"Language Enhancement using Adaptive Machine Learning"

A Major Project submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

in partial fulfillment of the requirements for the award of

degree of

Bachelor of Engineering

By

Kalash Jain

Under the guidance of

Prof. Sunil Kumar Kushwaha

Department of Computer Science & Engineering

Chameli Devi Group of Institutions, Indore
452 020 (India)

2015-16

DECLARATION

I certify that the work contained in this report is original and has been done by me under the guidance of my supervisor(s).

a. The work has not been submitted to any other Institute for any degree or diploma.
b. I have followed the guidelines provided by the Institute in preparing the report.
c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
d. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signature of the Student (s)

CERTIFICATE

Certified that the project report entitled, "**Language Enhancement using Adaptive Machine Learning**" is a bonafide work done under my guidance by Kalash Jain in partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in Computer Science & Engineering.

Date: _____

(Mr. Sunil Kumar Kushwaha)

Guide

_____ _____

(Mr. Surendra Shukla) (Dr. C.N.S. Murthy)

Head of the Department Dean

ACKNOWLEDGEMENT

I have immense pleasure in expressing my sincerest and deepest sense of gratitude towards my Guide Mr. Sunil Kumar Kushwaha (Associate Professor) for the assistance in preparing and presenting the Major Project. I also take this opportunity to thank Major Project Coordinator and Head of the department, for providing the required facilities in completing my Major Project report.

I am greatly thankful to my Parents, Friends and Faculty members for their motivation, guidance and help whenever needed.

Kalash Jain

0832CS121048

TABLE OF CONTENTS

List of Figures

CHAPTER- 1

Abstract

The task to write an impressive piece of writing with a meaningful matter embellished with good apt vocabulary is a devil of a job for people who are not seasoned enough in using the language. For them it becomes tough to make their thoughts rich in words. This paper focuses on deploying a system that will solve this problem of improving the quality of words in a sentence or paragraph and help people with less knowledge of synonyms and grammar. The paper also focuses on a brief review of various Machine Learning Algorithms and methods to be used in this project.

The aim is to use Machine Learning algorithms and design a system that will take normal sentences as input and by using the methods to break the sentences and tag them according to parts of speech and then running algorithms to search for words with similar meaning in the unstructured database, finally ranking the possible top options and at last giving the choices to the user to select the best fit answer. On the basis of choice selection by user, the system will be building its experience.

Several times it has been observed that the problem has been faced that people with ordinary knowledge of any language suffers the challenge of writing rich articles and thus it becomes tough for them to cope up with the language and spend hours of time searching words in dictionary for the correct meaning and apt vocabulary. In order to solve this issue, I have designed a system in which the input will be the normal trivial sentence and the output that will be generated, is going to be more complex and verbally stronger sentences.

The initial implementation of the system deals with just replacing the adjectives in the sentences and after that the system will add more features, coming on big data and unstructured information, and adopting machine learning key concepts, making the system more robust and efficient.

CHAPTER- 2

Introduction

2.1 Overview

“Language Enhancement using Adaptive Machine Learning” will be a software program, taking normal sentences as input and generating corresponding sentences rich in vocabulary and sentence structure, thus ameliorating the quality of language without changing the contextual meaning. The software will be dealing with database and will adopt learning algorithms to improve the accuracy of results generated.

2.2 Problems Solved

The software aims at giving a solution for the problem of undecorated and quite plain language by modifying it to a more verbally sound and attractive language without imparting any change in the meaning. When writing any essay or journal or novel, at times issue of poor language hinders the overall impression of the text. Thus, this software is designed to serve this purpose to the most satisfactory level with 80-85% accuracy.

2.3 Proposed Idea

The software will be designed in multiple stages, starting with the loading of datasets, with adequate information of words and sentences which will be including dictionaries, encyclopaedia links, other textual data through books, papers or interviews and when any sentence will be given, the meaning of the sentence will be matched with the stored sentences and their meanings, and then ranking will be done, suggesting the most appropriate sentence or word replacement for the text given. The output learning will be saved for future references and learning of the software.

CHAPTER- 3

Machine Learning Methods

MACHINE LEARNING METHODS:

3.1 ARTIFICIAL NEURAL NETWORKS (ANN)

In machine learning, artificial neural networks (ANNs) are a family of statistical learning algorithms inspired by biological neural networks. A neural network is a network of simulated neurons that can be used to recognize instances of patterns. Neural networks learn by searching through a space of network weights. It is used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which can compute values from inputs/output and are capable of machine learning as well as pattern recognition.

3.2 K-NEAREST NEIGHBOURS CLASSIFIER (K-NN)

Nearest neighbour classification are based on learning analogy i.e., by comparing given test tuple with training tuples that are similar. Each tuple represent a point in an n-dimensional space. Any training tuples are stored in an n-dimensional pattern space. It is a tuple-based classifier that can simply locate the nearest neighbour in tuple space and labelling the unknown tuple with the same class label as that of the known neighbour. The k-nearest neighbour classifier searches the pattern space for the k-training tuple that are closest to the unknown tuple. These training tuples are knearest neighbour classifier of the unknown tuple. Closeness can be defined as any distance metric such as Euclidean distance. Nearest neighbour classifiers are distance based comparisons intrinsically assign equal weight to each attribute. Therefore, they can suffer from poor accuracy if there is noisy or irrelevant attribute.

3.3 SUPPORT VECTOR MACHINES (SVM)

Support Vector Machine is a new generation learning system based on recent advances in statistical learning theory. It is an algorithm for both linear and non-linear data. It transforms the original data in a higher dimension, from where it can find a hyper plane for separation of the data using essential training tuples called support vectors. A Support Vector Machine is a discriminative classifier formally defined by a separating hyper plane. In other words, given labelled training Support vector machine constructs a hyper plane or set of hyper planes in a high or infinite-dimensional space, which can be used for classification, regression, or other

tasks. Intuitively, a good separation is achieved by the hyper plane that has largest distance to the nearest training data point of any class so called functional margin, since in general the larger the margin the lower the generalization error of the classifier.

3.4 NAÏVE BAYESIAN CLASSIFIER

Naïve Bayesian classification is called naïve because it assumes class condition independence. That is, the effect of an attribute value of given class is independence of the values of the other attributes. This assumption is made to reduce computational costs, and hence is considered naïve. A Naïve Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. The major idea behind naïve Bayesian classification is to try and classify data by maximizing

3.5 RANDOM FOREST

Random forest Leo Breiman (2001) is an ensemble of decision trees based classifiers. Each tree is constructed by a bootstrap sample from the data, and it uses a candidate set of features selected from a random set. It uses both bagging and random variable selection for tree building. Once the forest is formed, test instances are percolated down each tree and trees make their respective class prediction. The error rate of a random forest depends on the strength of each tree and correlation between any two trees. It can be used to rank the importance of variables in a regression or classification problem in a natural way.

3.6 BAGGING

Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid over fitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach. Main reason for error in learning is due to noise, bias and variance. Noise is error by the target function, Bias is where the algorithm cannot learn the target and Variance comes from the sampling, and how it affects the learning algorithm. Bagging minimizes these errors. Averaging over bootstrap samples can reduce error from variance especially in case of unstable classifiers.

3.7 BOOSTING

Boosting is a machine learning ensemble meta-algorithm for reducing bias primarily and also variance in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones. Boosting is based on the question posed by Kearns and Valiant (1988, 1989): Can a set of weak learners create a single strong learner? A weak learner is defined to be a classifier which is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

CHAPTER- 4

System Analysis

Proposed System:

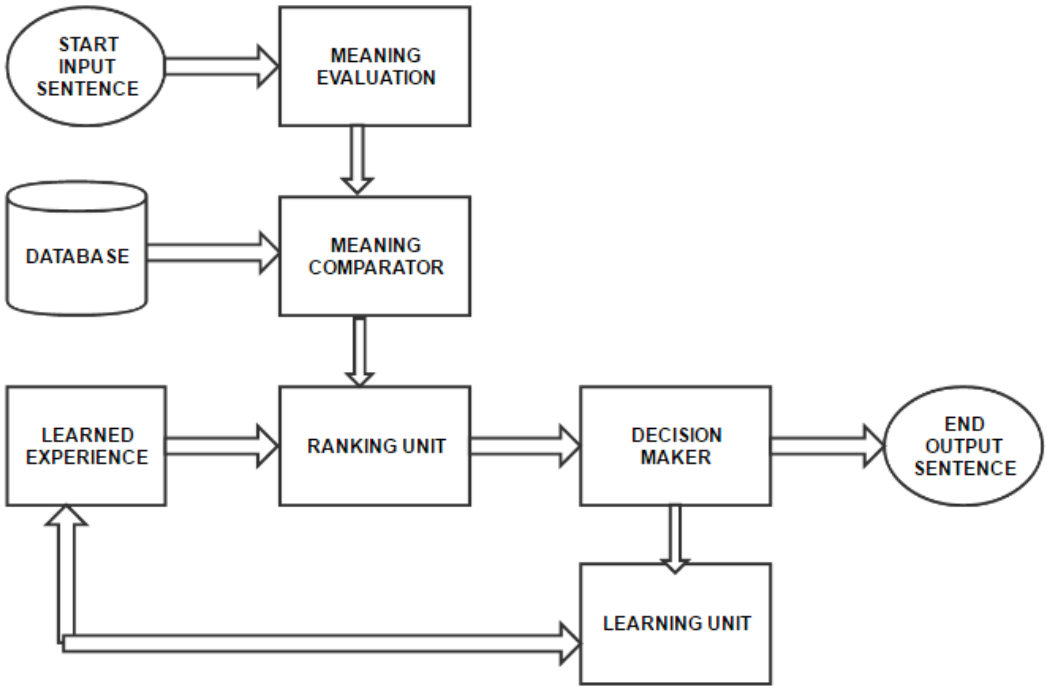

Fig 4.1 Proposed implementation of LEUAMC

- First of all the input message is processed for getting the meaning.
- The meaning derived is then compared with the database of information.
- On the basis of information along with the experience gained by the software ranking is done, generating some k number of answer nominations, and some percentage weight is given to them, depending on their relevance with the input sentence.
- Then the Decision Maker software on the basis of ranking decides the most apt tantamount sentence for the text input.
- On the basis of user feedback about the accuracy of the output, the experience is made in favour or oppose, which will assist the software to further take decisions for similar kinds of problems.

Present System:

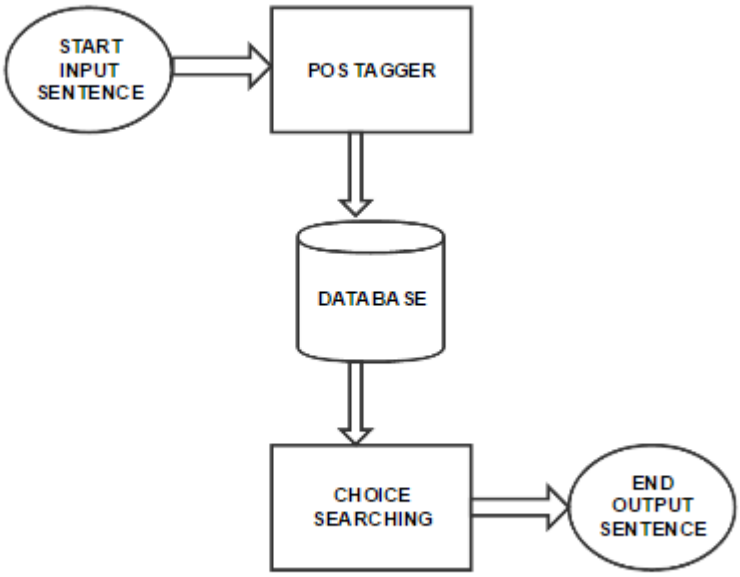

Fig 4.2 Present implementation of LEUAMC

- First of all the input message is processed for getting the meaning.
- Then the sentence is tagged using the Stanford POS tagger.
- The adjectives are extracted from the sentence and then their meaning is compared with the database of words.
- Then the most apt meanings are chosen from the database are chosen and replaced in the input sentence.
- Choices are given to the user to select from and then he picks his best suitable choice.
- On the basis of user feedback about the accuracy of the output, the experience is made in favour or oppose, which will assist the software to further take decisions for similar kinds of problems

CHAPTER- 5

Software Requirement & Specification

Software Required:

The project is implemented in Core Java , hence the software's required in the creation and execution of the project are j2sdk1.7 or Eclipse .As we know JAVA is a platform independent language so this software runs with JRE environment on any desired platform i.e. Linux ,windows 9x, XP, or 2000 or any operating system.

Hardware Required:

Any System with Pentium P2 or above processor, 32MB RAM, 1GB Hard Disk, a LAN Card, and a CDROM is sufficient. Its network based software so computers connected with any kind of mode (wireless, LAN connected etc) will suit its requirements. . . . It can also be run on a single machine for its demo use.

Best suited in laboratory where we can run its server on any machine and many clients can use it simultaneously.

Software Analysis Report

About java: Features JDK 1.7

Platform Independent:

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is closer to this feature. The programs written on one platform can run on any platform provided the platform must have the JVM.

Simple:

There are various features that make the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we can not say about the other programming languages. Java provides the bug free system due to the strong memory management. It also has the automatic memory allocation and de-allocation system.

Object Oriented:

To be an Object Oriented language, any language must follow at least the four characteristics.

- Inheritance : It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.
- Encapsulation: It is the mechanism of combining the information and providing the abstraction.
- Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
- Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

As the languages like Objective C, C++ fulfills the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. But in case of java, it is a fully Object Oriented language because object is at the outer most level of data structure in java. No stand alone methods, constants, and variables are there in java. Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.

Robust:

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other

programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features makes the java language robust.

Distributed:

The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

Portable:

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also have the standard data size irrespective of operating system or the processor. These features make the java as a portable language.

Dynamic:

While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

Secure:

Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The byte code Verifier checks the classes after loading.

Performance:

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

Multithreaded:

Java is also a multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer. Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs. This is how Multithreading works in Java which you will soon come to know in details in later chapters.

Interpreted:

we all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code. The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program. The versatility of being **platform independent** makes Java to outshine from other languages. The source code to be written and distributed is platform independent. Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

Architecture Neutral:

The term architectural neutral seems to be weird, but yes Java is an architectural neutral language as well. The growing popularity of networks makes developers think distributed. In the world of network it is essential that the applications must be able to migrate easily to different computer systems. Not only to computer systems but to a wide variety of hardware

architecture and operating system architectures as well. The Java compiler does this by generating byte code instructions, to be easily interpreted on any machine and to be easily translated into native machine code on the fly. The compiler generates an architecture-neutral object file format to enable a Java application to execute anywhere on the network and then the compiled code is executed on many processors, given the presence of the Java runtime

system. Hence Java was designed to support applications on network. This feature of Java has thrived the programming language.

ABOUT : JDK:

The **Java Development Kit** (**JDK**) is a Sun Microsystems product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java SDK. On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007[1] and the source code was contributed to the OpenJDK.

The primary components of the JDK are a selection of programming tools, including:

- Java – The loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, is no longer provided with Sun JDK.
- javac – The compiler, which converts source code into Java bytecode
- jar – The archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.
- javadoc – The documentation generator, which automatically generates documentation from source code comments
- jdb – The debugger
- javap – The class file disassembler
- appletviewer – This tool can be used to run and debug Java applets without a web browser.
- javah – The C header and stub generator, used to write native methods
- extcheck – This utility can detect JAR-file conflicts.
- apt – The annotation processing tool
- jhat – (Experimental) Java heap analysis tool
- jstack – (Experimental) This utility prints Java stack traces of Java threads.
- jstat – (Experimental) Java Virtual Machine statistics monitoring tool
- jinfo – (Experimental) This utility gets configuration information from a running Java process or crash dump.

- jmap – (Experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
- idlj – The IDL-to-Java compiler. This utility generates Java bindings from a given IDL file.
- policy tool – The policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources
- VisualVM – visual tool integrating several command line JDK tools and lightweight performance and memory profiling capabilities

The JDK also comes with a complete Java Runtime Environment, usually called a *private* runtime. It consists of a Java Virtual Machine and all of the class libraries that will be present in the production environment, as well as additional libraries only useful to developers, such as the internationalization libraries and the IDL libraries.

Also included are a wide selection of example programs demonstrating the use of almost all portions of the Java API.

Technologies and Requriments

IDE:

My Eclipse

Front End:

JSP, JDBC, Javascript, AJAX

Programming Language:

JAVA

Back End:

MySQL

CHAPTER- 6

Source Code

```
import java.io.IOException;
import edu.stanford.nlp.tagger.maxent.MaxentTagger;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;

public class Kalash
{
	public static void main(String[] args) throws IOException,ClassNotFoundException
	{
		MaxentTagger tagger = new MaxentTagger("taggers/english-left3words-distsim.tagger");
		Scanner input = new Scanner(System.in);
		System.out.println("Write the input line");
		String sample= input.nextLine();
		String tagged = tagger.tagString(sample);

		int count=0;
		System.out.println("Input: " + sample);
```

```
System.out.println("Output: "+ tagged);

String search;

String token;

StringTokenizer tokenizer = new StringTokenizer(tagged);

while (tokenizer.hasMoreTokens())

{

token = tokenizer.nextToken();

//System.out.println("==Token== : "+token);

char []str= token.toCharArray();

for(int i=0;i<str.length;i++)

{

if(str[i]=='_')

{

i++;

if(str[i]=='J')

{

i++;

if(str[i]=='J')

{

search=token.substring(0,token.indexOf('_'));

//System.out.println(search);

Connection con=null;
```

```
Statement st=null;
ResultSet rs=null;
try
{
Class.forName("com.mysql.jdbc.Driver");
//System.out.println("Driver Loaded");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/tagger","root","kalash");
//System.out.println("connected");
st=con.createStatement();
String query="select * from table1";
rs=st.executeQuery(query);
while(rs.next())
{
String word=rs.getString(1);
if(word.equalsIgnoreCase(search))
{
count=1;
String f=rs.getString(2);
String out2[]=f.split(", ");
System.out.println("The Answer Choices are:");
for(int z=0;z<out2.length;z++)
{
```

```
String out1=sample.replaceAll(search, out2[z]);
System.out.println((z+1)+") "+out1);
}
}
}
if(count==0)
System.out.println("No match found \nOver!");
else
System.out.println("Over!");
}
catch(ClassNotFoundException e)
{
System.out.print(e);
}
catch(SQLException e)
{
System.out.print(e);
}
}
}
}
}
```

```
                }
            }
        }
```

CHAPTER- 7

Output

CASE:1

Input : Rajesh always used to search for perfect articles.

Output : Rajesh_NNP always_RB used_VBD to_TO search_VB for_IN perfect_JJ articles_NNS ._.

The Answer Choices are:

1) Rajesh always used to search for consummate articles.

2) Rajesh always used to search for faultless articles.

3) Rajesh always used to search for flawless articles.

4) Rajesh always used to search for impeccable articles.

Over!

CASE:2

Input : The intelligent students don’t like to spend time with the lazy ones.

Output : The_DT intelligent_JJ students_NNS do_VBP n't_RB like_VB to_TO spend_VB time_NN with_IN the_DT lazy_JJ ones_NNS ._.

The Answer Choices are:

1) The bright students don’t like to spend time with the lazy ones.

2) The brilliant students don’t like to spend time with the lazy ones.

3) The knowing students don’t like to spend time with the lazy ones.

4) The quick-witted students don’t like to spend time with the lazy ones.

5) The smart students don’t like to spend time with the lazy ones.

6) The intellectual students don’t like to spend time with the lazy ones.

Over!

The Answer Choices are:

1) The intelligent students don’t like to spend time with the faineant ones.

2) The intelligent students don’t like to spend time with the idle ones.

3) The intelligent students don’t like to spend time with the indolent ones.

4) The intelligent students don’t like to spend time with the slothful ones.

Over!

CASE:3

Input : Ignorant attitude of people towards hygiene is leading to cause many diseases.

Output : Ignorant_JJ attitude_NN of_IN people_NNS towards_IN hygiene_NN is_VBZ leading_VBG to_TO cause_VB many_JJ diseases_NNS ._.

The Answer Choices are:

1) uneducated attitude of people towards hygiene is leading to cause many diseases.

2) untaught attitude of people towards hygiene is leading to cause many diseases.

3) unlearned attitude of people towards hygiene is leading to cause many diseases.

4) untutored attitude of people towards hygiene is leading to cause many diseases.

5) unlettered attitude of people towards hygiene is leading to cause many diseases.

6) illiterate attitude of people towards hygiene is leading to cause many diseases.

Over!

Over!

CASE:4

Input : At gym all the strong boys lift heavy weights.

Output : At_IN gym_NN all_PDT the_DT strong_JJ boys_NNS lift_VBP heavy_JJ weights_NNS ._.

The Answer Choices are:

1) At gym all the stout boys lift heavy weights.

2) At gym all the sturdy boys lift heavy weights.

3) At gym all the tough boys lift heavy weights.

4) At gym all the stalwart boys lift heavy weights.

5) At gym all the tenacious boys lift heavy weights.

Over!

The Answer Choices are:

1) At gym all the strong boys lift weighty weights.

2) At gym all the strong boys lift hefty weights.

3) At gym all the strong boys lift massive weights.

4) At gym all the strong boys lift ponderous weights.

5) At gym all the strong boys lift cumbersome weights.

Over!

CASE:5

Input : Young kids are very active in daily tasks.

Output : Young_NNP kids_NNS are_VBP very_RB active_JJ in_IN daily_JJ tasks_NNS ._.

The Answer Choices are:

1) Young kids are very energetic in daily tasks.

2) Young kids are very dynamic in daily tasks.

3) Young kids are very vigorous in daily tasks.

4) Young kids are very lively in daily tasks.

Over!

Over!

CHAPTER- 8

Bibliography

- **For Java installation**
    - https://www.java.com/en/download/
- **For Oracle DataBase installation**
    - http://www.oracle.com/index.html
- **Reference websites**
    - www.javatpoint.com
    - www.w3schools.com
    - http://www.tutorialspoint.com/java/index.htm
    - https://en.wikipedia.org/wiki/Machine_learning
    - http://www.sas.com/en_id/insights/analytics/machine-learning.html
    - http://nlp.stanford.edu/software/tagger.shtml