A PROJECT REPORT

ON

Automatic Text Categorization of Word Problems

SUBMITTED BY

Mr. Suraj Kumar, B120084330
Mr. Pranav Kanade, B120084257
Mr. Gaurav Shukla, B120084237
Mr. Sanket Deshmukh, B120084228

Under the guidance of

Mrs. Shanthi K. Guru

In partial fulfillment of the requirements for

Bachelors Degree in Computer Engineering of

SAVITRIBAI PHULE PUNE UNIVERSITY

[2016-2017]

Department of Computer Engineering

D. Y. PATIL COLLEGE OF ENGINEERING

Akurdi, PUNE-411044.

D. Y. Patil College of Engineering

Akurdi, Pune-411044

Department of Computer Engineering

CERTIFICATE

This is to certify that the Project Entitled

Automatic Text Categorization of Word Problems

Submitted by

Mr. Suraj Kumar, B120084330

Mr. Pranav Kanade, B120084257

Mr. Gaurav Shukla, B120084237

Mr. Sanket Deshmukh, B120084228

is a bonafide work carried out by Students under the supervision of Mrs. Shanthi Guru and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Mrs. Shanthi K. Guru Internal Guide Dept. of Computer Engg.	Dr. Neeta Deshpande H.O.D Dept. of Computer Engg.

Principal

D. Y. Patil College of Engineering, Akurdi Pune 44

Signature of Internal Examiner Signature of External Examiner

PROJECT APPROVAL SHEET

A Project Title

Automatic Text Categorization of Word Problems

Is successfully completed by

Mr. Suraj Kumar, B120084330

Mr. Pranav Kanade, B120084257

Mr. Gaurav Shukla, B120084237

Mr. Sanket Deshmukh, B120084228

at

DEPARTMENT OF COMPUTER ENGINEERING

D. Y. PATIL COLLEGE OF ENGINEERING, AKURDI, PUNE-44

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2016-2017

Mrs. Shanthi K. Guru	Prof. Dr. Neeta Deshpande
Internal Guide	H.O.D
Dept. of Computer Engg.	Dept. of Computer Engg.

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the final project report on ***'Automatic Text Categorization of Word Problems'****.*

We would like to take this opportunity to thank my internal guide ***Mrs. Shanthi K. Guru*** *for giving me all the help and guidance we needed. We are thankful to them for their kind support. Their valuable suggestions were very helpful.*

We are also grateful to ***Prof. Dr. Neeta Deshpande****, Head of Computer Engineering Department, D. Y. Patil College of Engineering, Akurdi for her indispensable support and suggestions.*

In the end our special thanks to ***Ms. Pallavi Khude, Mrs. Suvarna Kadam (Project Coordinator), Mr. Milind Deshpande and Persistent Systems Ltd. Pune*** *for providing various resources and guidance from time to time for our project.*

Pranav Kanade
Suraj Kumar
Gaurav Shukla
Sanket Deshmukh
(B.E. Computer Engg.)

ABSTRACT

Automatic Mathematical Word Problem Solver serves two purposes: First is in Intelligent Tutoring Systems and Second is in Cognitive Science to study way of thinking to solve these problems in children. To solve a MWP, First we need to classify the given problem into either Joint and Separate problem or Part-Part-Whole problem or Compare problem. After classification is done we need to recognize the function of each sentence and extract information from them. This extracted feature is finally used to form equations and then equations are solved to obtain the solution. So the proposed system consists of following main modules: Classification, Information Extraction, Equation Generator, and Equation Solver.

Contents

List of Figures

List of Tables

List of Abbreviations

- KNN : K-Nearest Neighbors
- SVM : Support Vector Machaine
- LSTM : Long Short Term Memory
- NLP : Natural Language Processing
- MWP : Mathematical Word Problem
- POS : Part of Speech
- NL : Natural Language
- DOL : Dolphine Language
- TF-IDF : Term Frequency–Inverse Document Frequency
- IE : Information Extraction
- PPW : Part-Part Whole
- J-S : Joint-Separate
- NE : Named Entity
- RNN : Recurrent Neural Network
- NLTK : Natural Language Toolkit
- JSON : JavaScript Object Notation
- CLI : Command Line Interface
- GPU : Graphics Processing Unit
- UML : Unified Modeling Language
- DFD : Data Flow Diagram

1 Synopsis

1.1 Project Title

Automatic Text Categorization of Word Problems

1.2 Project Option

Industry Sponsored

1.3 Internal Guide

Mrs. Shanthi K. Guru

1.4 Sponsorship and External Guide

Sponsor : Persistent Systems Ltd. Pune

Guide : Mr. Milind Deshpande

1.5 Technical Keywords (As per ACM Keywords)

- Arithmetic Word Problem
- Information Extraction
- Natural Language Processing
- Natural Text
- Semantic Analysis
- Syntactic Analysis
- Semantic Parsing

- Co-reference Resolution
- Word Vectors
- Word Embedding
- Verb Categorization
- Tf-Idf
- Template Based Approach
- Artificial Intelligence
- Machine Learning
- KNN
- SVM
- Automatic Classifier
- Automatic Solver
- Neural Networks
- Tensorflow
- LSTM

1.6 Problem Statement

Design a system which accepts an algebraic word problem as an input and classifies it into pre-defined domain (e.g. Part-Part-Whole, Join-Separate, Comparison, Equal Groups, Multiplicative-Compare etc.) using NLP (Information Extraction) and Machine Learning Techniques. Also generating equations for these problems and solving them automatically.

1.7 Abstract

Automatic Mathematical Word Problem Solver serves two purposes: First is in Intelligent Tutoring Systems and Second is in Cognitive Science to study way of thinking to solve these problems in children. To solve a MWP, First we need to classify the given problem into either Joint and Separate problem or Part-Part-Whole problem or Compare problem. After classification is done we need to recognize the function of each sentence and extract information from them. This extracted feature is finally used to form equations and then equations are solved to obtain the solution.So the proposed system consists of following main modules: Classification, Information Extraction, Equation Generator and Equation Solver.

1.8 Goals and Objectives

1. Building assistant tool for students to solve algebraic word problems
2. To study thought process while solving a algebraic word problem
3. To assist in teaching as Intelligent Tutoring System

1.9 Relevant mathematics associated with the Project

1.9.1 Problem Formulation

We address the problem of classifying and solving the math word problems that map to single equation and single variable form. Considering Following Representation for Problem formulation:

- P : set of all the Problems that map to single equation and single variable
- WP : specific word problem that we are trying to solve $WP \in P$
- w : set of all the words used in text explanation of the problem $w = \{w1, w2, \ldots., wn\}$

- V : set of all the variables identified
- quant: set of all the numerical quantities mentioned in the text description of problem
- x : unknown resultant state of the variable described in the question phrase

1.9.2 Problem Classification

- Cl : set of all classes identified for categorization of the set P
  Cl = { Join-Separate problem, Comparison Problem, Part-Part-Whole }
- $Feat$: feature-set identified for the categorization with "random forest" method

1.9.3 Problem Solution

- Q : set of all Q-set identified
- Op : set of all the binary operators used in the equation formation which maps math word problem statement
- Op = { +, -, /, *, =}
- verbs: set of all the verbs identified relevant to the operation on the variables E.g.- get, gave, fill, ate, etc.
- E : resultant equation that correctly maps all the information described in the text explanation of problem
- E = { V, Op }*

1.10 Names of Conferences / Journals where papers can be published

- National Conference on Emerging Trends in Computer Engineering and Technology, 24-25 March 2017, at MIT AOE, Pune

1.11 Review of Conference/Journal Papers supporting Project idea

Previous work to solve mathematical word problems falls into two categories: symbolic approaches and statistical learning methods. In symbolic approaches, math problem sentences are transformed to certain structure by pattern matching or verb categorization.Equations are then derived from the structure. Statistical learning methods are employed using probabilistic model. Some of these methods are discussed below.

- The paper [*Kushman et al., 2014*] proposes a system which automatically learns to solve algebra word problems. In this paper, a problem is mapped to equations in two steps: First, a template is selected to define structure of equation system. Next, template is instantiated with numbers and nouns from the text. It uses data from algebra.com, a crowd-sourced tutoring website. It can correctly answer 70% data in the data-set.

- Another paper [*Hosseini et al. 2014*] designs a system named ARIS which is based on state transition. It analyzes each sentence in the problem to find relevant variables and their values. Verb categorization is important during state transition. There are seven categories of verbs.This is 81.2% accurate in classifying verbs categories and could solve 77.7% of problems of standard primary school. Data-set consists of only addition and subtraction problems. No verb is shared between training and test data-sets.

- The paper [*Rik Koncel-Kedziorski et al., 2015*] designs a system called ALGES which solves multi-sentence algebraic word problems as that of generating and scoring equation trees. Integer Linear Programming(ILP) is used to generate

equation trees. Scoring is done via learning local and global discriminative models. Training data-set for this consists of problems and their equations. No manual annotation is done on training data. It can cover all mathematical operations (+, -, *, / , =).

- The paper [*Roy and Roth, 2015*] solves arithmetic word problems without using predefined templates or additional manual annotation. There is a notion of expression tree which is helpful in representing and evaluating the target expression. This approach outperforms existing system achieving state of art performance. The system is evaluated on three data-sets. Each data-set has different category of problems. The data-set consists of problems of all arithmetic types(+,-,*,/).

- The paper [*Suleyman Cetintas et al., 2009*] describes text categorization of mathematical word problems into multiplicative compare and equal group problems. It allows selective stop-word removal and stemming. POS tagging is used to distinguish between discriminative and non-discriminative words. It uses SVM classifier with selective pre-processing technique which outperforms default SVM classifier. This approach is totally based on uni-gram model of words.

- The paper [*Amnueypornsakul et al., 2014*] follows a systematic approach to solve mathematical word problems. There are three categories of problems which are handled by this: join and separate problems, part part whole problems and compare problems. Various stages to solve problems are: Firstly, the problem is classified, which is followed by sentence function identification and sign prediction. Next, equation is generated using information extraction followed by solving it.

- The paper [*Arindam Mitra et al.*] uses formulas to solve arithmetic word problems. The system developed identifies the variables and their attributes; and maps this information to higher level representation. This representation is then used to recognize the presence of formula along with associated variables. Finally, an equation is generated from formal description of the formula. It is able

to produce 86.07% accuracy. Add-Sub data-set is used which consists of 395 addition-subtraction arithmetic problems.

- The paper [*Shuming Shi et al., 2015*] presents a system called SigmaDolphin which automatically solves math word problems by semantic parsing and reasoning. DOL language is used as structured semantic representation of NL text. A semantic parser is used to transform math problem text into DOL trees. A reasoning module is used to derive math expressions from DOL trees and calculate final answers. Data-set used contains 1878 math problems.

1.12 Plan of Project Execution

Month	**Schedule**	**Project Task**
July	1st Week	Idea about Project Selection/Project Topic Selection
	3rd Week	Submission of Project Synopsis/Abstract
	5th Week	Guide Allocation & Literature Survey
August	1st Week	To Display List of Project Groups with Guide Names
	2nd Week	First presentation with Guide about idea of projects (Feasibility study)
	3rd & 4th Week	Requirement Analysis (SRS Document) Preparation & Submission
September	1st Week	Design of Project-Low level, High Level, Data Structure, Database tables & Algorithms
	3rd Week	Presentation –II with design
	4th Week	Preparation of preliminary report

Table 1: Project Plan

Month	**Schedule**	**Project Task**
October	1st Week	Submission-Prelim Report of the Project
November	2nd Week	University Exam on Preliminary Report
January	1st Week	Coding
	2nd Week	At lest 2 module should finish (30%) Total Work
February	1st Week	First demonstration on project work expected (60%) of total work
March	1st Week	Test Plan , Design & Installation
	3rd Week	Final Project Demonstration
	4th Week	Preparation of project report , Preparation of Installable Project & Manual
April	1st Week	Submission of Report (Final Submission)
May	2nd Week	Final University Examination

Table 2: Project Plan

2 Technical Keywords

2.1 Area of Project

In this project we are building a MWP solver. This system is built using python language. It uses various methodology and technology such as Machine Learning, Neural Networks, Natural Language Processing etc. It uses intersection of other techniques and tools to develop the system such as Tensorflow, Spacy, Numpy, Scikit, Sympy, Gensim etc.

2.2 Technical Keywords

- Arithmetic Word Problem
- Information Extraction
- Natural Language Processing
- Natural Text
- Semantic Analysis
- Syntactic Analysis
- Semantic Parsing
- Co-reference Resolution
- Word Vectors
- Word Embedding
- Verb Categorization
- Tf-Idf
- Template Based Approach
- Artificial Intelligence

- Machine Learning
- KNN
- SVM
- Automatic Classifier
- Automatic Solver
- Neural Networks
- Tensorflow
- LSTM

3 Introduction

3.1 Project Idea

In Mathematics, word problem is the term used to describe any mathematical exercise on which significant background information is presented as text rather than in mathematical notations. This information is critical to solving the problems as it presents relevant data which is useful in identifying problem type and to identify the components of problem and we are trying to solve MWP with computers.

3.2 Motivation of the Project

While participating for ACM ICPC 2015, we did not get much clue about how to proceed to solve a algorithmic problem. What approach to follow and which methods to use. So we thought to make a system which could automatically suggest data structures and algorithms for the problem. But the problem here was the topic was too vast to start with. So we thought to apply the same for the basic algebraic problems and thus in this way we got motivated for the proposed system.

3.3 Literature Survey

3.3.1 Information Extraction - By Ralph Grishman[10]

Key Points - Named Entities, Syntactic Analysis, Semantic Analysis, Co-reference Resolution, Relation Extraction etc.
The goal of information extraction (IE) is to make the text's semantic structure explicit so that we can make use of it. More precisely, IE is the process of analyzing text and identifying mentions of semantically defined entities and relationships within it. A well-studied example involves capturing all instances of executives starting or leaving their jobs. If a system reads "Oshkosh, 1 April–Mary Smith retired from Proctor and Gamble yesterday; she was succeeded as CTO by Jillian Jones," it should be

Name	Job title	Company	Date	In/out
Mary Smith	CTO	Proctor and Gamble	31 March	Out
Jillian Jones	CTO	Proctor and Gamble	31 March	In

Figure 1: Example of IE

able to build the example in Table below. IE is generally organized as the product of several analysis components: named entity (NE) tagging; syntactic analysis; [within-document] co-reference resolution;entity, relation, and event extraction (semantic analysis); and cross-document co-reference resolution. This decomposition is convenient for describing a typical IE system, but it doesn't require the steps to be done in sequence. In fact, judicious use of joint inference can provide a significant advantage. Named Entities: Suppose we wanted to build a system that could analyze reports of prizes being awarded, such as "Marie Curie, wife of Pierre Curie, received the Nobel Prize in Chemistry in 1911. She had previously been awarded the Nobel Prize in Physics in 1903." Our first task is to identify the individuals and entities referred to in the passage. To this end, we'll run a name tagger that recognizes and classifies the names "Marie Curie" and "Pierre Curie" as person names and "Nobel Prize in Chemistry" and "Nobel Prize in Physics" as prize names. This might seem straightforward, but in fact, it isn't so simple; if she had received the Nobel Prize in Stockholm, we wouldn't want to mark "in Stockholm" as part of the name of the prize. Researchers have used many different models for NE extraction, including hidden Markov models, maximum entropy Markov models, and conditional random fields. Syntactic Analysis- The next stage of analysis aims to capture some of the general linguistic relationships in each sentence, and it's essential for determining semantic relations (the "who did what to whom") later in the pipeline. Generally speaking, richer analysis at this stage simplifies the semantic analysis that follows. At a minimum, we perform partial parsing or chunking during syntactic analysis. This divides the input into a sequence of chunks, non-recursive linguistic units such as noun groups and verb groups. Co-reference Resolution- If an individual is mentioned several times, later mentions generally won't repeat the full name. Thus, in the example given earlier, we might see instead an abbreviated name ("Mdm. Curie"), a

pronoun ("she"), or a descriptive phrase ("the noted scientist"). We don't want a database entry to list "she," so we run co-reference resolution to link all mentions of an individual within a document to the most complete version of the name. Semantic Analysis- The next stage maps the syntactically analyzed sentence—consisting of a set of lexical items connected by syn- tactic relations—into a set of entities and database relations.

3.3.2 Machine-Guided Solution to Mathematical Word Problems - By Bussaba Amnueypornsakul, Suma Bhat[5]

Key Points– Problem Classification(Part-Part Whole Problems, Compare Problems, Join-Separate Problems), Equation Generation , Equation Solver etc.
This paper describes various types of problems like join-separate problems, compare problems etc. The approach followed to solve MWP is to classify the problems into predefined domain and then comes the step of equation generation which is done using information extraction. Finally these equations are solved using python module numpy. Problem-level features includes the features like length-related and document-related. The length of the problem in number of sentences is a feature that we consider at the problem level, noticing that on an average, J-S problems tend to have more sentences per problem than those of the C type, which in turn have more sentences than those of the PPW type. Sentence-level features: Mainly used for sentence-level classification into types, the features in this class are positional, structural or semantic. We observe that problems of the J-S type are characterized by significant action language that describe changes in the possession or condition of objects. Thus, we posit that the count of unique verb lemmas will serve as a discriminating feature. Entity-related features denotes the number of unique noun phrases.

3.3.3 Learning to Automatically Solve Algebra Word Problems - By Nate Kushman , Yoav Artzi , Luke Zettlemoyer, and Regina Barzilay[1]

Key Points-Situated Semantic Interpretation, Automatic Word Problem Solvers, Information Extraction.

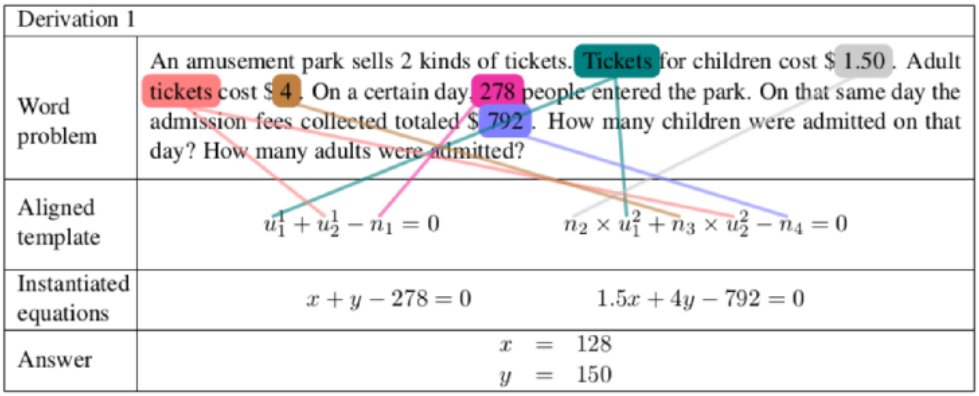


Figure 2: Example of MWP

This system reasons across sentence boundaries to construct and solve a system of linear equations, while simultaneously recovering an alignment of the variables and numbers in these equations to the problem text. It defines a two step process to map word problems to equations. First, a template is selected to define the overall structure of the equation system. Next, the template is instantiated with numbers and nouns from the text. During inference we consider these two steps jointly. The template dictates the form of the equations in the system and the type of slots in each: u slots represent unknowns and n slots are for numbers that must be filled from the text. In above figure, the selected template has two unknown slots, u1 and u2 , and four number slots, n1 to n4 . Slots can be shared between equations, for example, the unknown slots u1 and u2 in the example appear in both equations. A slot may have different instances, for example u11 and u21 are the two instances of u1 in the example. We align each slot instance to a word in the problem. Each number slot n is aligned to a number, and each unknown slot u is aligned to a noun. For example, Derivation aligns the number 278 to n1 , 1.50 to n2 , 4 to n3 , and 792 to n4 . It also aligns both instances of u1 (e.g., u 11 and u 21) to “Tickets”, and both instances of u2 to “tickets”.

3.3.4 Learning to Solve Arithmetic Word Problems with Verb Categorization - By Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni and Nate Kushman[2]

Key Points - Verb Categorization, Equation Generator and Solver, Sentence Categorization etc.

This system (ARIS) analyzes each of the sentences in the problem statement to identify the relevant variables and their values. ARIS then maps this information into an equation that represents the problem, and enables its solution as shown in Figure. The paper analyzes the arithmetic-word problems "genre", identifying seven categories of verbs used in such problems. ARIS learns to categorize verbs with 81.2% accuracy, and is able to solve 77.7% of the problems in a corpus of standard primary school test questions.

3.3.5 Automatic Text Categorization of Mathematical Word Problems - By Suleyman Cetintas, Luo Si, Yan Ping Xin, Dake Zhang, Joo Young Park[4]

Key Points- Text categorization for mathematical word problems(Multiplicative Compare and Equal Group problems), SVM classifier, Part of Speech Tagging (POS).

This paper uses Support Vector Machine (SVM) which is a learning approach that is based on structural risk minimization principle. The problem SVM deals with is to find a decision surface that best separates the data points in two classes over a vector space. SVM has been shown to achieve state-of-the-art generalization performance over a wide variety of application domains of which space limitations preclude mentioning and is one of the most accurate and widely used text categorization techniques. The commonly accepted text pre-processing method of removing stop-words should be avoided for classification of mathematical word problems. For the categorization of mathematical word problems, stemming hurts the categorization performance as it transforms some of the words that are associated with different mathematical word problems into word stems that can exist in any mathematical word problem type. It is important to distinguish the discriminative parts of speech from the non-

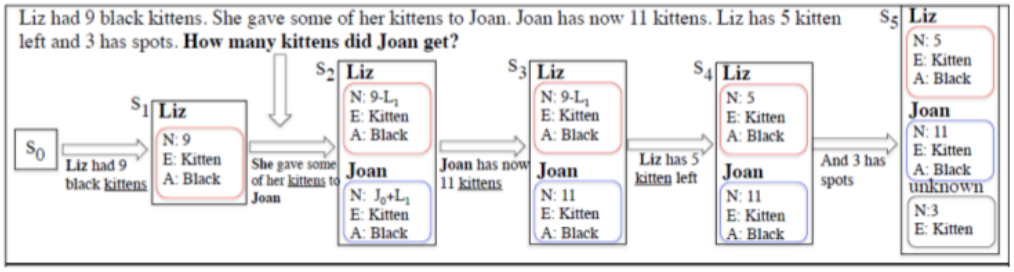


Figure 3: States of MWP

discriminative ones and eliminate the non- discriminative parts of speech before a classification algorithm is applied on the mathematical word problems.

4 Problem Definition and scope

4.1 Problem Statement

Design a system which accepts an algebraic word problem as an input and classifies it into pre-defined domain (e.g. Part-Part-Whole, Join-Separate, Comparison, Equal Groups, Multiplicative-Compare etc.) using NLP (Information Extraction) and Machine Learning Techniques. Also generating equations for these problems and solving them automatically.

4.1.1 Goals and objectives

1. Building assistant tool for students to solve algebraic word problems.
2. To study thought process while solving a algebraic word problem.
3. To assist in teaching as Intelligent Tutoring System.

4.1.2 Statement of scope

Since there could be variety of MWP and it will be a tough job to define a generic solver. So the system focuses on solving three types of problems:

1. Join and Separate Problems
2. Part-Part-Whole problems
3. Comparison Problems

4.2 Major Constraints

- This proposed system works on only single variable MWP.
- The hardware requirements are very of high capacity if hardware is not that much of power-full there will be performance degradation.

- If the problem does not fall in the specified category it would be solvable.

4.3 Methodologies of Problem solving and efficiency issues

4.3.1 Classifier

MWP describes the textual information of a problem. A student need to understand problem language and do analytical reasoning to generate and solve the problem. Similar procedure is applied for machine. Reasoning by machines to generate equation is relatively tough job as compared to humans. Generally two approaches are followed for solving MWP. First method involves Classification of problems into predefined categories like PPW, JS, Compare followed by equation generation and solver phases. Second method do not take into account problem type, it rather uses state transition model to parse each sentence and generate equation. Details of three types of problems are:

- **Part Part Whole :** Problems of this category contain two functional types of sentences: part and whole. Part sentence denotes some numerical or unknown values. Whole sentence denotes values which is sum of values of two or more part sentences.For example, DYPCOE, Akurdi organized a cultural fest. 500 students attended the fest. There are 1200 students in the college. How many students were absent for the cultural fest.

- **Join Separate :** Problems of this category contain three types of sentences: Given, Change and Result. Given denotes initial quantity in the problem. Change denotes updation to Given. Result denotes final quantity as a result of applying Change to Given. For example, DYPCOE, Akurdi organized a test. There are 10 questions to solve. Ram solved 6 questions. How many questions were not solved by him.

- **Compare :** These problems involves comparison of count of two sets. For example, DYPCOE, Akurdi has 100 faculties. DYPIEMR, Akurdi has 5 more faculties than DYPCOE. How many faculties are there in DYPIEMR.

Problem Classification is a special type of text classification which requires numeric features as input for learning and prediction. Length of problems in terms of number of sentences is a very basic feature which could be used for this purpose. On an average JS problems has more number of sentences than Compare and Compare has more number of sentences than PPW. Comparative adjectives like than could be a unique feature for Compare problems. Keywords like altogether characterizes PPW problems and action verbs describe JS problems which could be extracted using TF-IDF method.

Classification gives more accuracy provided stop-words are not removed and stemming is avoided. In normal text classification stop-words make no sense, so it is removed but for MWP this is not the case. Some stop-word like than, altogether are very useful for MWP Classification. Stemming also hurts performance of classifier. Some words like times gives time after stemming which could misclassify a problem. POS tagging is also used to distinguish words in discriminative POS from non-discriminative POS. Otherwise non-discriminative POS could lead to misclassification. Some words like working, speed, traffic etc can exist in any MWP. So they are non-discriminative. Details of stop-words removal, stemming, POS is covered in pre-processing of data-set.

Statistical Learning Algorithms like SVM, Random Forest are used for MWP Classification. Linear SVM is chosen as it is fast. Scikit-Learn in Python have implementation of both SVM, Random Forest which could be directly used for classification.

4.3.2 Solver

- Information Extraction : It is difficult to analyze information present in raw text, as it does not posses any explicit semantic structure. IE aims to make text's semantic structure explicit so that it can be used. Hence, IE could be defined as the process of semantically identifying mentions of semantically defined entities and relationships within the text. More specifically in the task of solving MWP, relationships play important role they make it easy to infer the facts and process of identification of entities involved in that relationship.

- Q-set (Quantified set) : In order to build the equation from the relation stated in MWP, we need some definitive data structure which could be used to store

the state of the variables and the relation between any two variables. Hence we use the concept of "Quantified set"(Q-set) proposed by *Koncel-Kedziorski et. al.(2015)*. The definition of quantified set is useful in modeling representation of quantities and their relationships in natural language. Q-sets are used for tracking and combining state of entity's containers to get the final result.

Definition 1: Given a MWP w, let Q be the set of all possible spans in w.

Q-set = (ent, qnt, adj, loc, verb, syn, ctr) with,

Q = $Q \cup \theta$: θ represents empty span, i.e. unspecified property.

ent $\in$ Q : entity noun (e.g. - Student, chocolate, etc.);

qnt $\in R \cup$ {x} : quantity or amount of the entity(e.g. - 4, x, etc.);

adj $\in Q$: adjective for ent in w;

loc $\in Q$: location of ent (e.g. 'in the drawer');

vrb $\in Q$: governing verb for ent (e.g. - 'get', 'ate', 'gave', 'filled');

syn $\in Q$: syntactic and positional information for ent. (e.g. - Noun in subject position);

ctr $\subseteq Q$: container of ent (e.g. 'box' might be a container for 'chocolates' Q-set);

Definition 2: combiner

As we have already defined the Q-set, there is also a need to define a function on them which combines Q-set according to the operation inferred from their relationship to model intermediate stages of the MWP.

combiner : Q-set $\times$ **Q-set** $\times$ **Op** $\rightarrow$ **Q-set**

combiner : Q-set $\times \Phi\{=\} \rightarrow$ **Q-set**

To be precise,

combiner : base_Q-set $\times$ *base_Q-set* $\times$ *Op* $\rightarrow$ *compound_Q-set*

This recursive definition allows us to take advantage of recurrent neural networks, as it satisfies both parameters of our problem domain sequential operations and recursive definition of quantities. **Verbs** are the single most important parameter which describes the policy of the actor over the quantity of entity, i.e. depending on the verb and its dependencies given in the text the operation is chosen to be performed on the Q-sets (e.g. - 'Ajay ate 4 chocolates from pack of 12', in adjoint sentence 'ate' is the verb of the span. Which depicts the

operation of **removal** i.e. '***ate***' $\rightarrow \{-\}$.

Based on above definition we propose construction of reinforced LSTM to solve MWP. The process involves identification of all the base Q-sets with the help of the **"subject, verb, object" triplet**(section : pre-processing). These base Q-sets are related to one another by some connecting verb. These Q-sets will be fed into system one per iteration of RNN. All of them individually emit operation = (assignment) to their respective container of subject (e.g. - {apples, 5, suma} : container of suma contains 5 apples initially.). After every iteration of LSTM, combine merges any two Q-sets depending on governing verb, and discards original Q-sets out of it's memory. Remaining set of Q-sets get passed to next iteration of the LSTM.

4.4 Outcome

The automatic problem solver defines following outcomes:

1. Assistance Tools for students to solve algebraic word problems
2. To study thought process while solving a algebraic word problem
3. To assist in teaching as Intelligent Tutoring System

4.5 Applications

- Intelligent tutor system
- algebraic MWP solver
- Basic model for smart system

4.6 Software Resources Required

Platform :

1. Operating System: Linux(Ubuntu)
2. Libraries Used: scikit, NLTK3, TensorFlow , Spacy, Stanford coreNLP, Numpy, gensim, numpy
3. Programming Language: Python3

4.7 Hardware Resources Required

Sr. No.	Parameter	Minimum Requirement
1	Intel CPU Speed	2 GHz
2	GPU(NVIDIA)	8 GB
3	RAM	16 GB

Table 3: Hardware Requirements

5 Project Plan

5.1 Project Estimates

While developing the proposed system, it was required to test each unit before adding another unit because the correctness of next unit depends on correctness of present unit. So **Incremental Method** of SDLC was best choice for development. Three units were developed using incremental approach- Pre-processing, Classification, Equation generator. Each unit defines accuracy of next unit. This method gives us flexibility to apply various methods for classification and equation generation. Also accuracy of the overall system can be improved using this technique.

5.1.1 Time Estimates

	Task Name	Start Date	End Date	Duration	Status	Assigned To
1	**Idea Prepration**	**07/04/16**	**08/02/16**	**22d**		
2	Discussion	07/04/16	07/18/16	10.5d	Complete	Pranav kanade, Suraj Kumar
3	Revision 1	07/18/16	07/20/16	2d	Complete	Gaurav Shukla, Sanket Deshmukh
4	Revision 2	07/20/16	07/25/16	3.5d	Complete	Sanket Deshmukh, Pranav Kanade, Gaurav Shukla, Suraj Kumar
5	Scope Limiting	07/26/16	08/02/16	6d	Complete	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh,
6	**Domain Identification**	**07/31/16**	**09/26/16**	**42d**		
7	Data Collection	07/31/16	08/17/16	14d	Complete	Gaurav Shukla, Sanket Deshmukh
8	Base Paper Study	08/03/16	08/15/16	9d	Complete	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh,
9	Reference Paper Study	08/05/16	09/26/16	37d	Complete	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
10	**Study of NLP**	08/31/16	09/14/16	11d		
11	NLP pipeline	08/31/16	09/05/16	4d	Complete	Sanket Deshmukh
12	NLP coreference Resolution	09/06/16	09/09/16	4d	Complete	Gaurav Shukla, Sanket Deshmukh
13	NLP word sence Disambiguation	09/06/16	09/14/16	7d	Complete	Pranav Kanade, Suraj Kumar
14	**Study of NLP libraries**	**09/12/16**	**10/18/16**	**27d**		
15	NLTK	09/12/16	09/23/16	10d	Complete	Gaurav Shukla
16	Spacy	10/06/16	10/18/16	9d	Complete	Gaurav Shukla, Sanket Deshmukh
17	Stanford NLP	09/19/16	09/28/16	8d	Complete	Pranav Kanade, Suraj Kumar
18	**Data Pre-processing and Cleanning**	10/16/16	10/31/16	12d	Complete	Gaurav Shukla, Sanket Deshmukh
19	**Sem 1 Report**	09/12/16	11/07/16	41d	Complete	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh,

Figure 4: Time Estimation

	Task Name	Start Date	End Date	Duration	Status	Assigned To
20	**Study of Neural Networks**	10/29/16	02/03/17	71d		
21	CNN	10/29/16	11/11/16	11d	Complete	Pranav Kanade
22	RNN	11/16/16	02/03/17	58d	Complete	Pranav Kanade, Suraj Kumar
23	LSTMs	11/16/16	12/27/16	30d	Complete	Pranav Kanade, Suraj Kumar
24	Bi-Directional LSTMs	12/28/16	01/06/17	8d	Complete	Pranav Kanade, Suraj Kumar
25	Study of Papers on Reinforcement Learning (DQn)	01/06/17	02/03/17	21d	Complete	Pranav Kanade, Suraj Kumar
26	**Feature Extraction from Text**	12/14/16	12/27/16	10d	Complete	Gaurav Shukla, Sanket Deshmukh
27	**Study of Word Vector and Context Isolation**	12/28/16	01/12/17	12d	Complete	Gaurav Shukla, Sanket Deshmukh
28	**Classification with KNN**	01/13/17	01/25/17	9d	Complete	Suraj Kumar, Sanket Deshmukh
29	**Study English Grammer**	01/24/17	02/23/17	23d	Complete	Pranav Kanade, Gaurav Shukla
30	Study Dependency Parsing and Relationship	01/24/17	02/23/17	23d	Complete	Pranav Kanade, Gaurav Shukla
31	**Study of Machine Learning and Deep Learning Libraries**	12/16/16	02/02/17	35d	Complete	Pranav Kanade, Suraj Kumar
32	Tensorflow	12/30/16	02/02/17	25d		
33	Numpy	12/16/16	01/09/17	17d		
34	Scikit and Sympy	12/26/16	01/20/17	20d		
35	**Q-set Identification**	02/24/17	03/07/17	8d	Complete	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
36	Study Q-set Parameters	02/24/17	03/07/17	8d	Complete	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
37	**Verb behavior Modeling**	03/08/17	03/20/17	9d	In Progress	Pranav Kanade, Suraj Kumar
38	**Equation Generation and Solution**	03/21/17	04/07/17	14d	In Progress	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh

Figure 5: Time Estimation

5.2 Risk Identification

S.no	1
Risk	Misclassification of MWP
Description	The classification module has accuracy of about 88%. So there is a chance of misclassification. If a problem is misclassified at this stage, the solution could never be correct.So classification is very crucial.
Category	Algorithmic Development
Source	Architectural Diagram
Chance	Low
Impact	High
Response	Mitigate
Strategy	Train classifier on large data-sets
Risk Status	Identified

Table 4: Risk 1

S.no	2
Risk	Word-Sense Disambiguation
Description	A word in English language could have different meaning. The meaning depends upon its context in the sentence. E.g. Bank could imply "River bank" or "Financial Institution" .
Category	Information Extraction
Source	Software Requirement Specification Description
Chance	Low
Impact	Medium
Response	Mitigate
Strategy	Word-Sense identification
Risk Status	Identified

Table 5: Risk 2

S.no	3
Risk	Co-reference Resolution
Description	The pronoun in a sentence will refer to some noun like name, object etc. It is important to resolve what does it refers to. E.g. Ram has 5 chocolates. He gave 2 to me. Here "He" refers to "Ram".
Category	Information Extraction
Source	Software Requirement Specification Description
Chance	Medium
Impact	Medium
Response	Mitigate
Strategy	Use Co-reference resolution methods
Risk Status	Identified

Table 6: Risk 3

S.no	4
Risk	Over-fitting:
Description	Testing data-set could be same as training data-set. It will mislead the system by showing 100% accuracy.
Category	Training and Testing Stage
Source	Test Cases
Chance	Low
Impact	High
Response	Mitigate
Strategy	Use separate data for testing and training.
Risk Status	Identified

Table 7: Risk 4

S.no	5
Risk	Balanced Data-set
Description	The number of training examples in each class should be same. If there is non-uniformity in distribution, then classifier would be biased towards classes having more examples.
Category	Data Preparation
Source	Software Requirement Specification Description
Chance	Low
Impact	Low
Response	Mitigate
Strategy	Use proper data-set distribution among all classes
Risk Status	Identified

Table 8: Risk 5

5.3 Project Schedule

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1 : Data-set collection
- Task 2 : Data cleaning
- Task 3 : Data processing
- Task 4 : NLP pipe-lining
- Task 5 : Removal of irrelevant text
- Task 6 : Word vectorization
- Task 7 : Classification with KNN
- Task 8 : Semantic Parsing
- Task 9 : Qset Generation
- Task 10 : Sign prediction using action verbs
- Task 11 : Equation generation
- Task 12 : Equation solution
- Task 13 : Interfacing

5.3.2 Task network

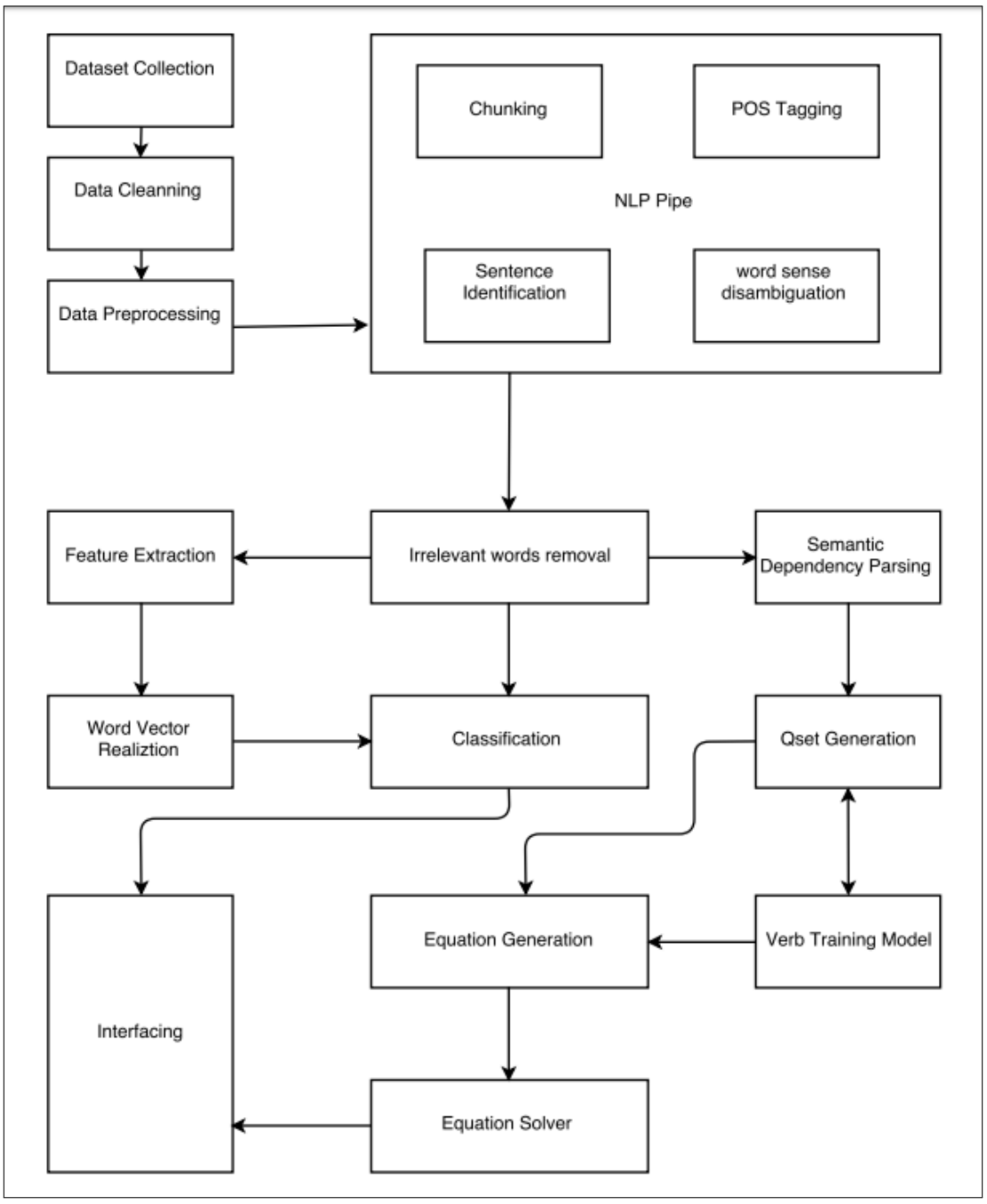


Figure 6: Task Network

5.3.3 Time-line Chart

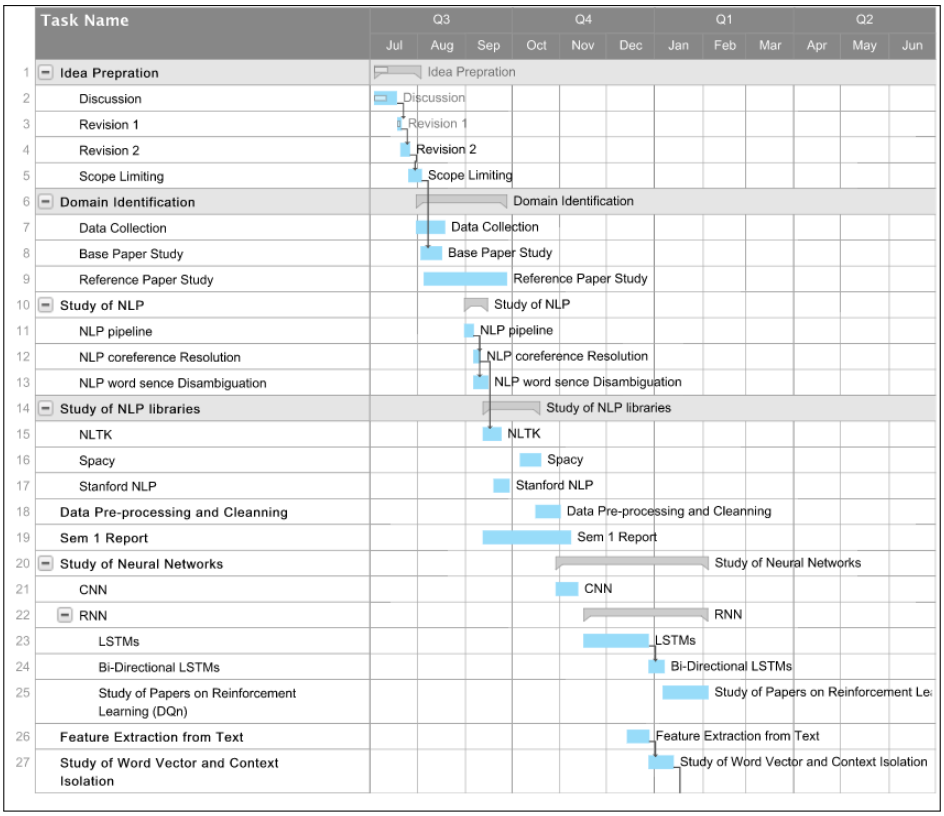


Figure 7: Time-line Chart

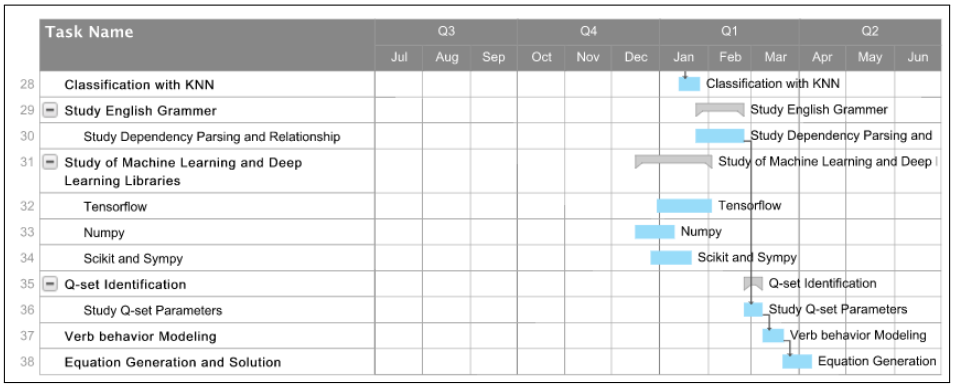


Figure 8: Time-line Chart

5.4 Team Organization

	Task Name	Assigned To
1	**Idea Prepration**	
2	Discussion	Pranav kanade, Suraj Kumar
3	Revision 1	Gaurav Shukla, Sanket Deshmukh
4	Revision 2	Sanket Deshmukh, Pranav Kanade, Gaurav Shukla, Suraj Kumar
5	Scope Limiting	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
6	**Domain Identification**	
7	Data Collection	Gaurav Shukla, Sanket Deshmukh
8	Base Paper Study	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
9	Reference Paper Study	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
10	**Study of NLP**	
11	NLP pipeline	Sanket Deshmukh
12	NLP coreference Resolution	Gaurav Shukla, Sanket Deshmukh
13	NLP word sence Disambiguation	Pranav Kanade, Suraj Kumar
14	**Study of NLP libraries**	
15	NLTK	Gaurav Shukla
16	Spacy	Gaurav Shukla, Sanket Deshmukh
17	Stanford NLP	Pranav Kanade, Suraj Kumar
18	**Data Pre-processing and Cleanning**	Gaurav Shukla, Sanket Deshmukh
19	**Sem 1 Report**	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh

Figure 9: Team Organization

	Task Name	Assigned To
20	**Study of Neural Networks**	
21	CNN	Pranav Kanade
22	RNN	Pranav Kanade, Suraj Kumar
23	LSTMs	Pranav Kanade, Suraj Kumar
24	Bi-Directional LSTMs	Pranav Kanade, Suraj Kumar
25	Study of Papers on Reinforcement Learning (DQn)	Pranav Kanade, Suraj Kumar
26	**Feature Extraction from Text**	Gaurav Shukla, Sanket Deshmukh
27	**Study of Word Vector and Context Isolation**	Gaurav Shukla, Sanket Deshmukh
28	**Classification with KNN**	Suraj Kumar, Sanket Deshmukh
29	**Study English Grammer**	Pranav Kanade, Gaurav Shukla
30	Study Dependency Parsing and Relationship	Pranav Kanade, Gaurav Shukla
31	**Study of Machine Learning and Deep Learning Libraries**	Pranav Kanade, Suraj Kumar
32	Tensorflow	
33	Numpy	
34	Scikit and Sympy	
35	**Q-set Identification**	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
36	Study Q-set Parameters	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh
37	**Verb behavior Modeling**	Pranav Kanade, Suraj Kumar
38	**Equation Generation and Solution**	Pranav Kanade, Suraj Kumar, Gaurav Shukla, Sanket Deshmukh

Figure 10: Team Organization

6 Software requirement specification

6.1 Introduction

Mathematical word problems (MWP) test critical aspects of reading comprehension in conjunction with generating a solution that agrees with the "story" in the problem. We design and construct an MWP solver in a systematic manner, as a step towards enabling comprehension in mathematics and teaching problem solving for children in the elementary grades. We build a multistage software prototype that predicts the problem type, identifies the function of sentences in each problem, and extracts the necessary information from the question to generate the corresponding mathematical equation

6.1.1 Project Scope

Since there could be variety of MWP and it will be a tough job to define a generic solver. So the system focuses on solving three types of problems:

1. Join and Separate Problems
2. Part-Part-Whole problems
3. Comparison Problems

6.1.2 User Classes and Characteristics

Two user classes exists for our project:

1. Young Learners-For intelligent tutoring system
2. Cognitive Scientists-To understand the cognitive aspects of problem solving in children

6.1.3 Operating Environment

For Automatic Problem Solver, operating environments are:

1. Operating System: Linux and Unix
2. Python and its NLP libraries
3. Training data-set is present in JSON file

6.1.4 Design and Implementation Constraints

Following are the Design and Implementation Constraints:

1. How well the classifier is predicting problem type?
2. How well a sentence function is predicted?
3. Whether equation generated is correct or not?
4. When the problem relates to time, money and distance, it needs quantity conversions before the arithmetic calculations?

6.1.5 Assumptions and Dependencies

We are assuming following to design automatic problem solver:

1. Questions do not have the complex sentence structure.
2. Only additions and subtractions type problems are solved by solver.
3. Classifier is good enough to predict problem type.

6.2 System Features

There could be lots of features for our system. Some of them are listed below:

1. Solver to MWP
2. Step by step representation of solution(optional)
3. Web interface to solver(optional)

6.2.1 Solver to MWP(Functional Requirements)

1. A large number of training data-set in JSON is required to train the classification model.
2. A well-trained information extractor

6.2.2 Step by step representation of solution(Functional Requirements)

1. A module to capture the intermediate processing steps of a problem and finally display them.

6.2.3 Web interface to solver(Functional Requirements)

1. Deploying the solver to web using Web Technologies.

6.3 External Interface Requirements

6.3.1 User Interfaces

It is the secondary part of the project as main emphasis is being given to design a solver using CLI.

6.3.2 Software Interfaces

1. Operating System: Linux(Ubuntu)
2. Programming Language: Python3
3. Libraries Used: scikit, NLTK3, TensorFlow , Spacy, Stanford coreNLP

6.3.3 Hardware Interfaces

Sr. No.	Parameter	Minimum Requirement
1	Intel CPU Speed	2 GHz
2	GPU(NVIDIA)	8 GB
3	RAM	16 GB

Table 9: Hardware Requirements

6.3.4 Communication Interfaces

Web interface is supported across all web browsers.

6.4 Non-functional Requirements

6.4.1 Performance Requirements

The Solver should solve the problem correctly. For this we weed a well-trained Classifier.

6.4.2 Safety Requirements

Classifier should be saved somewhere else(back up) for future use of similar kind of work and also for extending this work.

6.4.3 Security Requirements

No Security Threats as system does not contain any sensitive data of users.

6.4.4 Software Quality Attributes

1. AVAILABILITY: System should be available to as many customers who are using tutoring system.
2. CORRECTNESS: System should solve the MWP correctly.
3. MAINTAINABILITY: The administrators should be able to re-train the system if needed.
4. USABILITY: System should solve almost all problems of given domains.

6.5 Usage Scenario

S.no	1
scenario	System accepts current question with single variable.
Description	user queries a question which can be categorized to one of the classes.
Actor	User and system
result	System preserves its characteristics and efficiency property and takes action to classify and solve the given problem.

Table 10: Usage Scenario 1

S.no	2
scenario	System accepts the question which can be solved with single variable.
Description	User queries a question which can be solved but does not belongs to any of the class defined in the scope of the project.
Actor	User and system
result	Classification module misbehaves but solver will be able to solve the question and will preserve its efficiency.

Table 11: Usage Scenario 2

S.no	3
scenario	System accepts question which is multi-variable and does not belong to any of the classes.
Description	Question does not belong to the pre-constrained scope of the project.
Actor	User and system
result	System will fail to clarify and solve the given question.

Table 12: Usage Scenario 3

S.no	4
scenario	System accepts question which involves multiplication and division.
Description	Question has keywords such as "times" , "distributes" etc.
Actor	User and system
result	System will poorly classify but fail to solve.

Table 13: Usage Scenario 4

S.no	5
scenario	System intersects wrong concept.
Description	System identifies wrong dependency relation in given question text.
Actor	system
result	System will fail to identify the correct sign regarding the action "verb" described.

Table 14: Usage Scenario 5

6.5.1 Use Case View

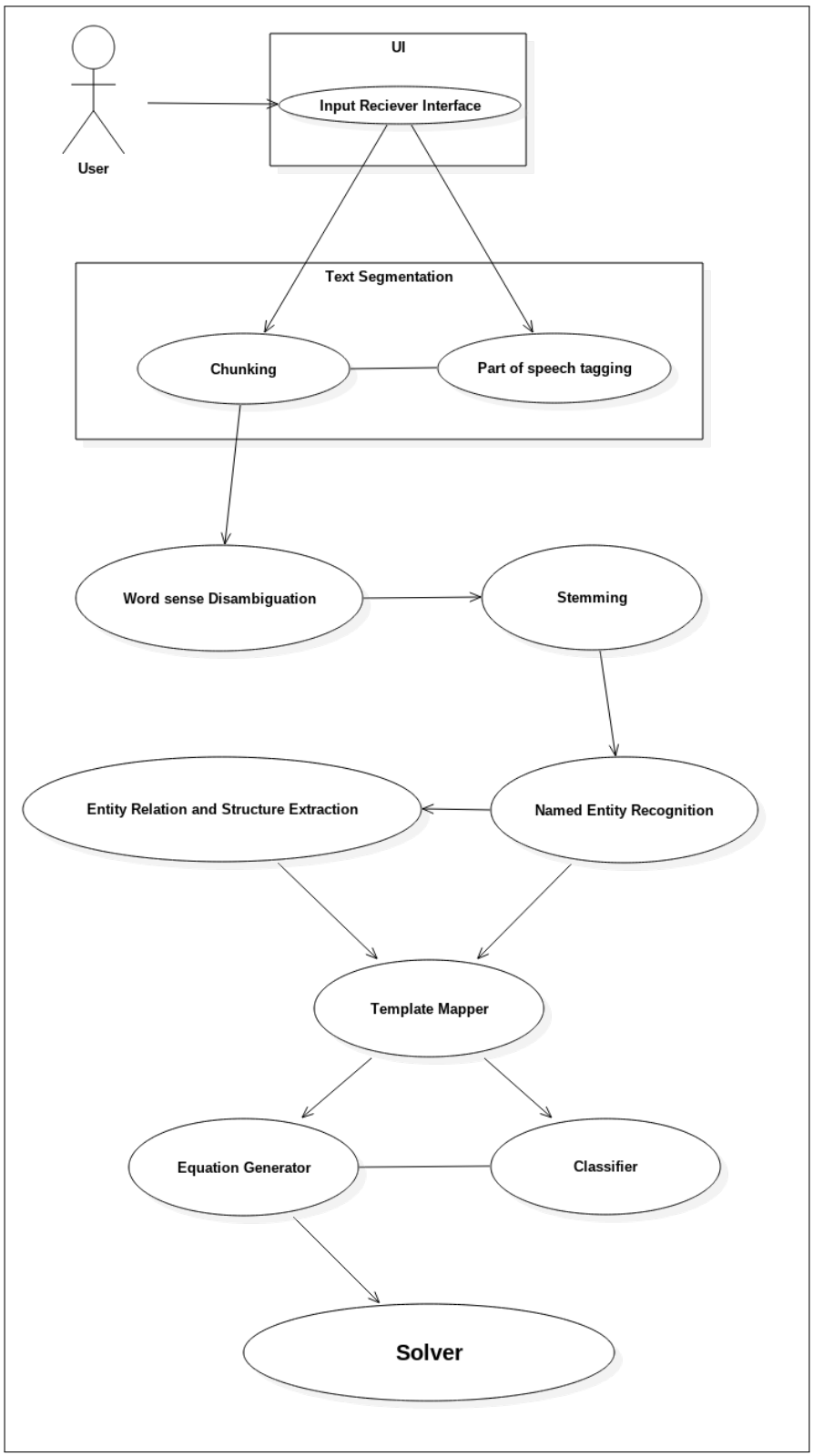


Figure 11: Use case diagram

6.5.2 Data Flow Diagrams

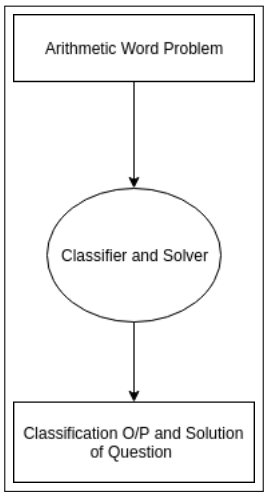


Figure 12: DFD-Level 0

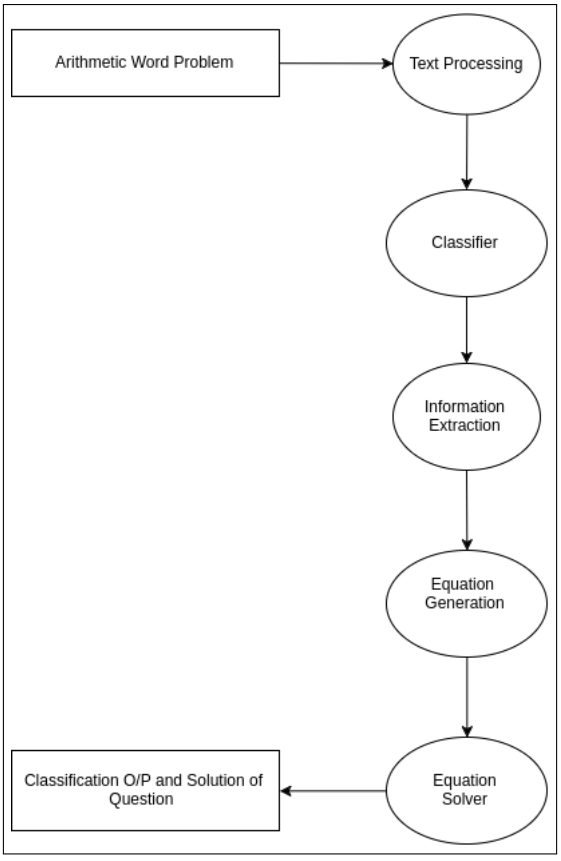


Figure 13: DFD-Level 1

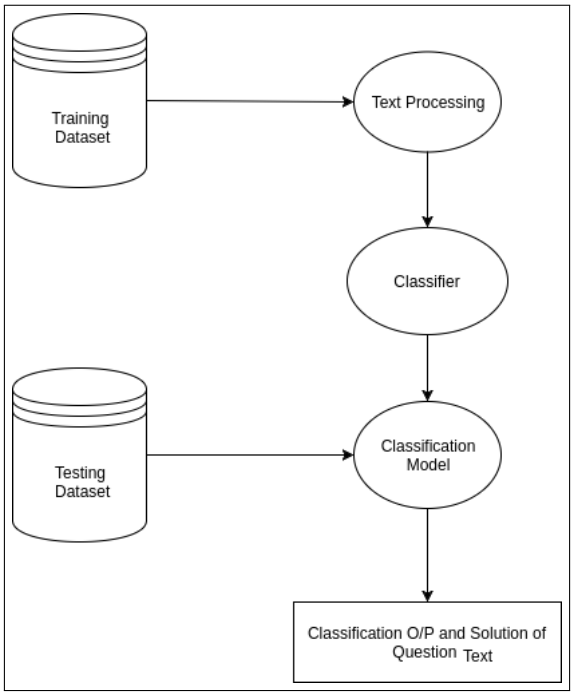


Figure 14: DFD-Level 2

6.5.3 Class Diagram

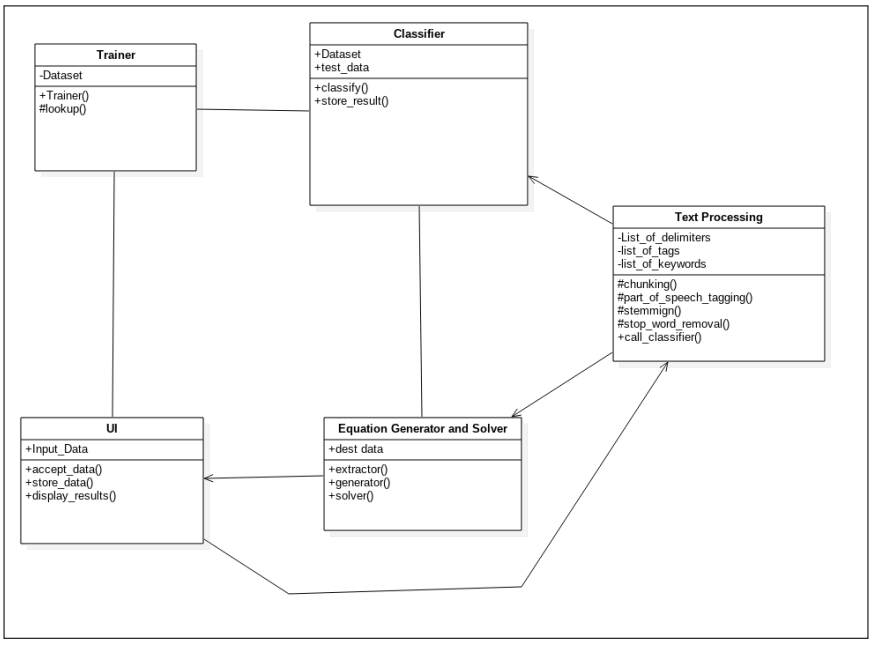


Figure 15: Class diagram

7 Detailed Design Document using Appendix A and B

7.1 Introduction

- UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering.
- The standard is managed, and was created by, the Object Management Group.
- The goal is for UML to become a common language for creating models of object oriented computer software.
- In its current form UML is comprised of two major components: a Meta model and a notation. In the future, some form of method or process may also be added to or associated with UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing objects oriented software and the software development process.
- The UML uses mostly graphical notations to express the design of software projects.

7.2 UML Diagrams

7.2.1 Sequence diagram

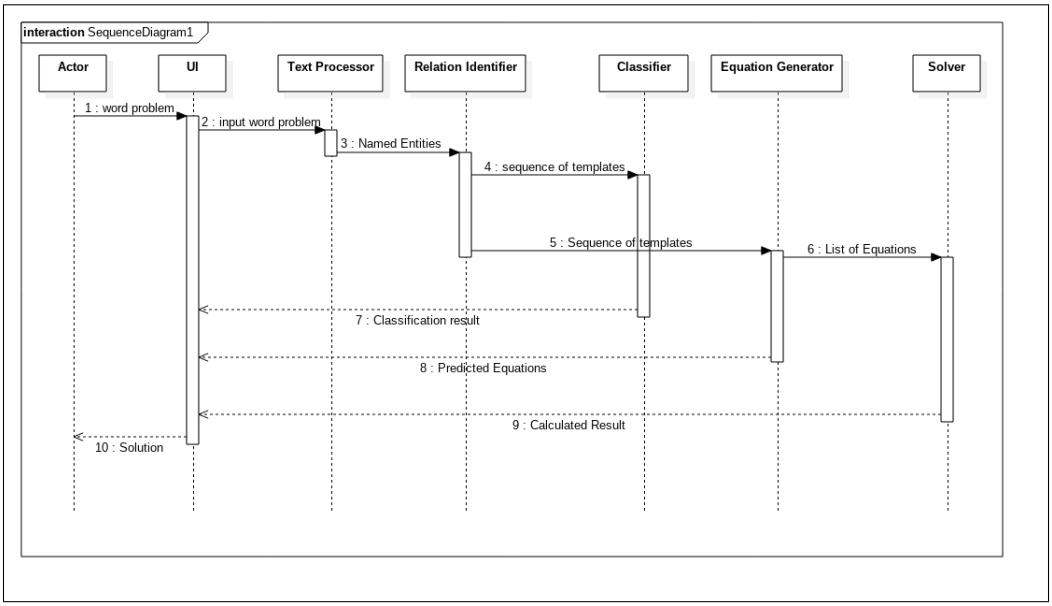


Figure 16: Sequence diagram

7.2.2 Component Diagram

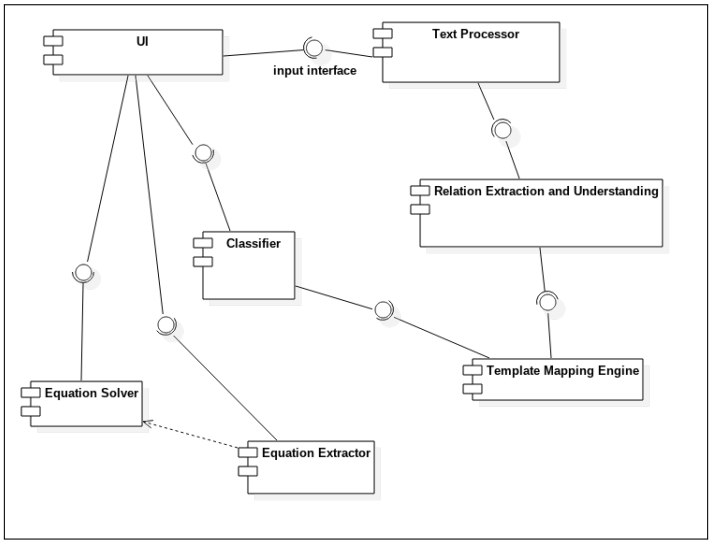


Figure 17: Component diagram

7.2.3 Activity diagram

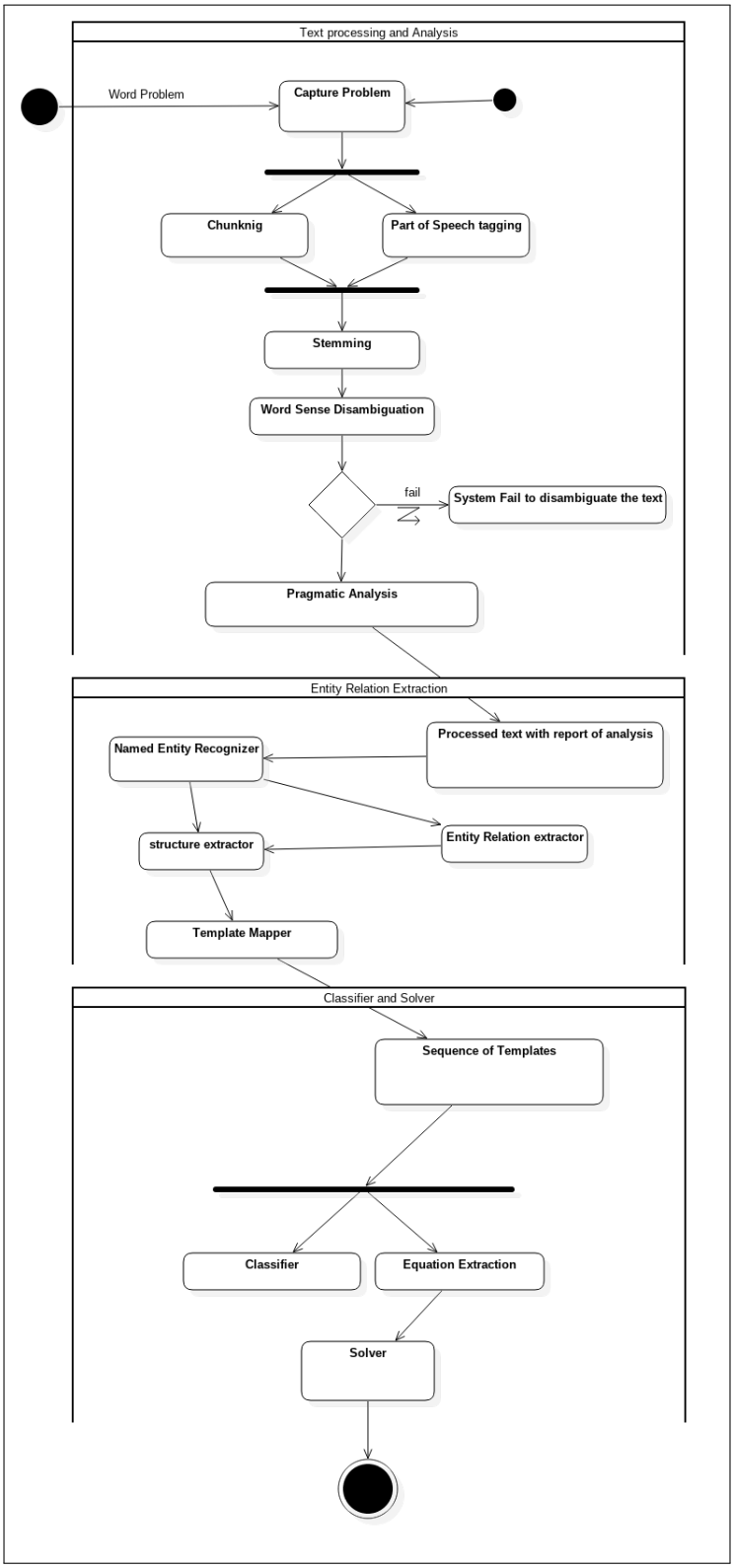


Figure 18: Activity diagram

7.3 Architectural Design

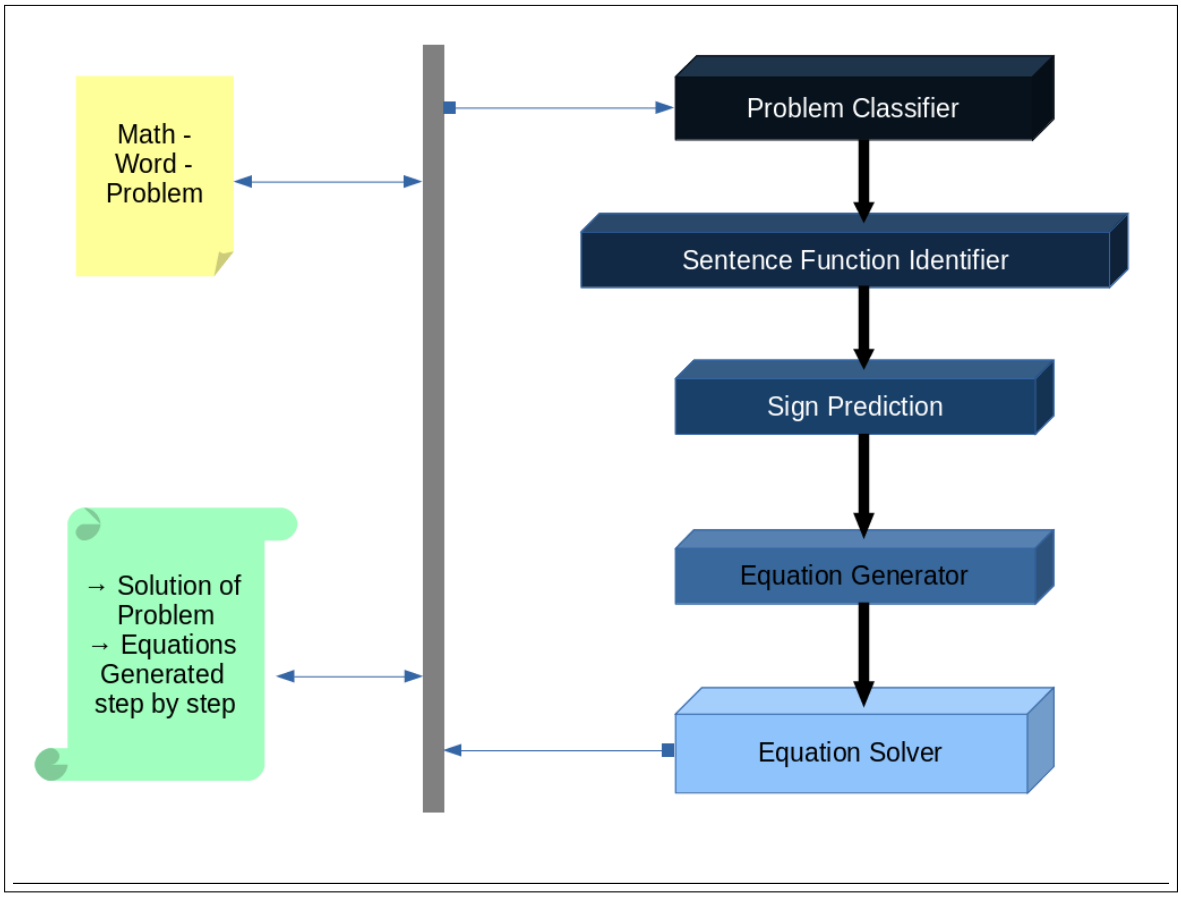


Figure 19: Architecture diagram

- The system takes mwp as input and gives solution to the problem as output.The system is divided into problem classifier,sentence function identifier,sign prediction,equation generator and solver.
- The classifier performs the task of classifying the mwp into part-part whole,joint separate and compare category.We can successfully perform the classification of certain kind of problem.We use random forest for classification of the mwp.It creates trees of several type of the given problem and select the best feasible tree for classification.
- Sentence function identifier performs the task of chunking and tagging.As input for the system is mwp,we have extract the information from the problem using

NLP technique.It identifies the noun,verb,and the entity.It track the transition of entity among the actors in the problem.

- Sign predictor identify the sign according to the context of the problem.In mwp,entity changes according to the action of the actors.The equation generator keeps all the transition of the entity in the given problem and generate equation for it.Solver performs the task of solving the equation and generates solution.

8 Project Implementation

8.1 Introduction

- Implementation of the system is distributed in two main modules namely "Classification" and "Solver"
- Classification module is responsible for distributing the queried question in predefined classes PPW ,J-S, Comparison .We try to employ KNN using the identified features using the text pre-processing modules.
- Solver module tries to derive the set of Q-sets. Identified state transfer verb gets mapped to the sign of the action. This generate the respective equation which represents the question and the unknown quantity mentioned as a variable.
- System takes help of the python library called "sympy" to solve the equation.

8.2 Tools,Technologies and Dataset Used

- **Tools and Technologies** : scikit, NLTK3, TensorFlow , Spacy, Stanford coreNLP, sumpy, gensim, numpy
- **DataSets** : MAWPS Data-set is used for experimental evaluation. It is an online repository of MWP. It consists of SingleEQ, MultiArith, SingleOp, AddSub, Data with Equations, Algebra.com Data-set. We can select data-set with reduced lexical overlap, minimized template overlap or improved grammatically characteristics because MAWPS backend supports these features. There are 3914 Questions in the data-set. Also, 3221 Questions are provided with their equations in the data-set.This data-set is used in several prominent work.

  Cognitively Guided Instruction CGI Scheme considers set of MWP which aims at developing a child's mathematical thinking. "Machine Guided Solution to MWP" by Suma Bhat et al., 2014 used this data-set. We will classify MWP on the same data-set. Dataset consists of three class of problems: JS, PPW and Compare. Details about this data-set is discussed along with classification.

8.3 Methodologies and Algorithm Details

8.3.1 Preprocessing

Data-set is available in JSON format. The data-set needs to be pre-processed before it is used in classification phase. Various pre-processing steps like sentence tokenization, word tokenization, POS tagging, Named Entity Recognition, Coreference Resolution, SVO triplet extraction etc are performed before feeding data-set for classification.
Sentence Tokenization is performed to find out problem length and this length serves as feature for classification phase. On an average JS problems has more number of sentences than Compare and Compare has more number of sentences than PPW.
POS tagging is used to find out discriminative and non-discriminative words. Some words like working, eating, speed, parking etc can be present in any problem type. So these words are non-discriminative and are compulsorily needed to be removed. If they remain there is a high chance of misclassification of problem type. POS tagger is used to eliminate nouns, verbs, modal auxiliaries, adverbs, adjectives, numerals, conjunctions, interjections etc that are all non-discriminative.
Stop-words removal and Stemming should not be performed as they are important for MWP classification. Stop-words like than, altogether are important for Compare and PPW problems type.But some stop-word like he,she ,it, they, you, how, what etc are non-discriminative. Due to above distinguishing indicative stop-words from non-distinguishing stop-words is an important step in MWP.
Coreference Resolution is used when a pronoun is used instead of a name or an entity. It links pronoun with name or entity which describes it the best. We prefer objects which are closer to the pronoun.Entities could consist of one or more words. Named Entity Recognition helps to find out those entities.

8.3.2 Learning And Inference

Reinforcement Learning : It is a remarkable technique which has proven to be powerful in solving control optimization problems. By control optimization problem we mean, problems where the task is to realize the best action to perform in every

state visited by system. In our case **realization of action means realization of the operation** to perform on two Q-sets from set of operators Op. In general RL is employed in cases where problem reflects complex stochastic structure.

Recurrent Neural Networks : It is popular Neural Network variant that considers the property of "sequential relations" and use incremental architecture to extract results. As text maintains the sequential behaviour RNN have shown promising results in the field of NLP. LSTMs(Long-Short term memory Networks) are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter and Schmidhuber (1997), and were refined and popularized by many people in following work.

Method : We present an approach based on the RNN(LSTMs) and Reinforcement Learning to solve single equation math word problems. We employ policy based RL to find the ***deterministic policy***.

Policy = f(q1, q2); f : Q-set $\times$ ***Q-set*** $\rightarrow$ ***Op***

Above function f determines the value of the operation (policy : Op = +, -, / , *, =). This also considers the verb vector embedded inside the definition of Q-set. Value function which acts as actor critic will determine the correction gradient.

gradient of a deterministic policy a= π***(s)***

$\delta L(u)/\delta u = E[(\delta \hat{Q}(s,a)/\delta a) * (\delta a/\delta u)]$

Bidirectional LSTMs are the most suitable choice for building such kind of models. As forward connections of LSTMs handle all the data coming from the previous node which will be of type compound_Q-set or base_Q-set. LSTM networks will either discard or collect the data which is required for the processing at the current node. Every node in the model employs ***combiner*** function which produces result of the current node and input for next one.

W = combiner(Q-set1 , Q-set2, Op);

combiner: Q-set $\times$ ***Q-set*** $\times$ ***Op*** $\rightarrow$ ***Q-set***

Reinforced Arrangement : It qualifies the correction gradient and if its value is nonzero it propagates this to resultant node which then uses backpropagation to correct its interpretation of policy function ***f***.

The training dataset also contains the equation which is then looked up by node of every iteration if mismatch is found then it corrects the bias of the current node and again calculates the result.

8.3.3 Algorithm

1. Accept a algebric word problem
2. Classify it into Join and Seperate problem or Part-Part-Whole problem or Comparison problem
3. Identify function of each sentence in the problem
4. Generate equations(with the help of Information Extraction)
5. Solve Equations
6. Display Solutions

9 Software Testing

9.1 Software Testing Tools

9.1.1 Unit Testing (unittest)

The Python unit testing framework, sometimes referred to as "PyUnit," is a Python language version of JUnit, by Kent Beck and Erich Gamma. Unit-test supports test automation, sharing of setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework. The unit-test module provides classes that make it easy to support these qualities for a set of tests. To achieve this, unit-test supports some important concepts:

1. test fixture:

   A test fixture represents the preparation needed to perform one or more tests, and any associate cleanup actions. This may involve, for example, creating temporary or proxy databases, directories, or starting a server process.

2. test case:

   A test case is the smallest unit of testing. It checks for a specific response to a particular set of inputs.unit-test provides a base class, Test-Case, which may be used to create new test cases.

3. test suite:

   A test suite is a collection of test cases, test suites, or both. It is used to aggregate tests that should be executed together.

4. test runner:

   A test runner is a component which orchestrates the execution of tests and provides the outcome to the user. The runner may use a graphical interface, a textual interface, or return a special value to indicate the results of executing the tests.

The standard work flow is:

1. You define your own class derived from unit-test.Test-Case.
2. Then you fill it with functions that start with 'test'.
3. You run the tests by placing unit-test.main() in your file, usually at the bottom.

Unit Testing with *unittest*

```
 """unitest testing suit"""
from data_preprocessing import nlp_pipeline
import unittest
class TestDataPreprocessing(unittest.TestCase):
.       def test_list_sentences(self):
.            que_string = "Mr.  Smith had 6 cookies.  Suzy gave
him 3 more cookies.  How many cookies does Mr.  Smith have
now?"
.            sent_list_expected = ["Mr.  Smith had 6 cookies.",
"Suzy gave him 3 more cookies.", "How many cookies does Mr.
Smith have now?"]
.            self.assertTrue(nlp_pipeline.list_sentences(que_str-
ing), sent_list_expected)
if __name__ == "__main__":
.     unittest.main()
```

9.1.2 Functional Testing (py.test)

The pytest framework makes it easy to write small tests, yet scales to support complex functional testing for applications and libraries. Due to pytest's detailed assertion introspection, only plain assert statements are used.

Features

1. Detailed info on failing assert statements (no need to remember self.assert* names);

2. Auto-discovery of test modules and functions;
3. Modular fixtures for managing small or parametrized long-lived test resources;
4. Can run unit-test (including trial) and nose test suites out of the box;
5. Python2.6+, Python3.3+, PyPy-2.3, Jython-2.5 (untested);
6. Rich plugin architecture, with over 150+ external plugins and thriving community;

Functional Testing with *py.test*

```
 """py.test testing suit"""
from data_processing import nlp_pipeline
import pytest
def test_list_sentences():
.     """Check if the function is working functionally fine or
not"""
.     que_string = "Mr.  Smith had 6 cookies.  Suzy gave him 3
more cookies.  How many cookies does Mr.  Smith have now?"
.     sent_list_expected = ["Mr.  Smith had 6 cookies.", "Suzy
gave him 3 more cookies.", "How many cookies does Mr.  Smith
have now?"]
.     assert nlp_pipeline.list_sentences(que_string) ==
sent_list_expected
```

10 Results

10.1 Result tables

- Comparison wrt time for various NLP toolkits in Python :

Toolkit	Word Tokenization	Sentence Tokenization	POS Tagging	Accuracy	Language
SPACY	Low	Medium	Very Low	More	Multiple Language
NLTK	Medium	Low	Very High	Less	English and German

Table 15: Comparison wrt time for various NLP toolkits in Python

- Clasification using Random Forest :

Algorithm	Accuracy
Baseline Classifier	62.47%
SVM	79.50%
Random Forest Classifier	86.67%

Table 16: Clasification using Random Forest

10.2 Screen shots

- Classification Result :

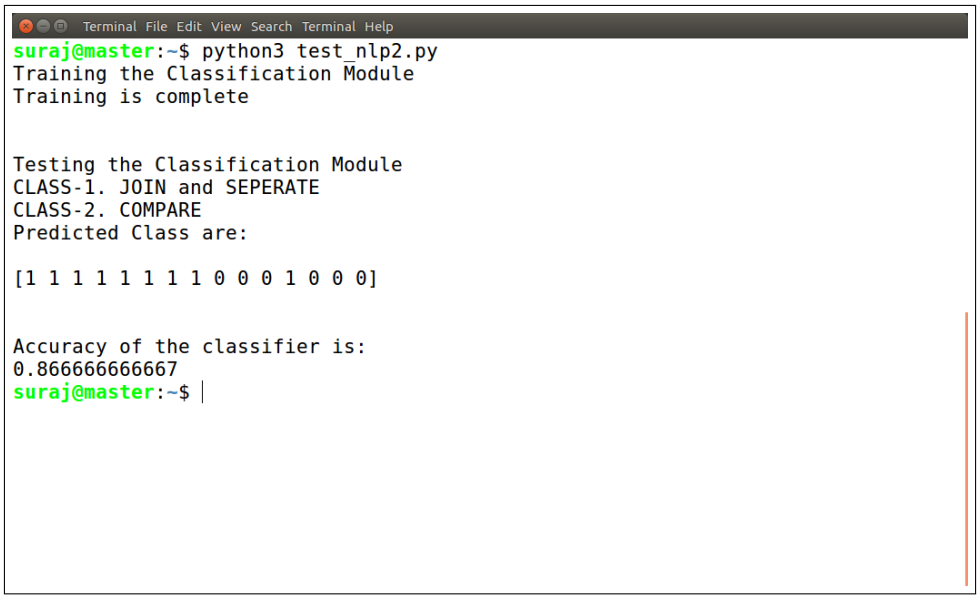


Figure 20: Result from Classifier

- Equation Generation Result :

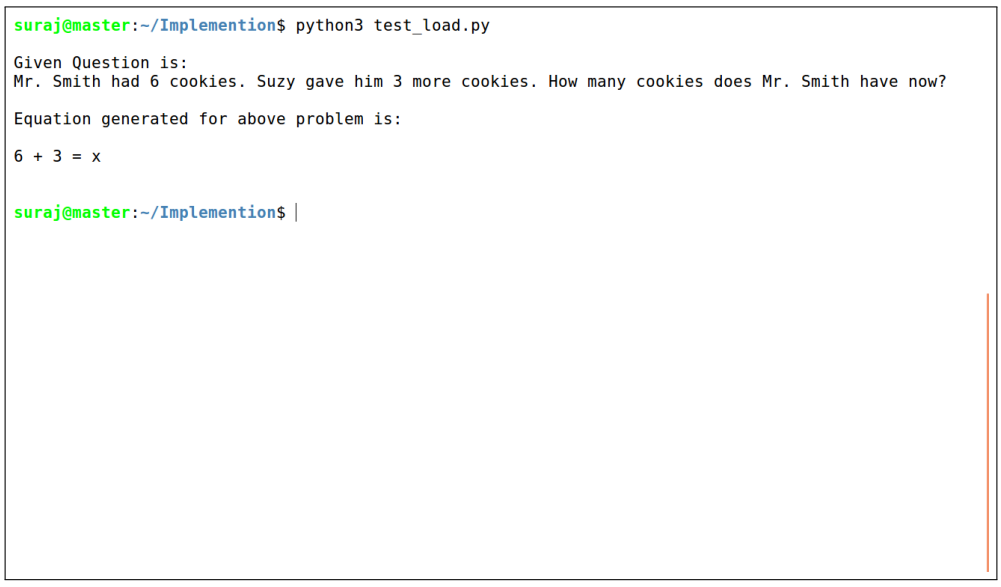


Figure 21: Equation Output

11 Conclusion and Future Scope

- We have built a system which is capable of solving single variable mathematical word problem of defined types.This system successfully classify,generate equation and solve them.
- The future of this system covers a very wide domain of the Artificial Intelligence system.We can extend the system to solve complex problem of multiple variable,geometry problems,calculus and the top most algorithmic problems.
- The system can be made for predicting the solution to the given problem.On top of that,it can be made to solve the complete problem itself.

Appendix A References

[1] Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014 - Learning to Automatically Solve Algebra Word Problems, In *proc. of the Annual Meeting of the Association for Computational Linguistics(ACL).*

[2] Mohammad Javad Hosseini , Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014 - Learning to Solve Arithmetic Word Problems with Verb Categorization, *In proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533 and *Association for Computational Linguistics (ACL).*

[3] Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015 - Parsing Algebraic Word Problems into Equations, In *proc. of the Annual Meeting of the Association for Computational Linguistics(ACL).*

[4] Suleyman Cetintas, Luo Si, Yan Ping Xin, Dake Zhang, Joo Young Park. 2009 - Automatic Text Categorization of Mathematical Word Problems, In *proc. of Association for the Advancement of Artificial Intelligence* (www.aaai.org)

[5] Bussaba Amnueypornsakul, Suma Bhat. 2014 - Machine-Guided Solution to Mathematical Word Problems, In *Proc. of Pacific Asia Conference on Language, information and Computation*

[6] Arindam Mitra, Chitta Baral. 2016 - Learning To Use Formulas To Solve Simple Arithmetic Problems, In *proc. of the Annual Meeting of the Association for Computational Linguistics(ACL)*

[7] Shuming Shi, Yuehui Wang, Chin-Yew Lin1, Xiaojiang Liu and Yong Rui. 2015 - Automatically Solving Number Word Problems by Semantic Parsing and Reasoning, In *proc. of the Conference on Empirical Methods in Natural Language Processing*

[8] Subhro Roy and Dan Roth. 2015 - Solving general arithmetic word problems, In *proc. of EMNLP.*

[9] Rik Koncel-Kedziorski, Subhro Roy , Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016 - MAWPS: A Math Word Problem Repository, In *proc. of NAALC*

[10] Ralph Grishman. 2015 - Information Extraction, In *New York University*

Appendix B Laboratory assignments on Project Analysis of Algorithmic Design

- Theory:

  Overall Idea: To design a system which accepts an algebraic word problem as an input and tries to classify it into pre defined domain (e.g. Part-Part- Whole, Join-Separate, Comparison, Equal Groups, Multiplicative-Compare etc.) using Machine Learning Techniques and NLP(Information Extraction). Later on more features like generating algebraic equations, methods to solve them could be added to the system. The main aim of project is to assist any people(Students, Teachers etc) to solve mathematical word problems. Proposed work has following modules:-

  1. Problem Type Classification
  2. Sentence Function Identification and Sign Prediction
  3. Equation Generate
  4. Equation Solver

- Feasibility Study:

  The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

  Three key considerations involved in the feasibility analysis are

  1. Economical Feasibility

     This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within

the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

S.no	Technology	How it will be used	Materials that need to be created	Time allocated	Responsible person in the group
1.	Python	Language to develop modules	for using different libraries of python	1 week	1.Gaurav Shukla 2.Sanket Deshmukh 3.Pranav Kanade 4.Suraj Kumar
2	NLTK/ Stanford NLP/ spacy	To develop classifier and equation	Classifier and Information Extractor generator	4 weeks	1.Pranav Kanade 2.Gaurav shukla
3	Scikit Learn/ Tensor Flow	For machine learning	Classifier Training	2 weeks	1.Suraj Kumar 2.Sanket Deshmukh
4	Texstudio	To develop report	Document creation	1 week	1.Gaurav Shukla 2.Sanket Deshmukh 3.Pranav Kanade 4.Suraj Kumar

Table 17: IDEA Matrix

2. Technical Feasibility

   This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have

a modest requirement, as only minimal or null changes are required for implementing this system.

3. Social Feasibility

   The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Appendix C Reviewers Comments of Paper Submitted

- **Name of the Conference/Journal where paper submitted : NCETCET'17**

1. Paper Title : Comparative Study of Approaches Used To Solve Mathematical Word Problems.
2. Paper accepted/rejected : Accepted

Two Days National Level Conference on
"Emerging Trends in Computer Engineering and Technology"
(NCETCET'17)
24th and 25th March 2017

Review Form of NCETCET'17

Paper ID	NCETCET-80
Paper Title	**Comparative Study of Approaches Used To Solve Mathematical Word Problems**

Evaluation:

Decision about paper is divided into following Categories , Select the proper option using Radio Button Image •

Categories	Poor	Fair	Good	Very Good	Outstanding
Originality				•	
Innovation			•		
Technical Merit				•	
Applicability				•	
Presentation And English		•			
Match To Conference Area			•		

Comments About Paper to Authors:

Comment 1 Affiliation should be written properly and also all authors must mention their mail id
Comment 2 Check grammatical corrections.
Comment 3 . Equations & Tabulation should be editable (No Image or Figure)
Comment 4 Not in format (Refer the IEEE Template and make correction i.e. margin, font size, reference style)
Comment 5 Cite the references in increasing numerical sequences [1,2,3 etc...] and list them accordingly under "references
Comment 6 Figures should be clearer and align properly

Track Under Paper is going to considered (Select Any One) :
- Track 1: Advanced Computing and Cloud. ☐
- Track 2: Computer Network Communications ☐
- Track 3: Informatics and Embedded System √ ☐

Figure 22: Reviewer Report from NCETCET'17

- **Name of the Conference/Journal where paper submitted : International Journal of Science and Research (IJSR)**

  1. Paper Title : Comparative Study of Approaches Used To Solve Mathematical Word Problems.
  2. Paper accepted/rejected : Accepted

International Journal of Science and Research (IJSR)

www.ijsr.net
ISSN (Online): 2319-7064

Reviewer Evaluation Report

Date: 03.03.2017

Paper ID: ART20171457

Paper Title: Comparative Study of Approaches Used To Solve Mathematical Word Problems

Reviewer Report

Evaluation Criteria	Score (0-10)
Relevance of Topic	8
Scholarly Quality	8
English Usage	7
Use of Theory	9
Novelty and Originality of the idea	8
Technical Content and Correctness	8
Critical Qualities	9
Clarity of Conclusions	7
Use / Quality of References Used	9
Other Aspects	8
Total Score	81

*75-100: Accept (No Modification Required), *55-75: Accept with Minor Revisions

*35-54: Accept with Major Revisions, *Below 35: Reject

Reviewer Decision

Accepted (No Modification Required)

Figure 23: Reviewer Report from IJSR

Appendix D Plagiarism Report

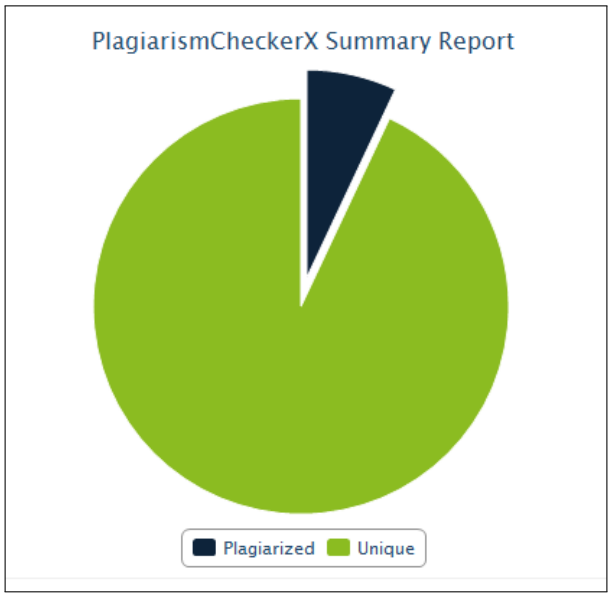


Figure 24: Plagiarism report of Literature survey

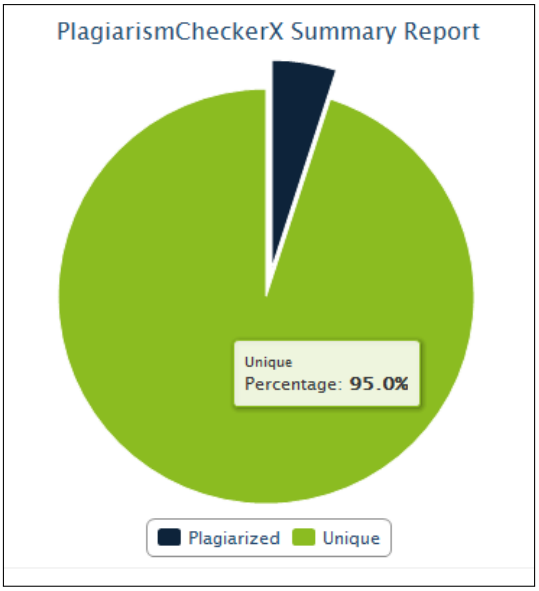


Figure 25: Plagiarism report of Proposed system

Appendix E Term-II Project Laboratory Assignments

E.1 Project workstation selection, installations along with setup and installation report preparations.

NUMPY: It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Installation Command:

```
$ sudo pip3 install numpy
```

SYMPY: It is a python library for symbolic mathematics. It aims to keep code as simple as possible in order to be comprehensive and easily extensible.

Installation Command:

```
$ sudo apt-get install python-sympy
```

SCIKIT-LEARN: It is a python library for machine learning algorithm.

Installation Command:

```
$ pip3 install scikit-learn
```

TENSORFLOW: TensorFlow is an open source software library for numerical computation using data flow graphs.

Installation Command:

1. For CPU:

```
$ pip3 install tensorflow
```

2. For GPU

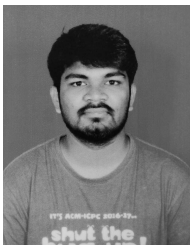
```
$ pip3 install tensorflow-gpu
```

SPACY: It is a library for advanced natural language processing in Python and Cython. It currently supports English and German, as well as tokenization for Chinese, Spanish, Italian, French, Portuguese, Dutch, Swedish, Finnish, Hungarian, Bengali and Hebrew.

Installation Command:

```
$ pip3 install spacy
```

Appendix F Information of Project Group Members



1. Name : Pranav Kanade
2. Date of Birth : 09/06/1995
3. Gender : Male
4. Permanent Address : 66-B, Ratnadeep Nagar, Chopda, Dist - Jalgaon
5. E-Mail : pkanade493513@gmail.com
6. Mobile/Contact No. : +919545581207
7. Placement Details : Persistent Systems Ltd. Pune

Department of Computer Engineering
In Association with
CSI (Computer Society of India)

MIT Academy of Engineering
An Autonomous Institute affiliated to Savitribai Phule Pune University

National Conference on Emerging Trends in Computer Engineering and Technology
(NCETCET'17)
24th-25th March 2017

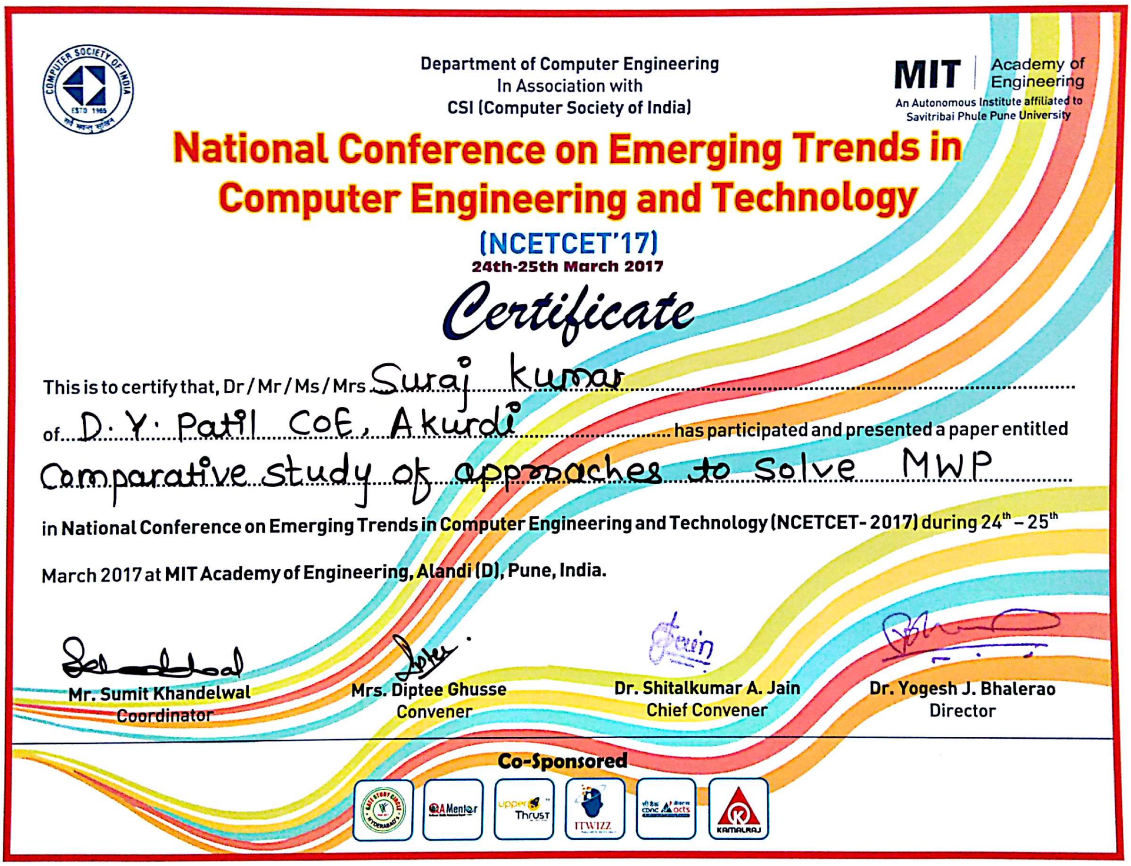
Certificate

This is to certify that, Dr / Mr / Ms / Mrs Pranav kanade
of D. Y. Patil COE, Akurdi has participated and presented a paper entitled
Comparative Study of approaches to solve MWP
in National Conference on Emerging Trends in Computer Engineering and Technology (NCETCET- 2017) during 24th – 25th March 2017 at MIT Academy of Engineering, Alandi (D), Pune, India.

Mr. Sumit Khandelwal, Coordinator
Mrs. Diptee Ghusse, Convener
Dr. Shitalkumar A. Jain, Chief Convener
Dr. Yogesh J. Bhalerao, Director

Co-Sponsored

1. Name : Suraj Kumar
2. Date of Birth : 18/08/1994
3. Gender : Male
4. Permanent Address : Khand Par, Sheikhpura, Bihar.
5. E-Mail : surajdypcoe@gmail.com
6. Mobile/Contact No. : 7757889510
7. Placement Details : Amdocs Inc.



Department of Computer Engineering
In Association with
CSI (Computer Society of India)

MIT Academy of Engineering
An Autonomous Institute affiliated to Savitribai Phule Pune University

National Conference on Emerging Trends in Computer Engineering and Technology
(NCETCET'17)
24th-25th March 2017

Certificate

This is to certify that, Dr / Mr / Ms / Mrs Suraj kumar
of D. Y. Patil CoE, Akurdi has participated and presented a paper entitled
Comparative study of approaches to solve MWP
in National Conference on Emerging Trends in Computer Engineering and Technology (NCETCET- 2017) during 24th – 25th March 2017 at MIT Academy of Engineering, Alandi (D), Pune, India.

Mr. Sumit Khandelwal, Coordinator
Mrs. Diptee Ghusse, Convener
Dr. Shitalkumar A. Jain, Chief Convener
Dr. Yogesh J. Bhalerao, Director

Co-Sponsored

1. Name : Gaurav Shukla
2. Date of Birth : 04/04/1995
3. Gender : Male
4. Permanent Address : S. No. 133/1 Gurudwara Chowk, Akurdi
5. E-Mail : gauravshuklapilani@gmail.com
6. Mobile/Contact No. : 8087153341
7. Placement Details : Cybage Software Pvt. Ltd.

Department of Computer Engineering
In Association with
CSI (Computer Society of India)

MIT Academy of Engineering
An Autonomous Institute affiliated to Savitribai Phule Pune University

National Conference on Emerging Trends in Computer Engineering and Technology

(NCETCET'17)

24th-25th March 2017

Certificate

This is to certify that, Dr / Mr / Ms / Mrs Gaurav Shukla
of D. Y. Patil COE, Akurdi has participated and presented a paper entitled
Comparative study of approaches to solve MWP
in National Conference on Emerging Trends in Computer Engineering and Technology (NCETCET- 2017) during 24th – 25th March 2017 at MIT Academy of Engineering, Alandi (D), Pune, India.

Mr. Sumit Khandelwal, Coordinator
Mrs. Diptee Ghusse, Convener
Dr. Shitalkumar A. Jain, Chief Convener
Dr. Yogesh J. Bhalerao, Director

Co-Sponsored

1. Name : Sanket Deshmukh
2. Date of Birth :13/10/1995
3. Gender : Male
4. Permanent Address :257,Kasaba peth,Phaltan-415523,Dist. Satara
5. E-Mail : sanket77deshmukh@gmail.com
6. Mobile/Contact No. :7875389516
7. Placement Details :TCS

Department of Computer Engineering
In Association with
CSI (Computer Society of India)

MIT Academy of Engineering
An Autonomous Institute affiliated to Savitribai Phule Pune University

National Conference on Emerging Trends in Computer Engineering and Technology
(NCETCET'17)
24th-25th March 2017

Certificate

This is to certify that, Dr / Mr / Ms / Mrs Sanket Deshmukh
of D. Y. Patil CoE, Akurdi has participated and presented a paper entitled
Comparative study of approaches to solve MWP
in National Conference on Emerging Trends in Computer Engineering and Technology (NCETCET- 2017) during 24th – 25th March 2017 at MIT Academy of Engineering, Alandi (D), Pune, India.

Mr. Sumit Khandelwal, Coordinator
Mrs. Diptee Ghusse, Convener
Dr. Shitalkumar A. Jain, Chief Convener
Dr. Yogesh J. Bhalerao, Director

Co-Sponsored

Appendix G Letter of Sponsorship

September 19, 2016

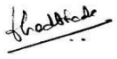
To Whomsoever it May Concern

This is to certify that following students from D.Y. PATIL COLLEGE OF ENGINEERING, AKURDI are undergoing their final year B.E. project at Persistent Systems Ltd. for academic year 2016-17 under the title 'AUTOMATIC CATEGORIZATION OF WORD PROBLEMS'

Name of Students:

i. SURAJ KUMAR
ii. KANADE PRANAV
iii. DESHMUKH SANKET
iv. SHUKLA GAURAV

For Persistent Systems Ltd.



Kaustubh Bhadbhade
Senior Manager - Human Resource