INSTRUCTION CLASSIFICATION USING

NATURAL LANGUAGE PROCESSING

A PROJECT REPORT

Submitted by

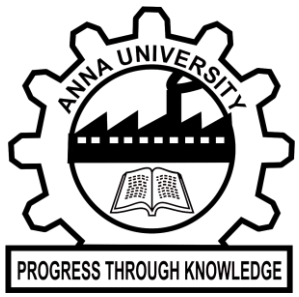
DAMODHARAN R 2019115028

SRI SIVA MURUGAN V 2019115103

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment for the award of the degree
of
BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY



DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

DECEMBER 2021

BONAFIDE CERTIFICATE

Certified that this project report titled INSTRUCTION CLASSIFICATION USING NATURAL LANGUAGE PROCESSING is the bonafide work of Damodharan R (2019115028) , Sri Siva Murugan V (2019115103) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: CHENNAI
DATE:

DR. S. BAMA
ASSISTANT PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. SASWATI MUKHERJEE
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Robots play a major role in the advancement of technology. What we feed a robot to make it do a task is an instruction. An instruction is a detailed information on how you should do something, use something, etc. In this project we identify instructions from a given passage and then classify those instructions into 8 custom types.

Simple instruction, negative instruction, instruction with purpose, instruction in sequence, mandatory instruction, optional instruction, instruction with reason, and instruction with tool are the eight types of instruction. Thus this project will help you to quickly go through a paragraph and comprehend, by filtering out the instructions and categorizing them into eight types.

சுருக்கம்

தொழில்நுட்ப வளர்ச்சியில் ரோபோக்கள் முக்கிய பங்கு வகிக்கின்றன. ஒரு ரோபோவை ஒரு பணியைச் செய்ய நாம் ஊட்டுவது ஒரு அறிவுறுத்தலாகும். அறிவுறுத்தல் என்பது நீங்கள் எதையாவது எப்படிச் செய்ய வேண்டும், எதையாவது பயன்படுத்த வேண்டும், போன்றவற்றைப் பற்றிய விரிவான தகவலாகும். இந்தத் திட்டத்தில் கொடுக்கப்பட்ட பத்தியில் உள்ள வழிமுறைகளைக் கண்டறிந்து, அந்த வழிமுறைகளை 8 தனிப்பயன் வகைகளாக வகைப்படுத்துகிறோம்.

எளிய அறிவுறுத்தல், எதிர்மறை அறிவுறுத்தல், நோக்கத்துடன் கூடிய அறிவுறுத்தல், வரிசையில் அறிவுறுத்தல், கட்டாய அறிவுறுத்தல், விருப்ப அறிவுறுத்தல், காரணத்துடன் கூடிய அறிவுறுத்தல் மற்றும் கருவி மூலம் அறிவுறுத்தல் ஆகியவை எட்டு வகையான அறிவுறுத்தல்கள். இவ்வாறு, வழிமுறைகளை வடிகட்டி, எட்டு வகைகளாகப் பிரிப்பதன் மூலம், ஒரு பத்தியை விரைவாகச் சென்று புரிந்துகொள்ள இந்தத் திட்டம் உதவும்

ACKNOWLEDGEMENT

I wish to record my deep sense of gratitude and profound thanks to my project supervisor **Dr. S.Bama** , Assistant Professor, Department of Information Science and Technology, College of Engineering, Guindy for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this project into fruition.

I am extremely indebted to **Dr. Saswati Mukherjee**, Professor and Head of the Department of Information Science and Technology, Anna University, Chennai, for extending the facilities of the Department towards my project and for her unstinting support.

I express my sincere thanks to the Faculty In-charge **Ms. C. M. Sowmya**, Teaching Fellow, Department of Information Science and Technology, Anna University, Chennai, for her reviews throughout the course of my project.

I thank my parents, family, and friends for bearing with me throughout the course of my project and for the opportunity they provided me in undergoing this course in such a prestigious institution.

DAMODHARAN R
SRI SIVA MURUGAN V

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
POS	Parts Of Speech
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
TF-IDF	Term Frequency–Inverse Document Frequency
SVM	Support Vector Machines
VADER	Valence Aware Dictionary for Sentiment Reasoning

CHAPTER 1

INTRODUCTION

We read a lot of passages every day, and some of them may be helpful as a guide to perform something, such as how to act in an examination hall. The length of these paragraphs might range from a few lines to many pages. We can't skip these sections since they're so essential. Have you ever been in a scenario where you needed to read through a guide booklet as quickly as possible while still making sure you didn't miss any key lines? Then you must focus purely on the instructions in those paragraphs. Instructions are the sentences that tell you whether or not to do something. As a result, the focus of this project is on finding the instructions in the given passage.

This project isn't over yet. It also categorizes the instruction into eight different types. These eight types are simple instruction, negative instruction, instruction with purpose, instruction in sequence, optional instruction, mandatory instruction, instruction with reason and instruction with tool. It will be easier to spot a specific instruction from a text if the instruction is classified.

Natural Language Processing is used to identify and categorize instructions. Natural language processing (NLP) is a branch of computer science, and more specifically, a branch of artificial intelligence (AI), concerned with teaching computers to interpret text and spoken words in the same manner that humans do. NLP integrates statistical, machine learning, and deep learning models with computational linguistics rule-based modeling of human language. These technologies, when used together, allow computers to interpret human language in the form of text or speech data and understand its full meaning, including the speaker's or writer's purpose and mood.

1.1 PROBLEM STATEMENT

One method of summarizing a paragraph is to identify the instructions included within it. The importance of instructions is that they inform you what to do and what not to. By reading these instructions, one can quickly skim through the passage. Even though the instructions are identified, it will be more convenient if they are classified into various types according to the need. As a result, our problem statement is to detect the instructions in a section and classify them into eight different types in order to help people comprehend them better.

1.2 OBJECTIVE

Our objective is to identify the sentences which are instructions from a given passage and classifying these instructions into the following eight different types using natural language processing.

1. Simple instruction
2. Negative instruction
3. Instruction with Purpose.
4. Instruction in Sequence.
5. Mandatory Instruction
6. Optional Instruction
7. Instruction with Reason
8. Instruction with Tool

1.3 MOTIVATION

Imagine hurrying to an exam hall to write an exam and suddenly noticing the bulletin board where the exam guide is placed, and it extends for several pages; you don't have time to read it all, but you can't ignore it too. In that case, it will be extremely beneficial to you if

only the most crucial sentences are taken out. We created the instruction identifier to deal with such scenarios. You may now quickly skim through the instructions. We have also divided the instructions into eight separate categories, which helps you to narrow your search to a certain piece of information. As an example, you may use the ‘instruction with tool’ to check what tool you will need to write the test.

1.4 NOVELTY IN THE PROJECT

Many NLP projects have already been accomplished, such as stop word detection, email spam detection, sentiment analysis, and so on, but no one has yet tried to identify the instructions from a series of texts and classified them into these eight types. Hence, this concept is our innovation.

CHAPTER 2

LITERATURE SURVEY

This chapter deals about the existing work carried out in the field of Natural Language Processing which makes use of the POS tagger which plays a crucial role in our project.

2.1 BLOGSPOT

2.1.1 Text Classification with Flair

https://blog.jcharistech.com/2020/10/04/text-classification-with-flair-pytorch-nlp-framework/

This blogspot sheds light on how to use Flair for text classification. They have built a model to predict or classify text as either offensive or non-offensive.

2.1.2 Sentiment Analysis in Python

https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair

This blog focuses on classification of text, where the given input text is classified into positive, neutral, or negative sentiment. It explains how to build a sentiment analysis model from scratch using TensorFlow. At the end, comparison of accuracies produced by using various algorithms like VADER, Flair and Textblob has also been shown.

2.2 WEBSITE

2.2.1 Text Classification

https://github.com/kk7nc/Text_Classification#word-embedding

This github repository provides an overall view on various aspects of text classification like tokenization, spelling correction, word frequency, etc.

2.2.2 Text Summarization

https://github.com/akashp1712/nlp-akash/tree/master/text-summarization

This github repository demonstrates the summarization of a given text, using word frequency, and TF-IDF algorithm, and fetching top sentences using TextRank algorithm.

2.3 RESEARCH PUBLICATION

2.3.1 Article classification using natural language processing and machine learning

https://ieeexplore.ieee.org/document/9044227

The above research paper describes classification of a list of articles into different topics, using text classification, natural language processing, and machine learning algorithms like Naive Bayes, & K-Nearest neighbors.

2.3.2 Clinical Report Classification Using Natural Language Processing and Topic Modelling

https://ieeexplore.ieee.org/document/6406751

This research explains how NLP can be used to analyze the Computed Tomography (CT) reports and other clinical reports, to aid in medical decision making. Here text classification is done by SVM and decision tree algorithm.

2.3.3 Survey on Parts of speech Tagger Techniques

https://ieeexplore.ieee.org/document/8550884

This paper discusses the survey of different types of POS taggers designed by the researchers and organizations.

2.3.4 The Effect of POS Tag Information on Sentence Boundary Detection in Turkish Texts

https://ieeexplore.ieee.org/document/8554031

Here, a corpus of texts are analysed and checked whether it is a sentence, non-sentence or end of sentence using POS tagger.

2.4 CONCLUSION FROM LITERATURE SURVEY

Various research papers and blogspots from the last few years were studied and a conceptual understanding of the same was developed. By going through all these resources, knowledge about various techniques used in Natural Language Processing was gained. On reading the above research works and studying the code, it is evident that identification of instructions from a given passage and classifying those instructions into such types haven’t

been done so far. This proves that our project is rich with novelty and hasn't been done before.

CHAPTER 3

SYSTEM DESIGN

Figure 3.1 depicts the overall architecture diagram of the project. The flow of the project is discussed below

1. Start
2. Read input passage from the user
3. Tokenize the passage into sentences
4. Iterate through the list of sentences
    a. If the sentence is an instruction
        i. Classify it into one of the 8 types
    b. Else
        i. Go to the next sentence
5. Display the list of instructions
6. End

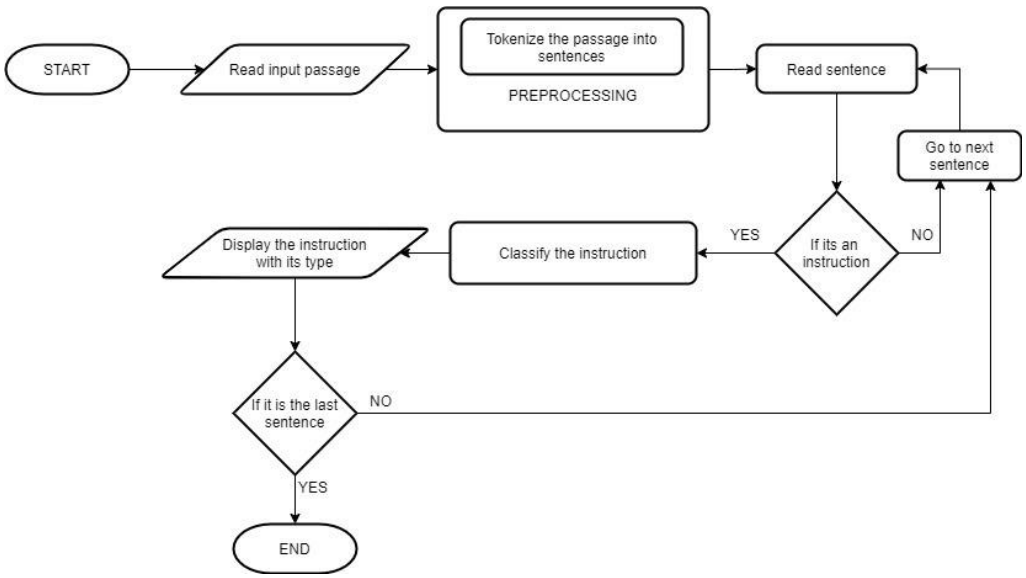


Figure 3.1 Overall flow diagram of the project

3.1 MODULE DESCRIPTION

.

3.1.1 LIST OF MODULES

Following are the list of modules that we have implemented in this project

- IsInstruction
- IsInstructionWithPurpose
- IsInstructionWithTool
- IsInstructionInsequence
- IsSimpleInstruction
- IsNegativeInstruction
- IsInstructionWithReason
- IsOptionalInstruction
- IsManditoryInstruction

3.1.2 IsInstruction:

Description: To check whether a given statement is an instruction or not.

Input: Statements (given by the user)

Output: Instructions

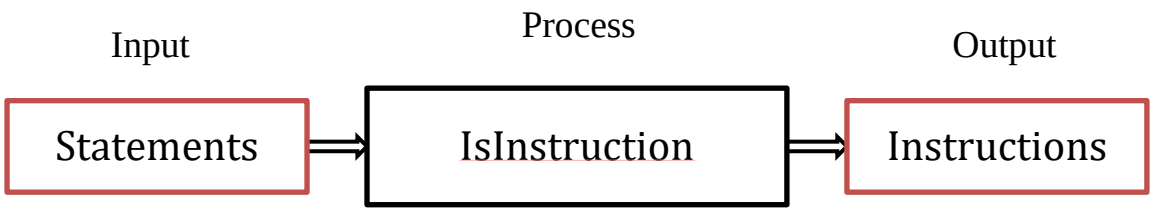


Figure 3.2 IsInstruction Module

3.1.2 IsInstructionWithPurpose:

Description: To check whether a given statement is an instruction with purpose.

Input: Instructions (output of isInstruction module)

Output: Instructions with purpose.

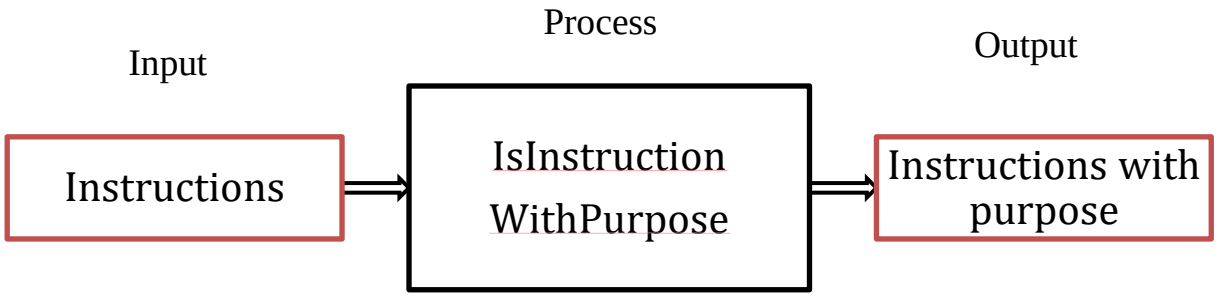


Figure 3.3 IsInstructionWithPurpose Module

3.1.3 IsInstructionInSequence:

Description: To check whether a given statement is an instruction in sequence.

Input: Instructions (output of isInstruction module)

Output: Instructions in sequence

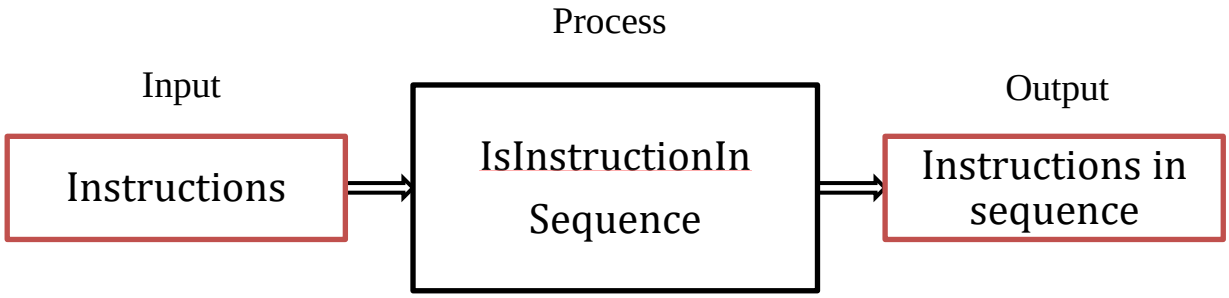


Figure 3.4 IsInstructionInSequence Module

3.1.4 IsInstructionWithTool:

Description: To check whether a given statement is an instruction with tool.

Input: Instructions (output of isInstruction module)

Output: Instructions with tool.

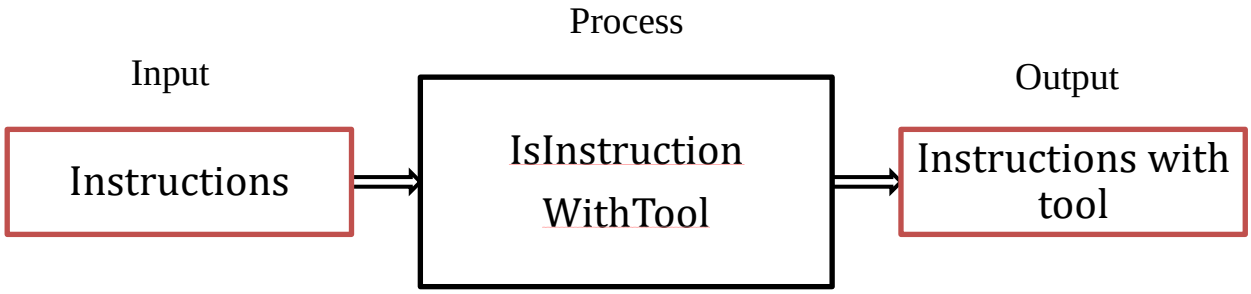


Figure 3.5 IsInstructionWithTool Module

3.1.5 IsSimpleInstruction:

Description: To check whether a given statement is a simple instruction.

Input: Instructions (output of isInstruction module)

Output: Simple instructions

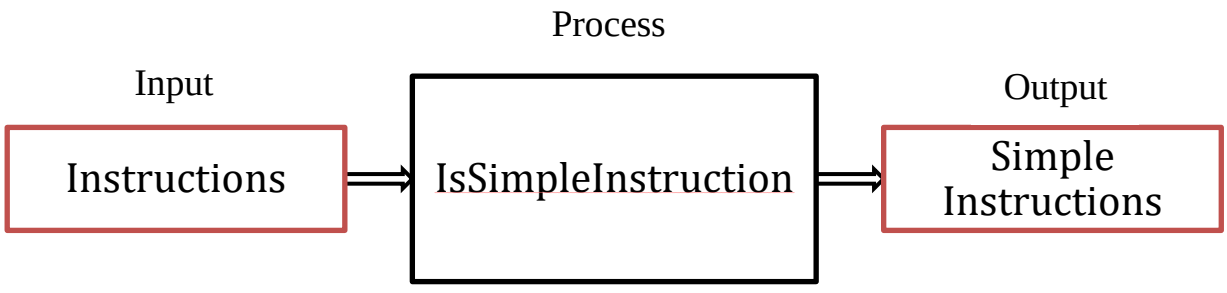


Figure 3.6 IsSimpleInstruction Module

3.1.6 IsInstructionWithReason:

Description: To check whether a given statement is an instruction with reason.

Input: Instructions (output of isInstruction module)

Output: Instructions with reason

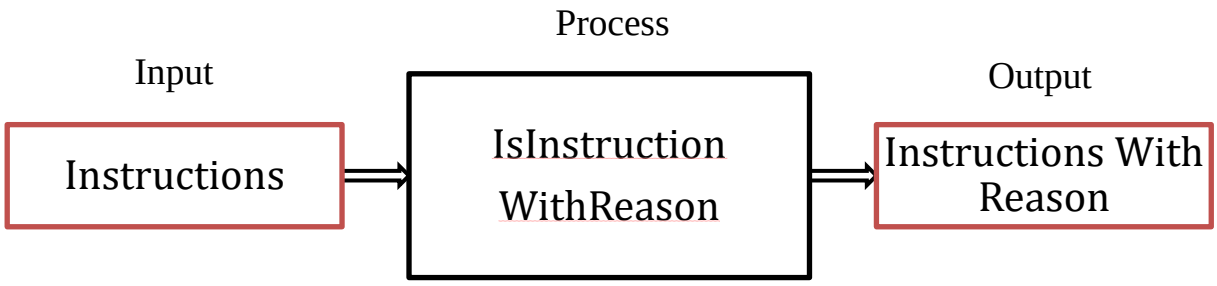


Figure 3.7 IsInstructionWithReason Module

3.1.7 IsMandatoryInstruction:

Description: To check whether a given statement is a mandatory instruction.

Input: Instructions (output of isInstruction module)

Output: Mandatory instructions

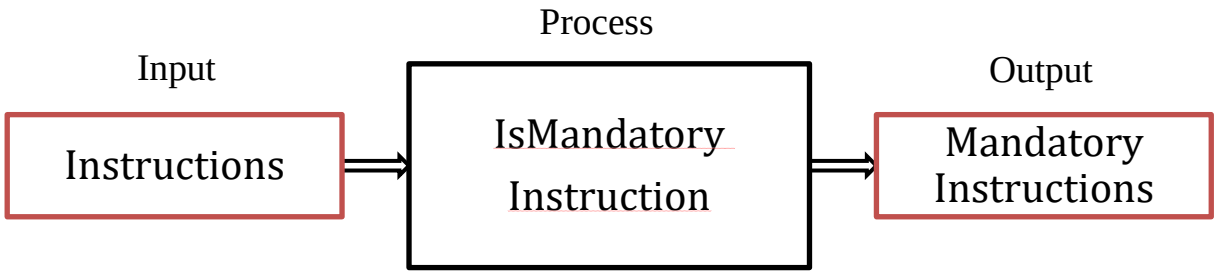


Figure 3.8 IsMandatoryInstruction Module

3.1.8 IsOptionalInstruction:

Description: To check whether a given statement is an optional instruction.

Input: Instructions (output of isInstruction module)

Output: Optional instructions

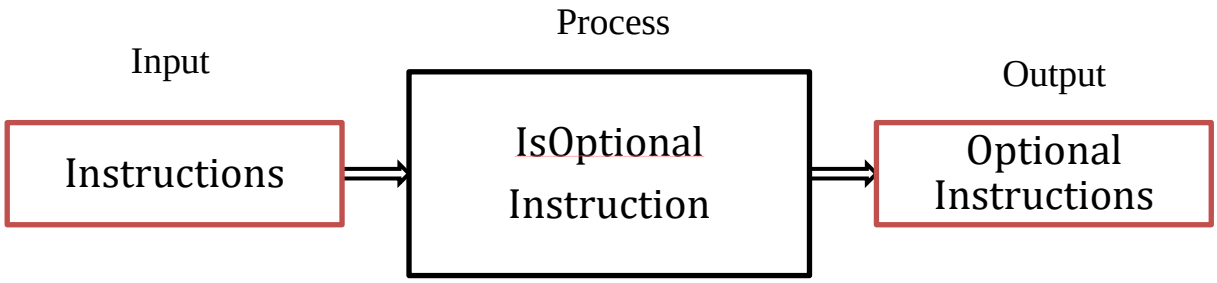


Figure 3.9 IsOptionalInstruction Module

3.1.9 IsNegativeInstruction:

Description: To check whether a given statement is a negative instruction.

Input: Instructions (output of isInstruction module)

Output: Negative instructions

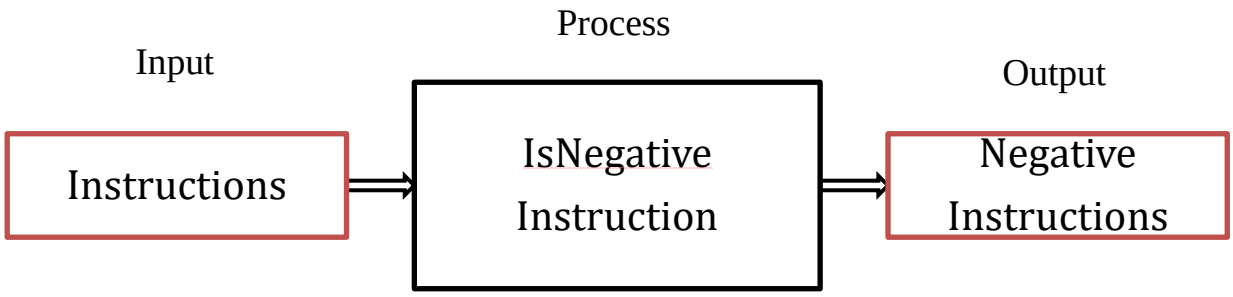


Figure 3.10 IsNegativeInstruction Module

3.2 ARCHITECTURE DIAGRAM

Figure 3.10 depicts the overall architecture of the project, which shows the various processes involved in this project.

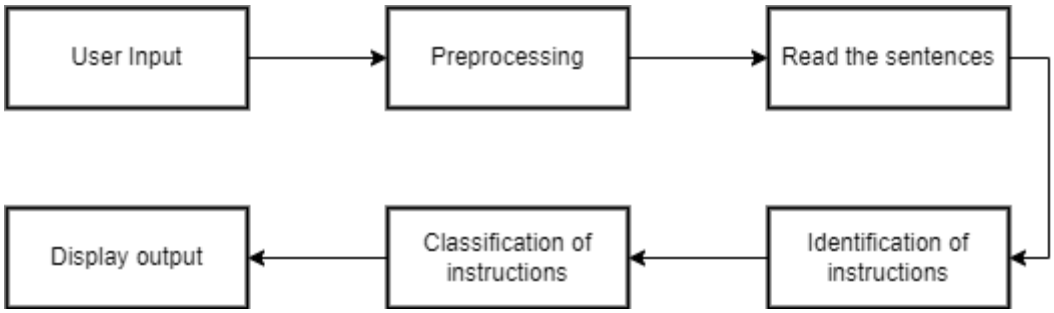


Figure 3.11 Overall Architecture Diagram

3.3 TOOLS AND APPLICATIONS

3.3.1 Integrated Development Environment

IDE used is Google Colab. Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

3.3.2 Programming Language

Programming language used is PYTHON. Python is a popular programming language. It is the simplicity and easy to understand syntax of python language which makes it more popular among programmers. Python has a rich variety of libraries which have various inbuilt functions and thus make the job of programming a lot easier.

3.3.3 Libraries

Libraries used are Flair, Spacey, and Pandas. We make abundant use of the POS taggers of the Flair library. We use the pandas library to manipulate csv files.

3.3.4 Frontend

We have used the Bootstrap framework to design the frontend of our website, where we get input passage from the user. Bootstrap is a free, open-source CSS framework used to create responsive web pages.

3.3.5 Backend

The backend framework used is Flask. Flask is a web application framework that provides libraries to build lightweight web-applications in Python. Here, we make use of flask framework and run our python scripts on the user input, process the input and send the classified list of instructions to the website frontend, as our output.

CHAPTER 4

IMPLEMENTATION & RESULT

4.1 ALGORITHM DESCRIPTION & FLOW CHART

We have designed our own algorithms for identifying the instructions from series of sentences and classifying them into eight different types of instructions. In this chapter we have discussed the algorithms that we have designed in our project.

4.1.1 IsInstruction(s):

1. Start
2. Read the sentence s.
3. Tag the sentence with POS tagger of flair.
4. If the first word is a verb then
    a. return true.
5. If the given sentence is a question
    a. return false.
6. If the first word is 'If' or 'While' or 'In order to'
    a. comma = s.find(',')
    b. if the substring of s, after the comma and till the end of s, is an instruction
        i. return True
    c. else
        i. return False
7. Else
    a. iterate through the POS tagged sentence
        i. if there is a past tense or past participle
            1. return False

ii. if there is a modal verb and it is neither must nor should
    1. return False
iii. if there is a modal verb and it is either must or should
    1. return True
iv. if there is a 3rd or non-3rd person singular present verb and it is among [‘am’, ‘is’, ‘are’]
    1. return False

b. initialize verb_found = 0, noun_found = 0
c. iterate through the POS tagged sentence
    i. if the word is a verb
        1. verb_found = 1
    ii. else if the word is a noun
        1. noun_found = 1
    iii. if verb_found equals to 1 and noun_found equals to 0
        1. return True
    iv. else if verb_found equals to 0 and noun_found equals to
        1. return False
d. return False

8. End

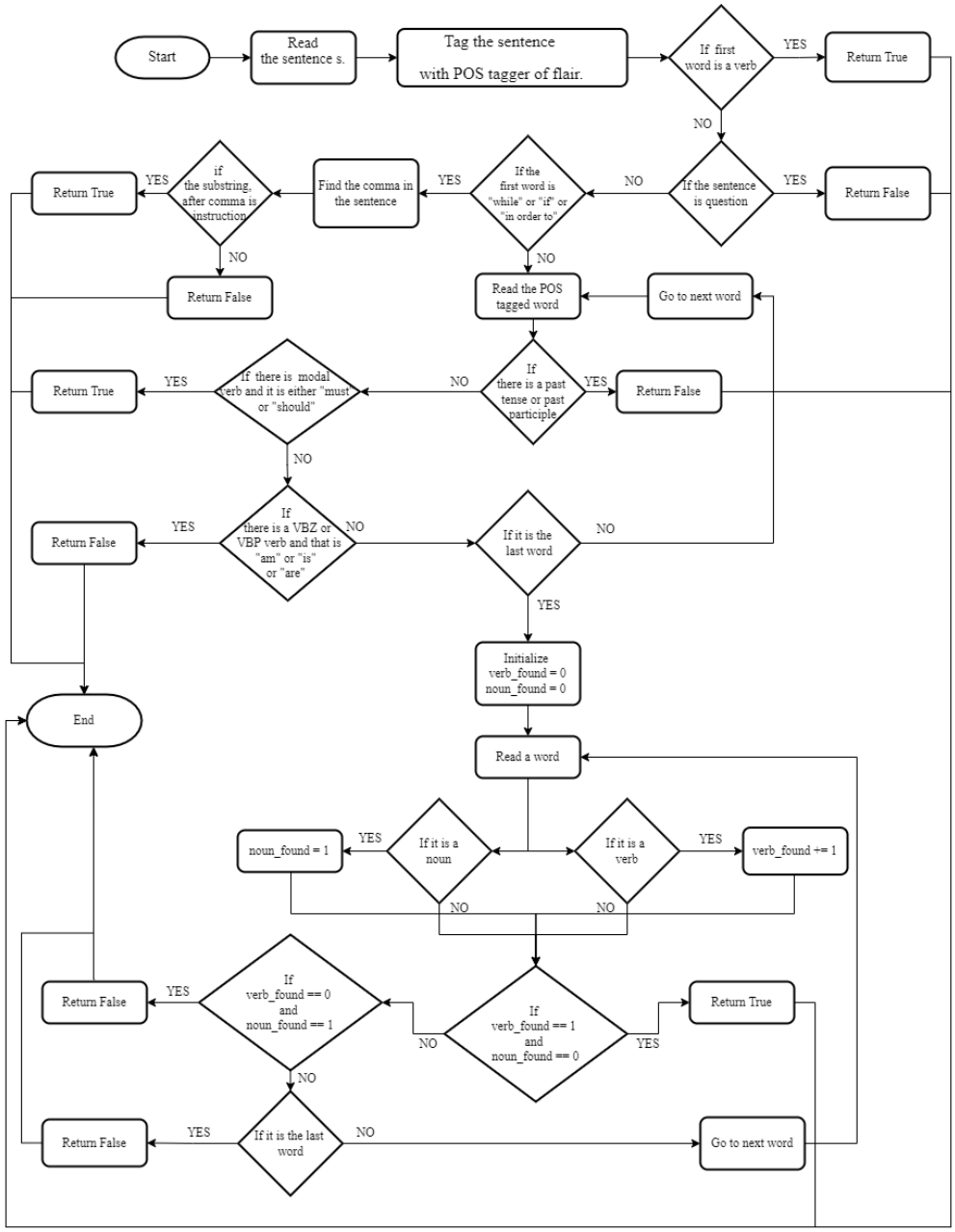


Figure 4.1 IsInstruction() Flow Chart

4.1.2 IsInstructionWithPurpose(s):

1. Start
2. Read instruction s.
3. initialize a = "hello word" , b = "good morning".
4. if the first word is "to" then

   a. remove the first word and keep the remaining as g.
   b. if there is a "," in the sentence g then
      i. a = substring of the sentence g before ","
      ii. b = substring of the sentence g after ","
5. else if second word is "to" then
   a. remove the first 2 words and keep the remaining as g.
   b. if there is a "," in the sentence g then
      i. a = substring of the sentence g before ","
      ii. b = substring of the sentence g after ","
6. else
   a. if there is a "to" in the middle of the sentence then
      i. a = substring of s before "to"
      ii. b = substring of s after "to"
7. if (isInstruction(a) and isInstruction(b)) then
   a. return True
8. else
   a. return False
9. End

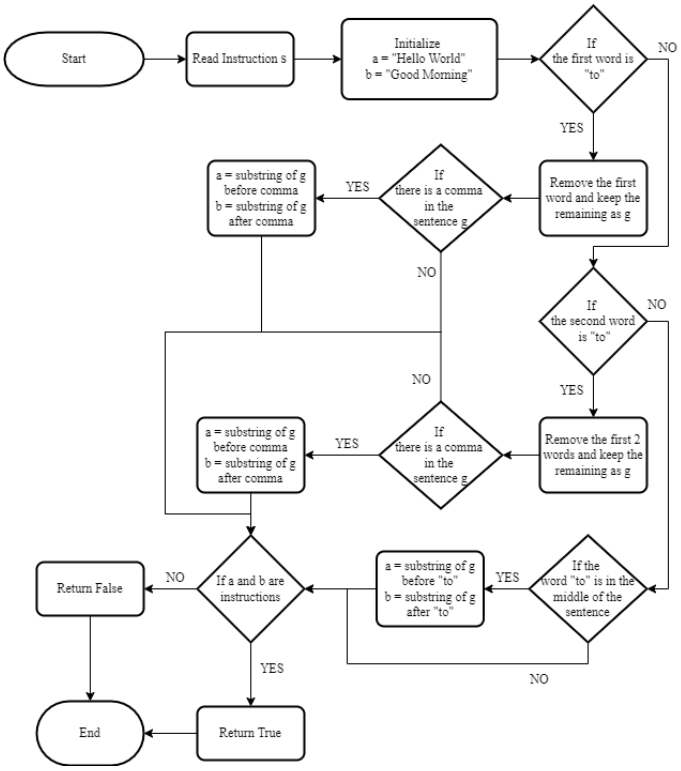


Figure 4.2 IsInstructionWithPurpose() Flow Chart

4.1.3 IsInstructionInSequence(s):

1. Start
2. Read instruction s.
3. if there is a "," in the sentence then
    a. a = substring of s before ","
    b. b = substring of s after ","
    c. if (isInstruction(a) and isInstruction(b)) then
        i. return True
4. else if there is a "then" in the sentence then
    a. a = substring of s before "then"
    b. b = substring of s after "then"
    c. if (isInstruction(a) and isInstruction(b)) then
        i. return True
5. else if there is a "and" in the sentence then
    a. a = substring of s before "and"
    b. b = substring of s after "and"
    c. if (isInstruction(a) and isInstruction(b)) then
        i. return True
6. else
    a. return False
7. End

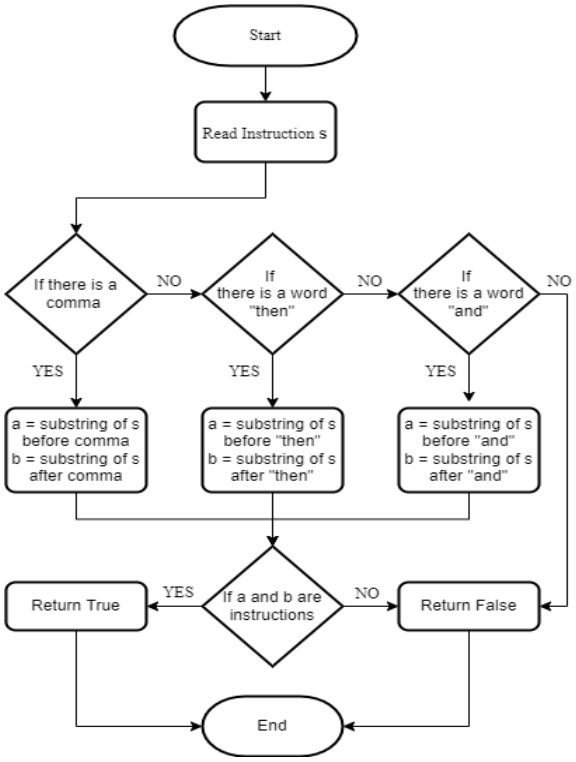


Figure 4.3 IsInstructionInSequence() Flow Chart

4.1.4 IsInstructionWithTool(s):

1. Start
2. Read instruction s.
3. initialize object_count = 0;
4. load the English language tagger from spacy library.
5. using the tagger, tag the sentence s.
6. for each word in the sentence s
    a. if the tag_name of the word is "pobj" or "doj" then
        i. object_count += 1
7. if (object_count >= 2) then
    a. return True
8. else
    a. return False
9. End

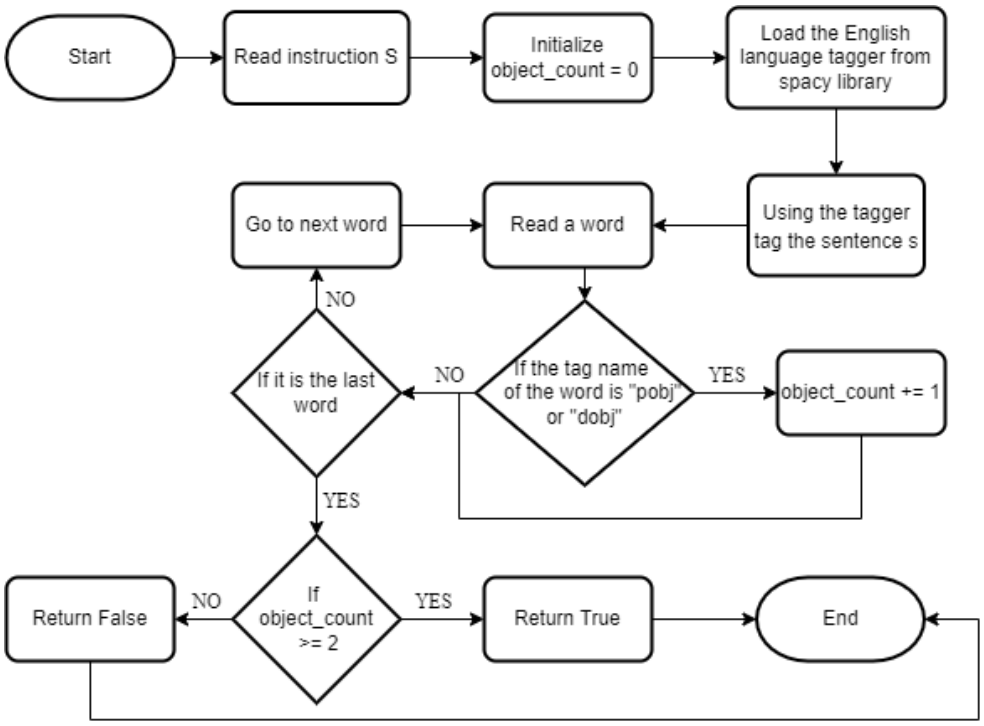


Figure 4.4 IsInstructionWithTool() Flow Chart

4.1.5 IsSimpleInstruction(s):

1. Start
2. Read instruction s.
3. initialize verb_count = 0;
4. Tag the words in the sentence using the pos tagger of flair.
5. for each word in the sentence s
    a. if the pos tag of the word belongs to any form of verb then
        i. verb_count += 1;
6. if (verb_count == 1) then
    a. return True
7. else
    a. return False
8. End

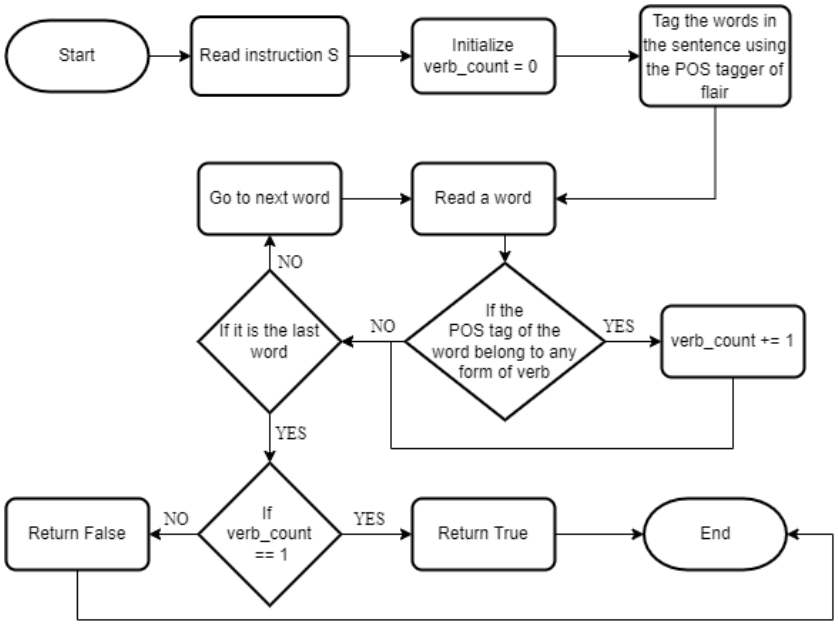


Figure 4.5 IsSimpleInstruction() Flow Chart

4.1.6 IsOptionalInstruction(s):

1. Start
2. Read instruction s.
3. i = s.find(" or ")
4. If i not equal to -1
    a. str1 = substring of the sentence s, before the word " or ".
    b. str2 = substring of the sentence s, after the word " or ".
    c. If either str1 or str2 is an instruction:
        i. return True
5. return False
6. End

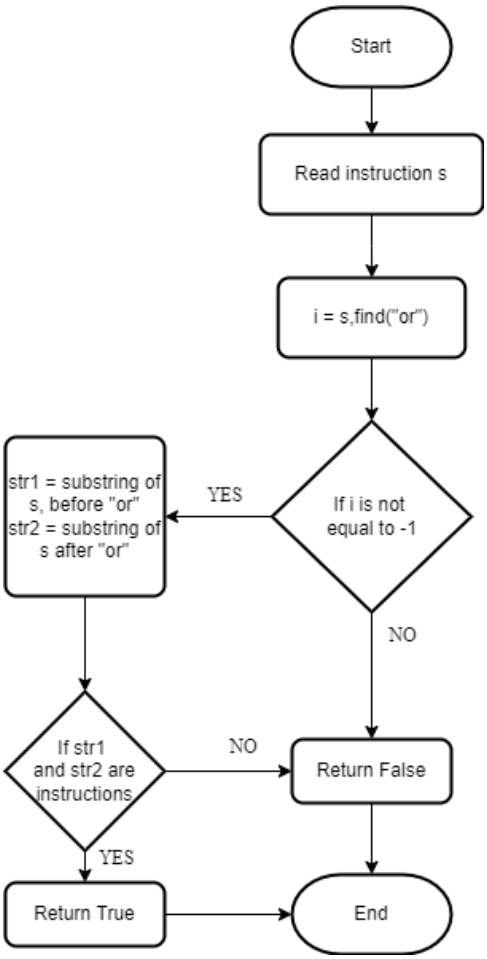


Figure 4.6 IsOptionalInstruction() Flow Chart

4.1.7 IsInstructionWithReason(s):

1. Start
2. Read instruction s.
3. i = s.find(',')
4. If i not equal to -1
    a. If the first word is "If"
        i. st = substring of the sentence s, after the ', '.
        ii. If st is an instruction:
            1. return True
    b. Else
        i. st = substring of the sentence s, before the ', '.

ii. If st is an instruction:
    1. return True

5. return False
6. End

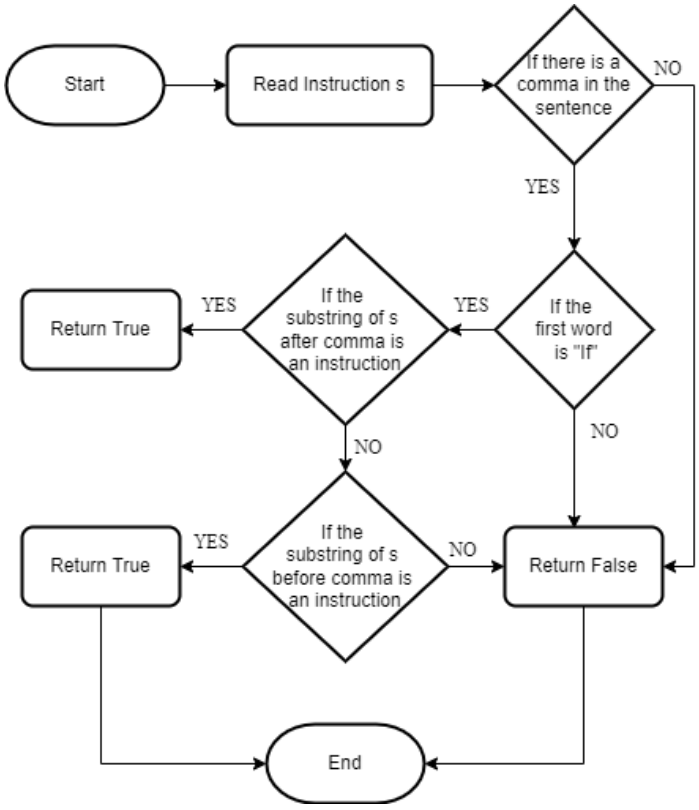


Figure 4.7 IsInstructionWithReason() Flow Chart

4.1.8 IsManditoryInstruction(s):

1. Start
2. Read instruction s.
3. mustIndex = s.find('must')
4. shouldIndex = s.find('should')
5. if either mustIndex or shouldIndex is not equal to -1:

   a. return True
6. return False
7. End

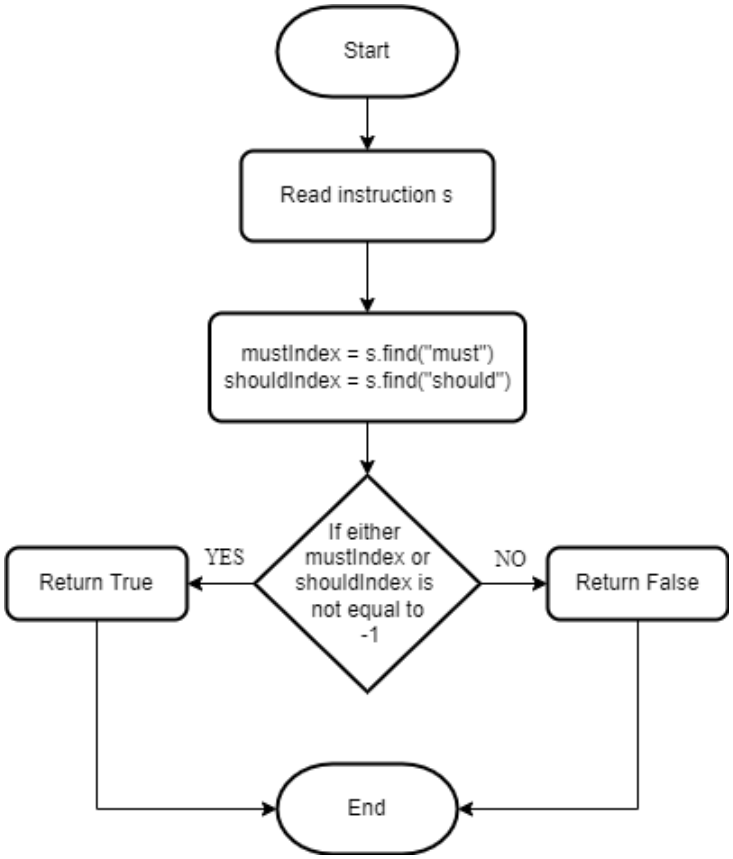


Figure 4.8 IsMandatoryInstruction() Flow Chart

4.1.9 IsNegativeInstruction(s):

1. Start
2. Read instruction s.
3. ls = list(s.split(" "))
4. Find the index of the words "don't", "do not", "not", "mustn't", "shouldn't", "must not", "should not", "must stop", "should stop", "must avoid", "should avoid"
5. If any of the words are present in the sentence
   a. return True
6. If the first 2 words are "Do" and "not"

   a. If the substring of s, after "not" till the end, is an instruction
      i. return True
7. If the first word is either "Don't" or "Never"
   a. If the substring of s, after "Don't" or "Never" till the end, is an instruction
      i. return True
8. If the first word is either "Stop" or "Avoid"
   a. return True
9. return False
10. End

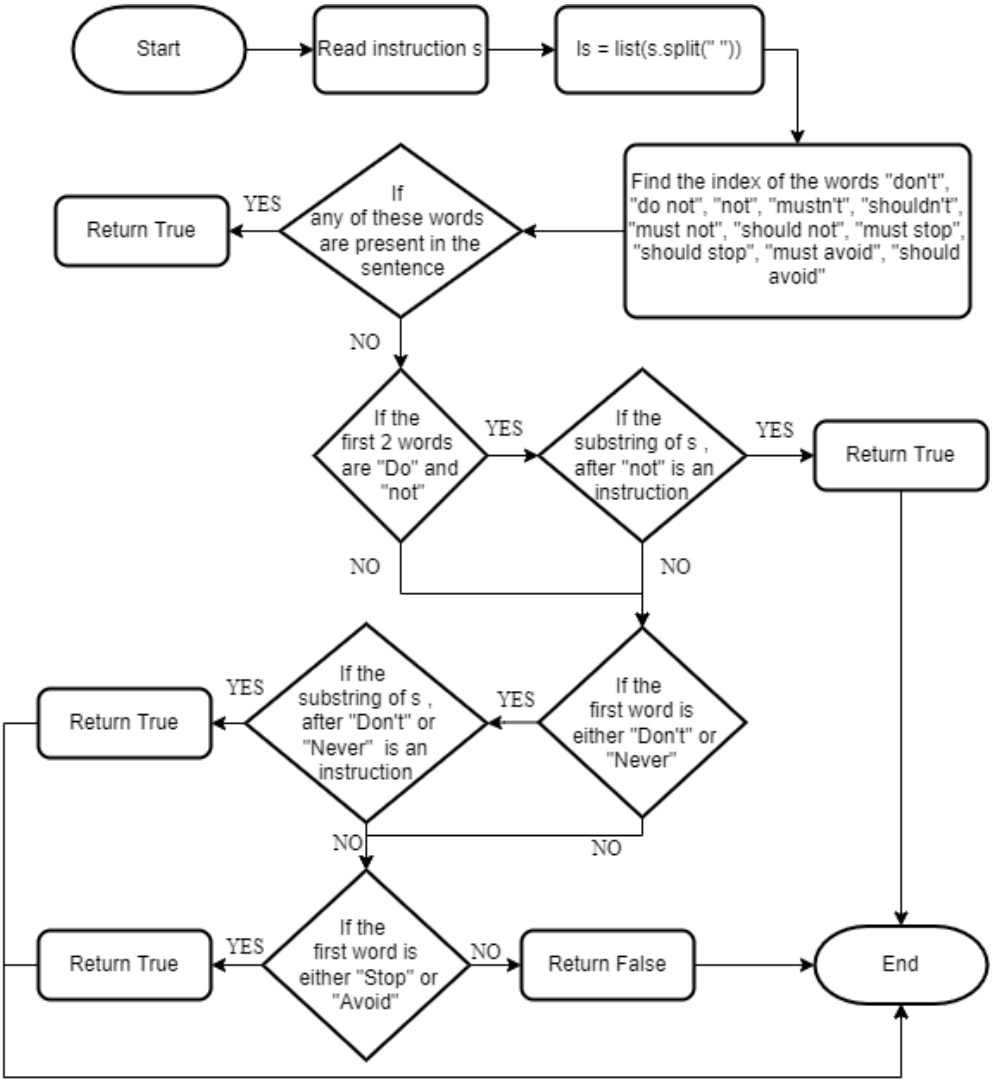


Figure 4.9 IsNegativeInstruction() Flow Chart

4.2 RESULT AND DISCUSSION

When the instructions are identified from a paragraph, we may simply read them to understand what has to be done and what not to. It essentially sums up the passage. It will be more beneficial in the future with robotics, as robot commands are primarily instructions. As a result, our project will aid in quick understanding of a passage in a short period of time.

4.2.1 OUTPUT SCREENSHOTS

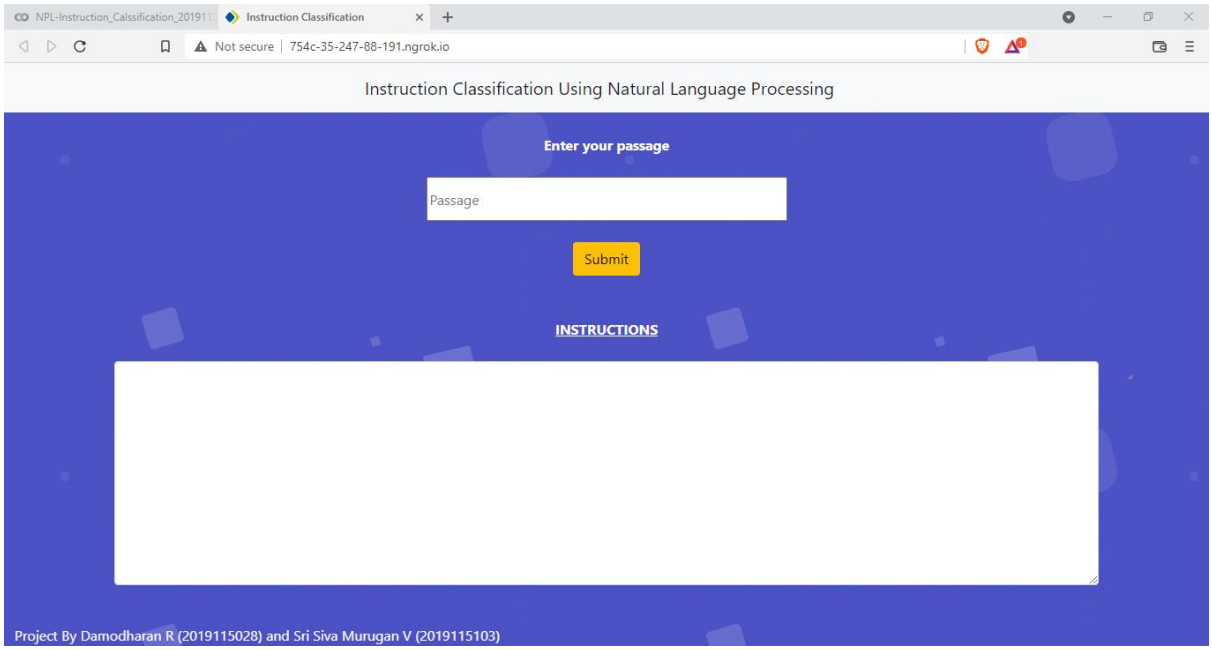


Figure 4.10 Output Screenshot - GUI

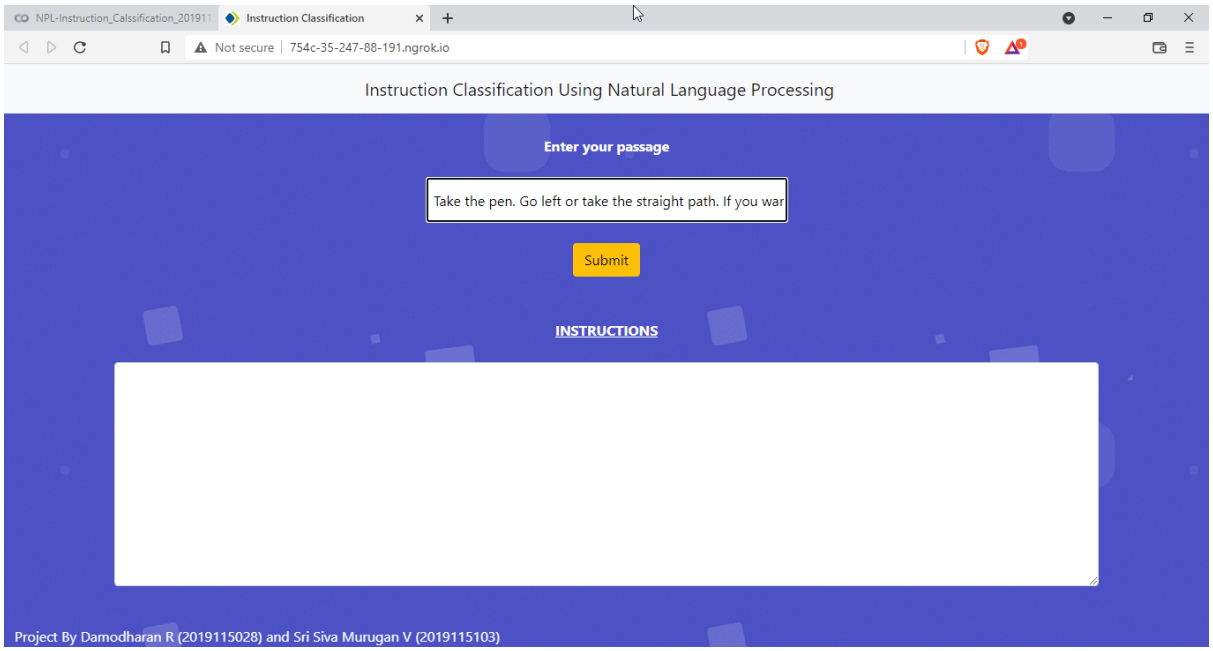


Figure 4.11 Output Screenshot – User Input

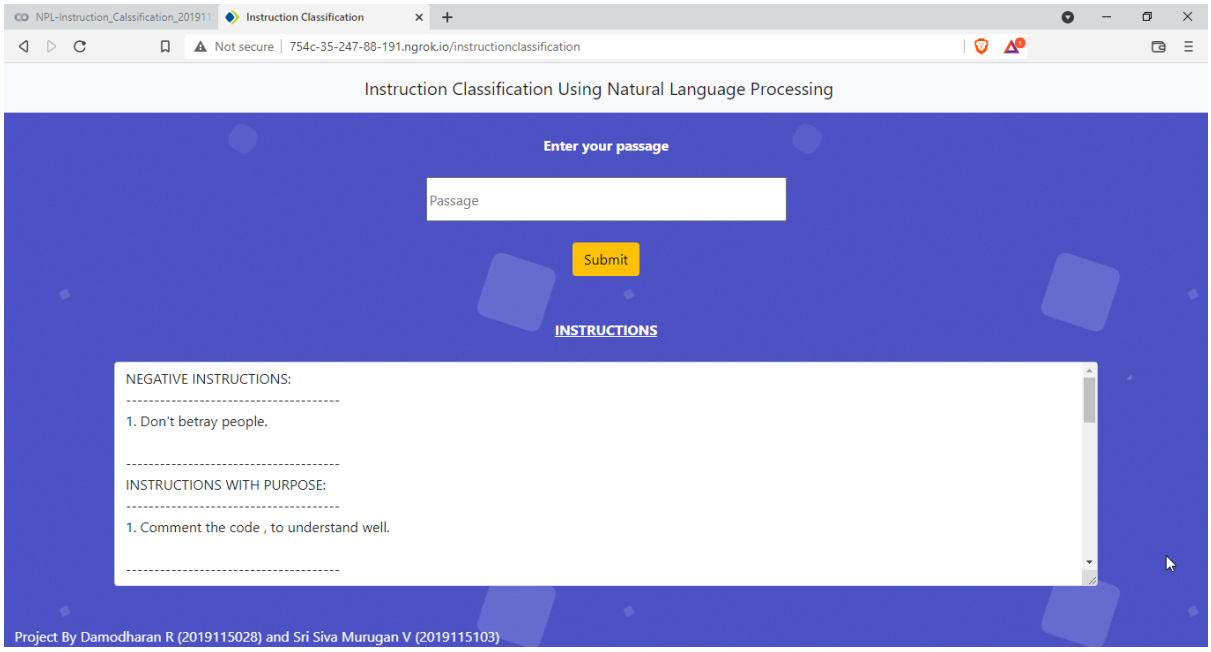


Figure 4.12 Output Screenshot – Negative Instruction and Instruction with purpose

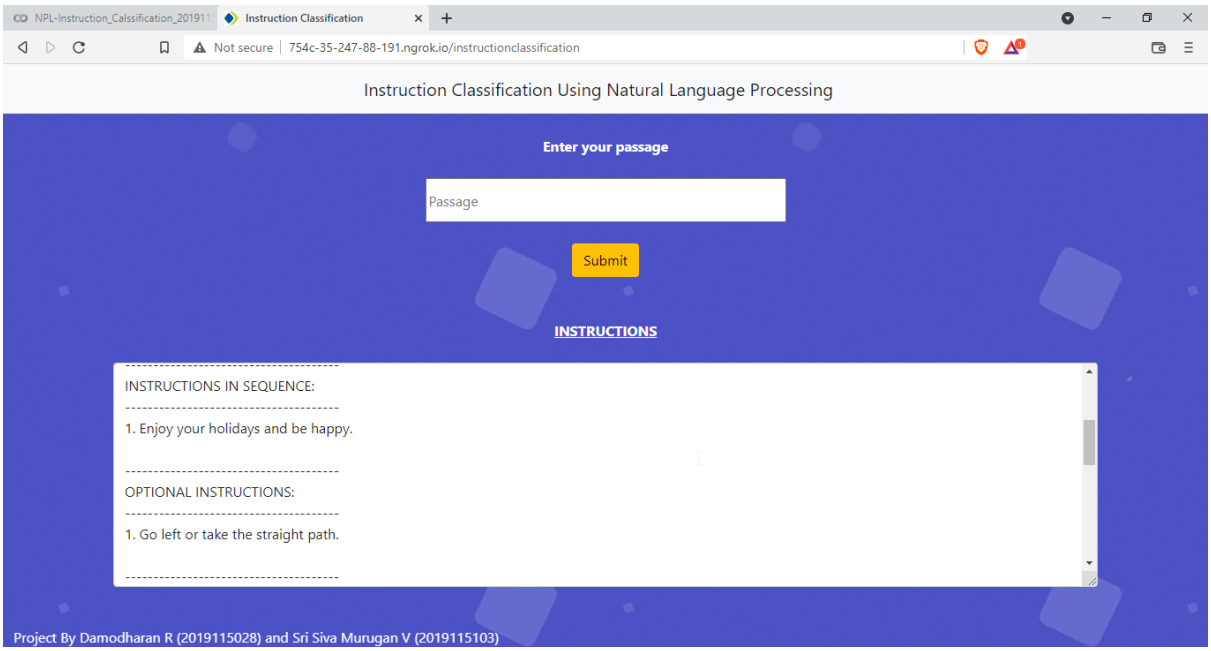


Figure 4.13 Output Screenshot – Instruction in sequence and Optional instruction

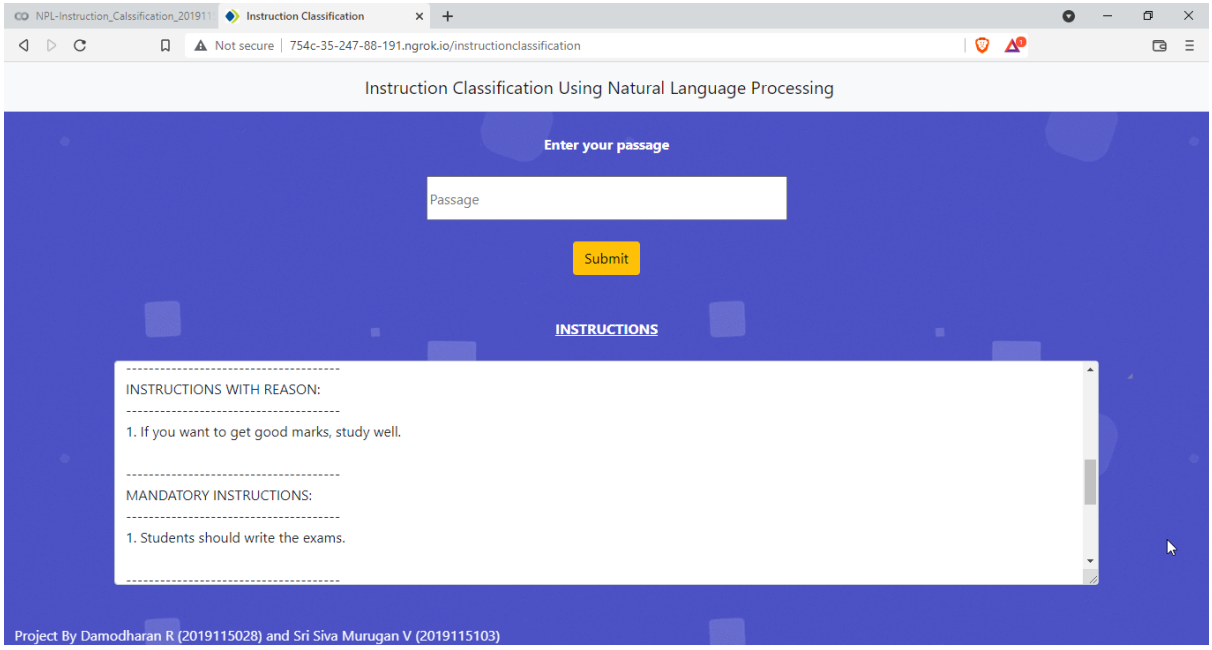


Figure 4.14 Output Screenshot – Instruction with reason and Mandatory instruction

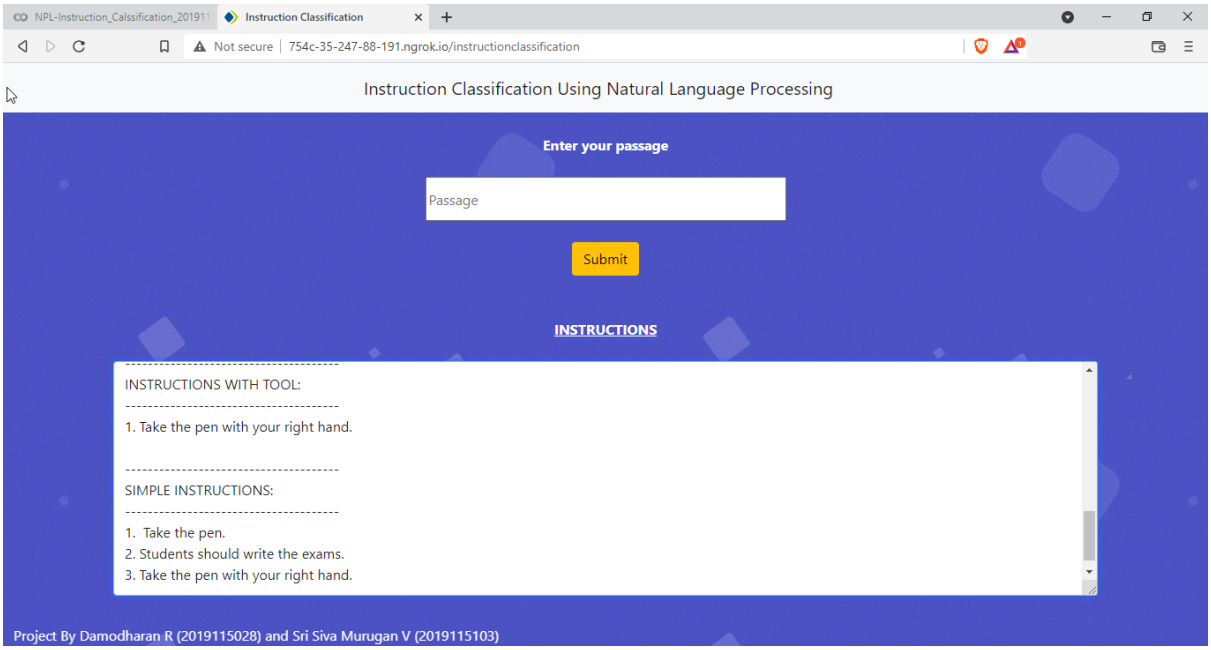


Figure 4.15 Output Screenshot – Instruction with tool and Simple instructions

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This documentation explains the identification of instructions from a given passage and classifying them into eight different types, done with the help of the POS tagger of Flair and the algorithms we developed. This is definitely a time saver as this will help people to extract the crucial information i.e., instructions from a given text and get the crux of it, without reading the whole text

5.2 FUTURE ENHANCEMENT

Since we have not given much attention to the accuracy of this project, in the future, people can attempt to improve the accuracy of this project and then, with the help of this project, they can try to make a text summarizer, to read a piece of text in short time without missing the important information in it.

REFERENCES

[1] Pengpeng Zhou and Hao He. Translating Natural Language Instructions for Behavioral Robot Indoor Navigation with Attention-History Based Attention. 4th International Conference on CSAI (2020).

https://dl.acm.org/doi/10.1145/3445815.3445857

[2] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer and Dieter Fox. Learning to Parse Natural Language Commands to a Robot Control System. University of Washington, Seattle, USA.

https://homes.cs.washington.edu/~lsz/papers/mhzf-iser12.pdf

[3] Article classification using natural language processing and machine learning

https://ieeexplore.ieee.org/document/9044227

[4] Clinical Report Classification Using Natural Language Processing and Topic Modelling

https://ieeexplore.ieee.org/document/6406751

[5] Survey on Parts of speech Tagger Techniques

https://ieeexplore.ieee.org/document/8550884

[6] The Effect of POS Tag Information on Sentence Boundary Detection in Turkish Texts

https://ieeexplore.ieee.org/document/8554031

[7] https://blog.jcharistech.com/2020/10/04/text-classification-with-flair-pytorch-nlp-framework/

[8] https://neptune.ai/blog/sentiment-analysis-python-textblob-vs-vader-vs-flair

[9] https://github.com/kk7nc/Text_Classification#word-embedding

[10] https://github.com/akashp1712/nlp-akash/tree/master/text-summarization