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

1. **Abstract:**

This project aims to automate the process of comparing the two trees generated from natural language by the parser. It is a part of research at the 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.

Communicator tool is a multilingual translation tool in which we give natural languages as an input which gets translated 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 parser is a compiler or interpreter component that breaks data into smaller elements for easy translation into another language. The parser generates parse tree of a particular sentence containing parent node and child node and having relations i.e. cases, compound, VBS, NNS etc. between them.

Thus, we have used Irshad Parser and Stanford Parser for generating CONLL format through which we generate trees from natural language

At one stage, this tool uses IRSHAD Parser and STANFORD Parser for comparing the trees and the output is in CONLL format. A parser takes the input in the form of a sequence of tokens or programs and instructions and usually build the data structure in the form of a parse tree or an abstract syntax tree. The parse trees are then compared to resolve the obligations arise while comparing the nodes of the tree. The output generated is in the compressed form of the sentence.

1. **Hardware & Software requirements:**

Operating System: Ubuntu

Software Tools: Python 3.6 and above.

Processor: Core i5(equivalents or above)

Internal Memory: 6 GB (minimum)

GPU: Nvidia/AMD(optional)

1. **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 discusses about Apertium Platform,

which is an open-source shallow transfer machine translation system, which was

initially designed for translation between related language pair.

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

in India ” it has been written about all the rows of user CSV’s as well as about

concept dictionary. 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.

a) Each field in a row should be separated by comma(,)

b) If a field has no value then the only comma needs to be typed.

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

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

2. Concept dictionary: 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 is in the respective language columns. The concept dictionary as of now has Hindi, Eng, German, Japanese, Tamil and Marathi.

3. Indexing: The 3rd row is for IDs for words.

4. Ontological information: In the user CSV, the user has to give the information about the definiteness of the noun in 4th row.

5. Morphology: GNP information is mentioned for nouns, proper nouns and pronouns.

6. Intra chunk: The intra chunk relation is the dependency relation between the tokens of a chunk.

7. Inter chunk: 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.

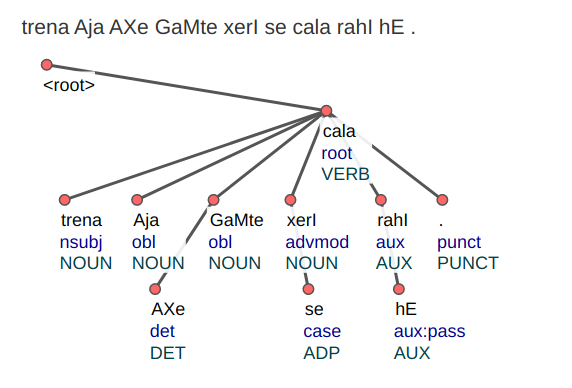
8. Discourse relation: Relations such as emphasis, delimitation, co-referencing are marked in this row.

9. Type of sentence: The 8th row shows the type of the sentence.

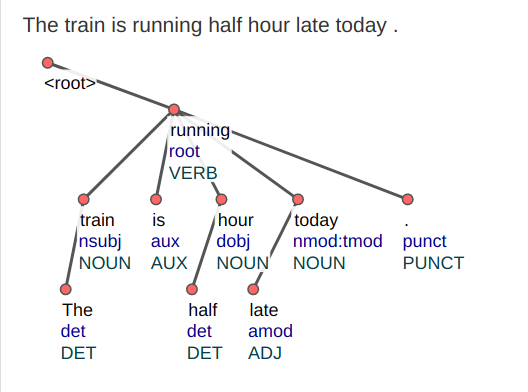
**4) Example:**

**Natural Language:**

1. **Hindi(Represented in WX notation)**

****

1. **English**

****

**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.

