Project Outline NLP
Neural Dependency Parser

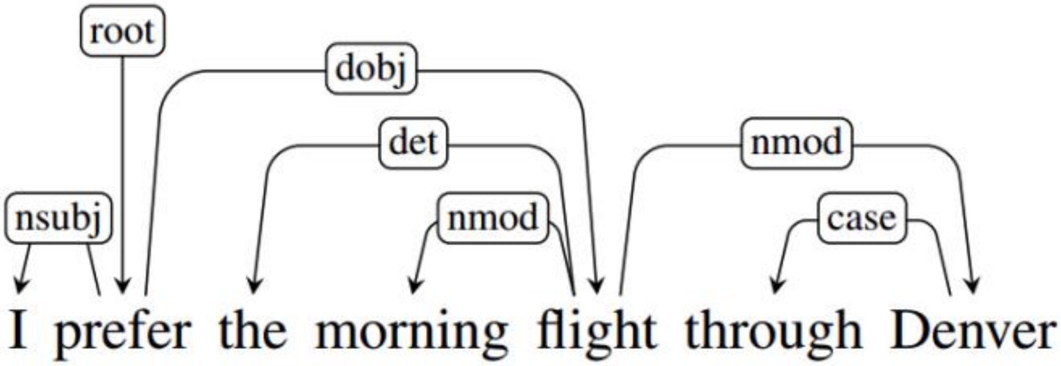
Team No:53
Team Name: NSU_NLP
Member 1: Nikhil Chawla (2022201045)
Member 2: Shubham Deshmukh (2022201076)
Member 3: Udrasht Pal (2022201020)

Problem Statement

In this project, we are creating a dependency parser for the **Hindi Language**. A dependency parser identifies the syntactic dependency between words in a sentence. Although there is considerable literature on dependency parsing within the NLP community, there is a scarcity of robust tools designed specifically for Indian languages.

Neural Dependency Parser

A dependency parser analyzes the grammatical structure of a sentence, establishing relationships between "head" words and words that modify those heads. The figure below shows a dependency parse of a short sentence. The arrow from the word “flight” to the word “Denver” indicates that “Denver” modifies “flight”, and the label “nmod” assigned to the arrow describes the exact nature of the dependency.



A dependency parsing mechanism can give a parsed dependency tree for a given sentence, which involves a set of relations over its words such that one is a ‘dependent’ of the other.

Our Methodology

We are developing a dependency parser for the Hindi language. The parsing can be either **Transition-based or Graph-based**. In this project, we are using a **Transition-based parsing approach.**

Graph-based Dependency Parsing: A method where we score individual connections between words in a sentence to find the best overall structure, like a tree, that shows how words relate to each other.

Transition-based Dependency Parsing: This parser creates sentence structures by reading words one by one. It uses a stack to track words being processed and a buffer for words to process. It applies three actions—LEFT-ARC, RIGHT-ARC, and SHIFT—to build the structure. A neural network decides which action to take based on the current state. The training uses an oracle to determine the correct actions for each step in the training data.

Why did we choose Transaction-based parsing?

Advantages of Transition-based Parsing:

- Faster processing.
- Easier implementation.
- Works well for incremental parsing.

We are still exploring more about Transition-based Dependency Parsing by reading the research papers

Dataset

We are planning to use two types of datasets which are:

1. Data corpus taken from LTRC, IIIT Hyderabad
   Link to the dataset: https://ltrc.iiit.ac.in/treebank_H2014/
   The Hindi Dependency Treebank (HDTB) is a linguistic resource created to facilitate the study of the syntactic structure of Hindi sentences. It consists of manually annotated sentences with syntactic dependencies, providing a detailed analysis of the grammatical relationships between words in Hindi.
   The HDTB was developed by the Language Technology Research Center (LTRC) at the International Institute of Information Technology, Hyderabad (IIIT-H), India. Creating the HDTB aims to support research and development in natural language

processing (NLP) tasks specific to Hindi, such as parsing, machine translation, information retrieval, and sentiment analysis.

2. Data corpus from UD Hindi HDTB
   Link to the dataset:
   https://universaldependencies.org/treebanks/hi_hdtb/index.html
   UD Hindi HDTB refers to the Universal Dependencies (UD) Treebank for Hindi, which follows the Hindi Dependency Treebank (HDTB) annotation guidelines. Universal Dependencies is a framework for cross-linguistically consistent grammatical annotation of languages, aiming to provide a unified syntactic representation across different languages.It consists of manually annotated sentences with syntactic information such as word lemmas, part-of-speech tags, and dependency relations. The annotations in the HDTB follow the UD guidelines, ensuring consistency with other treebanks in the Universal Dependencies project.

Note: All the data will be used only after preprocessing.

Evaluation Metrics:

Two types of evaluation Metrics will be used:-

1. Unlabeled Attachment Score (UAS)
   UAS measures the percentage of correctly predicted dependencies between words in a sentence, regardless of the specific dependency labels. It considers a dependency as correct if the predicted head of a word matches the actual head.

$$\mathrm{UAS} = \frac{\text{Number of Correctly Predicted Dependencies}}{\text{Total Number of Dependencies}} \times 100\%$$

2. Labeled Attachment Score (LAS):
   LAS extends the concept of UAS by considering not only the correctness of predicted dependencies but also the correctness of the dependency labels. It measures the percentage of correctly predicted dependencies with correct labels.

$$\text{LAS} = \frac{\text{Number of Correctly Predicted Dependencies with Correct Labels}}{\text{Total Number of Dependencies}} \times 100\%$$

Timeline:

The tentative timeline for the project is as follows:

Date	Target
11 March 2024	Read related research papers and explore different datasets
25 March 2024	Data Collection and Preprocessing, Reading more research papers if required
4 April 2024	Design and implement the model on base case.
15 April 2024	Train and evaluate the model
20 April 2024	Fine Tuning the model and comparison with different older models.
6 May 2024	Final submission along with report and presentation.

References:

1) A Fast and Accurate Dependency Parser using Neural Networks
   https://aclanthology.org/D14-1082.pdf
2) Transition-based Dependency Parsing with Rich Non-local Features
   https://aclanthology.org/P11-2033.pdf
3) DEEP BIAFFINE ATTENTION FOR NEURAL DEPENDENCY PARSING
   https://arxiv.org/pdf/1611.01734.pdf

Dataset:

1. HDTB Data corpus taken from LTRC, IIIT Hyderabad
   https://ltrc.iiit.ac.in/treebank_H2014/
2. Data corpus from UD Hindi HDTB
   https://universaldependencies.org/treebanks/hi_hdtb/index.html