Semantic Role Labeling in Hindi

Mohit Sharma - 2022201060

Neeraj Asdev - 2022201056

Hrishikesh Deshpande- 2022201065

Introduction

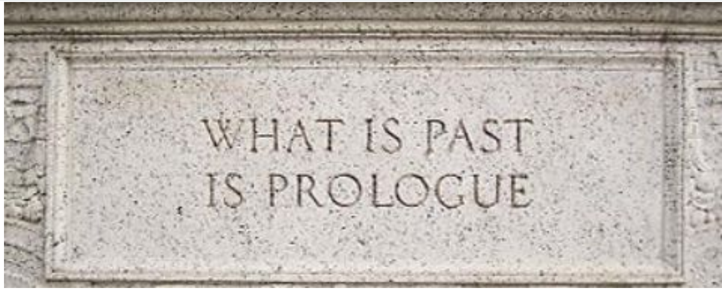
Semantic Role Labeling (SRL) in NLP identifies word relationships and roles in events which is vital for tasks like question-answering and inference etc.

Our SRL project focuses on Hindi language specifically aiming to enhance existing systems by accurately assigning roles.

Our goal is to develop a Hindi SRL system using statistical/neural models to label arguments in sentences.

Why?

1. SRL improves various NLP applications like ***question-answering***, ***inference,*** and ***knowledge graph creation*** by providing deeper insights into sentence structures and meanings.
2. SRL aids in better translation by preserving the intended meaning and syntactic structure across languages, leading to more accurate and *contextually relevant* translations.
3. Greater accessibility in multiple languages.
4. Catering to diverse linguistic communities (especially in nations like India)



Previous Work

➔ Introduction of statistical method by first identifying the arguments related to a given verb in the input sentence, then categorizing them into roles.(*Nomani, et al.*).

This study achieved 58% precision and 42% recall for classifying constituents into their semantic roles.

➔ Proposal of new features and modifications over baseline with introduction of supervised semantic role labeler. (*Shrivastava, et al.*)

Datasets

- **Hindi Propbank**
  - This dataset, specifically designed for Semantic Role Labeling tasks in Hindi
  - It is a part of the Hindi-Urdu PropBank project which involved building a multi-representational and multi-layered Treebank for Hindi-Urdu
- **Created custom dataset based on propbank**
  - Comprising around 14,000 tokens of Hindi text
  - Information like head POS, dependency from Propbank
- **Collected additional dataset of 1.3k Hindi sentences along with arguments, SRL labels and dependency relations**

Experiments

1. Dataset Preparation
2. Statistical models
3. Neural models
4. Evaluation
5. Final Results

Dataset Preparation

- Collected data from sources such as Propbank, available in a tree-like structure from GitHub of previous similar works
- Filtered relevant tokens for our experiments, ensuring to gather additional information about each token
- Extracted sentences from the Propbank dataset, making sure to retain all associated labels, including argument POS tags, head POS, and SRL tags.
- Dependency/कारक relation signify the relationship between words in a sentence in Hindi.

Statistical models

- Linear Support Vector Classifier (LinearSVC)
- Known for its effectiveness in handling high-dimensional data
- Commonly used for text classification tasks
- Three different sets of input features, each capturing different aspects of the input data:
    - Includes features such as the word, whether it's an argument, the predicate, and the postposition.
    - Feature include the word, postposition, whether it's an argument, the predicate, and the head-POS.
    - Comprises features such as the word, dependency, postposition, whether it's an argument, the predicate, and the head-POS.

Neural network models

1. FastText Embeddings (Non-contextual) + BiLSTM Classifier
2. FastText Embeddings(Non-contextual) + Dependency Relation + BiLSTM Classifier
3. Indic-Bert (Contextual) + MLP Classifier
4. Indic-Bert (Contextual) + BiLSTM Classifier
5. Indic-Bert + Dependency Relation + Bi-LSTM Classifier

FastText Embeddings + BiLSTM Classifier

- Incorporated FastText embeddings into a Bidirectional Long Short-Term Memory (BiLSTM) classifier
- FastText embeddings offer a computationally efficient means of representing words in vector space.
- Non-contextual.

FastText Embeddings + Dependency Relation + BiLSTM Classifier

- Augmented the input features with dependency relations extracted from the dataset
- Incorporated these relations alongside FastText embeddings
- Dependency relations provide valuable syntactic and semantic information about the relationships between words in a sentence.

Indic-Bert + MLP Classifier

- Leveraged the powerful representations learned by the Indic-Bert model from *AI4Bharat*.
- It is a transformer-based architecture specifically trained for Indian languages
- Embeddings obtained from Indic-Bert were fed into a Multilayer Perceptron (MLP) classifier for SRL classification
- This architecture allowed us to benefit from the contextual understanding encoded by Indic-Bert while maintaining flexibility in the classification layer.

Indic-Bert + BiLSTM Classifier

- Combined Indic-Bert embeddings with a BiLSTM classifier
- This architecture aimed to capitalize on the strengths of both transformer-based models and RNNs.
- Indic-Bert embeddings provided rich contextual information, while the BiLSTM layer offered additional sequential modeling capabilities
- Trained models were evaluated on a separate test dataset to assess their performance

Indic-Bert + Dependency Relation + Bi-Lstm Classifier

- Introduced the integration of Indic-Bert embeddings with dependency relations into a Bidirectional Long Short-Term Memory (BiLSTM) classifier
- Aimed to enrich the model's understanding of syntactic and semantic relationships within sentences
- BiLSTM layer further enhanced the model's ability to capture sequential dependencies, complementing the contextual knowledge encoded by Indic-Bert

Evaluation - Baseline Model

- Utilized a straightforward approach by considering only the most fundamental features :
    - Word
    - POS Tag
    - Is Argument (or NOT)
- These features serve as the foundation for our classification task, providing essential information about each word's identity, grammatical function, and role within the sentence.

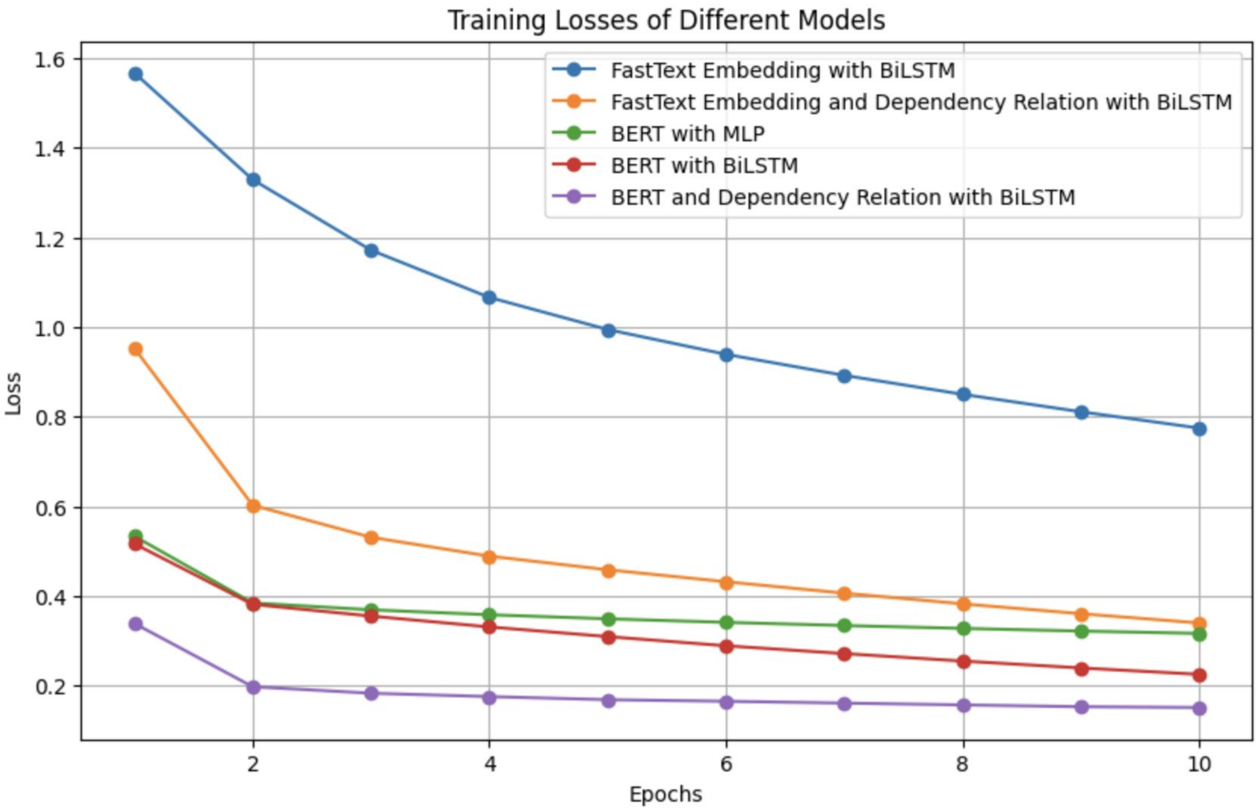
Evaluation - Statistical models

- Experimental Results

Model	Features	Accuracy	Precision	Recall	F1 Score
Model 1	Baseline	76.94%	69.27	76.94	71.95
Model 2	Baseline + 'Head POS Tag'	77.22%	73.41	77.22	72.75
Model 3	Baseline + 'Head POS Tag' + 'Dependency'	81.27%	77.53	81.27	78.00

Evaluation - Neural network models

- Losses



Results

Model	Accuracy	Precision	Recall	F1 Score
Fast-Text Embedding with Bi-LSTM	0.7041	0.6400	0.7000	0.6600
Fast-Text Embedding and Dependency Relation with Bi-LSTM	0.8686	0.8600	0.8700	0.8600
BERT with MLP	0.8987	0.8524	0.8987	0.8681
BERT with Bi-LSTM	0.9007	0.8670	0.9007	0.8823
BERT and Dependency Relation with Bi-LSTM	0.9534	0.9510	0.9534	0.9516

Challenges

- Scarcity and availability of suitable datasets for SRL tasks in Hindi
- Shortage of resources and references in this domain
- More understanding and clarity about Hindi grammar required and probably about more Indian languages.
- A small dataset is not so reliable at all. Some metrics MAY exhibit unusual behavior (like *accuracy* and *recall* in our case).

Future Works

- Exploring Different BERT Layers for Embeddings.
- Testing different layers helps identify which ones capture the most important semantic features for SRL.
- Typically, layers closer to the input focus on fine syntactic details, while deeper layers capture broader, higher-level meanings.
- By carefully choosing BERT layers to extract embeddings, SRL models can balance computational efficiency with semantic richness.

Thank You