GROUP NO: 02

Project Report

on

Automatic Text Categorization of Word Problems

SUBMITTED BY

SURAJ KUMAR, B120084330

PRANAV KANADE, B120084257

SANKET DESHMUKH, B120084228

GAURAV SHUKLA, B120084237

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 **Mr. Suraj Kumar, Mr. Pranav Kanade, Mr. Sanket Deshmukh and Mr. Gaurav Shukla** have satisfactorily completed the project work entitled **"Automatic Text Categorization of Word Problems"** which is a bonafide work carried out by him under the supervision of **Mrs. Shanthi K. Guru** and it is approved for the partial fulfillment of requirement of Savitribai Phule Pune University, for the award of the degree of Bachelors of Engineering (Computer Engg.) for the academic year 2016-17 Sem-I.

Mrs. Shanthi K. Guru	Dr. Mrs. Neeta Deshpande
(Project Guide)	HOD (Computer)

Place: Pune

Date:

ACKNOWLEDGEMENT

With immense pleasure, we present the project report as part of the curriculum of the B.E. Computer Engineering. We wish to thank and express deep sense of gratitude to our guide Prof. Mrs. Shanti K. Guru for her consistent guidance, inspiration and sympathetic attitude throughout the project work. Without her this project could not have even been imagined. This project would not have been feasible without encouragement and guidance of our Project Co-ordinator, Mrs. Shanti K. Guru. We are heavily indebted to her. She patiently discussed ideas with us and gave indispensible suggestions.

We are grateful to Dr. Mrs. Neeta Deshpande, Head of Department of Computer Engg., D.Y.P.C.O.E. for always being ready to help with the most diverse problems that we have encountered along the way. We express our sincere thanks to all our staff and colleagues who have helped us directly or indirectly in completing this project.

Last but not the least, We thank Persistent Systems Ltd. who sponsored this project and had immense faith in us.

SURAJ KUMAR	B120084330
PRANAV KANADE	B120084257
SANKET DESHMUKH	B120084228
GAURAV SHUKLA	B120084237

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 Seperate problem or Part-Part-Whole problem or Compare problem. After clssification 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

1 Problem Definition

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

1.1 Problem Outcome

The automatic problem solver defines following outcomes:

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

1.2 Motivation

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 algebric problems and thus in this way we got motivated for the proposed system.

2 Literature Survey

2.1 Information Extraction - By Ralph Grishman

Key Points - Named Entities, Syntactic Analysis, Semantic Analysis, Coreference 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 able to build the example in Table below.

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

IE is generally organized as the product of several analysis components: named entity (NE) tagging; syntactic analysis; [within-document] coreference resolution; entity, relation, and event extraction (semantic analysis); and cross-document coreference resolution. This decomposition is convenient for describ- ing 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 be- ing 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 cap- ture 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 speak- ing, richer analysis at this stage simplifies the semantic analysis that follows. At a minimum, we perform partial parsing or chunking during syntac- tic analysis. This divides the input into a sequence of chunks, nonrecursive linguistic units such as noun groups and verb groups.

Coreference Resolution- If an individual is mentioned several times, later mentions generally won't repeat the full name. Thus, in the exam- ple 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 coreference resolution to link all mentions of an individual within a doc- ument 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.

2.2 Machine-Guided Solution to Mathematical Word Problems - By Bussaba Amnueypornsakul, Suma Bhat

Key Points– Problem Classification(Part-Part Whole Problems, Compare Problems, Join-Seperate Problems), Equation Generation , Equation Solver etc.

This paper describes various types of problerms like join-seperate 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 informtion extraction. Finally these equtions 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 prob- lem 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 lan- guage that describe changes in the possession or condi- tion 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.

2.3 Learning to Automatically Solve Algebra Word Problems - By Nate Kushman , Yoav Artzi , Luke Zettlemoyer, and Regina Barzilay

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

This system reasons across sentence boundaries to construct and solve a system of linear equations, while simultane- ously 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 num- bers and nouns from the text. During inference we consider these two steps jointly. The

Derivation 1		
Word problem	An amusement park sells 2 kinds of tickets. Tickets for children cost \$ 1.50 . Adult tickets cost \$ 4 . On a certain day, 278 people entered the park. On that same day the admission fees collected totaled \$ 792 . How many children were admitted on that day? How many adults were admitted?	
Aligned template	$u_1^1 + u_2^1 - n_1 = 0$	$n_2 \times u_1^2 + n_3 \times u_2^2 - n_4 = 0$
Instantiated equations	$x + y - 278 = 0$	$1.5x + 4y - 792 = 0$
Answer	$x = 128$ $y = 150$	

Figure 2: Example of MWP

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".

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

Topics - Verb Categorization, Equation Genarator 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.

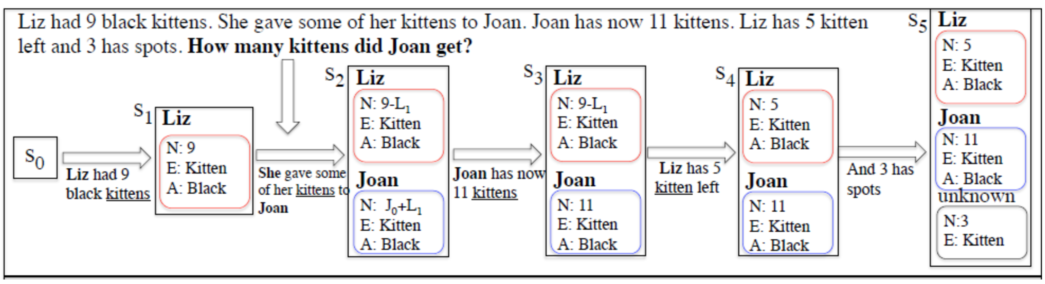

Figure 3: States of a MWP

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

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

3 Software Requirements Specification

3.1 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.

3.1.1 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 Seperate Problems
2. Part-Part-Whole problems
3. Comparison Problems

3.1.2 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

3.1.3 Operating Environment

For Automatic Problem Solver, operating environments are:

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

3.1.4 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

3.1.5 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

3.2 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)

3.2.1 Solver to MWP(Functional Requirements)

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

3.2.2 Step by step representation of solution(Functional Requirements)

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

3.2.3 Web interface to solver(Functional Requirements)

1. Deploying the solver to web using Web Technologies.

3.3 External Interface Requirements

3.3.1 User Interfaces

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

3.3.2 Hardware Interfaces

1. INTEL CPU
2. GPU(NVIDIA) for parallel processing(8 GB)
3. RAM (16 GB)

3.3.3 Software Interfaces

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

3.3.4 Communication Interfaces

Web interface is supported across all web browsers.

3.4 Non-functional Requirements

3.4.1 Performance Requirements

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

3.4.2 Safety Requirements

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

3.4.3 Security Requirements

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

3.4.4 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 sytem if needed.

4. USABILITY: System should solve almost all problems of given domains.

3.5 Analysis Models

Various diagrams of the system like Data Flow Diagrams, Class Diagrams, State-transition Diagrams are useful to analyze the system functionaly and implementation details.

3.5.1 Data Flow Diagrams

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

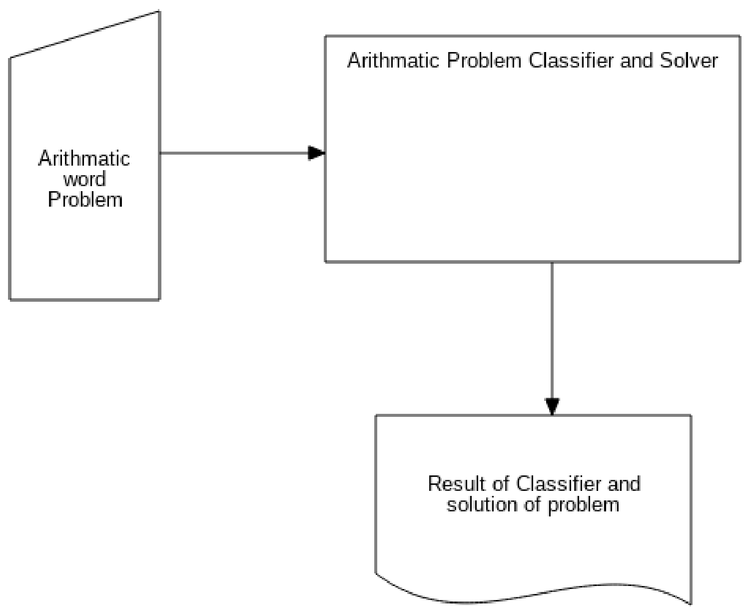

Figure 4: Level-0 DFD

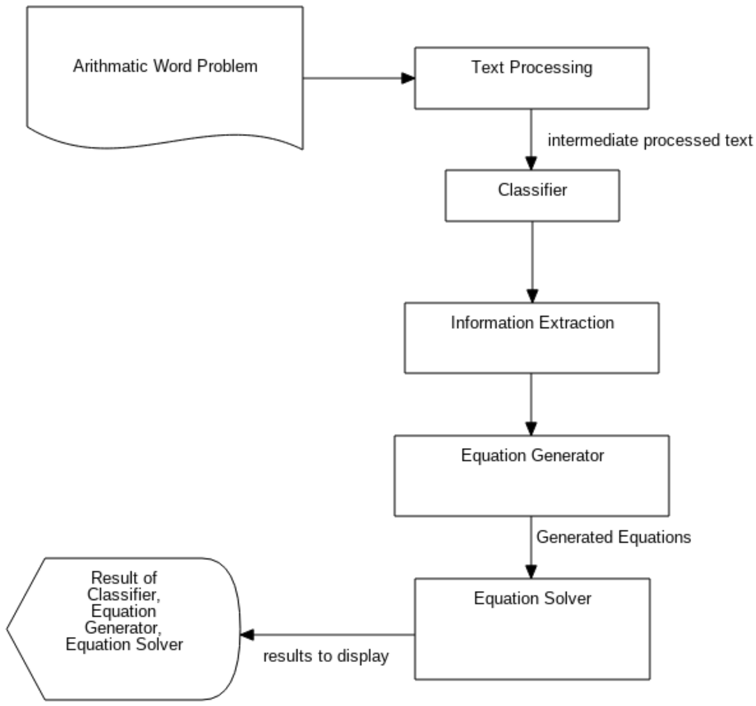

Figure 5: Level-1 DFD

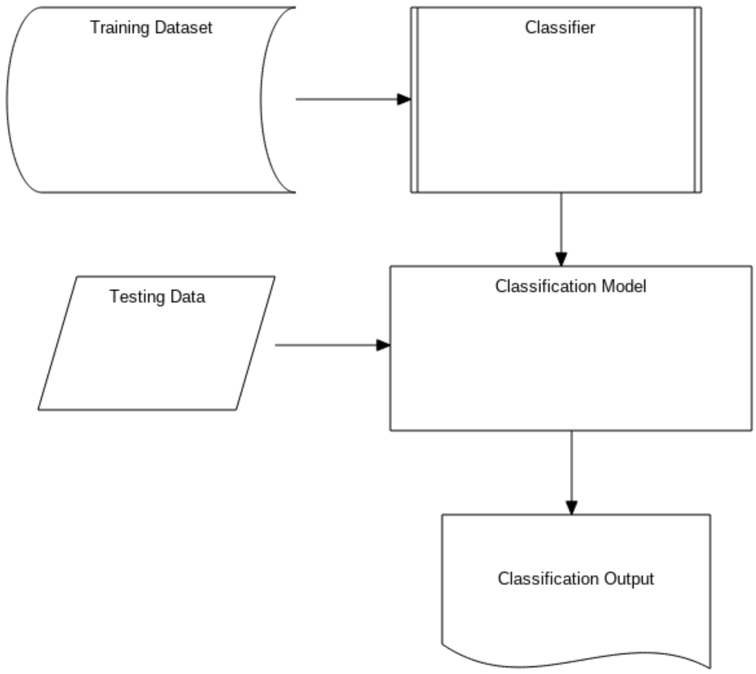

Figure 6: Level-2 DFD

3.5.2 Class Diagrams

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

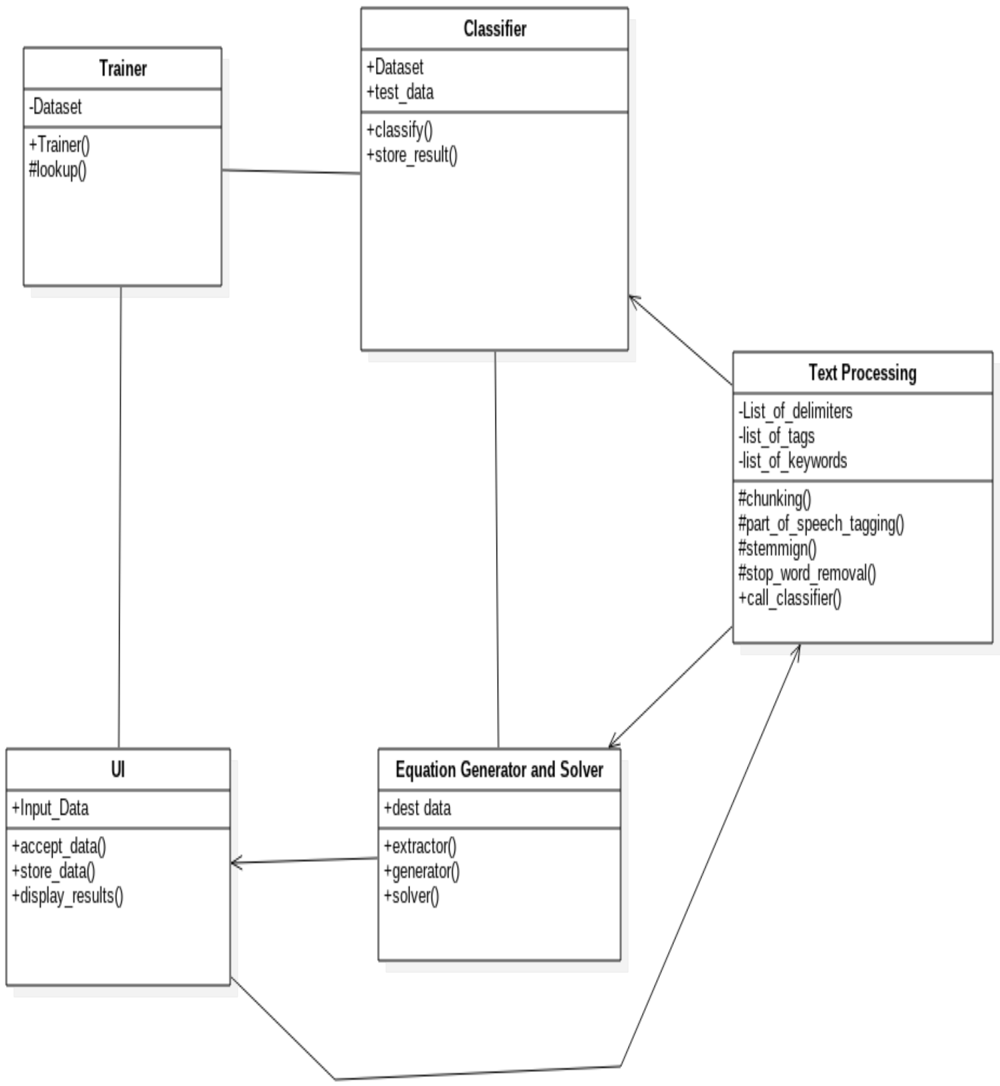

Figure 7: Class Diagram

3.5.3 State-transition Diagrams

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

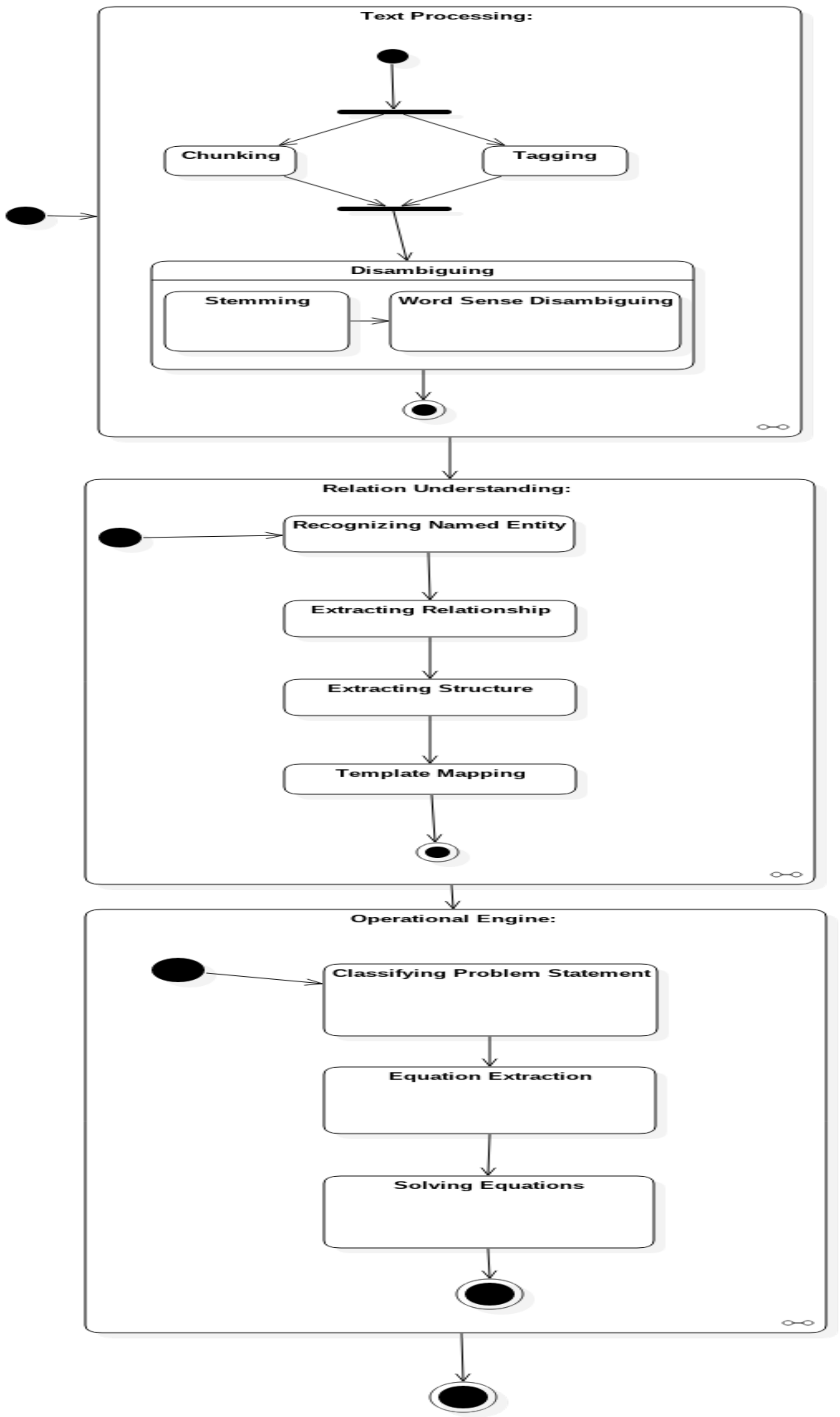

Figure 8: State Transition Diagram

3.6 System Implementation Plan

Table 1: System Implementation Plan

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
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	2n Week	Final University Examination

4 System Design

4.1 System Architecture

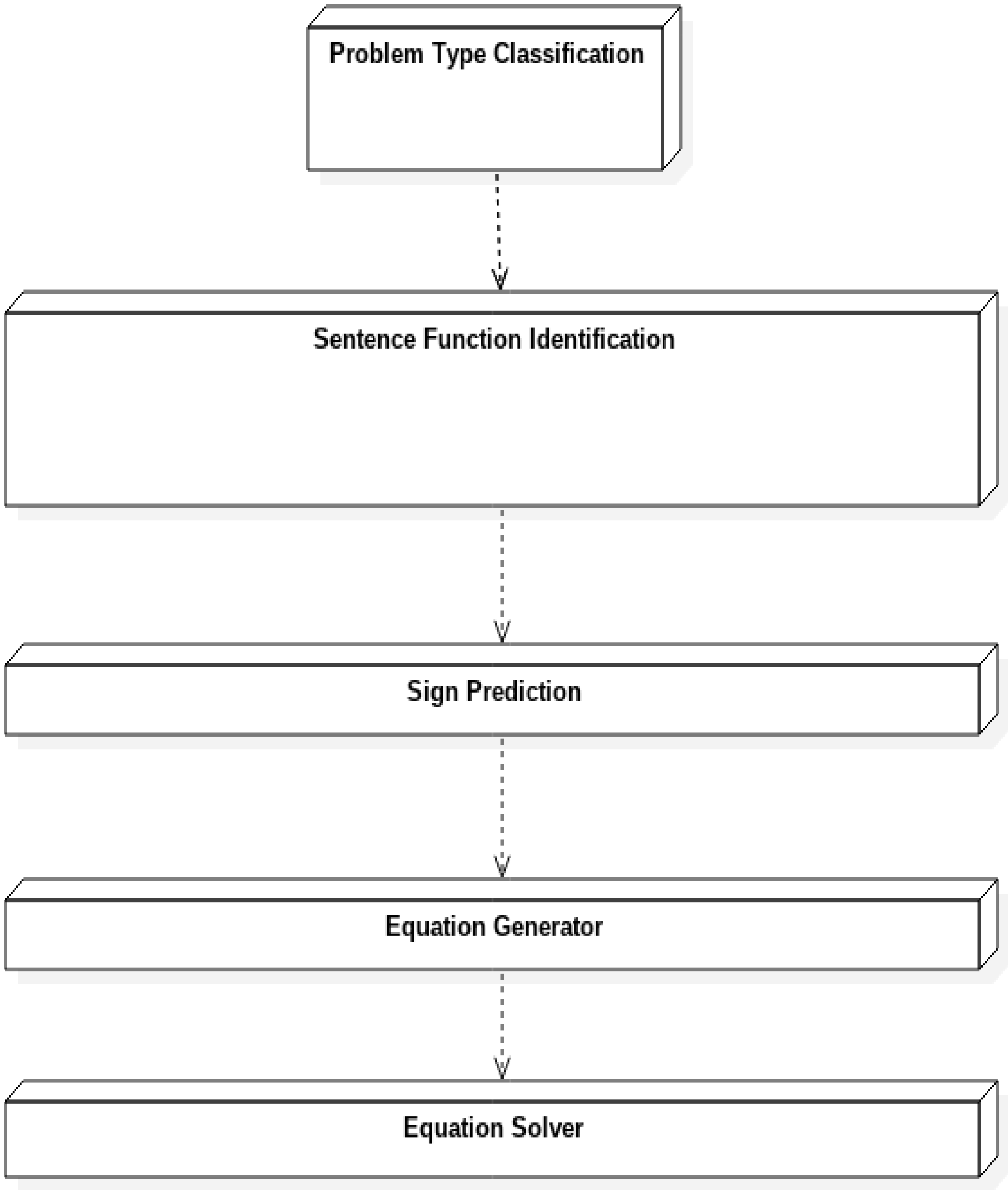

Figure 9: System Architecture

4.2 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

4.3 UML Diagrams

4.3.1 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system.

4.3.2 Use Case Diagram

Use case diagram is dynamic in nature and there should be some internal or external factors for making the interaction.These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application.

4.3.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

4.3.4 Package Diagram

Package diagram is UML structure diagram which shows packages and dependencies between the packages.

4.3.5 Component Diagram

Component diagrams are different in terms of nature and behaviour. Component diagrams are used to model physical aspects of a system. Now the question is what are these physical aspects? Physical aspects are the elements like executables, libraries, files, documents etc which resides in a node.

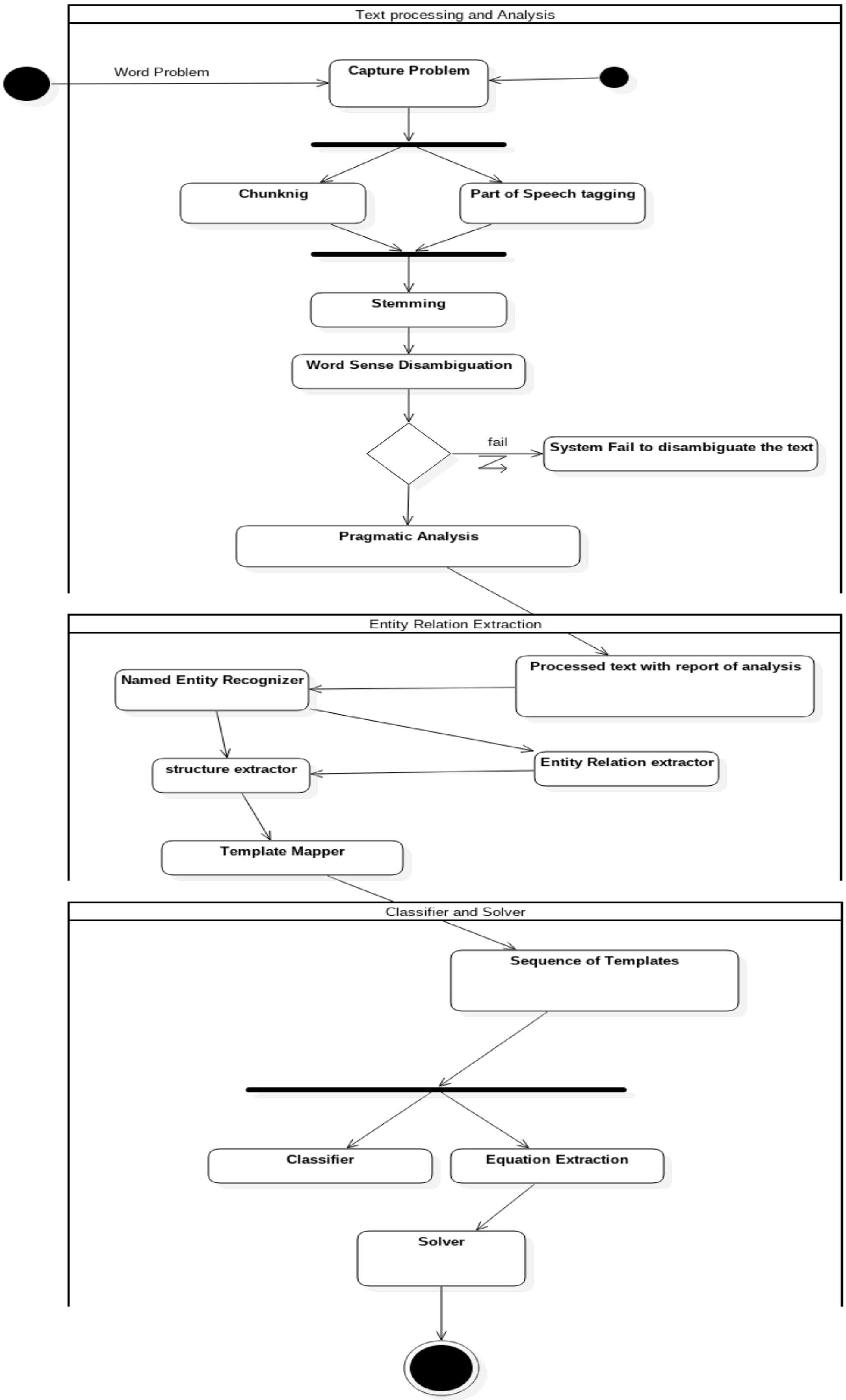

Figure 10: Activity Diagram

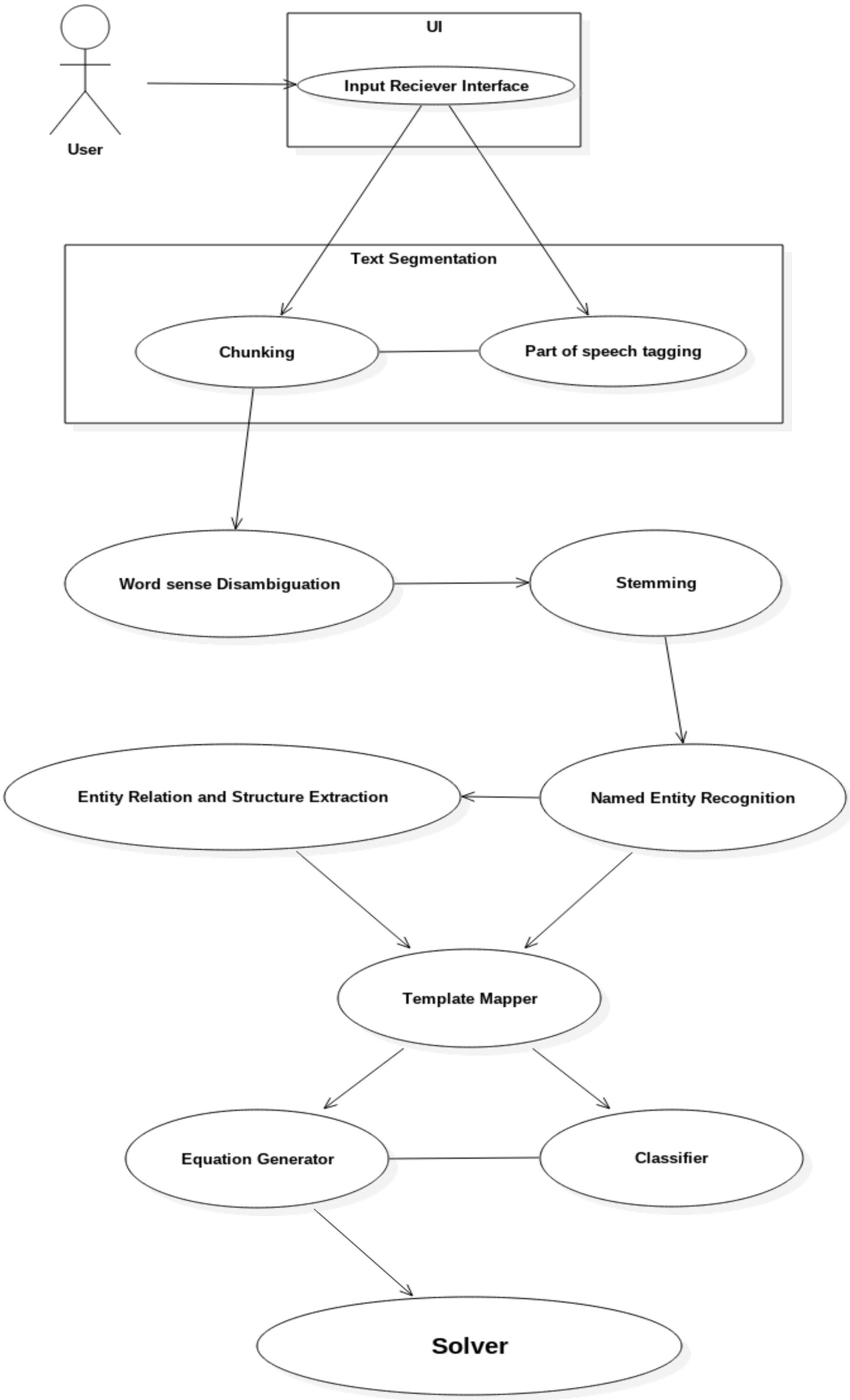

Figure 11: Use Case Diagram

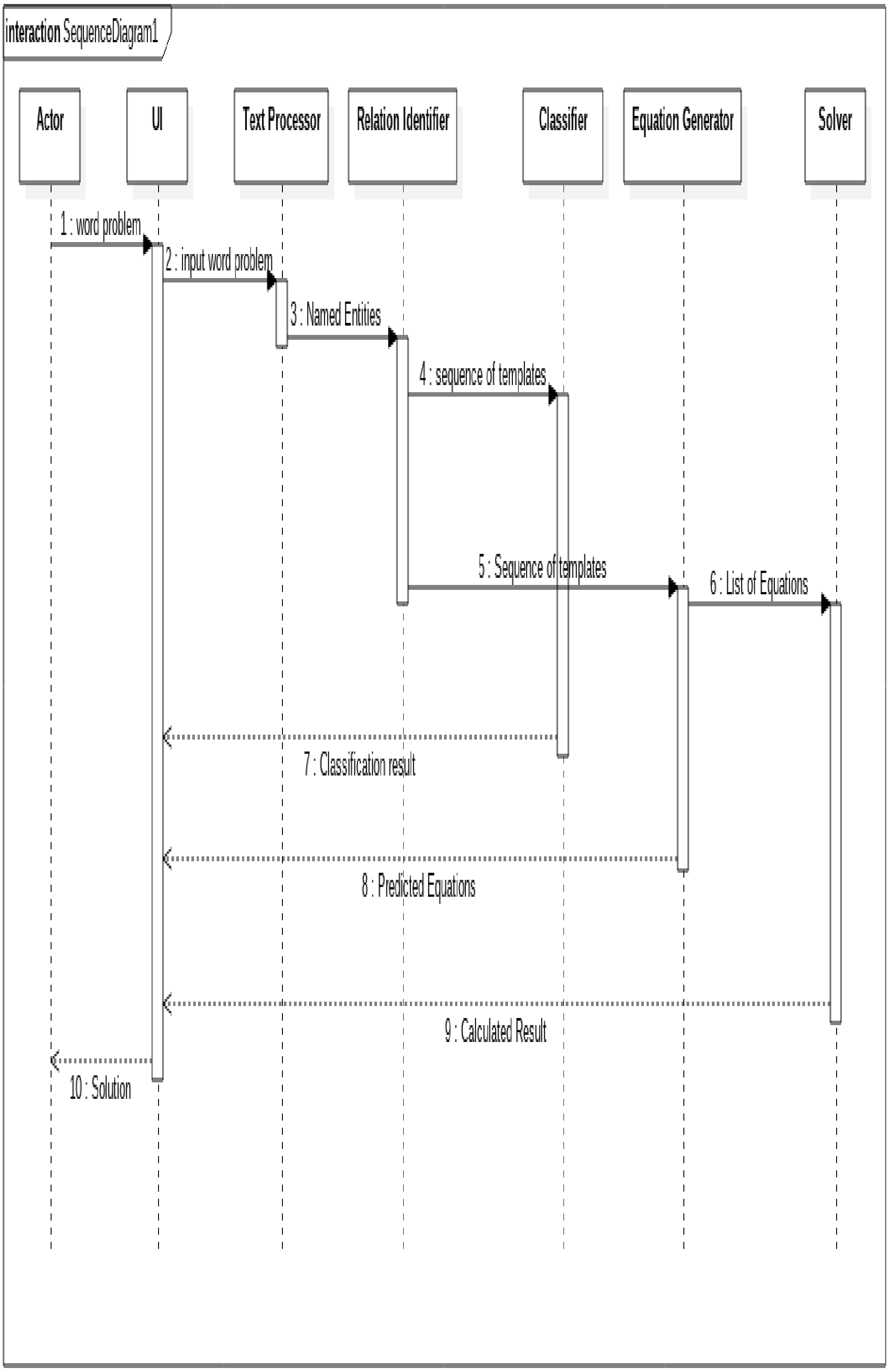

Figure 12: Sequence Diagram

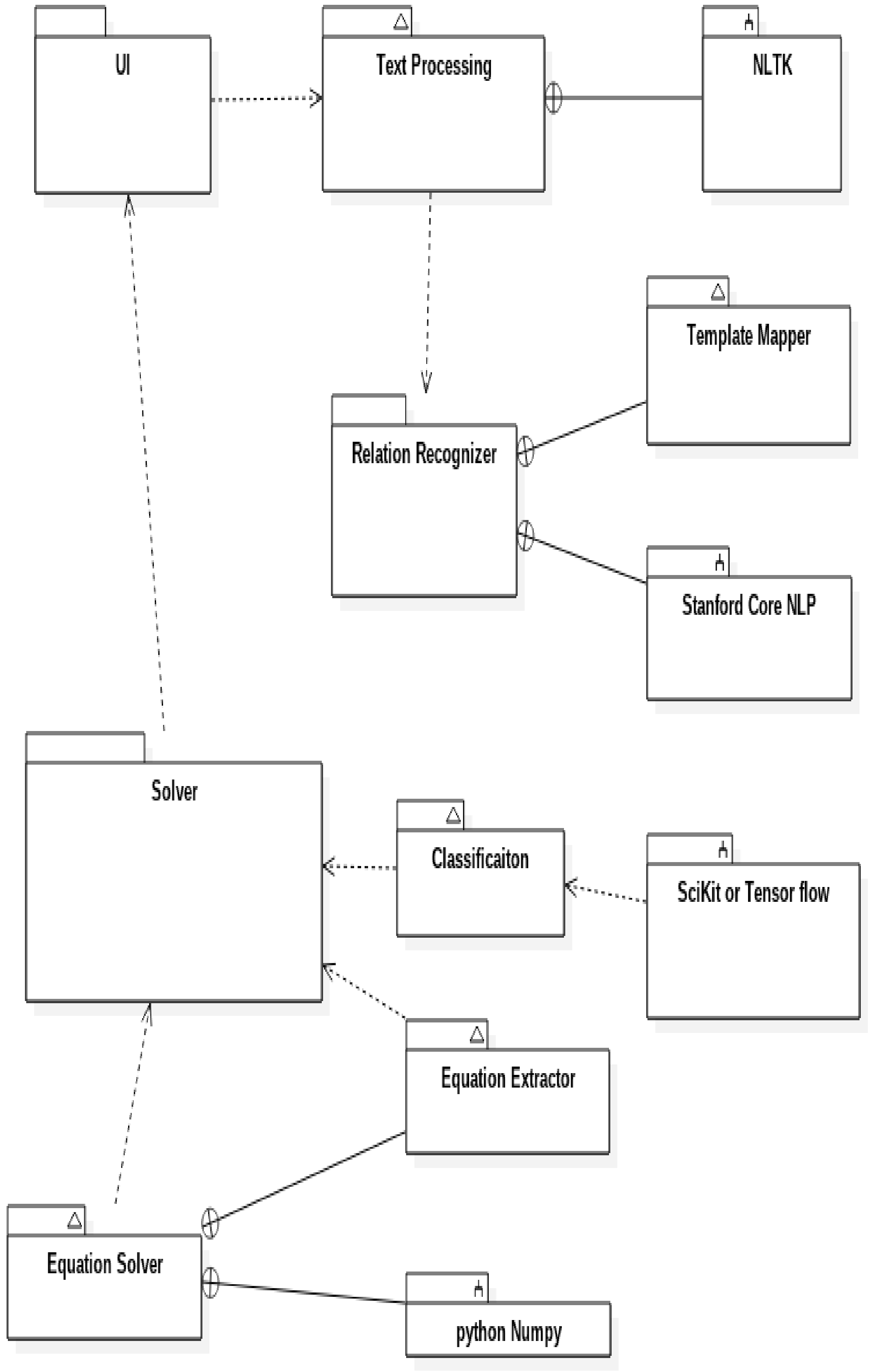

Figure 13: Package Diagram

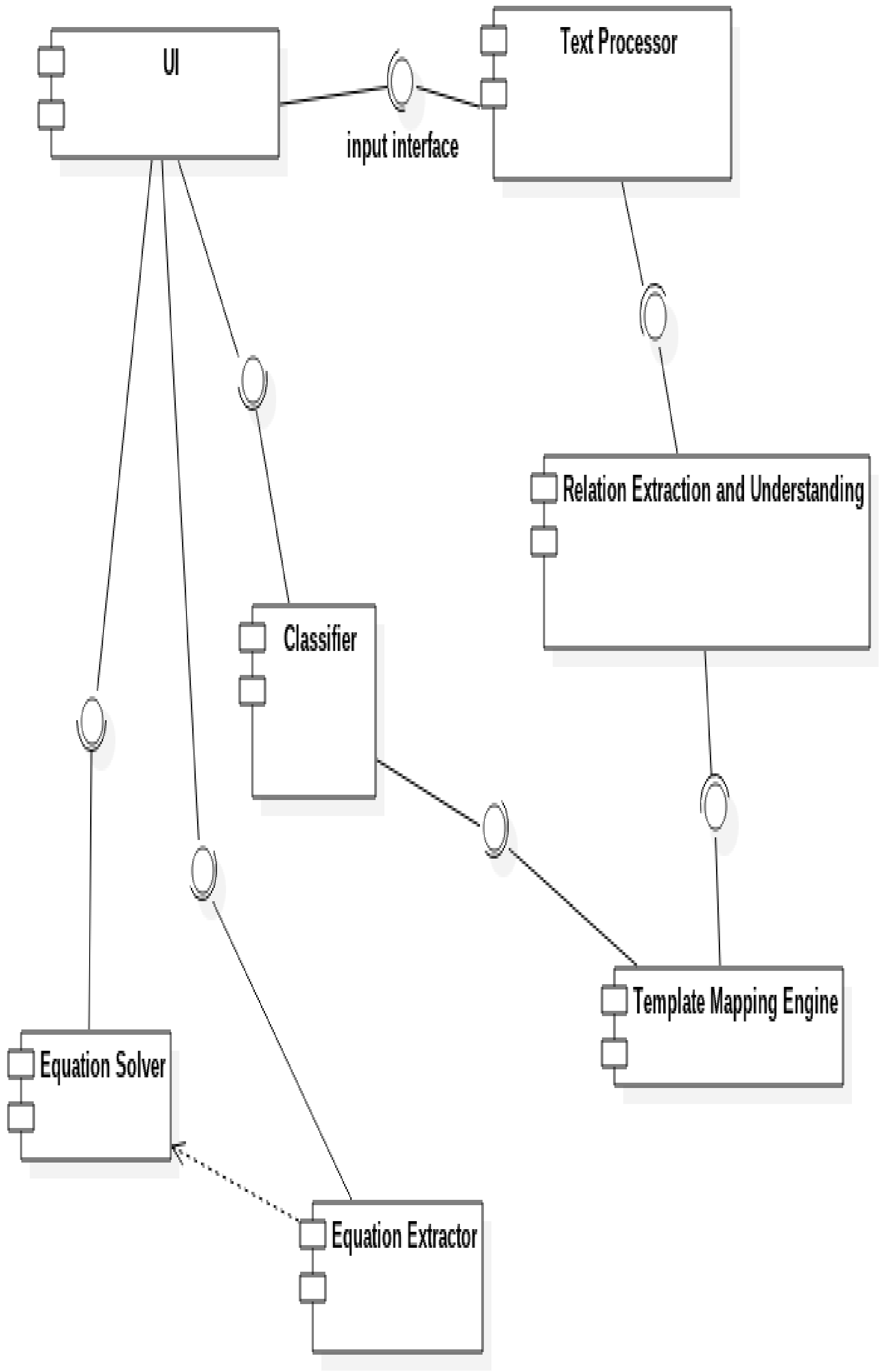

Figure 14: Component Diagram

5 Technical Specification

5.1 Advantages

1. Assists students to solve MWP
2. Assists cognitive scientists to understand problem solving in children
3. More similar systems could be developed in future using analogy like automatic algorithm and data structure prediction for computer problems.
4. It covers in detail the concept of Classification and Information Extraction which is useful in many NLP projects.

5.2 Disadvantages

1. Chances of wrong classification of problem type for a MWP
2. Chances of wrong equation generation
3. Chances of wrong sign prediction in Join and Seperate problems

5.3 Applications

1. Intelligent Tutoring Systems
2. To study how children solve elementary MWP by Cognitive Scientists
3. Web based assistant for MWP

6 Appendix A: Assignments

7 Appendix B: Glossary

List of Acronyms:

1. MWP - Mathematical Word Problem
2. NN - Neural Network
3. SVM - Support Vector Machine
4. NER - Named Entity Recognition
5. IE - Information Extraction
6. UML - Unified Modelling Language

8 References

[1] Ralph Grishman, "Information Extraction" in 1541-1672/15/2015 IEEE

[2] Bussaba Amnueypornsakul, Suma Bhat, - University of Illinois, Urbana Champaign, "Machine-Guided Solution to Mathematical Word Problems" in 28th Pacific Asia Conference on Language, Information and Computation pages 111–119

[3] Nate Kushman , Yoav Artzi , Luke Zettlemoyer, and Regina Barzilay, Computer Science and Articial Intelligence Laboratory, Massachusetts Institute of Technology, Computer Science and Engineering, University of Washington, " Learning to Automatically Solve Algebra Word Problems"

[4] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni and Nate Kushman, University of Washington, Allen Institute for AI, Massachusetts Institute of Technology, "Learning to Solve Arithmetic Word Problems with Verb Categorization"

[5] Suleyman Cetintas, Luo Si, Yan Ping Xin, Dake Zhang, Joo Young Park, Department of Computer Science, Department of Statistics, Department of Educational Studies Purdue University, West Lafayette, IN, 47906, USA , "Automatic Text Categorization of Mathematical Word Problems"

[6] Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu and Yong Rui, "Automatically Solving Number Word Problems by Semantic Parsing and Reasoning" in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1132–1142, Lisbon, Portugal, 17-21 September 2015

[7] Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang, "Parsing AlgebraicWord Problems into Equations" in Transactions of the Association for Computational Linguistics, vol. 3, pp. 585–597, 2015. Action Editor: Regina Barzilay.

[8] S.R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, Regina Barzilay, ”Reinforcement Learning for Mapping Instructions to Actions”

[9] Arindam Mitra, Chitta Baral, ”Learning to Use Formulas To Solve Simple Arithmetic Problems” in Proceddings of the 54th Annual Meeting of the Association of Computational Linguistics, 2016.

9 Sponsorship letter

PERSISTENT

September 19, 2016

<u>To Whomsoever it May Concern</u>

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'

<u>Name of Students:</u>

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

For Persistent Systems Ltd.

Kaustubh Bhadbhade

Figure 15: Sponsorship letter