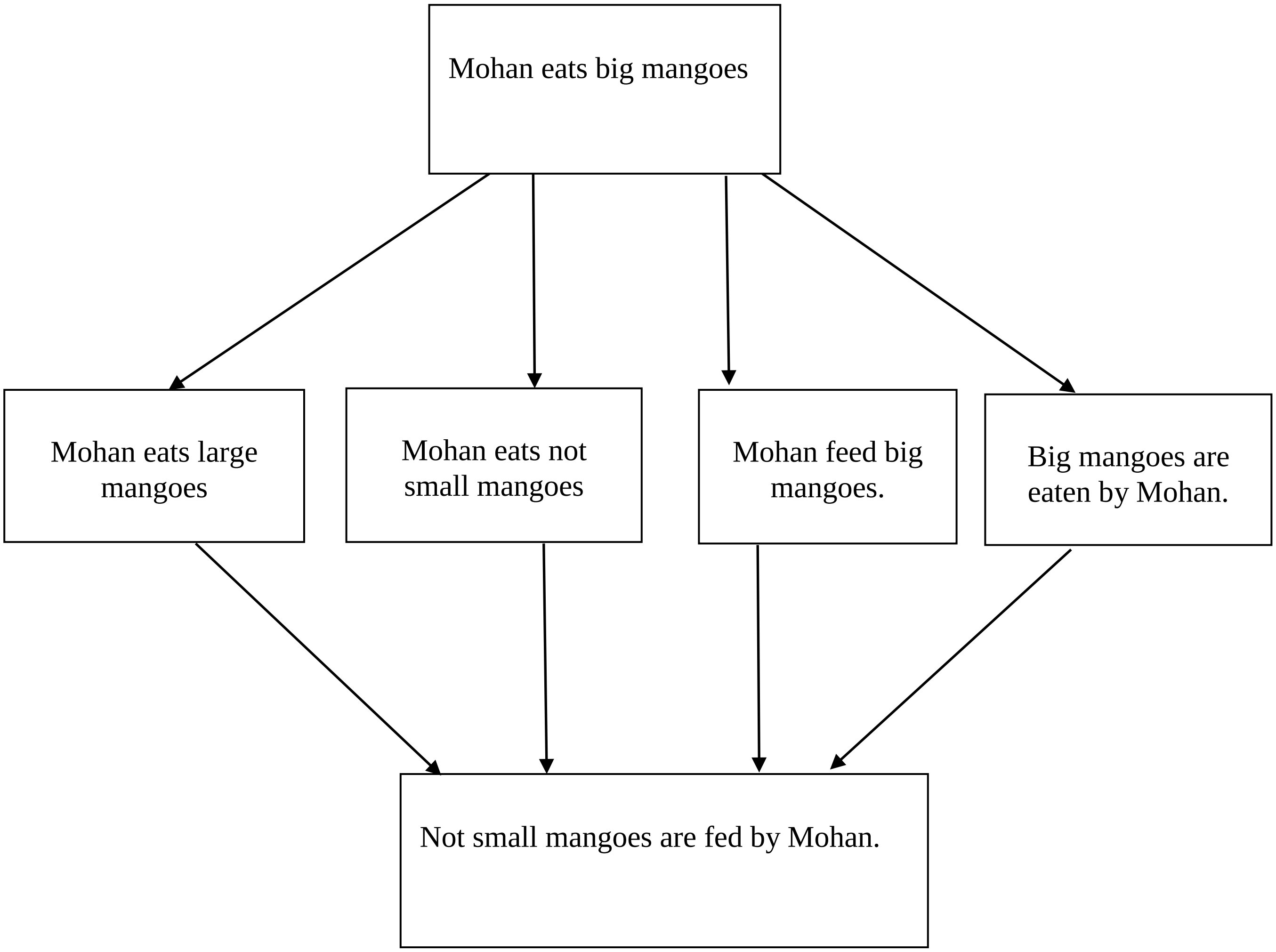

Mohan eats big mangoes
Mohan eats large mangoes
Mohan eats not small mangoes
Mohan feed big mangoes.
Big mangoes are eaten by Mohan.
Not small mangoes are fed by Mohan.

Figure 1.3: Reframing types

Here , the sentence “Mohan eats big mangoes” is , reframed using synonym replacement rules , antonym replacement rules and active to passive voice conversion rule. Here the actual meaning of the sentence is kept same as the original sentence. There is either change in the sequence of the word or , word's synonym or synonym is used. The reframing do not accompany the direct reference , it serves as a source to new reframed paragraph.

1.5 Java and Java Libraries

1.5.1 Introduction to Java

Java is an object-oriented programming language developed by Sun Microsystems of USA in 1991. Java is also called Oak by James Gosling in starting. Java was invented for the development of software for consumer electronic devices like TVs, toasters, etc. Java is a simple, portable and reliable programming language. Java is the only language which does not depend on any operating system or hardware. Program can be executed on any system where java installed.

Java applications are firstly compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM). Java is reliable language that is specifically designed so that it has few dependencies. Therefore, once it is written it can be run on any system and in any type of environment[12].

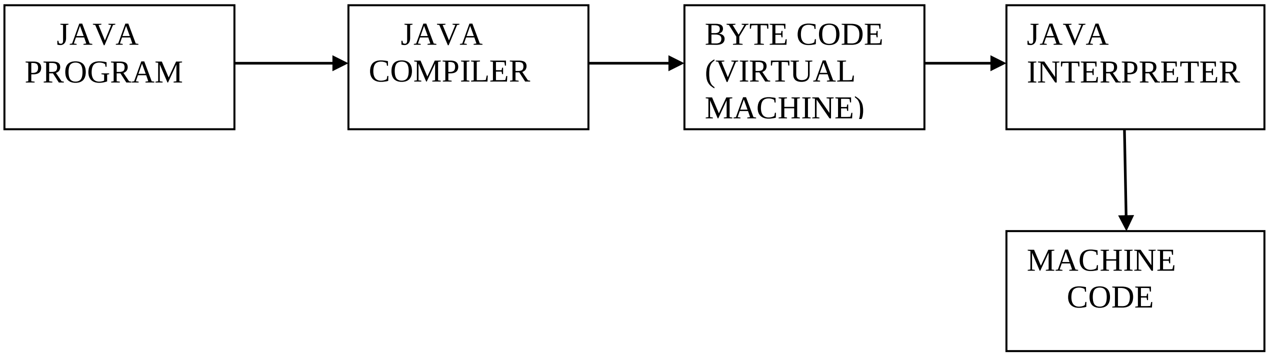


Figure 1.4: JVM[12]

1.5.2 JAVA Features

1.5.2.1 Compiled and Interpreted: All languages are either compiled or interpreted but not both. Java is the only language which uses both. Java compiler translates java code into bytecode whereas interpreter translate java code into machine code.

1.5.2.2 Platform Independent and portable: Java supports portability i.e. can be easily moved from one computer system to another. Java sure portability due to following reasons:

Java compiler generate the bytecode and that can be executed on any machine.

Primitive data types sizes are machine independent.

1.5.2.3 Object- oriented: Java is object-oriented language. Everything is an Object in Java. All code exists in objects and classes. Java has many classes organize in packages.

1.5.2.4 Robust and secure: Java is a most secure language, so it is used in the internet. It helps the programmers virtually from the problems of all memory management. Java also uses exception handling, to detect serious errors and all kind of threats.

1.5.2.5 Distributed: Java is a Distributed language for applications on networks which can contribute both data and programs. Java applications access remote objects very easily. It means multiple programmers can work together on single task present at multiple remote locations.

1.5.2.6 Simple and Small: Java is very simple and small language. Java does not header files, goto statements, pointer, operator overloading and multiple inheritance.

1.5.2.7 Multithreaded and Interactive: Multithreaded means run many programs at a time as a instance. In Java, we implement multithreaded programs means there is no need to wait to finish one task to start new task. Now both task are able to run on the same time. This feature is used in graphics applications.

1.5.2.8 High performance: Java uses intermediate bytecode, therefore performance is very high for an interpreted language as compared to others. It also reduces runtime. Execution speed is also very high due to multithreading.

1.5.2.9 Dynamic and Extensible: Java dynamically links new class, libraries, methods and objects. Support functions written in other language such as C and C++ are Java program, called as native methods.

1.5.3 WHY JAVA?

1.5.3.1 Swing: Java Swing is a lightweight component. It includes many packages and has controls to build trees, image buttons, tabbed panes, sliders, toolbars, tables and text areas. Swing is a platform independent. Swing has more additional functions and is faster as compared to the awt. The default layout the Swing is the Border Layout. It provides pluggable look and feel so we can change the look and feel of every component without affecting the coding of the application. We can also change both visual and physical behavior of the GUI component. Swings are used to make simple as well as comprehensive GUI projects. Applications written in Swings are easy to write and also easy to understand.

We choose Swing because:

i) Swing has an extensively rich set of user interface elements which are convenient to use.

ii) Swing gives a user experience which is consistent.

iii) Swing is a platform independent.

1.5.4 Java Event Delegation Model: In GUI, users can click, input data using the keyboard drag and drop components of the interface. All these actions generate an event which is handled by code written by the user.

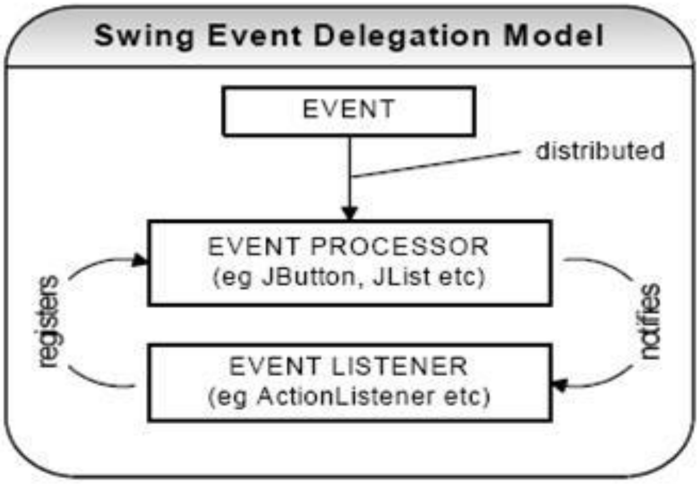


Figure 1.5: Event delegation model[12]

Event delegation model has 3 components: Event, Listener and Source.

i) **Events:** In the delegation model, an event is an object which occurs when state change in a source. Event is generated when user interact with GUI i.e. user can drag, drop, click, press a button of the mouse, input character from keyboard and select an item all creates event. It is not important that event can occur only when user interact with GUI, it can also occur due to hardware or software failure, when operation is completed or time expires.

ii) **Event Source:** A source can be any object that generates an event. When the internal state of that object changes, it creates an event. Sources can generate different types of event. Each type of event has its own registration method. Event Object extends java.util.EventObject subclasses. For example, if a user clicks an icon, we can add up the mouse-clicked event and the mouse-moved event into one event object.

iii) **Event Listener:** When an event occurs, listener is informed. It extends java.awt.event class. Two conditions of listener is :

a) registered with any source.

b) implement method so that it can take and process notifications.

For example, For example, the MouseListener interface handles mouse events, and the ActionListener interface handles action events generated by buttons and other components.

1.5.5 Java Bean: A Java Bean is a Java technology class with at minimum following features:

i) A no-argument Constructor.
ii) No public class instance variable.
iii) class implements java.io.Serializable interface.
iv) Properties defined with mutators and accessors.

1.5.6 Java Platform

The Java is platform independent environment. All the programs in Java run in a virtual environment called Java Virtual Machine(JVM). A Java virtual machine (JVM) is a connected planning machine which is known for its detail, execution, and case. The determination portrays the required of a JVM execution. It promises all use are interoperable if it has single specific. A JVM utilization is a PC program that meets the essentials of the JVM detail in a wonderful and in a perfect world each formant way. An event of the JVM is a system where PC framework amassed into Java “Bytecode” gets executed. The Java virtual machine is a theoretical (virtual) PC delineated by a specific. This specific releases execution unassuming parts that are not key to ensure interoperability. Like the , the memory design of run-time data districts, the waste social undertaking number used, and any internal advancement of the Java virtual machine controls (their understanding into machine code). The key enlightenment behind this oversight is to not pointlessly compel powers. Any Java application can be run simply inside some strong execution of the sensible specific of the Java virtual machine[12].

1.5.7 Java Environment

The best thing about using Java is it is platform and environment independent. The same codes can run in windows or linux environment without making any changes. The “Byte code” generated by Java JVM compiler is independed and can run in any JVM environment.

1.5.8 Java Library JWI

Java library JWI is an interface provided by MIT for accessing Wordnet in Java platform. The principle interface for getting to word reference information is the edu.mit.jwi.IDictionary interface. The standard usage of this interface is the edu.mit.jwi.Dictionary class. In the least complex case, where you are utilizing Wordnet with the information Files on the same File system as our Java program, we can instantiate the Lexicon class with a solitary contention, a Java URL object that indicates the index where the Wordnet word reference information Files are found[21].

1.5.9 Java Library SimpleNLG

Simplenlg is a straightforward Java API intended to encourage the era of Natural Language. It was initially created at the University of Aberdeen's Department of Computing Science.

Simplenlg is planned to capacity as an "acknowledgment motor" for Natural Language Generation architectures, and has been utilized effectively as a part of various ventures, including the BabyTalk and BabyTalk-Family Projects. The simplenlg java library contains the different verb forms , which is used for conversion from active to passive voice[10].

1.5.10 Java Library Regex

The Regex of Java or the regular expression in Java is an API that specifies the pattern for determining the matching string. In websites regex is used to form validation. Regex restricts the form in a login website using applied pattern. In java REGEX is used by including the java.util.regex that incorporates one interface and three classes.

It provides following classes and interface for regular expressions. The Matcher and Pattern classes are widely used in java regular expression.

i). Match Result interface

ii). Matcher class

iii). Pattern class

iv). Pattern Syntax Exception class

A search pattern for strings is defined by general expression. The shorten form for regular expression is regex. The search pattern that is described by the regex can be in any form either a simple character or a fixed string or any complex expressions which contains some special character that describes the patterns, which might match one or a few times or not in the least for a given string. General expression can be utilized to alter, control and search the text.

A pattern object is an accumulation of regular expressions. In a pattern class there is no public constructor available, hence, whenever a pattern need to be created, first of all a call is made to one of public static method, which then returns a pattern object. These static methods accept a regular expression as the first argument.

A Matcher object is a tool that understands the pattern and carries out match operations against a string which is given as an input. In Matcher also, there is no definition of public constructors. So, to obtain a matcher object, a matcher method is invoked on a pattern object[12].

CHAPTER 2
REVIEW OF LITERATURE

This chapter will survey the different types of algorithm and software for morphological analysis of words. Here we will try to explain certain techniques for determining the importance of words in a sentence on the basis of different factors. Here we will also see summery techniques, about Google translators and Term frequency-inverse document frequency (tf-idf) algorithm.

Guido Minnen, John Carroll and Darren Pearce (2001) et al. [9], proposed a system of Morphological study in which internal structure of words were studied .An inflected word, which is composed of a root word , with some prefix or suffix , to make it a plural form (e.g. affix -s) or past tense (e.g. affix -ed).In this proposed system inflected words were reduced to their stems by removing the affixes and root word was found. In this proposed system some flex rules were discussed such as:

```
{A}+{C} “ied”        {return (lemma (3, “ed”));   }            (analyzer)
{A}+{C} “y+ed”       {return (glemma (4, “ied”)); }            (generator)
```

For example:

```
“boogi+ed”          {return (lemma(3, “”,“ed”)); }
“boogi+yed”         {return (lemma(4, “”,“ied”));}
```

By applying flex rules like this affixes were separated out from inflected word and stem word was found.

Maria Soledad and Pera Yiu-Kai Ng (2009) et al. [7]**,** proposed a system in which, CorSum technique was used to summarize web documents. CorSum technique summarize web documents based on words similarity. It used I Bayes classifier to train the system. The factor of correlation of two words wi and wj was used to find the distance between two words were based on two factors namely, frequency of co-occurrence and the relative distance of wi and wj in any document D and is defined as follows:

$$wcf(w_i, w_j) = \sum_{x \in V(w_i)} \sum_{y \in V(w_j)} \frac{1}{d(x, y)}$$

… (1)

where d(x, y) denotes the distance (i.e., the number of words in) between x and y in D plus 1, and V (wi) (V (wj).

Then degree of similarity of any two sentences S1 and S2 by was calculated

$$Sim(S_1, S_2) = \sum_{i=1}^{n} \sum_{j=1}^{m} wcf(w_i, w_j)$$

… (2)

where wi,wj , is a word in S1,S2 respectively and n ,m is the number of words in S1,S2 respectively), and wcf(wi, wj) is given in Equation 1.

After CorSum was used to calulate if a sentence is to be included in a summery by using

OS(Si) = Σj=1,i=jn Sim(Si, Sj)

where n is number of different sentences in D, Si and Sj are sentences in D, and Sim(Si, Sj) is defined in Equation 2.

Finally, Odds ratio was used to the Rank value of Si in D

$$Rank(S_i) = \frac{OS(S_i)}{1 - OS(S_i)}$$

… (3)

where OS(Si) as the positive evidence of Si in representing the content document D.

Working of Google Translator [19], Google translator generates translated documents by comparing with hundreds of documents which has already been translated by humans. Google believes that if someone wants to learn a new language, then he must start by learning vocabulary and grammatical rules which are further used to construct grammatically correct sentences. A machine could learn foreign language in the similar manner like humans by consulting to vocabulary and set of grammar rules. But in spite this fact languages are difficult to learn as there are many exception to any language and in computer programs exceptions reduces the quality of translations. So Google uses different approach to translate documents. Google Translate makes guesses of translation using a process of seeking patterns in large amounts of text. When the computer recognizes a pattern, it uses similar already translated documents by humans and saves the translated document for future use. It even takes users suggestions for improvement in translation. These user suggestions are further reviewed by the developer team and transition algorithm is further modified to match translations with human translated documents.

Term frequency–inverse document frequency (tf-idf) algorithm [18], tells the importance of a word in a document. It assigns a weighting factor for getting information from the text and for text mining. The value of tf-idf increases proportionally for repetition of word in a document. The tf–idf is the product of two values, namely term frequency and inverse document frequency.
TF: Term Frequency, which measures how frequently a term occurs in a document.

TF(t) = (Number of times a word t appears in a document) /
(Total number of words in the document).

IDF: Inverse Document Frequency, thats measures the importance of a word in a document.
IDF (t) = log_e(Total number of documents / Number of documents with word t in it).

Girish Mishra, Sohan Lal Nitharwal and Sarvjeet Kaur (2010) et al. [8]**,** proposed a system in which**,** Fuzzy- SVM method was used for recognition of five Indian languages i.e. Urdu, Kashmiri, Bangla, Manipuri, and Hindi,. In this proposed system, different patterns are extracted and represented it on normalized scale which was then given to SVM (Support Vector Machine) as input. The proposed Pattern Recognition system was capable of identifying even identify plaintext which was used for automatic separation of plain text while analyzing intercepted, multiplexed, and interleaved Speech, Data and Fax correspondence. This proposed system took only the set of different features of languages in Romanized texts. Than fuzzy sets were compose for features like frequency of definite characters, double characters, and the frequency of diagraph starting and ending with a letter. The proposed system was processing texts and it was independent of dictionaries in spite when the boundaries of word were unknown. SVM was then performed for classification by constructing a hyper plane of an N-dimensional that separated the data into two categories optimally. SVM have advantages like robustness, not require any parameter tuning, and high dimensional feature space for generalization.

Rekha S. Sugandhi, Ritika Shekhar, Tarun Agarwal, Rajneesh K. Bedi, Vijay M. Wadhai (2011) et al[11], proposed system that focused on the development of a Machine Aided Translation (MAT) system i.e. translate the text from English to Hindi. To generate parse tree for Indian languages was very difficult therefore, the proposed system present a parser that identify the syntactical elements of English and translate it to Hindi. The machine translation based on the following techniques:

i). Rule-Based – Rules for parsing and translation are written in form, like context free grammars.

ii). Example-Based – Principle of analogy between two or more parallel corpora of different languages involved in the translation process.

iii). Statistical Method – Finding probabilities of various words of source text converted to target language, word which has the highest probability was chosen as the target word.

In this proposed system focused on the grammar rules, written for the parsing of the source sentence. Firstly the source sentence was parsed for syntactic correctness and then translation rules were applied.

In all languages, sentences are organized by defining rules for three major constituents- Subject, Object and Verb. The syntactic structures of sentences are represented in one of the following forms:

Subject-Verb-Object (S-V-O)
Subject-Object-Verb (S-O-V)
Object- Subject-Verb (O- S-V)

For e.g.: Mohan eats mango.
(Subject) (Verb) (Object)

Then for parsing, set of production rules were written with the help of grammar rules and then parse tree was generated. i.e.

<S> -> <NP><VP>
<NP> -> <Det><NN>
<VP> -> <VB><NP> | <VB>

CHAPTER 3
SCOPE OF STUDY

Scope of the research is the limit that we set for our research work. A research work cannot have all be applicable for all areas and as it must have some restrictions and it may lead to biased conclusions. Time and money constraints for the resources are some of the constraints in research. Therefore we must define the scope of research for correctness and authenticity. For the research to be meaningful it is necessary to mention areas that will be covered and what are the areas that will not be covered in the research.

This research work deals with reframing of English sentences without changing the actual meaning of the sentence. An English sentence can be written in many ways expressing the same meaning or idea. The main purpose of this research work is for generated all the grammatically correct sentence.

The scope of reframing sentences is very vast. Reframing sentences concept can be used to write articles from example of earlier articles in new words keeping the meaning the meaning same as earlier one. Reframing of sentences is helpful in making a robot understand different forms of sentences. It can also help English scholars to get the idea about different form of sentences. It is also helpful in designing an intelligent system that can take decisions like humans.

This work can be further extended to make a decision support system that will work is a similar manner like humans and can respond just like humans by understanding the different forms of sentences given to it as an input.

The efficiency and performance of any intelligent system entirely depends upon the size of knowledge base being used in that intelligent system. Thus, the main limitation of this research is its working in case of complex sentence. The reframing of complex sentences will entirely depend upon the size of database being used in the research work. And sometimes it might not work in reframing of complex sentences. But this limited can be removed in future by expanding the size of database and dictionary that is being used in the system

CHAPTER 4
OBJECTIVES OF THE STUDY

The main objective of the system is to reframe English sentences using Natural Language Processing techniques of artificial intelligent. English sentences can be reframed using various grammar rules such as active to passive voice conversion, direct to indirect conversion, replacing words with its synonyms, replacing negative words with its antonyms.

In this system input may be taken as a paragraph document and after processing new documents with new English sentences will be given as output. This system can also be used to validate Subject - Verb Agreement , sentence tokenization , parts of speech tagging , discourse integration, word stemming and sentence classification.

In the present scenario most of the plagiarism checker software works on syntactic analysis, but through this research work it would be possible do semantic analysis of document. In syntactic analysis the actual words of a sentence are checked during plagiarism check, but in sematic analysis whole structuring of a sentence are check during analysis.

A sentence can be formed in many ways and reframing plays a key role to make a machine work like humans. Thus this research work will play an important in decision support system that will work in similar manner like human based decision support system.

Sometimes this tool can also be used for article writing which will avoid the changes of plagiarism. In companies a lot of time and recourses are wasted in production promotion by making tag lines and writing review articles. In those scenarios this tool can generate automated articles based of the sample input and will boost product sales.

This research work can also be used for generating automated blog articles which will help the blog writers to improve the content quality of the articles. Thus this tool will help language scholars, technical writers, business man, artificial intelligent industry and as a plagiarism checker tool.

CHAPTER 5
RESEARCH METHODOLODY

5.1 Natural Language Processing (NLP)

The entire idea of this thesis work is based of Natural Language Processing. The main concept to be used in this thesis work is taking a paragraph or text as input. Then sentence tokenization and work tokenization rules to be applied. Then different sentences will processed separately after discourse integration and new sentences will be generated based on already defined rules line the knowledge base.

5.2 Feasibility Study

The feasibility study is review of the facts to see if it proceeds to the analysis phase. The feasibility analysis is the primary tool to calculate whether to proceed to the next phase or to reject the project. Depending on the results of the initial investigation, it is expanded to a more detailed. Feasibility study is a test of a system proposal according to impact ability to meet user needs, workability and effective use of resources. The main objective of a feasibility study is to find whether the proposed system is feasible but not to solve the problem. After determining that project is feasible, analyst finalizes the requirements and starts preparing for the next phase. It has a time constraint i.e. written or oral feasibility report submit by the user. The observations and content decide whether to proceed, cancel or postpone the project.

5.2.1 Types of feasibility analysis

i) Economic Feasibility

ii) Technical Feasibility

iii) Operational Feasibility

iv) Behavioral Feasibility

5.2.1.1 Economic Feasibility: The economic feasibility see if the project has an acceptable return on investment. The rough projected costs will only be estimate. There is no need to calculate exact cost. Project is feasible if and only if it costs remains within the budget. The software "Reframing of Sentences" is economically feasible because

The software requires very less time factors.

The software will be easy to use and there will be no cost required for running the software.

There is no training cost required to learn it.

The software will not require expensive hardware to develop it.

5.2.1.2 Technical Feasibility: The technical feasibility means technical requirements. The technical requirements are then compared to the technical capability of the organization. The systems project is technically feasible if the internal technical capability is sufficient to support the project requirements. In this project, the programming should be done in such a way that the application is fast and uses less resource. As the technology used for application is Java therefore, require only jdk and jre installed in the system.

5.2.1.3 Operational Feasibility: The operational requirement means management commitment to the proposed project. If management initiated the request, it means there is management support and the system will be accepted and used. It is operationally feasible because it can be helpful to those who want to learn Hindi. People generally know Hindi words but they don't know exactly about the prefix, suffix and the root word because when any affix can be added to root word it changes the meaning of the word. Therefore, this software will prove helpful for all.

5.2.1.4 Behavioral Feasibility: The software that is being developed is user friendly and easy to learn. In this way, it will be a whole good experience for the "Stemming of Hindi words" without knowing a lot about the computer system.

5.3 System Analysis

The aim of the system analysis is to understand the requirements of the customer and to document them properly. It consists of two activities:

i) Requirements gathering and analysis

ii) Requirements specification

The requirements gathering activity is done to collect all relevant information from the customer regarding the product so that we can clearly understand the customer requirements to remove incompleteness and inconsistencies from the project[20].

5.4 Gathering Information

Information about the present system needs to be gathered as possible. There are many techniques to gather information like:

i) Observation

ii) Interviews

iii) Questionnaires

iv) Collecting Documents

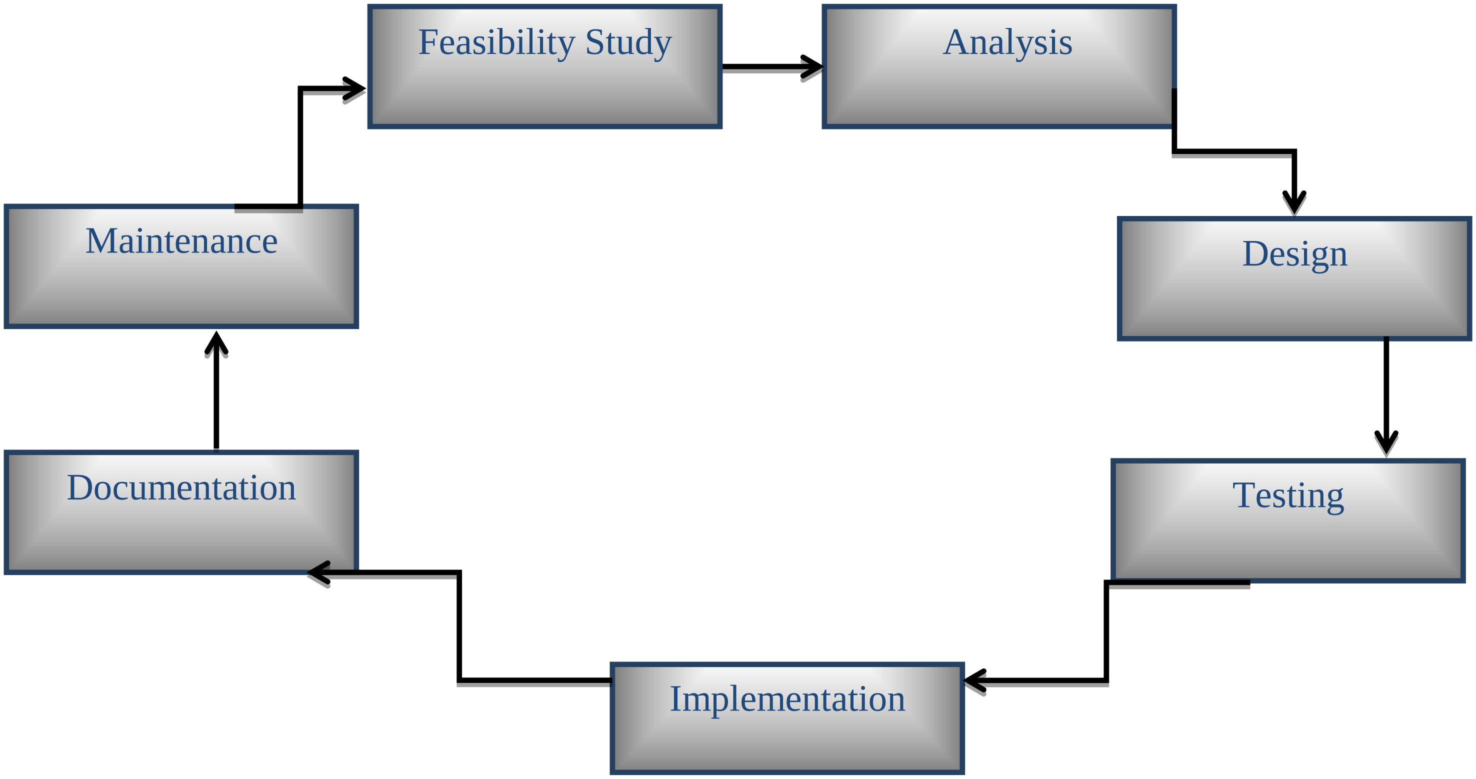


Figure 5.1: Software Development Life Cycle[20]

Block diagram for reframing of Sentences

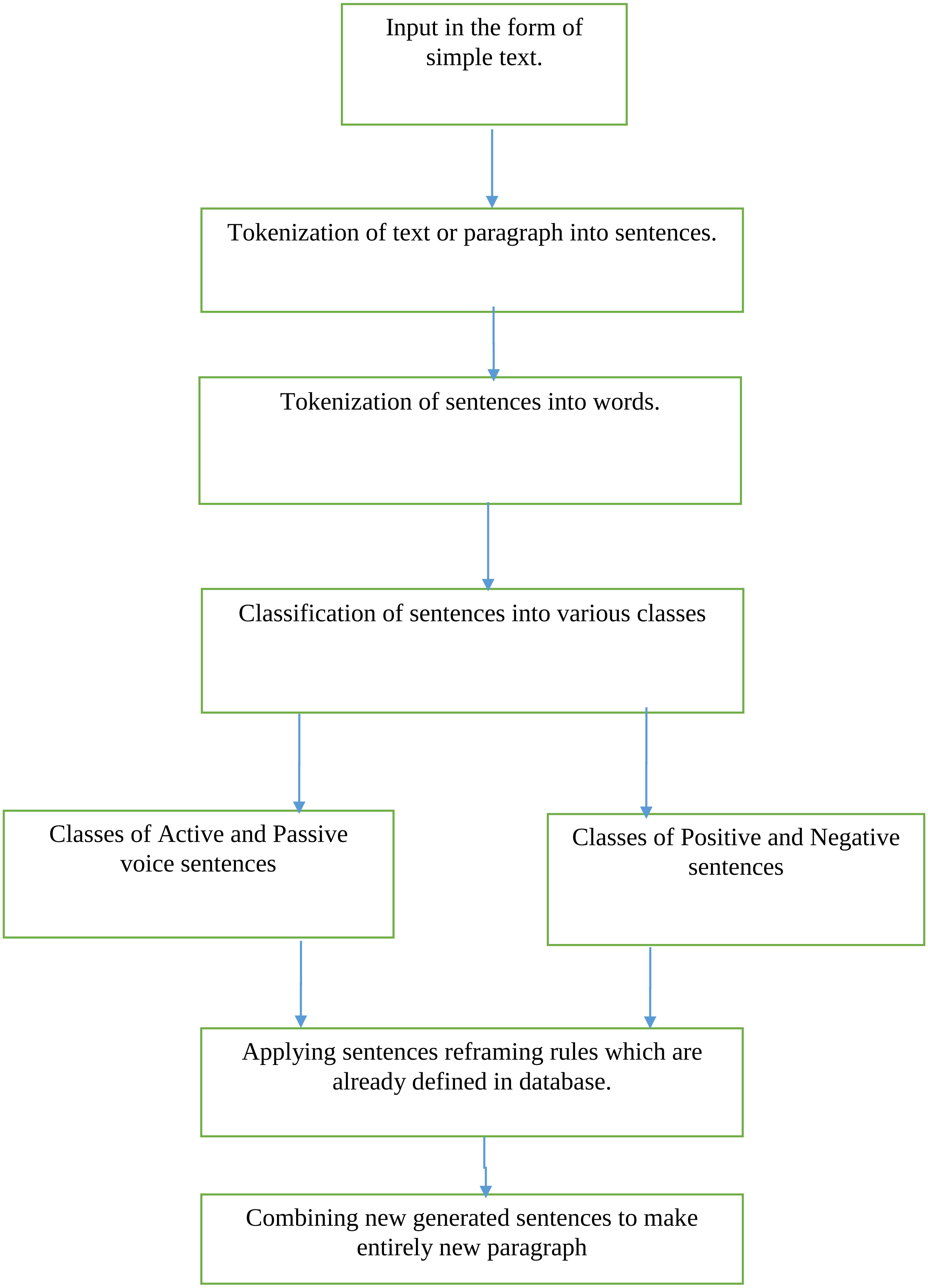

Input in the form of simple text.
Tokenization of text or paragraph into sentences.
Tokenization of sentences into words.
Classification of sentences into various classes
Classes of Active and Passive voice sentences
Classes of Positive and Negative sentences
Applying sentences reframing rules which are already defined in database.
Combining new generated sentences to make entirely new paragraph

Figure 5.2: Block Diagram of sentence Reframing

5.5 General Rules for active to passive voice conversion

Table 5.1: Active and passive voice condition table

Active voice	Passive voice
The starting of the sentence is subject.	Subject is not used.
Verb form is used as per tense.	Always 3rd form of verb is used.
Subjective form of subject is used in starting.	Objective form of subject is used in end.
Helping verb is used according to tense and condition.	Helping verb is used according to tense and condition.

Table 5.2: Pronoun in subjective and objective form

Person	Subjective form	Objective form
1st	I We	Me Us
2nd	You	You
3rd	He She It They	Him Her It Them

5.6 Rules and Pattern in Database

After input of a paragraph is segmented into various lines and POS tagging of each sentence is found. Then these tagging are matched with the specified rules which are already defined in the database. In this proposed system regex pattern matcher is used to match the pattern of a sentence and apply the rules accordingly.

Some of the patterns defined in this system on which pattern matching is done to apply desired case are below:

1. Pattern for Simple Present and Past Tense

(|JJ)(|NNS |NNPS |NNP |NN)(NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB|VBD) (.| DT | JJ)(NNS|NNPS|NNP|NN

Matching case examples

i) Mohan plays cricket.

ii) Mohan played cricket.

2. Pattern for Simple Present and Past Tense (Negative)

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB|VBD) (RB) (VB|VBZ|VBD) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan do not play cricket.

ii) Mohan did not played cricket.

3. Pattern for Simple Present and Past Tense (Interrogative sentences)

(VBZ|VB|VBP|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Do Mohan play cricket?

ii) Did Mohan play cricket?

4. Pattern for Simple Present and Past Tense (Interrogative sentences and negative)

(VBZ|VB|VBP|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBZ|VBP|VB) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Does Mohan not play cricket?

ii) Did Mohan not play cricket?

5. Pattern for Simple Present and Past Tense (Double Interrogative sentences)

(WRB|WDT) (VBZ|VBP|VB|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When does Mohan play cricket?

ii) When did Mohan play cricket?

6. Pattern for Simple Present and Past Tense (Double Interrogative sentences and negative)

(WRB|WDT) (VBZ|VBP|VB|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBZ|VBP|VB) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When does Mohan not play cricket?

ii) When did Mohan not play cricket?

7. Pattern for Present and past continuous tense

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VBD) (VBG) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan is playing cricket.

ii) Mohan was playing cricket.

8. Pattern for Present and past continuous tense (Negative)

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VBD) (RB) (VBG) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan is not playing cricket.

ii) Mohan was not playing cricket.

9. Pattern for Present and past continuous tense (Interrogative)

(VBZ|VBP|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$)

(VBG)(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Is Mohan playing cricket?

ii) Was Mohan playing cricket?

9. Pattern for Present and past continuous tense (Interrogative negative)

(VBZ|VBP|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBG)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Is Mohan not playing cricket?

ii) Was Mohan not playing cricket?

10. Pattern for Present and past continuous tense (Interrogative negative)

(WRB|WDT) (VBZ|VBP|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBG)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When is Mohan playing cricket?

ii) When was Mohan playing cricket?

11. Pattern for Present and past continuous tense (Double Interrogative negative)

(WRB|WDT) (VBZ|VBP|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBG)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When is Mohan not playing cricket?

ii) When was Mohan not playing cricket?

12. Pattern for Present and past perfect tense

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB|VBD) (VBN)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan has played cricket.
ii) Mohan had played cricket.

13. Pattern for Present and past perfect tense (negative)

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB|VBD) (RB) (VBN)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan has not played cricket.
ii) Mohan had not played cricket.

14. Pattern for Present and past perfect tense (Interrogative)

(VBZ|VBP|VB|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBN|VBD)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Has Mohan played cricket?
ii) Had Mohan played cricket?

15. Pattern for Present and past perfect tense (Interrogative negative)

(VBZ|VBP|VB|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBN|VBD)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Has Mohan not played cricket?
ii) Had Mohan not played cricket?

16. Pattern for Present and past perfect tense (Double Interrogative negative)

(WRB|WDT) (VBZ|VBP|VB|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBN|VBD) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When has Mohan played cricket?

ii) When had Mohan played cricket?

17. Pattern for Present and past perfect tense (Double Interrogative negative)

(WRB|WDT) (VBZ|VBP|VB|VBD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBN|VBD) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When has Mohan played cricket?

ii) When had Mohan played cricket?

18. Pattern for Simple future tense

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (MD) (VB|VBZ|VBP) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan will play cricket.

19. Pattern for Simple future tense (negative)

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (MD) (RB) (VB|VBZ) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan will not play cricket.

20. Pattern for Simple future tense (Interrogative)

(MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VB|VBZ|VBP)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Will Mohan play cricket?

21. Pattern for Simple future tense (Interrogative negative)

(MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VB|VBZ|VBP)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Will Mohan not play cricket?

22. Pattern for Simple future tense (Double Interrogative)

(WRB|WDT) (MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When will Mohan play cricket?

23. Pattern for Simple future tense (Double Interrogative)

(WRB|WDT) (MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VBZ|VBP|VB)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When will Mohan not play cricket?

24. Pattern for Future continues tense

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (MD) (VB|VBZ|VBP) (VBG)

(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan will be playing cricket?

25. Pattern for Future continues tense (negative)

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (MD) (RB) (VB|VBZ|VBP) (VBG)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan will be playing cricket?

26. Pattern for Future continues tense (Interrogative)

(MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VB|VBZ|VBP) (VBG)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Will Mohan be playing cricket?

27. Pattern for Future continues tense (Interrogative negative)

(MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VB|VBZ|VBP) (VBG)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Will Mohan be playing cricket?

28. Pattern for Future continues tense (Double Interrogative)

(WRB|WDT) (MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VB|VBZ|VBP) (VBG)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When will Mohan be playing cricket?

29. Pattern for Future continues tense (Double Interrogative negative)

(WRB|WDT) (MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (RB) (VB|VBZ|VBP) (VBG)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When will Mohan not be playing cricket?

30. Pattern for Future perfect tense

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (MD) (VBZ|VBP|VB) (VBN)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan will had played cricket?

31. Pattern for Future perfect tense (negative)

(|JJ)(NNS|NNPS|NNP|NN|PRP|PRP$) (MD) (VBZ|VBP|VB) (RB) (VBN)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Mohan will had not played cricket?

32. Pattern for Future continues tense (Interrogative)

(MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB) (VBN)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Will Mohan had played cricket?

33. Pattern for Future continues tense (Interrogative negative)

(MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB) (RB) (VBN)
(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) Will Mohan had not played cricket?

34. Pattern for Future continues tense (Double Interrogative)

(WRB|WDT) (MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB) (VBN)(.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When will Mohan had not played cricket?

35. Pattern for Future continues tense(Double Interrogative negative)

(WRB|WDT) (MD) (NNS|NNPS|NNP|NN|PRP|PRP$) (VBZ|VBP|VB) (RB) (VBN) (.| DT | JJ)(NNS|NNPS|NNP|NN|PRP|PRP$)

Matching case examples

i) When will Mohan had not played cricket?

The Java regex is used for pattern matcher. When any of the above pattern is matched the corresponding case rules is applied for conversion from active to passive voice.

Table 5.3: Valid tense cases for conversion from active to passive voice

Tense and form	**Type**	**Valid**
Present and Past tense		
S+V+O	Simple	Yes
S+HV+not+V+O	Negative	Yes
HV+S+V+O	Interrogative	Yes
HV+S+not+V+O	Interrogative Negative	Yes
WH+HV+S+V+O	Double Interrogative	Yes
WH+HV+S+not+V+O	Double Interrogative negative	Yes

Present and Past continuous		
S+HV+V+O	Simple	Yes
S+HV+not+V+O	Negative	Yes
HV+S+V+O	Interrogative	Yes
HV+S+not+V+O	Interrogative Negative	Yes
WH+HV+S+V+O	Double Interrogative	Yes
WH+HV+S+not+V+O	Double Interrogative negative	Yes
Present and Past Perfect		
S+V+O	Simple	Yes
S+HV+not+V+O	Negative	Yes
HV+S+V+O	Interrogative	Yes
HV+S+not+V+O	Interrogative Negative	Yes
WH+HV+S+V+O	Double Interrogative	Yes
WH+HV+S+not+V+O	Double Interrogative negative	Yes
Future Tense		
S+HV+V+O	Simple	Yes
S+HV+not+V+O	Negative	Yes
HV+S+V+O	Interrogative	Yes
HV+S+not+V+O	Interrogative Negative	Yes
WH+HV+S+V+O	Double Interrogative	Yes
WH+HV+S+not+V+O	Double Interrogative negative	Yes
Future continuous tense		
S+V+O	Simple	Yes
S+HV+not+V+O	Negative	Yes
HV+S+V+O	Interrogative	Yes
HV+S+not+V+O	Interrogative Negative	Yes
WH+HV+S+V+O	Double Interrogative	Yes
WH+HV+S+not+V+O	Double Interrogative negative	Yes

Future Perfect tense		
S+V+O	Simple	Yes
S+HV+not+V+O	Negative	Yes
HV+S+V+O	Interrogative	Yes
HV+S+not+V+O	Interrogative Negative	Yes
WH+HV+S+V+O	Double Interrogative	Yes
WH+HV+S+not+V+O	Double Interrogative negative	Yes
Imperative sentences	Command	No
	Request	No
Intransitive verb sentences	All Types	No
Sentence with more than one subject	All Types	No
Complex sentences	All Types	No

5.7 Synonyms Replacement

Two or more interrelated words that can be changed in a context are synonyms. In this system user will be provided with a list of options of synonyms to choose from. A user can select any of synonym from given drop down list. The synonyms in the available list would be arranged according to similarity index. The word which are nearest to the matched words are replaced accordingly.

Table 5.4: Synonym Replacement

WORD	SAMPLE SYNONYMS
Good	Beneficial , Goodness, Thoroughly, Well
Heaven	Eden, Paradise , Nirvana
World	Earth , Universe, Reality
Human	Human being , Man
Dull	Dumb , Muffled

5.8 Antonym Replacement

When a word express the meaning opposite to the given word, it is referred as anonym. In this proposed system negative sentences can be converted to positive sentences by using specific antonym.

Table 5.5: Antonym Replacement

WORD	SAMPLE REPLACEMENT
Not good	Bad
Not ugly	Beautiful , Lovely
Not poor	Rich
Not genuine	Counterfeit
Not bad	Good

5.9 Active – Passive voice Conversion

When the subject actively participates in sentences then that sentence is in active voice. These sentences can be converted to passive voice using various grammar rules. In active voice the subject always precedes the object. But in passive voice the object is before the subject. Active voice sentences is used when something is directly spoken , whereas passive voice is used when somebody else refer to that sentence.

Table 5.6: Active Passive conversion

TENSE	SENTENCE	VOICE
Simple Present	Active	Mohan plays cricket.
Simple Present	Passive	Cricket is played by Mohan.
Simple Past	Active	Mohan ate a mango.
Simple Past	Passive	A mango was eaten by Mohan
Present Continuous	Active	John is watching a movie.
Present Continuous	Passive	A movie is being watched by John.

5.10 Proposed algorithm

```
Step 1-INPUT Text directly or through text file.

Step 2- Apply Coreference resolution by replacing pronoun with its individual entity.

Step 3- Apply Text Segmentation.

Step 4- Apply POS tagging to individual sentences

Step 5- Set count=1

Step 6- Repeat Step 7 to 10
        While (count < number_of_sentences )

Step 7- Convert Active to passive voice as per matched pattern from database

Step 8- Replace similar words with Synonym.

Step 9- Replace negative words with Anyms

Step 10- Incement count by 1

Step 11- End
```

5.11 Database Structure

In this proposed system, WordNet 3.0 by Princeton University, is used as dictionary for synonym and antonym extraction using JAW-BIN Java library. Jave library Simplenlp is used in this system to extract different forms of a verb such as base form, past form, past participle form, present participle form. Different verb forms are used while conversion of active voice sentences to passive voice sentences

Table 5.7: Different forms of a verb

Verb Form	Verb
Base form	Eat
Past form	Ate
Past participle form	Eaten
Present participle form	Eating

5.12 POS Tagger

Various grammatical rules are applied after text segmentation and POS tagging of sentences. In this system Stanford POS tagger is used. In this proposed system english-left3words-distsim tagger model is used to tag the words.

Stanford CoreNLP: A Suite of Core NLP Tools

Stanford CoreNLP is a facilitated framework that gives a game plan of customary tongue examination gadgets which can take plain substance information and give the base sorts of words, their parts of talk, whether they are name of the individual, affiliation or any establishment, institutionalize dates, times, and numeric sums, and examination the structure of sentences in regards to articulations and word conditions, that show which thing representations insinuate the same substances, show conclusion, et cetera. It joins various general vernacular contraptions that contain the tokenizer, the linguistic highlight (POS) tagger, the named component recognizer (NER), the parser, the coreference determination system, the inclination examination, and the bootstrapped sample learning instruments. The urgent movement gives model records to the examination of English, yet the engine is great with models for distinctive lingos likewise like models for Chinese and Spanish, and Stanford NLP models for German and Arabic are usable inside CoreNLP. Its target is to make it easy to apply a pack of etymological examination instruments to a touch of substance. Starting from plain substance, just make two lines out of code and run all the gadgets on it. It is expected to be exceptionally versatile and extensible. With a single decision you can change which mechanical assemblies should be enabled and which should be handicapped. Its examinations give the foundational building pieces to bigger sum and range specific substance appreciation applications.

Flow chart for reframing

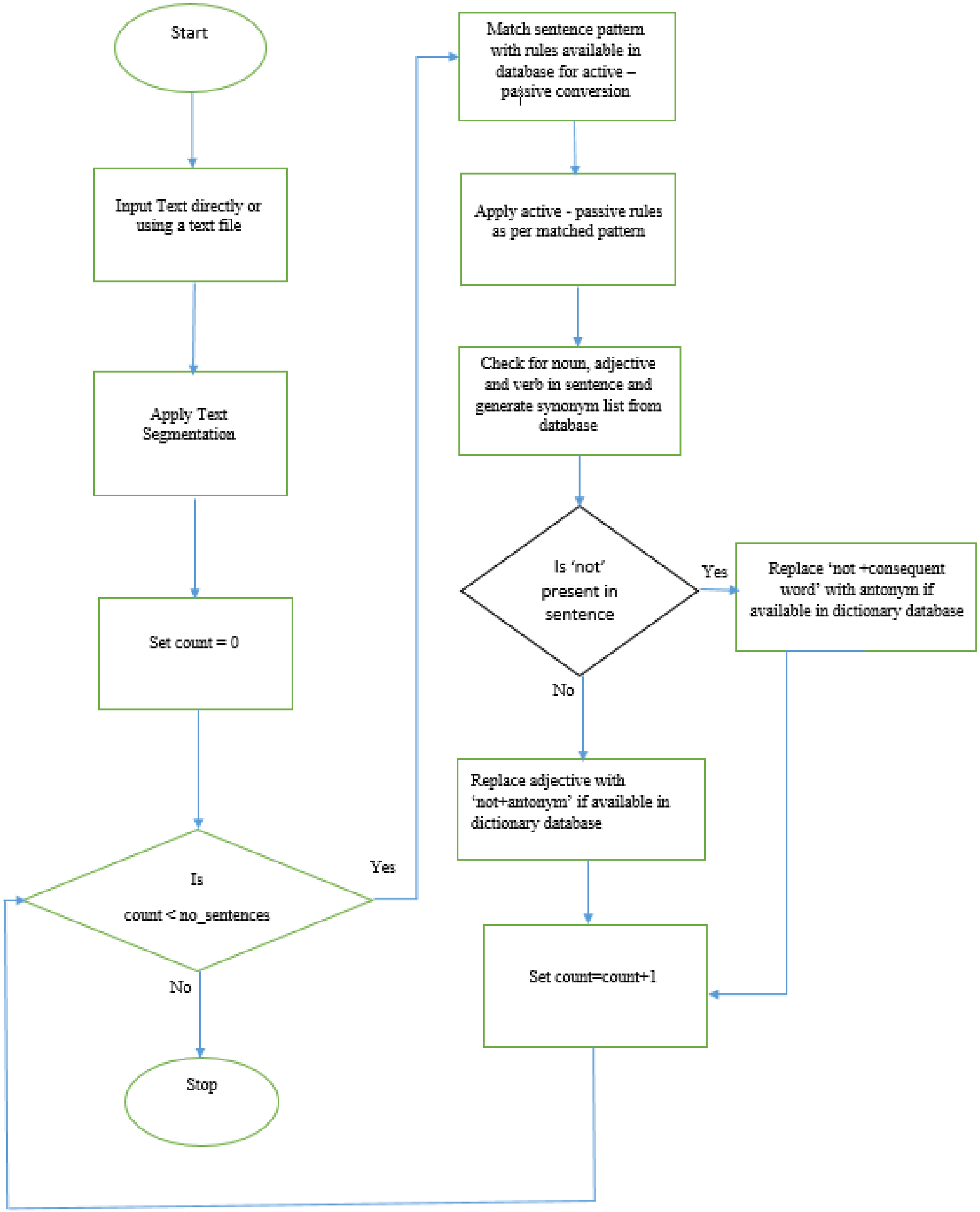
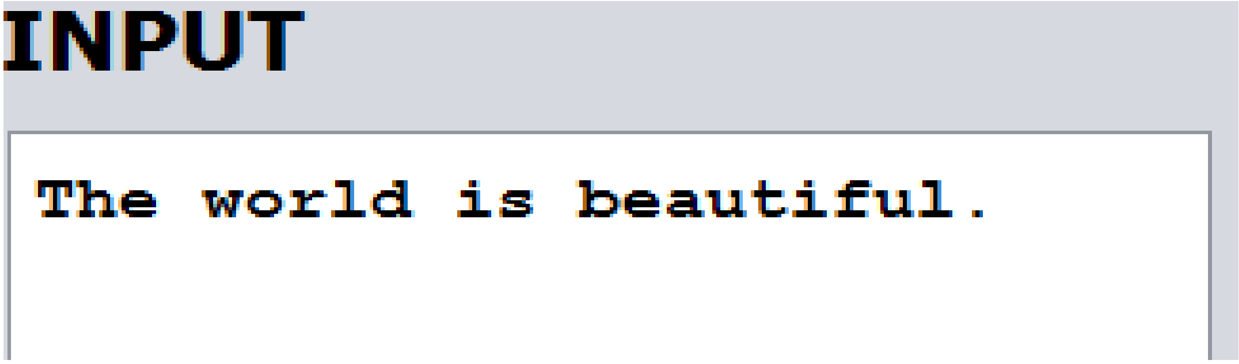

Start
Input Text directly or using a text file
Apply Text Segmentation
Set count = 0
Is count < no_sentences
Yes
No
Stop
Match sentence pattern with rules available in database for active – passive conversion
Apply active - passive rules as per matched pattern
Check for noun, adjective and verb in sentence and generate synonym list from database
Is 'not' present in sentence
Yes
Replace 'not +consequent word' with antonym if available in dictionary database
No
Replace adjective with 'not+antonym' if available in dictionary database
Set count=count+1

Figure 8: Flow chart for reframing

CHAPTER 6
RESULTS AND DISCUSSIONS

This thesis work will generate new sentences based on specified rules and after synonyms and antonyms replacements. The following snapshots shows some results of the present work.

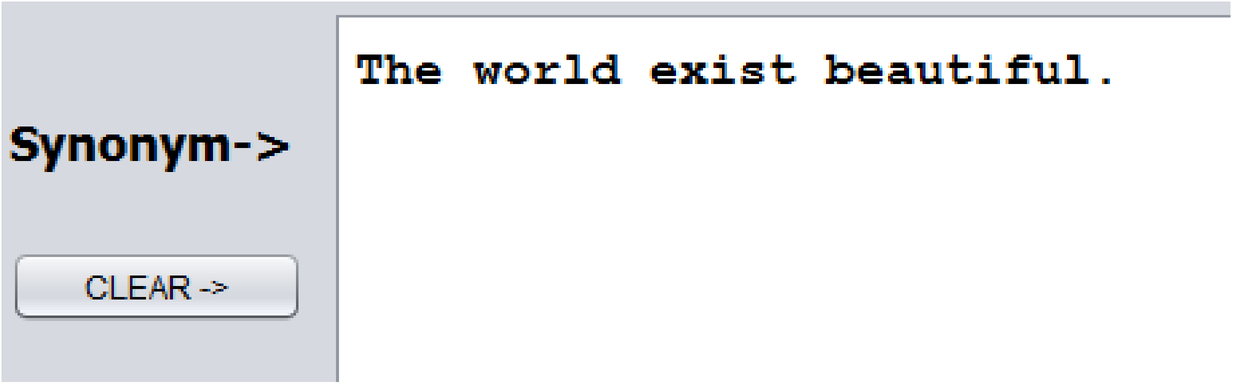
6.1 Sample Input (1) (Synonym and Antonym case)



The world is beautiful.

Figure 6.1: Input for Reframing(1)

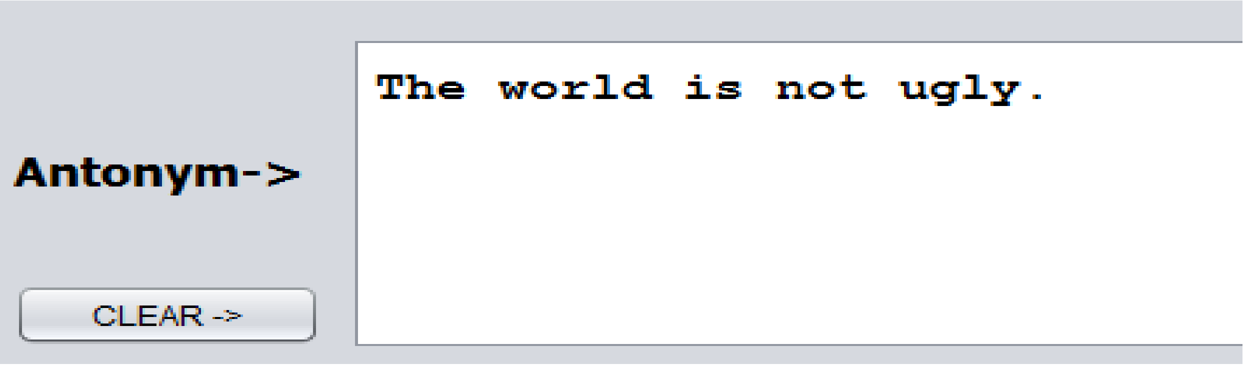
6.1.1 Synonyms Replacement



The world exist beautiful.

Figure 6.2: Synonym replacement(1)

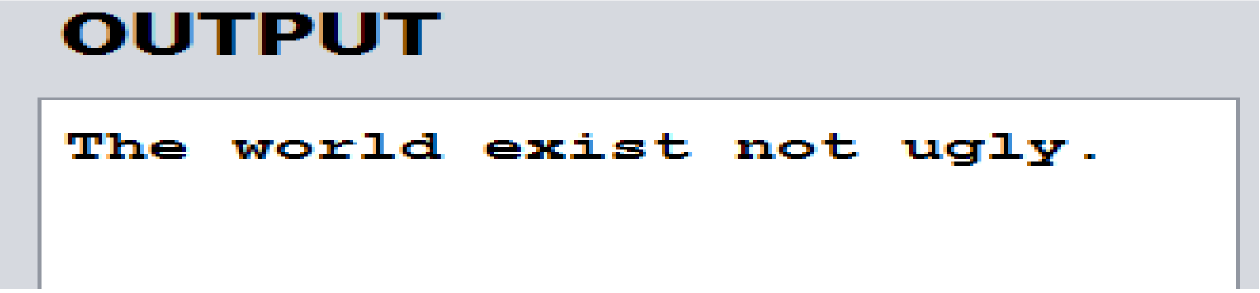
6.1.2 Antonym Replacement



The world is not ugly.

Figure 6.3: Antonym Replacement(2)

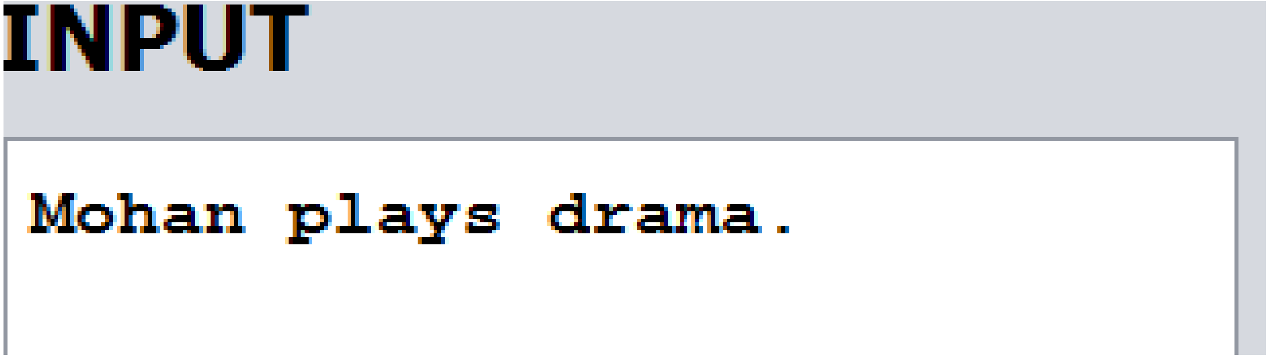
6.1.3 Overall Output



The world exist not ugly.

Figure 6.4: Synonyms replacement and antonym replacement overall result(1)

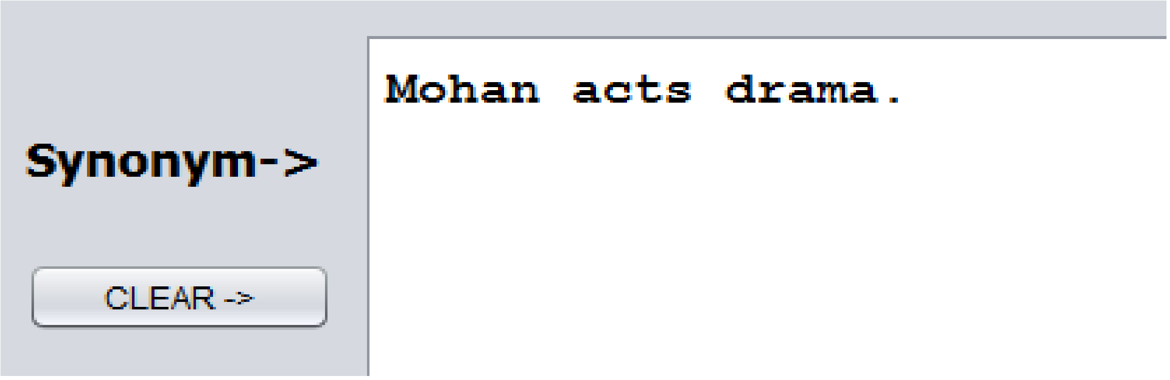
6.2 Sample Input (2) (Synonym and active passive voice case)



Mohan plays drama.

Figure 6.5: Input for Reframing(2)

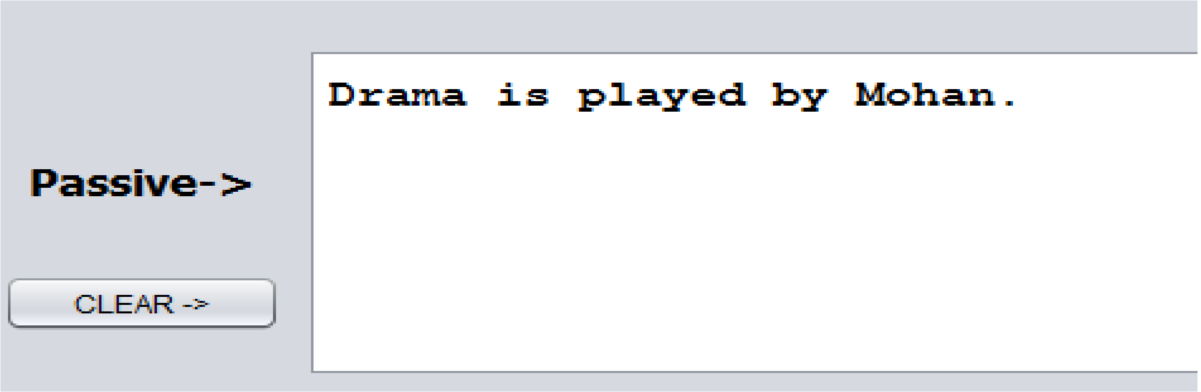
6.2.1 Synonyms Replacement



Mohan acts drama.

Figure 6.6: Synonym replacement(2)

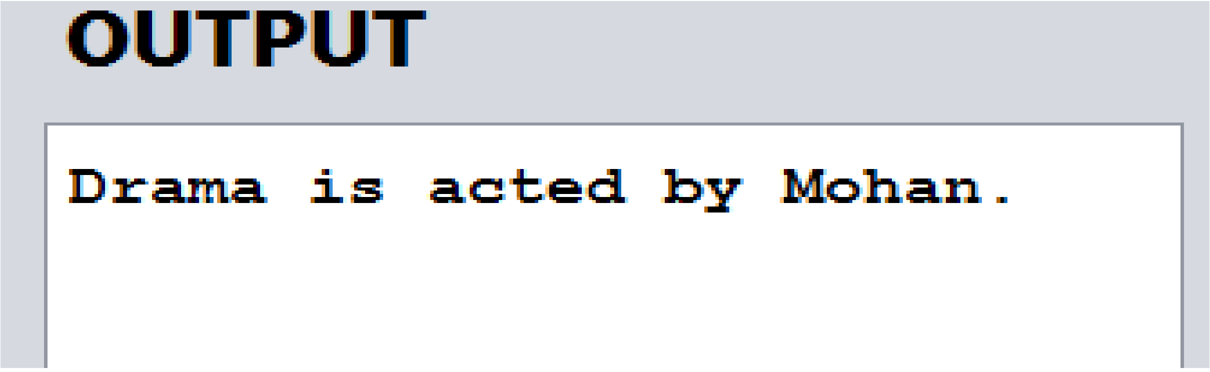
6.2.2 Active to passive voice conversion



Drama is played by Mohan.

Figure 6.7: Antonym Replacement(2)

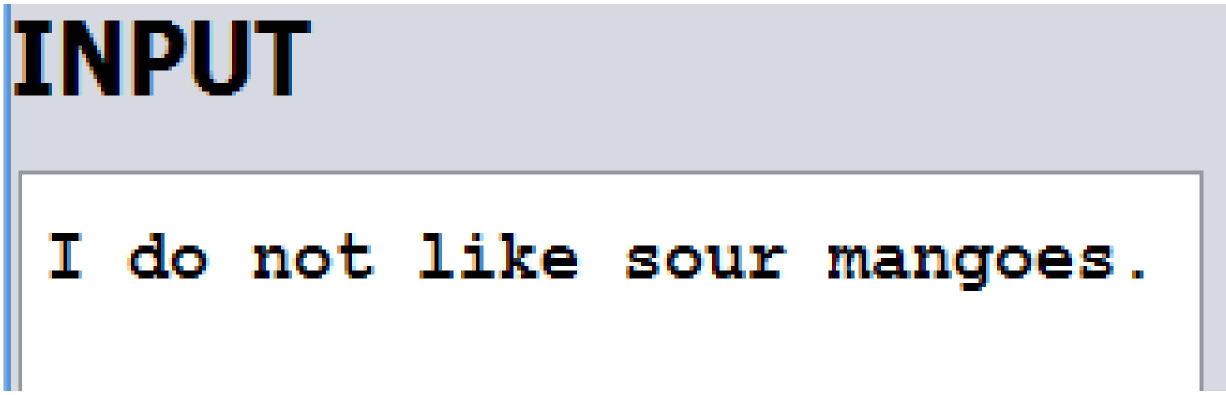
6.2.3 Overall Output



Drama is acted by Mohan.

Figure 6.8: Synonyms replacement and active to passive voice overall result(2)

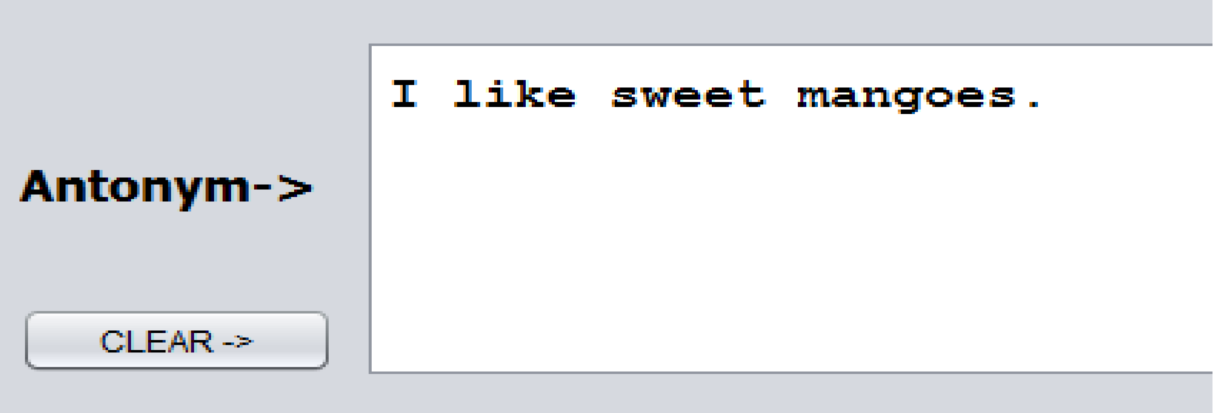
6.3 Sample Input (3) (Antonym and active passive)



I do not like sour mangoes.

Figure 6.9: Input for Reframing(3)

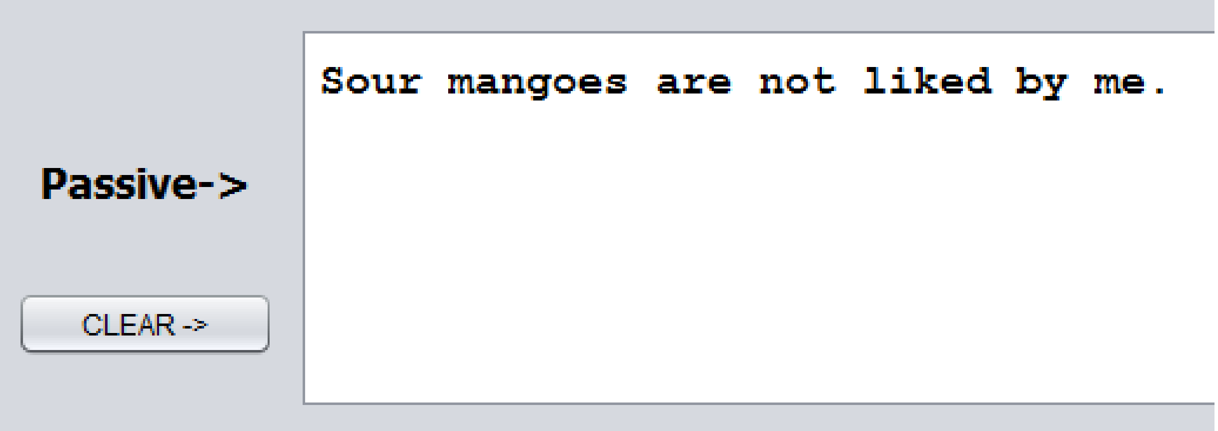
6.3.1 Antonym Replacement



I like sweet mangoes.

Figure 6.10: Antonym replacement(3)

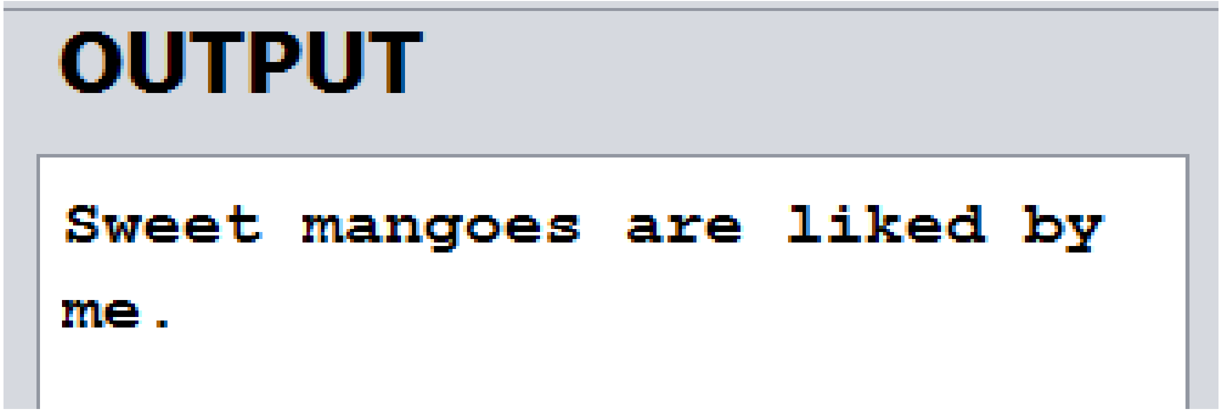
6.3.2 Active to passive voice conversion



Sour mangoes are not liked by me.

Figure 6.11: Active to Passive Voice(3)

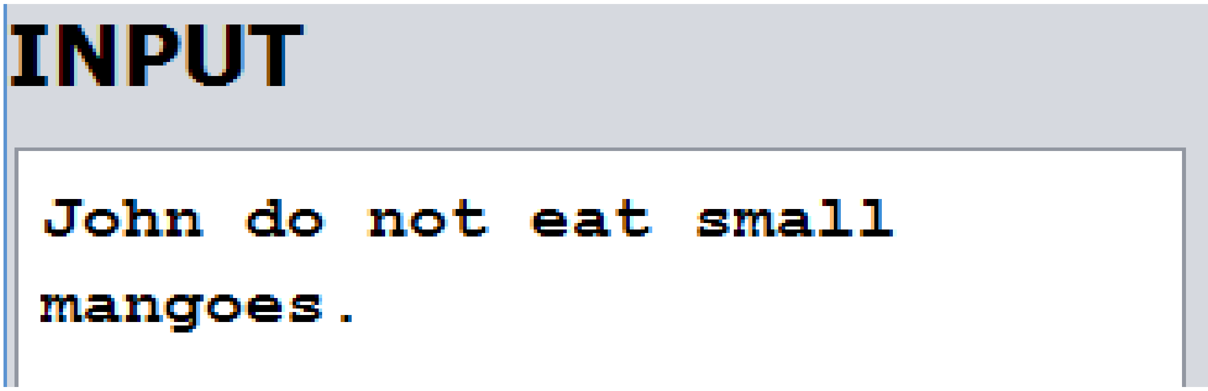
6.2.3 Overall Output



```
Sweet mangoes are liked by
me.
```

Figure 6.12:Antonym replacement and active to passive voice overall result(3)

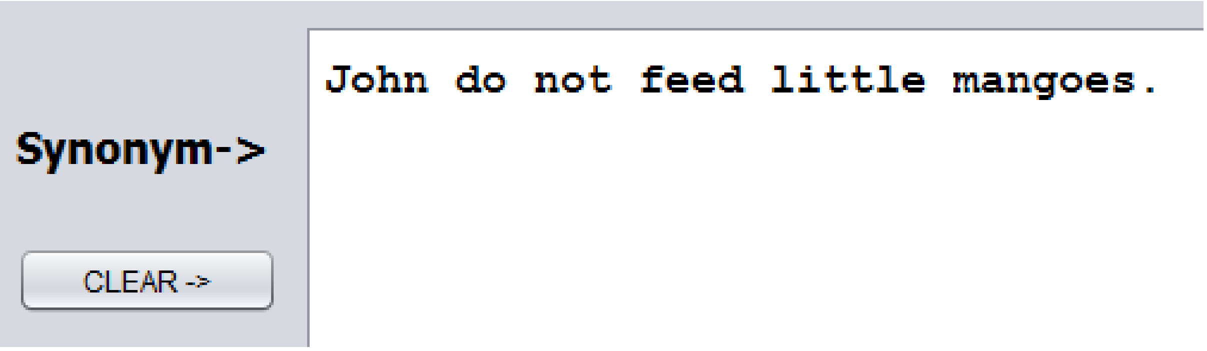
6.4 Sample Input (4) (Synonym , Antonym and active passive)



```
John do not eat small
mangoes.
```

Figure 6.13: Input for Reframing(4)

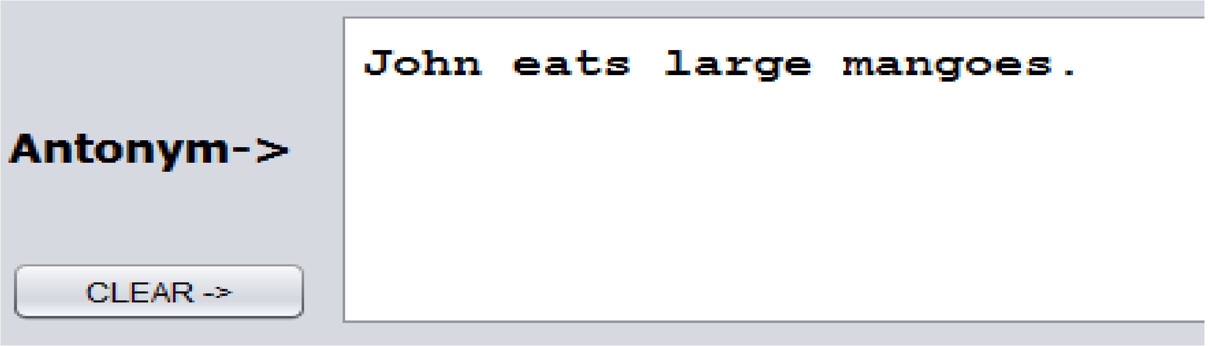
6.4.1 Synonym Replacement



```
John do not feed little mangoes.
```

Figure 6.14: Synonym replacement(4)

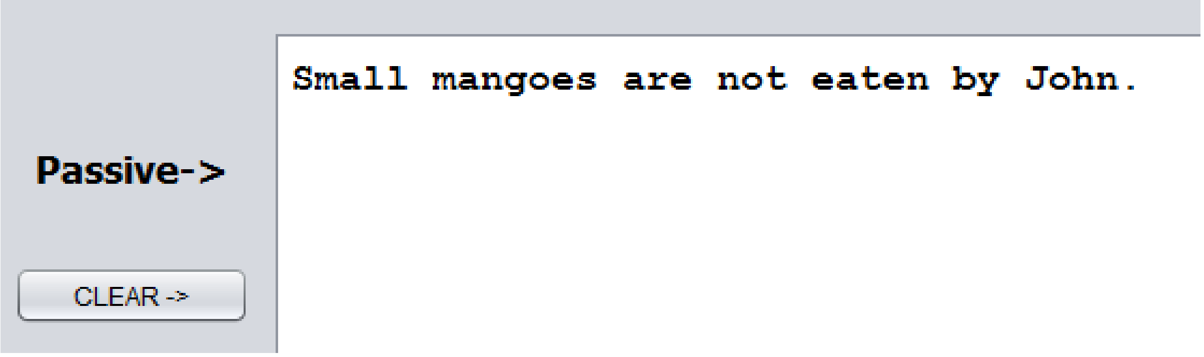
6.4.1 Antonym Replacement



John eats large mangoes.

Figure 6.15: Antonym replacement(4)

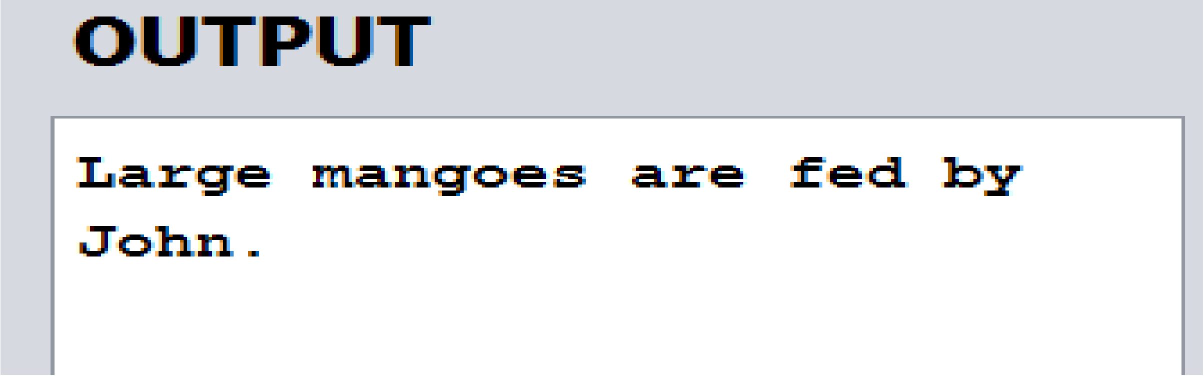
6.3.2 Active to passive voice conversion



Small mangoes are not eaten by John.

Figure 6.16: Active to passive(4)

6.2.3 Overall Output



Large mangoes are fed by John.

Figure 6.17:Synonym , Antonym replacement and active to passive voice overall result(4)

Table 6.1: Result Analysis

Input	Replacement and Reframing	Result
The world is beautiful.	Synonym	The world exist beautiful.
	Antonym	The word is not ugly
	Synonym+Antonym	The world exist not ugly.
Mohan plays drama	Synonym	Mohan acts drama
	Active to passive	Drama is played by Mohan.
	Synonym+Active Passice	Drama is acted by Mohan.
I do not like sour mangoes.	Antonym	I like sweet mangoes.
	Active to passive	Sour mangoes are not liked by me.
	Antonym+Active Passice	Sweet mangoes are liked by me.
John do not eat small mangoes	Synonym	John do not feed little mangoes.
	Antonym	John eats large mangoes.
	Active Passive	Small mangoes are not eaten by John
	Synonym+Antonym+Active Passive	.Large mangoes are eaten by John.

6.5 Accuracy of the system

Since the proposed system is fully dependent of Word web dictionary and Stanford POS tagger so the accuracy of the system the number of words available in the dictionary.

6.6 Efficiency of the system

The efficiency of the system entirely depends upon the number of patterns defined in the rule base system. More rules may be added to the system for wider use.

6.7 Comparison With Existing System

In the existing sentence rephrasing system, such as Ginger software, there is only synonym, idioms and phrases replacement. There is no scenario for antonym and active to passive voice conversion. Also most of the existing system are static and only one form of new paragraph is given as output. But in the proposed system there are different output screens for antonyms , synonyms , active to passive voice and one overall result including the these three things.

CHAPTER 7
CONCLUSION AND FUTURE SCORE

The overall idea of this thesis is to reframe English sentences and form new sentences without changing the actual meaning of the sentence.

7.1 Summary

This thesis will provides the information about different components of a sentence and how a sentence can be reframed. It also provides the details about sentence classification, sentence tokenization, parts of speech tagging of a sentence , word stemming and discourse integration of a sentence. This report also provides details about how Natural Language Processing toolkit can make a machine understand different sentences and can interact with humans. This report also discuss the idea to implement sentence reframing concept. In this report a brief working of Google translator is also discussed also with certain summarization techniques of web documents and language identification using fuzzy system.

7.2 Conclusion and Future Scope

In this proposed system we have discussed sentence reframing technique using NLP. This system can be used by scholars, technical writers and researchers. This system can be further extended to develop a software for semantic analysis for information extraction and other. This system can be helpful in making a robot understand different forms of sentences. It can also be used to develop an intelligent system that can take decisions like humans. This work can be extended to make a decision support system that will work is a similar manner like humans and can respond just like humans by understanding the different forms of sentences given to it as an input.

CHAPTER 8
LIST OF REFERENCES

REFERENCES

I. Books

[1]. D.W.Patterson, Introduction to AI & Expert Systems, Prentice Hall.

[2]. Mark Lutz (2009, September), Learning Python, O’Reilly Media , 4th edition.

[3]. Mark Lutz (2009, September), Python Poket Reference, O’Reilly Media , 5th edition.

[4]. Perkins Jacob (2010, November), Python Text Processing with NLTK 2.0 Cookbook, Packt Publishing, 1st edition.

[5]. Rich ,Knight,” Artificial Intelligence”, Tata McGraw Hill, 2009(Third edition)

[6]. Steven Bird, Ewan Klein, and Edward Loper (2009 , July) Natural Language Processing with Python, O’Reilly Media , 1st edition.

II. Research Papers

[7]. Maria Soledad Pera and Yiu-Kai Ng, “Classifying sentence based summaries of web documents”, 21st IEEE International Conference on Tools with Artificial Intelligence, 2009.

[8]. Mishra, Girish, Sohan Lal Nitharwal, and Sarvjeet Kaur. "Language identification using fuzzy-SVM technique." In Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on. IEEE, 2010 , pp. 1-5

[9]. Minnen, Guido, John Carroll, and Darren Pearce. "Applied morphological processing of English." Natural Language Engineering 7, no. 03 (2001), pp. 207-223.

[10]. Marcel Bollman (2011). Adapting SimpleNLG to German. To appear in Proceedings of the 13th European Workshop on Natural Language Generation (ENLG-11)

[11]. Rekha S. Sugandhi, Ritika Shekhar, Tarun Agarwal, Rajneesh K. Bedi, Vijay M. Wadhai, “Information and Communication Technologies (WICT)”, 2011 , 754 - 759.

III. Websites

[12]. Java basics
www.tutorialspoint.com/java/

[13]. Natural Language Processing.
http://www.mind.ilstu.edu/curriculum/protothinker/natural_language_processing.php

[14]. NLP Application
http://www.dmi.unict.it/~faro/tesi.php?req=02

[15]. Sentence definition,
https://www.englishclub.com/grammar/what-is-a-sentence.htm

[16]. Subject Verb Agreement
http://grammar.ccc.commnet.edu/grammar/sv_agr.htm

[17]. Standford POS tagger
http://nlp.stanford.edu/wiki/Main_Page

[18]. Term frequency–inverse document frequency(tf-idf) ,
http//nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html

[19]. Working of Google Translator,
http://translate.google.co.in/about/intl/en_ALL/
http://www.themarysue.com/how-does-google-translate-work/

[20]. Various stages of Software development
http://www.veracode.com/security/software-development-lifecycle

[21]. Wordnet API for Java
http://projects.csail.mit.edu/jwi/

APPENDIX

List of Abbreviations:

AI: Artificial Intelligence

API: Application Programming Interface

HV : Helping Verb

NLP: Natural Language Processing

O : Object

POS: Part of Speech

S: Subject

V : Verb

WH : Which , When , What , Why