Semantic Role Labeling for Indian languages
Final Submission

Team Name: Semantic Sentinels
Team Number: 55

1. D Priyanka - 2023814003
2. S Monica - 2023802005
3. CV Thirumala Kumar - 2023702020

1 Introduction

Semantic Role Labeling (SRL) is the task of assigning labels to words or phrases in a sentence that indicate their roles in relation to the main verb, such as ”agent”, ”patient”, ”instrument”, etc. In other words, Semantic role labeling is the study of determining WHO did WHAT to WHOM, WHERE, HOW, WHY, and WHEN in a sentence, adding a layer of semantic annotation that helps in understanding the meaning and structure of sentences in natural language processing tasks. The main verb/action in a phrase is referred to as the predicate. Arguments are words or phrases that are semantically related to the predicate, representing entities or roles involved in the expressed action or state. These arguments include subject, object, indirect object modifiers, and other syntactic constituents like adjuncts that contribute to the meaning and structure of the sentence. The identification and classification of arguments relative to the predicate form the basis of semantic analysis tasks such as Semantic Role Labeling (SRL).

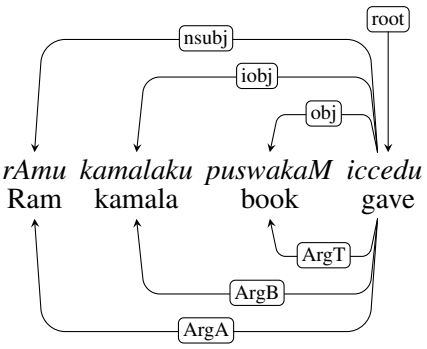

Figure 1: ’Ram gave a book to Kamala’

Let’s understand SRL more clearly with an example explained in Figure 1. In the sentence, the verb ‘gave’ being a ditransitive verb takes 3 arguments i.e. subject, object and indirect object. These arguments are the core arguments of the verb and can be assigned semantic roles based on their syntactic structure. To elaborate more, nsuj ‘Ram’ being the doer of action i.e. Agent [ArgA], iobj ‘Kamala’ being the indirect object is the Beneficiary [ArgB] of the event and the action of giving a book i.e. object is the theme of the sentence. In such a way semantic roles can be assigned to all arguments of predicate. Table 1 explains the core semantic arguments of the predicate along with the description.

Table 1: Core Semantic role labels

Tags	Core Arguments	Description
ArgA	AGENT	The volitional causer of an event
ArgT	THEME	The participant most directly affected by an event
ArgEx	EXPERIENCER	The experiencer of an event
ArgB	BENEFICIARY	The beneficiary of an event
ArgR	RECIPIANT	Receive something (whether good or bad) in an event
ArgI	INSTRUMENT	An instrument used in an event
ArgLOC	LOCATION	A locative or path prepositional phrase introduces an underspecified location.
ArgTOP	TOPIC	Conversation or message transfer is dealt with using communication verbs
ArgG	GOAL	Verb's destination or goal argument that indicates the motion's endpoint.
ArgS	SOURCE	arguments that can be thought of as a source or beginning point for the verb's event

2 Scope of SRL

The task of SRL is to identify the roles in the sentence so that downstream NLP tasks can "understand" it. An idea can be expressed in several ways. Consider the following two phrases which indicate the same thing: "Yesterday, Helen hit the Ball with a bat "; "Ball was hit by Helen yesterday with the bat. With a Bat, Helen Hit the Ball Yesterday. Helen hit the Ball with a Bat Yesterday". These sentences will be syntactically analyzed by either constituent or dependency parsing. However, syntactic relationships aren't always helpful in determining semantic roles. An analogy is one method of understanding SRL. In image captions, we figure out what are the main objects, how they're connected, and what's happening in the background. SRL does a similar job but with regular unstructured text instead of pictures. This type of comprehension is more than just a matter of syntax. Parsing, on the other hand, is helpful for SRL. A parse tree is used in a traditional SRL pipeline to identify the predicate arguments. SRL can be used in any NLP application that requires semantic understanding, including machine translation, information extraction, text summarization, and question answering. For example, Predicates and heads of roles help in document summarization. SRL is used to construct extraction rules for information extraction. Many NLP tasks benefit from semantic knowledge, such as question answering, dialogue systems, machine reading, machine translation, text-to-scene generation, and social network analysis.

3 Literature review

There has been much attention and a lot of contribution to the semantic analysis of languages in the last decade. For major languages such as English, a large amount of work has been done, including efforts to create semantically annotated data such as The Proposition Bank (PropBank) [7]. The Proposition Bank project proposes a practical approach to semantic representation by layering predicate-argument information, or semantic role labels, on top of the Penn Treebank's syntactic structures. Each sentence in the PropBank can be thought of as an event(s) with participants, similar to a predicate with arguments. This is done at the phrase (chunk) level. However, in Indian languages, very few works exist to date for languages such as Hindi, Urdu and no work exists for other languages. We saw a need for improvement in this domain, therefore we attempt to build a new system with a new set of features that significantly improve the classification of Telugu semantic roles.

Maaz et al. [1] build a statistical system for identifying the semantic relationships or semantic roles for two major Indian languages, Hindi and Urdu. The approach is a 2-stage architecture in which, first the arguments pertaining to a predicate in a sentence are identified by the system and then those identified arguments are classified into one of the PropBank semantic labels. Their system uses a basic Logistic Regression machine

learning algorithm for identifying the predicates and Support Vector Machines to classify the arguments of a predicate into semantic labels. Aishwary et al. [6] extended the work on the same dataset and their experiments showed a reasonable improvement with respect to the current baseline for Hindi [1], mainly for the classification step and significant improvements for the Argument Identification task in Urdu. Creates a new baseline for the Hindi using 5-fold cross-validation. In [**?**], a Rule-Based Probability Assigner is proposed, which assigns probability values for the tags of phrases based on heuristics specific to the Tamil language.

4 Dataset

In the project outline, we mentioned that we will work on SRL (Semantic Role Labeling) for the Telugu language. But after searching for the dataset, we couldn't find any dataset in the Telugu language. We found 2 datasets which are for languages Hindi and Urdu. Hindi full dataset is not publicly available, only a part of the dataset is available in a GitHub repo (https://github.com/tanvi2612/SRL) which consists of around 1100 sentences which is small. Whereas for Urdu full dataset is available publicly at http://ltrc.iiit.ac.in/hutb_release. The total number of sentences in this dataset is 9949, which are available in SSF format. There are 22 roles labeled in this dataset that are not balanced. As full Urdu dataset is available publicly, we are using Urdu data for this project.

5 Baseline

Following the works [1, 6], we have implemented a baseline based on a 2-stage cascaded system, which has **stage 1**: an argument identifier to identify arguments and **stage 2**: an argument classifier for the classification of roles. Both of these classifiers are SVM-based classifiers. Features used for training the classifiers are 1) Predicate, 2) Head word as an embedded vector, 3) Chunk type, 4) Head word PoS (Parts of speech), and 5)Dependency relation. For extracting head word embeddings we have used fasttext [5] pre-trained models available at https://fasttext.cc/docs/en/crawl-vectors.html. The embedding dimension is 300. As there are no pre-trained models available, we have implemented this baseline from scratch and the code used for training is provided in the submission zip file. We have trained baseline SVM using this dataset with train and test splits as 85:15 and the performance is measured in terms of usual classification metrics such as precision, recall, and f1-score, since classes are not balanced we report weighted metrics in Table 3. We have also trained SVM for the Hindi data mentioned above following the same setup for reference purposes. As the dataset is small we are not reporting the performance in this submission.

6 Proposed Approach

Due to the lack of datasets, SRL is not well explored for Indian languages. So far dataset is available for only 2 Indian languages Hindi and Urdu. Existing works used a 2-stage method using SVM-based classifiers which are trained on hand-crafted features based on pos tags and dependency relations. These systems require an automatic pos tagger and dependency parser when testing on unseen data. Since the datasets have pos and dependency annotations, existing works didn't explore the effect of automatic pos tagging and dependency parser. Considering all these challenges/drawbacks of existing systems, we propose an end-to-end solution for SRL in Indian Languages.

In recent times, transformer models such as Bert become popular in the field of NLP for providing contextual embeddings of words and are proven to perform better for many downstream NLP tasks such as PoS tagging, NER etc. In this project, we propose to finetune the Indic-Bert [3] model as an initial attempt towards end-to-end SRL for Indian languages. Since these transform models require a decent amount of data to perform better, we propose data augmentation to create multilingual SRL data from Urdu. All the steps are explained in detail as follows.

1. **Multilingual SRL Data creation (Data augmentation)**: We translate Urdu sentences to 3 other Indian languages such as Hindi, Tamil and Telugu using available machine translation models.

2. **Labelling Multilingual SRL**: To label the translated data, we first align the translated sentences with Urdu sentences using awesome-align [4] (https://github.com/neulab/awesome-align). And transfer the roles from Urdu to other languages using these alignments. This will generate multilingual labeled SRL data.

3. **Finetuning Indic-Bert**: We finally fine-tune the multilingual Bert based models using this multilingual labeled data. We fine-tune with SRL data as a token classification task where each word will be classified as one of the output classes. Since we are assigning a tag to each token, we will append a few extra classes such as Not-An-Argument (NAA) (for chunk head which don't have srl label), Not-A-Head (NAH) (for words which are not chunk heads) and Predicate to the existing SRL tags.

4. **Evaluation**: We report classification performance language-wise in terms of f1-score. Since we cannot retrieve chunk information etc. from Urdu to other languages as alignments will not be straightforward, we will compare the performance of only Urdu with the above-mentioned SVM baseline for comparison.

7 Experimental setup

We choose to fine-tune 2 models for srl task which are (1) Indic-Bert[3] and (2) Multilingual-Bert [2]. Both of these has Indian languages in their training data and are suitable for multilingual tasks. We fine-tuned these models using Hugging Face toolkit with batch size of 16 per device and learning rate of 0.00002 utilising 2 GPUs. We trained models for 20 epochs with early stopping with patience of 5 to avoid overfitting. Since these models are trained on subword-level which are based on BPE and has their own tokenizers, we finetune each of these models in 2 configurations, in which we label all subwords of the words to the same tag as one configuration and labelling only the initial subword with word tag as another configuration and name them as mentioned in the Table 2 for referring. We split the data as 70:15:15 for train, validation and test sets respectively and we make sure that all test sentences are translated versions of same urdu sentences, so that there won't be any data leakage of test set.

Table 2: Model Descriptions

Model Name	Base Model	Decsription
Indic-bert	Indic-bert	labelling only first subtoken of the word
Indic-bert -lat	Indic-bert	labelling all subtokens of the word
multilingual-bert	multilingual bert	labelling only first subtoken of the word
multilingual-bert-lat	multilingual bert	labelling all subtokens of the word

Table 3: Comparison of SVM with the proposed approach

Model	Weighted F1		Macro F1	
	Argument Identification	Argument Classiffication	Argument Identification	Argument Classiffication
Indic-bert	0.60	0.54	0.59	0.22
Indic-bert-lat	0.65	0.60	0.63	0.23
multilingual-bert	**0.71**	**0.68**	**0.70**	**0.34**
multilingual-bert-lat	0.70	0.67	0.69	0.31
SVM	**0.77**	**0.64**	**0.77**	**0.32**

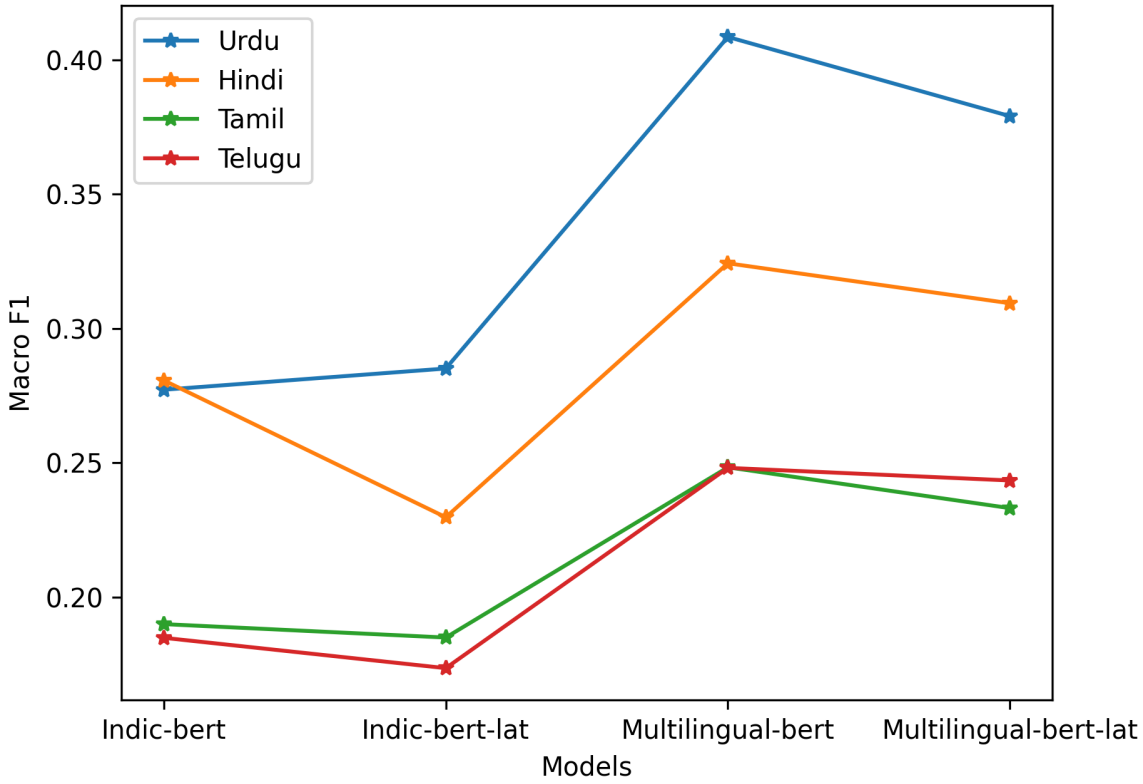

Figure 2: Language-wise performance comparison (Macro F1)

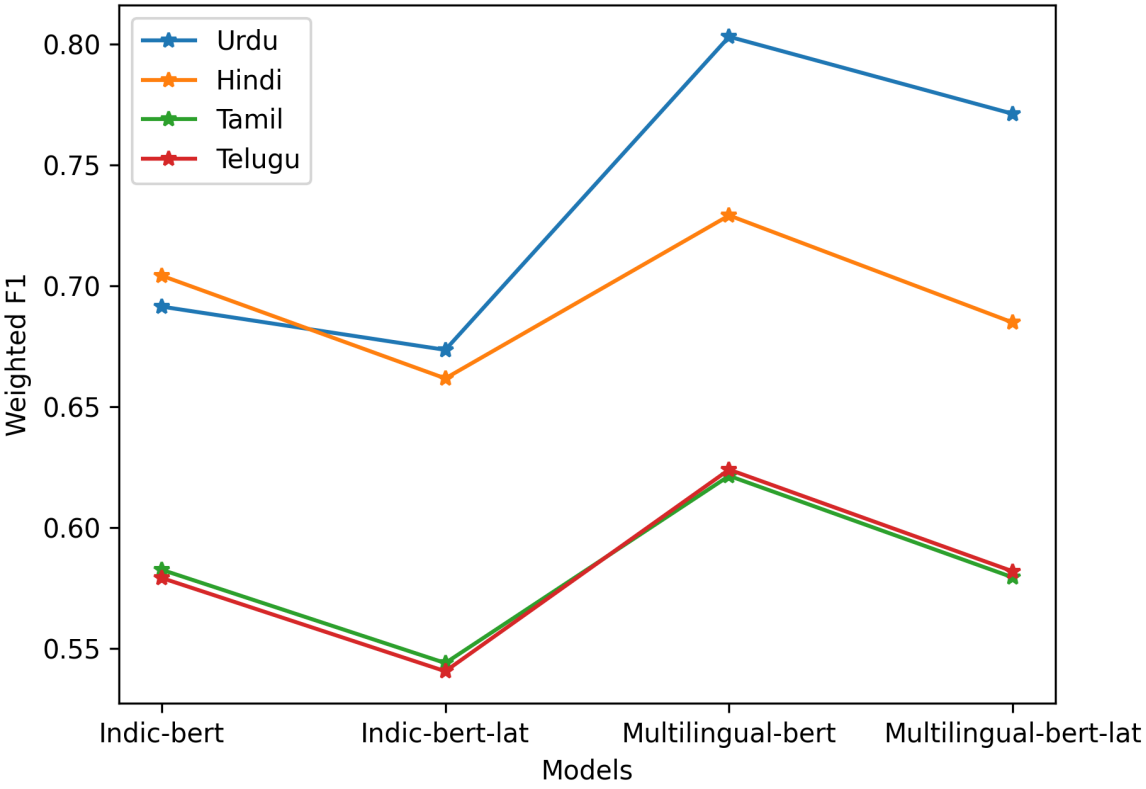

Figure 3: Language-wise performance comparison (Weighted F1)

8 Results

We report performance in terms of macro and weighted f1 scores, as classes are not balanced and this impact the performance of the models in classifying the tags. So we report the earlier mentioned 2 metrics for comparison.

8.1 Baseline vs Proposed approach

We compare the proposed approach and baseline for Urdu language since baseline can be trained on only Urdu as it requires additional information such as PoS tags, dependency relations etc which cannot be obtained for the multilingual augmented data. Figure 3 shows the comparison of performance of baseline SVM and proposed approaches. Though proposed models are trained for end-to-end token classification task, for comparison purpose we generated results for argument identification by considering all srl tokens as arguments and NAA (Not an argument) as non-arguments and argument classification by considering only srl tags by leaving non-srl tags such as predicate, NAH (Not a head), NAA (Not an argument). It can be seen that multilingual-bert model shows performance which is close to svm baseline and the difference is not significant. The reason for the small decrement in the proposed approach can be hypothesized as data insufficiency. As the available data is very small for token classification and also the task is now complex since it has to predict for each word which is not the case in SVM baseline. Also SVM is utilising extra information like pos tags, dependency relations etc., which are not used for proposed approach.

8.2 Language-wise comparison

Since the proposed method is an end-to-end token classification approach, we compare the performance of the models language wise in terms of overall macro and weighted f1 scores in contrast to usual argument identification and classification for srl task. Figures 2 and 3 presents the performance of models across individual languages. We can infer that Multilingual-Bert shows better performance than others. It can be observed that all models show similar performance for languages Urdu, Hindi and Tamil, Telugu. Also there is a significant performance gap between Urdu-Hindi and Tamil-Telugu with Urdu-Hindi being highest. Since Urdu and Hindi belong to the same language family i.e. Indo-Aryan, which share similar kinds of syntactic structures they have quite similar kind of results when compared to Telugu and Tamil which are from the Dravidian language family. These linguistic variations between the languages might have impacted the performance. Over all the performance of the models across languages in decreasing order is Urdu, Hindi, Tamil and Telugu. And the best performing model is multilingual-bert.

9 Conclusion

In this project we propose multilingual end-to-end approach for semantic role labelling along with a data augmentation approach for generating multilingual srl data from a single language. The proposed approach is validated using languages Urdu, Hindi, Tamil and Telugu with source language being Urdu. Results showed that the proposed approach is performing similar to the baseline. Since, the baseline is a cascaded approach using statistical machine learning based method i.e SVM and uses information such as PoS tags, dependency relations etc., which again is the output of another models which is not a practical approach. The proposed approach is the initial attempt for multilingual end-to-end srl. Although the performance is comparable with baseline, language properties like language families impacts this approach when augmented data from only one language.

All codes were submitted in the zip file. Due to size constraints model files are made available in this drive folder

References

[1] Maaz Anwar and Dipti Misra Sharma. Towards building semantic role labeler for indian languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4588–4595, 2016.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreyansh Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *ArXiv*, abs/2212.05409, 2022.

[4] Zi-Yi Dou and Graham Neubig. Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

[5] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[6] Aishwary Gupta and Manish Shrivastava. Enhancing semantic role labeling in hindi and urdu. In *The 13th Workshop on Asian Language Resources*, page 52, 2018.

[7] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.