**Mini Project Report**

**On**

**TO AUTOMATE COMPARISON OF TREES GENERATED FROM NATURAL LANGUAGE USING PARSER**

**VIth SEMESTER**

**INFORMATION TECHNOLOGY**

***Submitted by***

**GAURAV BHISIKAR**

**SAGAR MALIK**

**ADITYA DALE**

**Under the guidance of**

**PROF. POOJA WALKE**

**ASSISTANT PROFESSOR**

**Academic Year 2019-20**

**Department of Information Technology**



**ST. VINCENT PALLOTTI COLLEGE OF**

**ENGINEERING AND TECHNOLOGY**

Wardha Road, Gavsi Manapur, Nagpur

**ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING AND TECHNOLOGY**

Wardha Road, Gavsi Manapur, Nagpur

**Department of Information Technology**

**CERTIFICATE**

Certified that this project report **“TO AUTOMATE THE COMPARISON OF TREES GENERATED FROM NATURAL LANGUAGE USING PARSER”** is the bonafide work of “**GAURAV BHISIKAR, SAGAR MALIK, ADITYA DALE**” who carried out the mini project work under my supervision in partial fulfilment of VI Semester, Bachelor of Engineering in **INFORMATION TECHNOLOGY** of RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR.

**PROF. MANOJ V. BRAMHE**

**PROF. POOJA WALKE**

Associate Professor

Assistant Professor

**HEAD OF THE DEPARTMENT**

**GUIDE**

**PRINCIPAL**

**ST. VINCENT PALLOTTI COLLEGE OF ENGINEERING AND TECHNOLOGY**

**ACKNOWLEDGEMENT**

Our mini-project titled, **“TO AUTOMATE THE COMPARISON OF TREES GENERATED FROM NATURAL LANGUAGE USING PARSER”**. Any project seminarrequires a lot of hard work, sincerity and systematic work methodologies. We express our deepest gratitude to our Project Guide, **Prof. Pooja Walke**, for giving us an opportunity to be a part of this project seminar and for the guidance provided throughout the span of the first phase of our project.

We would also like to thank **Prof. M. V. Bramhe, Head of the Department** **of Information Technology** and all our faculty members who regularlyevaluated our project and pointed out the shortcomings in the projects. They also gave us important feedback for the further improvement of our project. We are highly indebted to them.

We owe our gratitude to the industry mentor **Dr. Soma Paul, research** **Assistant professor, IIIT Hyderabad** and **Dr. Vineet Chaitanya at Language Translation Research Center(LTRC), IIIT Hyderabad** fortheir immense support and guidance throughout the project.

We are also grateful to the **Management of the College, Dr. Surendra** **Gole, Principal & Prof. R. B. Gowardhan, Vice-Principal** for theoverwhelming support in providing us the facilities of computer lab and other required infrastructure. We would like to thank our Library Department for providing us useful books related to our project.

**GAURAV BHISIKAR**

**SAGAR MALIK**

**ADITYA DALE**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGEMENT** |  |
|  | **ABSTRACT** |  |
|  | **LIST OF FIGURES** |  |
| **1.** | **INTRODUCTION**  1.1 OVERVIEW  1.2 PROBLEM STATEMENT  1.3 OBJECTIVES  1.4 ORGANIZATION OF REPORT | **1** |
| **2.** | **REVIEW OF LITERATURE**     |  |  | | --- | --- | | 2.1 | INTRODUCTION | | 2.2 | LITERATURE SURVEY | | 2.3 | FEASIBILITY STUDY | | **3** |
| **3.** | **PROPOSED SYSTEM**  3.1 DRAWBACK OF CURRENT SYSTEM AND NEED OF PROPOSED SYSTEM  3.2 SYSTEM DESCRIPTION AND SRS SYSTEM  3.2.1 PURPOSE  3.2.2 SCOPE OF PROJECT  3.2.3 PROJECT PLANNING  3.2.4 WORK-BREAKDOWN STRUCTURE  3.3 SYSTEM ANALYSIS  3.3.1 SYSTEM ARCHITECTURE   |  |  | | --- | --- | | 3.3.2 FLOWCHART OF SYSTEM | | | 3.3.3 DATA FLOW DIAGRAM | | | 3.3.4 USE CASE DIAGRAM | | | 3.3.5 HARDWARE REQUIREMENTS | | | 3.3.6 SOFTWARE REQUIREMENTS | | | 3.4 | SYSTEM DESIGN | |  | 3.4.1 CLASS DIAGRAM  3.4.2 COMPONENT DIAGRAM  3.4.3 INTEGRATION DIAGRAM / DEPLOYMENT DIAGRAM  3.4.4 USER STORIES | | **8** |
| **4.** | **CONCLUSION & FUTURE SCOPE** | **19** |
| **5.** | **REFERENCES** | **20** |
|  |  |  |

**ABSTRACT**

This project aims to automate the process of comparing the two trees generated from natural language by parser. It is a part of research at International Institute of Information Technology Hyderabad. Processing of any natural language requires analysis to be done at multiple levels like word-level, phrase-level, sentence-level, semantic-level and higher levels of pragmatic and discourse. The sentence level, which in linguistics terms, is regarded as Syntactic Parsing. Syntactic parsing involves establishing relations between different words of a sentence to convey the possible meaning.In this work we are presenting our efforts of making new advancements in finding errors and making communicator tools more efficient.

Syntactic Parsing establishes the representational relations pertaining to a grammatical framework like phrase structure grammar, dependency grammar, categorical grammar, etc. In general, syntactic parsing is of two types:-

1. **Constituent Parsing:** It explicitly represents phrases (non-terminal nodes: Noun Phrase, Verb Phrase, etc.), structural categories (non-terminal labels like Part-of-Speech) and possibly some functional categories (grammatical functions) i.e. using Phrase Structure Grammar (PSG).

2. **Dependency Parsing:** It explicitly establishes relationships between words as functional categories (directed labels) and possibly some structural categories as head-dependent relations (directed arcs) i.e. using Dependency Grammar (DG).

Thus, we have used Irshad Parser and Stanford Parser for generating CONLL format through which we generate trees from natural language, which will help to develop the communicator tool for future needs.

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO.** | **FIG NO.** | **FIGURE NAME** | **PAGE NO.** |
|  |  |  |  |
| 1 | 3.3.1 | System Architecture | 11 |
|  |  |  |  |
| 2 | 3.3.2 | Flow Chart | 11 |
|  |  |  |  |
| 3 | 3.3.3.1 | Data Flow Diagram Level 0 | 12 |
|  |  |  |  |
| 4 | 3.3.3.2 | Data Flow Diagram Level 1 | 13 |
|  |  |  |  |
| 5 | 3.3.4 | Use Case Diagram | 14 |
|  |  |  |  |
| 6 | 3.4.1 | Class Diagram | 16 |
|  |  |  |  |
| 7 | 3.4.2 | Component Diagram | 17 |
|  |  |  |  |
| 8 | 3.4.3 | Deployment Diagram | 17 |
|  |  |  |  |

**1. INTRODUCTION**

**1.1 OVERVIEW**

This project aims to develop an authoring tool which helps users to write in one own’s language which can be converted into another target language through the mediation of the control languages which is very close to the source language. Communicator tool is a multilingual translation tool in which we give natural languages as an input which gets translation into the other foreign languages with the help of Parsers, data dictionary, User CSV. Language communicator allows users to see the translation in multiple languages. The languages that are currently supported by communicator tools are English, Japanese, German. The aim of this project is to automate the process of comparing two trees generated from two different natural languages.

The Communicator tool takes semantically disambiguated and user verified language as an input. It is the Universal Semantic Representation (USR) of the sentence with layers of information. The MRS (Minimal Recursion Semantics) mapper or the transformations engine takes it as the input it as the input and maps the Paninian and non-Paninian relations and all other ontological information to that of MRS representation. Using the MRS representation the ACE generator provides the final output for the Targeting Language.

ṣ

**1.2 PROBLEM STATEMENT**

To automate the process of comparing the trees generated from two different languages using irshad and stanford parser.

**1.3 OBJECTIVE**

The main aim of the project is to create a tool which can compare the language tree of natural language generated using parser, so that we can make new advancements in finding errors and making communicator tool more efficient.

**1.4 ORGANIZATION OF REPORT**

Our report is divided into several chapters which includes introduction, review of literature, proposed system, conclusion and reference. The chapter 1 include

The complete introduction, chapter 2 consists of the review of the literature that discusses what has been derived from papers, online forums and what previous work is done and the need of the proposed system. We have analyzed the various methods for learning word embeddings that have been proposed till now. Chapter 3 consists of the proposed system that describes the desired features and functions of the system. System analysis and design with the help of class diagrams and component diagrams has been explained in detail. User stories have also been considered in this section. Hardware and software requirements are also mentioned in this chapter. Explanation of all the modules with their work breakdown structure with the help of Flowcharts and Data Flow Diagrams is mentioned properly. Chapter 4 and Chapter 5 have the conclusion & future scope of our system and references for the project.

**2. REVIEW OF LITERATURE**

**2.1 INTRODUCTION**

A literature review is an evaluative report of information found in the literature related to your selected area of study. The review should describe, summarize, evaluate and clarify this literature. it should give a theoretical base for the research and help you to determine the nature of your research.

When conducting research, a literature review is an essential part of the project because it covers all previous research done on the topic and sets the platform on which the current research is based. no new research can be taken seriously without first reviewing the previous research done on the topic.

**2.2 LITERATURE SURVEY**

In paper [1] we have studied “TRANSLATION OF RELATED LANGUAGE PAIR USING APERTIUM “research paper which was written by Sriram Chaudhary, “Anusaaraka: An Expert System based MT System”, in the proceeding of IEEE conference on Natural language processing and knowledge management (IEEE-NLP KE 2010) along with some researchers. This paper discuss about Apertium Platform, which is an open source shallow transfer machine translation system, which was initially designed for translation between related language pairs.

In paper [2] ”Akshar Bharati, “Anusaaraka: Overcoming the Language Barrier

in India” it has been written about all the rows of user csv as well as about

concept dictionary. The details of user csv are mentioned below. Universal Semantic Representation(USR) must be explicit and free of all the linguistics ambiguities. While the machine can go up to some point, the speaker/user is at the best place to notice them and rectify as well. Other information such as discourse relations, morphological information, and sentence type is also part of the representation. It uses the concept dictionary which has unique Ids for all the content words. This representation is an extended form of controlled language.

As the name suggests CSV file is a Comma Separated Values.

1. Each field in a row should be separated by Comma (,)
2. If a field has no value, then only comma needs to be typed.

The user csv file will contain 9 rows. The rows are

1. Grouping
2. Concept from Concept dictionary
3. Index for Concepts
4. Lexical Category and Ontological Information about nouns
5. GNP (Gender, Number, Person) Information
6. Intrachunk Dependency Relations
7. Intrachunk Dependency Relations
8. Discourse Relations

9. Type of sentence

The 1st row in user csv is about grouping of local word grouping (LWG). the modifiers of a noun such as adjective, determiners, numbers (cardinal and/or ordinal), emphatic markers, measurement units and all other possible modifiers will be grouped together.

The concept dictionary represents the concepts and the respective language specific representations for those concepts. Each concept can have one or more than one sense which is disambiguated by the concept ids in the respective language columns. The concept dictionary as of now has Hindi, Eng. German, Japanese, Tamil and Marathi.

We are only keeping the ontological information about the nouns. As the user/speaker is in the best place to provide the definiteness information of a noun, the machine will ask the user/speaker to provide the same. If the user is convinced that the noun s/he is talking about is a one and not generic, then it is a definite noun; else it is not. In the user csv, the user must give the information about the definiteness of the noun in 4th row.

The intra chunk relation is the dependency relation between the tokens of a

chunk. Here, while grouping, will be grouped as one chunk. The relation between and

is known as Intrachunk relation. There are a number of intrachunk relations possible.

The inter chunk relation is the dependency relation between the chunk heads of the sentences. Many of the inter chunk relations are represented with respect to the main verb of the sentence.

In the user csv, we will represent the relations present between the chunk heads and the main verb (here). These interchunk relations are otherwise known as karaka relations. Relations such as emphasis, delimitation, co-refencing, negation are marked in this row. Types of sentences are described in this row like assertive, question, imperative.

TAM is one of the three interesting and toughest parts to be made in linguistics because they are obligatorily even when a simple sentence is produced. Tense and Aspect are both properties of the verb, but they affect the verb and meaning of the sentence in different ways. TAM is separated from the root by “-”. When the root is in pure form we postulate a 0.TAM is written with an underscore if the root word is separated from the tam words.

Naskar S. and Bandyopadhyay S. in paper [3] “Use of machine translation in India have proposed several techniques for improving statistical machine translation between closely related languages with scarce resources. Closely related languages such as Hindi and Marathi exhibit a large overlap in their vocabulary and strong syntactic and lexical similarities. Spelling convention in such related languages can still be different. However, the differences often constitute consistent regularities that can be generalized when translating.

In paper [4]” Multi-two level steps for Catalan morphology. In Demo Proceedings of the Conference on Applied Natural Language Processing” by Badia, T. ,Egea & T. Tuells CATMORF we studied the analysis of morph words. Marathi morph is a software which separates the bare form and the root form of the word. Any word which we insert into Marathi morph, it separates its bare word and root word with the help of the dictionary.

**2.3 FEASIBILITY STUDY**

A feasibility study aims to objectively and rationally uncover the strengths and weakness of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. it focuses on these major questions:

1. What are the user’s demonstrable needs and how does a candidate system meet them?
2. What resources are available for a given candidate system?
3. What are the likely impacts of the candidate system on the organization?
4. Whether it is worth it to solve the problem?

During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.

**2.3.1 TECHNICAL FEASIBILITY**

Technical feasibility is a study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

In our system, we have used Python which is a high-level programming language for general-purpose programming. An interpreted language, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale.

**2.3.2 ECONOMICAL FEASIBILITY**

Economic feasibility is the cost and logistical outlook for a business project or endeavor. Prior to embarking on a new venture, most businesses conduct an economic study, which is a study that analyses data to determine whether the cost of the prospective new venture will ultimately be profitable to the company. Sometimes, companies hire an external company that specializes in conducting economic feasibility studies for them.

**2.3.3 OPERATIONAL FEASIBILITY**

It is mainly related to human organizations and political aspects. It is the measure of how well a proposed system solves the problem, and takes advantage of the opportunities identified during scope definitions and how it satisfies the requirement. It also measures how well the proposed system solves problems and takes advantage of the opportunities identified during scope definitions.

The system is operationally feasible as ours is an unsupervised approach that reaches state-of-the-art supervised approach. Thus, satisfactorily solving the problems of existing or previous systems. The resources have been utilized judiciously and removed the operational feasibility of the system.

**2.3.4 SCHEDULE FEASIBILITY**

Time evaluation is the most important consideration in the development of a project. The time schedule required for the development of this project is very important since more development time affects machine time, cost and cause delay in the development of other systems. The modules were completed as per the schedule. It shows the estimated time to complete the project.

This includes the schedules of each process in a project and the total project time. This can change if unexpected challenges occur. Schedule feasibility ensures that a project can be completed before the project or technology becomes obsolete or unnecessary. As our system is taking care of present requirements we need to ensure each and every time that each module of the system should be completed on time and will be more efficient than previous systems. Thus schedule feasibility has been recovered.

**3. PROPOSED SYSTEM**

**3.1 DRAWBACKS OF CURRENT SYSTEM AND NEED OF PROPOSED SYSTEM**

In the current version of communicator tools we have to manually compare two language trees of different natural languages to find grammatical mistakes and in the newer version this process will be automated which will increase the efficiency and accuracy of the Communicator Tool.

**3.2 PROJECT PLANNING**

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. Project planning is often used to organize different areas of a project, including project plans, workloads and the management of teams and individuals.

Our project plan includes the study of language communicators for the language pair Hindi-English. Also, the new communicator tool master which bypasses the DMRS generation and directly goes to the MRS generation. PyDelphin is a set of Python libraries for the processing of DELPH-IN data. We got the knowledge of how to generate English from Hindi User\_csv. And, we have studied Marathi Morph and its working. For this we require two programming languages that are python and clips (C Language Integrated Production System) used for building an expert system.

**3.3 SYSTEM DESCRIPTION AND SOFTWARE REQUIREMENT SPECIFICATION**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and includes a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between users & admin on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It also provides a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirement specifications help prevent software project failure. The software requirements specification document enlists enough and necessary requirements that are required for the project development.

The SRS document is prepared as per SRS IEEE Template. This document is followed by three sections viz. project introduction consisting of purpose, scope of project, and glossary. Second section consists of an overall description which includes product perspective, product functions, operating environment, and design, implementation constraints. The third section comprises functional requirement specification.

**3.3.1.1 PURPOSE**

The purpose is to automate the process of comparison of language trees. The comparison could be of trees of any language but currently we are focusing on Indian languages.

**3.2.2 SCOPE OF PROJECT**

The project is a part of research at International Institute of information Technology Hyderabad for their open source machine translation tool “Communicator

Tool”. We work on the user csv which will going to be helpful for creating the

communicator tool more accurate.

**3.3 SYSTEM ANALYSIS**

The system analysis requirements gathering process is intensified and focused specifically on software. To understand the nature of the program to be built, the software engineer ("analyst") must understand the information domain for the software, as well as required function, behaviour, performance, and interface. Requirements for both the system and the software are documented and reviewed with

the customer. System analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for studying a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

Here the work of the system is to generate languages which can be done by using python and clips. Hindi morph readme was not available, so to overcome this we have created the readme which contains the installation steps.

**3.3.1 SYSTEM ARCHITECTURE**

The system architecture is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. The architecture is not the operational software. Rather, it is a representation that enables a software engineer to (1) analyse the effectiveness of the design in meeting its stated requirements, (2) consider architectural alternatives at a stage when making design changes is still relatively easy, and (3) reducing the risks associated with the construction of the software. A bottom-up architecture is used in the system which involves piecing together systems to give rise to more complex systems, thus making the original systems subsystems of the emergent system. It is also a composite of the design architectures of products and their life cycle processes.

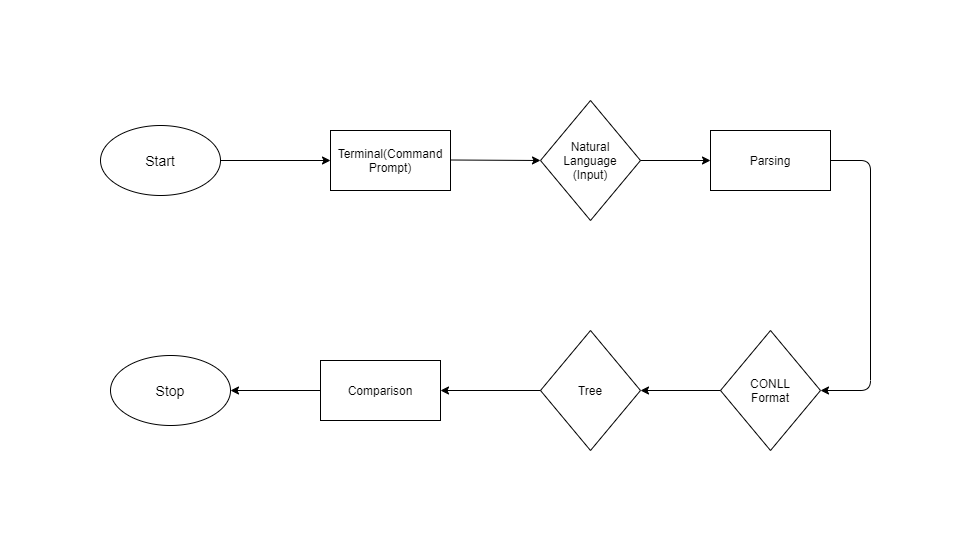
|  |
| --- |
|  |

**Fig: 3.3.1 System Architecture**

**3.3.2 FLOWCHART OF SYSTEM**

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows.

The below diagram shows the detailed working of the communicator tool and further use of output data in the language generator.

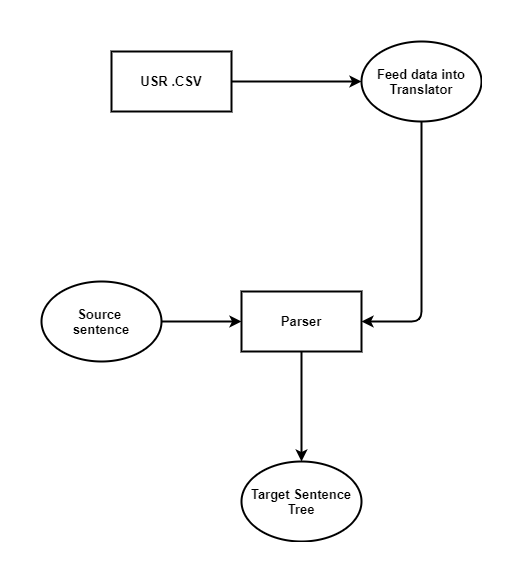


**Fig: 3.3.2 Flow Chart**

**3.3.3 DATA FLOW DIAGRAM (DFD) OF THE SYSTEM**

A data flow diagram is a graphical representation of the ‘flow’ of data through an information system, modelling its process aspects. The functional model consists of multiple DFD. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. A level 0 data flow diagram (DFD), also known as a context diagram, shows a data system and emphasizes the way it interacts with external entities.

In level 0 DFD diagram there will be an entity script which will provide required output data which is then feed to the translator. The following figure shows the Level 0 DFD of the system.



**Fig: 3.3.3.1 DFD Level 0**

A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into subprocesses that can then be analysed and improved on a more intimate level.

The level 1 DFD consists of entities developer’s user will input source language and the generator will give the target language as output.

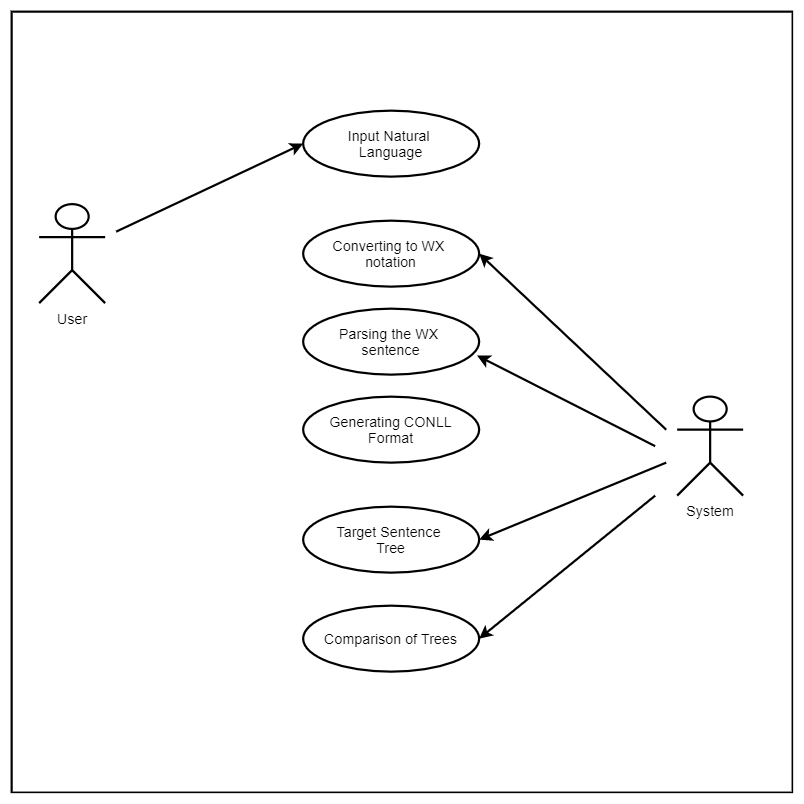
|  |
| --- |
| DFD level 1.png |

**Fig: 3.3.3.2 DFD Level 1**

**3.3.4 USE CASE DIAGRAM**

A Use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

Here the developer will develop scripts to extract data from websites which is given to the system. The user will provide input source language and the system will process and give the output target language.



**Fig: 3.3.4 Use case Diagram**

**3.3.5 HARDWARE REQUIREMENTS**

* Processor: Intel Celeron or higher
* RAM: 512 MB Ram or higher
* Disk Space: 1 GB of free disk space

**3.3.6 SOFTWARE REQUIREMENTS**

* Windows XP or later, Ubuntu
* Python 3.6 or above

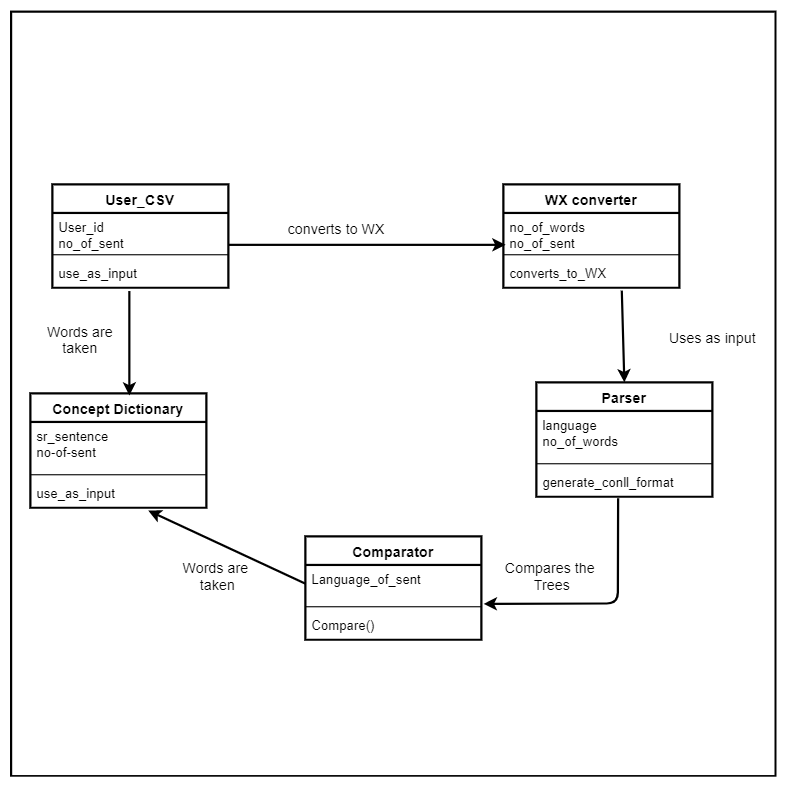
**3.4 SYSTEM DESIGN**

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. The logical system design arrived because systems analysis is converted into physical system design. Normally, the design proceeds in two stages: (1) Preliminary or General Design, (2) Structured or Detailed Design.

* + - 1. In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to the detailed design stage.
  1. In the detailed design stage, computer-oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structure design is a blueprint of a computer system solution to a given problem having the same components and inter-relationships among the same components as the original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. These tools and techniques are: (1) Flowchart, (2) Data Flow Diagram, (3) Data Dictionary, (4) Structured English, (5) Decision Table, and (6) Decision Tree etc.
  2. The system design is of three types which are architectural design, logical design and physical design. The architectural design of a system emphasizes on the design of the systems architecture which describes the structure, behaviour, and more views of that system and analysis. The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system.

**3.4.1 CLASS DIAGRAM**

The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling.



**Fig: 3.4.1 Class Diagram**

In the class diagram, there are five classes: User csv, concept dictionary, parser, WX converter and language comparator. The user will have attributes known language and user\_id and operation as insert sentence and view output. The concept dictionary is a script which has attributes such as output file and filename and operations such as extraction of data and formatting of data. Parser class will have the attribute as language and number of words in the sentence. The WX converter class have attributes as number of sentences and words and operation to convert natural language in to WX notation.The comparator have attribute language of the sentence and operation is comparing trees.

**3.4.2 COMPONENT DIAGRAM**

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system.

Here, in this the source sentence and the user csv both are sent to the language translator which is used to generate targeted language.

|  |
| --- |
| C:\Users\SUMEET SINGH\AppData\Local\Microsoft\Windows\INetCacheContent.Word\component.jpg |

**Fig: 3.4.2 Component Diagram**

**3.4.3 DEPLOYMENT DIAGRAM**

Deployment diagrams are typically used to visualize the physical hardware and software of a system.

This below diagram shows how our project communicator tool deals with the different servers which are used in this.

|  |
| --- |
| C:\Users\SUMEET SINGH\AppData\Local\Microsoft\Windows\INetCacheContent.Word\1.jpg |

**Fig: 3.4.3 Deployment Diagram**

**3.4.4 USER STORIES**

**Example: Visit to America by an Indian**

Suppose an Indian man went to America to spend his vacation for the first time with his family. After he arrives in America he needs some language generator tool so that he can communicate easily with the hotel staff or locals of the area. Whenever he visits some local place or a nearby market, of course he needs to communicate in English as the locals cannot talk to him in Hindi. Thus, at that time he can use our communicator tool to resolve his problem.

This Project aims to bridge this language barrier by allowing a user to enter a Hindi or Marathi text into the Communicator tool and generate it, in an English language.

1. USER FUNCTIONS

* Enter a Sentence in Hindi or Marathi
* Enter an Idiom/Phrase

1. COMMUNICATOR TOOL FUNCTIONS

* Generate Hindi, Marathi sentence to English Sentence.

**4. CONCLUSION AND FUTURE SCOPE**

The phase of development of the Communicator tool for generating language which is more accurate and has less ambiguity is well performed. The main aim behind this is to make the system able to find out errors, grammatical mistakes and ambiguity in the language and generate precise translation of the source language into target language. It attempts to give birth to a new efficient system that satisfies the current needs of the user and the scope for future growth within the organizational constraints.

The project is a part of research at International Institute of Information Technology, Hyderabad for their indigenously developed open source language translator tool” COMMUNICATOR TOOL”.

Our work is the first step in the process to incorporate these methods for Marathi-English language pairs in near future. The coming generations at LTRC labs will not be facing issues regarding the manual analysis and manual correction/error finding for generation of a multilingual languages.

**5.REFERENCES**

1. Sriram Chaudhary, “Anusaaraka: An Expert System based MT System”, in the proceeding of IEEE conference on Natural language processing and knowledge management, Published 2010.
2. Akshar Bharati, “Anusaaraka: Overcoming the Language Barrier in India”,

appeared in “Anuvad”, Published: New Delhi, 2002.

1. Naskar S. and Bandyopadhyay S. “Use of machine translation in India: status.

machine translation Review 15”, Published: Dc 2005.

1. Badia, T., À. Egea & T. Tuells CATMORF: Multi-two level steps for Catalan morphology. In Demo Proceedings of the Conference on Applied Natural Language Processing. Published: Washington ,1997

**STUDENT INFORMATION**



|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **ROLL** | **MOBILE** | **EMAIL ID** |
|  | **NO.** | **NO.** |  |
|  |  |  |  |
| GAURAV | 44 | +91 | gaurav.bhisikar@gmail.com |
| BHISIKAR |  | 7798827901 |  |
|  |  |  |  |
| ADITYA | 36 | +91 | adityadale39@gmail.com |
| DALE |  | 9503553426/  8788833074 |  |
|  |  |  |  |
| SAGAR | 56 | +91 | shorya848army@gmail.com |
| MALIK |  | 9370756840 |  |
|  |  |  |  |