IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Marathi Semantic Role Labeling Using Support Vector Machines

[1]Pallavi R. Deore, [2]Nita V. Patil, [3]Ajay S. Patil

[1]Research Student, [2]Associate Professor, [3]Professor

[1,2,3] School of Computer Sciences, KBCNMU, Jalgaon 425001(MS), India

Abstract: This paper describes an SRL system custom designed for the Marathi language. It employs SVM, a powerful and well-established machine learning technique for classification tasks. The system first extracts a set of linguistic features from its training data, then trains the SVM classifier on those features. It tests the accuracy and performance of the SVM classifier in relation to the test dataset which is independent of the training dataset. Therefore, in order to increase the robustness of the system, datasets tagged manually from Marathi textbooks that have rich annotation quality have been used for testing. One of the biggest strengths in this SRL system is that its feature extraction is efficient and makes it easy for the classifier to capture linguistic patterns and relationships inside the text. The more elaborate recognition techniques make the identification of the proper semantic roles along with their correct labels possible in this system. This method has a disadvantage in that it is fixed on a set of features that might limit its adaptability to other linguistic structures or types of languages. In this implementation, the system was trained on a dataset of 38,929 tokens and tested on another set of 9,631 tokens. The findings from the evaluation suggest that the methodology is highly effective, as evidenced by the system achieving a notable average recall of 84.00%, precision of 85.00%, and an F1-score of 84.00%. Such results signify that the system exhibits strong performance in accurately categorizing the semantic roles in Marathi text, demonstrating significant potential for support vector machines in semantic role labeling tasks for Indian languages.

Index Terms- Semantic Role Labeling, Support Vector Machine (SVM), Marathi, Machine Learning.

I. INTRODUCTION

NLP is a subset of AI and deals with providing machines with the ability to process, understand, and interpret human language. Inasmuch as the basis of communication between humans is on natural language, NLP therefore proves to be an extremely powerful tool for handling big texts and speech information in a more structured and meaningful way. NLP's work is marked by a methodical step through a sequence of layers of analysis, that at each step add to a more profound understanding of the language by succeeding one another based upon that gained in the previous one. These would commonly be the elementary linguistic analyses of NLP, whose processes typically entail breaking up written text into words or sentences, identifying parts of speech, and recognizing prefix and suffix morphemes. The basic steps lay down the foundation for further complex activities: syntactic analysis, wherein different sentence structures are analyzed in order to understand grammatical relationships, and semantic analysis, where meanings related to words, phrases, and sentences are retrieved.

The methods for modeling context and analyzing discussions, as the analysis progresses, further improve to capture the subtlety of language, like tone, feeling, or purpose. These NLP systems are often machine learning and deep learning algorithms that greatly increase their capacity to understand patterns in language and to

perform tasks of greater complexity, such as language translation, sentiment analysis, finding information, and answering questions.

This basic analysis consists of simple processes: understanding characters by themselves as in morphological analysis and the classification of words in a sentence as independent parts of speech since they provide the foundation for more complex analyses of sentence structure or meaning decomposition, whose results can be sentencing segments or syntactic trees that have been parsed. In a nutshell, Semantic Role Labeling is the process of giving meaning to words or phrases by answering who, what, to whom, where, and when. This process goes through two important steps: predicate identification which refers to identifying the main verb called the predicate, and, finally, identification of the components of a sentence and their labeling to conclude which roles are played by the entities involved. SRL is an essential component for applications including Question Answering, Information Extraction, Situation Recognition, Machine Translation, Opinion Role Labeling, and coreference resolution. The system has worked on a large number of projects in languages that include English, Chinese, among a few Indian languages like Hindi, Tamil, Bengali, and Urdu.

The paper's structure is given below: Section 2 introduces the definitions, methodologies, and resources involved with SRL. Section 3 explains the usage of SVM for SRL. Section 4 outlines the experimental setting used for the model. Section 5 compares the experimental results of the proposed methods against those obtained using SVM. Finally, Section 6 concludes the paper.

II. RELATED WORKS

Daniel Gildea and Daniel Jurafsky [6] pioneered automatic Semantic Role Labeling (SRL) using FrameNet data and statistical learning methods in 2002. Following that, Martha Palmer et al. [9] came up with PropBank, an extensive English corpus annotating argument structures that facilitated the development of SRL systems. They used algorithms like Support Vector Machines, Maximum Entropy Models, and Conditional Random Fields for supervised learning to predict semantic roles. Palmer and Xue [10] improved the process further by filtering out unwanted data in the preprocessing stage in 2009.

The mid 2000's saw improvements made to machine learning models involving feature engineering along with image annotations for better accuracy in SRL. The SRL task contributed significantly to their advancement in the year 2004/05 share task of the CoNLL [2]. The neural architecture of the 2010s, through RNN's, CNNs, and later transformers like BERT and GPT, continues to enhance accuracy in SRL. They expanded their focus on single-lingual to multilingual and cross-lingual SRL, which developed machine learning approaches, such as decision trees, and ensemble approaches to efficiently train SRL models.

Some methodologies divided SRL into two stages: identification of the arguments by a binary classifier and classification of them; others viewed SRL as multi-class classification or sequence tagging. Others focused more on labeling major syntactic constituents, using dependency parsing that is less demanding in terms of computational resources and time. For instance, in the year 2004, Hacioglu [7] effectively used semantic role labeling due to dependency parses being computationally cheap. Collobert and Weston [3] came forward with a model of neural network using multitask learning for SRL in 2007. In the year 2014, Roth and Woodsend [13] took the systems for SRL forward by utilizing representations of words for arguments and predicates. Then, Fitzgerald and others [5] further extended this in 2015 using neural networks that combine candidate arguments and semantic roles in a shared vector space.

In 2009, Pandian and Geetha [11] proposed an SRL system for the Tamil language. It was divided into two: the Learning Phase and the Evaluation Phase. The training of the system in the Learning Phase used a Maximum Entropy Model (MEM), utilized by important language data. In the Evaluation Phase, several specialized modules were added, which included the MEM Evaluator, Verb Frame Invoker, Rule-Based Probability Assigner, and Expectation Maximizer Component. All these modules interacted, assisting the system to power the process of identifying and correctly assigning semantic roles in Tamil texts.

Das et al. [4] took the concept of SRL forward for Bengali in 2010, mainly focusing on semantic role labeling for Bengali nouns. It contributed to a systematic understanding of semantic roles by applying the "5W" framework - who, what, when, where, and why-to Bengali. This method helped to clearly show the various

roles of meaning, thereby allowing a better and more detailed understanding of what language parts mean in Bengali.

Recently, in 2019, Pal and Sharma [8] presented an automatic SRL system for Hindi-English code-mixed tweets. It was an attempt to address the specific challenges that code-mixed languages posed. The system utilized a hybrid approach that handled the intricacies of mixing two languages in informal communication. Their work illustrated the flexibility of SRL techniques toward various linguistic and contextual settings, thereby emphasizing their suitability to modern, multilingual settings. This progression indicates the changing range of SRL systems in multiple languages, forms, and types of challenges that arise in computational linguistics.

III. SVM

Support Vector Machine [14] is a classifier intended to maximize an n-dimensional separation hyperplane such that the class would be separated perfectly from each other. Its working principal is close to neural networks; this is especially valid when applying sigmoid kernel, where the kernel represents almost a two-layer neural network. SVM is a type of supervised learning algorithm. It learns the input features along with the output labels to separate data points into two classes that are usually marked as +1 and -1. It learns the "max-margin" hyperplane so as to maximize the separation between two classes and increases generalization power by minimizing errors of classification. The hyperplane and its boundary planes are defined mathematically w.x + b =0 , (w is the weight vector, x is the data points and b is bias) and data points must be classified according to those equations. The optimization process ensures that the weight vector is minimized in magnitude and maximized for separation between classes expressed as minimize ½ ||w||2 + C∑ξi. SVM, with some tolerable errors allowed, makes use of slack variable ξ and a control parameter C to equate acceptable errors and the boundary equations, thereby modifying the classification margins [12]. In case data cannot be separated using a hyperplane, SVM can map data in a high dimension space using a non-linear kernel function so that a hyperplane can be created in this high dimension space. SVM also extends to deal with multiclass problems by breaking up the feature space into more than one subspace for dealing with problems with more than two classes.

IV. METHODOLOGY

Support Vector Machine, SVM, effectively used in SRL, sequence-based tasks. At first, there is a data loading of some pre-tagged dataset, organized into sentences. Features can be extracted over each token at the sentence for the word as such, POS, prefixes, and suffixes attributes, and others based on word neighbours. These features are then aggregated into a feature vector for each token, thereby creating a structured representation for classification.

Token labels are reduced for the purpose of easy labelling and data is divided into training and test subsets. Then, an SVM model using linear kernel is fitted for the purpose of classification on every token depending on its feature vector. The model is assessed on the test dataset using a variety of metrics, including accuracy, precision, recall, and F1-score, after learning patterns from the training dataset. To give a clear visual depiction of the model's performance in relation to these evaluation metrics, the results are usually shown in a bar plot. In the case of SVM, the core concept involves identifying the optimal hyperplane that effectively separates the different classes, or in this context, the tags. The "max-margin" strategy thus involves robust classification boundaries. Support vectors, the particular data points important in defining the hyperplane, have been of vital use in the learning mechanism that enables the model to generalize well with the precision of sequence data labeling. With a structured mechanism of feature extraction and optimization potential, SVM actually has been a very reliable technique to pursue while doing SRL tasks.

4.1 Data Preparation

For this research, a Marathi-tagged dataset was not available, so it was created using Balbharti Marathi textbooks, which were manually tagged with both pos and SRL tags. For Part-of-Speech (POS) tagging, we used the tagset developed by IIT Hyderabad [1]. Additionally, for Semantic Role Labeling (SRL), common SRL tags such as Agent, Predicate, Theme, Location, and Time were tagged using the IOBES tagging scheme. A total number of 6147 sentences are tagged.

4.2 Data preprocessing

The tagged sentences are organized as per the model requirement for that pre-processed data by tokenizing, removing punctuation marks, and stop words. Also did stemming and lemmatization. In data pre-processing tokenization and cleaning are done. Data is divided for training and testing Table 1 summarizes the dataset details, with the training set consisting of 4918 sentences, 28,000 tokens, and 13,630 distinct tokens, while the testing set has 1229 sentences, 9,731 tokens, and 5,059 distinct tokens.

Table 1. Dataset

Dataset	Number of Sentences	Number of Tokens	Number of Distinct tokens
Training	4918	38,929	13630
Testing	1229	9,731	5,059

4.3 Feature Extraction

Token-specific features:

Word (word): The token itself.

POS (pos): Part-of-speech tag of the token.

Prefix (prefix-1, prefix-2): The first 1 or 2 characters of the word.

Suffix (suffix-1, suffix-2): The last 1 or 2 characters of the word. Marathi has suffixes that indicate tense, number, gender, and case. suffixes like -ला, -ती, -ने are very indicative of grammatical role.

Is digit (is_digit): Checks if the token is numeric.

Is uppercase (is_upper): Checks if the token is in uppercase. (Less relevant for Marathi unless transliterated.)

Is title case (is_title): Checks if the first character is capitalized. (Relevant for proper nouns in transliterated text.)

Contextual features:

Previous word and POS (prev_word, prev_pos): The token and POS tag of the preceding word.

Next word and POS (next_word, next_pos): The token and POS tag of the following word.

Beginning of sentence (BOS): Marks the start of a sentence.

End of sentence (EOS): Marks the end of a sentence.

4.4 Experiment

Our final results for the manual tagged Marathi dataset is listed in Table 3. Result shows the label wise precision , recall and f1 score. Our system achieved overall 85.00% precision, 84.00% recall, and 84.00 F1 score. Fig. 1 shows the detailed workflow of our system in first step from Marathi textbooks dataset is created by tagging pos tags and SRL tags. After creating preprocessed data and it looks like below.

समीर,NNP,S-A
आकाशात,NN,S-L
पतंग,NN,S-T
उडवतो,VB,S-P
.,SYM,SYM

Feature extraction is done in next step and table 2 shows detailed output of feature extraction.

Table 2. Feature Extraction Example

Word	Features
समीर	{'word': 'समीर', 'pos': 'NNP', 'prefix-2': 'स', 'suffix-2': 'ीर', 'is_digit': False, 'is_upper': False, 'is_title': False, 'BOS': True}
आकाशात	{'word': 'आकाशात', 'pos': 'NN', 'prefix-2': 'आ', 'suffix-2': 'त', 'prev_word': 'समीर', 'prev_pos': 'NNP', 'is_digit': False, 'EOS': False}
पतंग	{'word': 'पतंग', 'pos': 'NN', 'prefix-2': 'प', 'suffix-2': 'ग', 'prev_word': 'आकाशात', 'prev_pos': 'NN', 'is_digit': False, 'EOS': False}
उडवतो	{'word': 'उडवतो', 'pos': 'VB', 'prefix-2': 'उ', 'suffix-2': 'तो', 'prev_word': 'पतंग', 'prev_pos': 'NN', 'is_digit': False, 'EOS': False}
.	{'word': '.', 'pos': 'SYM', 'prefix-2': '.', 'suffix-2': '.', 'prev_word': 'उडवतो', 'prev_pos': 'VB', 'is_digit': False, 'BOS': False, 'EOS': True}

After this process of feature extraction, the vectors produced have to undergo the process of numerical vectorization via a vectorization method that specifies the representation for the conversion into numerical vectors. After this data is split, with 80% going toward training and the remaining 20% going toward testing, the training and test datasets are also created. The training set comprises tokens, POS tags, and SRL tags; the testing set comprises only tokens and POS tags. The processed data is then used to feed into the SVM algorithm. Each word was to be classified under one of the predefined SRL tags by the SVM algorithm.

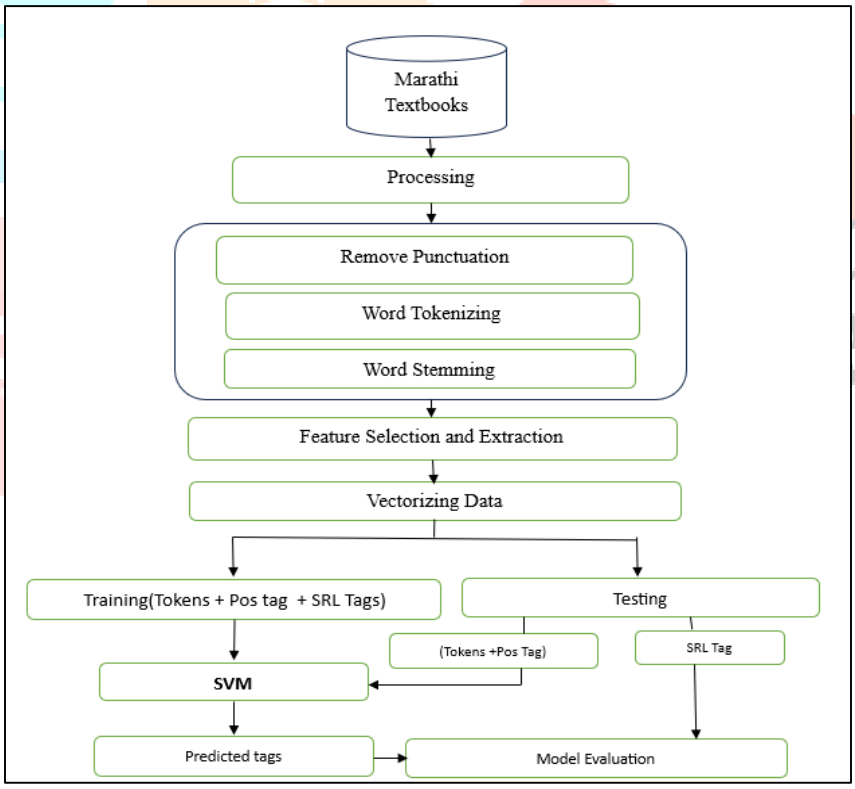

Fig 1 overall workflow of system

SVM finds that best hyper-plane, or boundary of decision, separating a set of the feature vectors across the tags by categorizing the word with respect to the associated feature. Then by this learning approach, after giving its prediction related to the SVM, these predicted tags are matched by comparing the actually tagged SRL in the training data set. In performance evaluation, this model is graded in terms of some standard evaluation metrics such as precision, recall, and the F1 score that give an estimation of how effectively the model's accuracy predicts the SRL tags. The general F1 score is computed so that it captures the overall balance between precision and recall for performance evaluation of SVM in this particular task of classification.

V. Evaluation and Result Analysis

To measure how well a system performs when predicting semantic roles for words in a sentence. The common metrics are:

Precision: Accuracy, precision, recall, and F1-score are among the metrics used to assess the model after it has learned patterns from the training dataset and been applied to the test dataset. In order to give a clear visual depiction of the model's performance in relation to various evaluation measures, the results are usually shown in a bar plot. In the case of SVM, the core concept involves identifying the optimal hyperplane that effectively separates the different classes, or in this context, the tags.

$$Precision = \frac{TP}{(TP+FP)} \quad (1)$$

Recall: These measures how well the system does in predicting which of the real correct roles really exist. In another word, how well the right roles are determined from all actual roles in data. Recall was calculated by just dividing the sum of correct predictions (the True Positives, TP) divided by the total count of actual roles (True Positives + False Negatives).

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

F1 Score: This is a combined score that balances both precision and recall. It gives a single number that shows how well the system is doing overall. The F1 Score is calculated by taking the harmonic mean of precision and recall.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

These metrics help evaluate the performance of the SRL system.

Table 3. Results of Testing data

Labels	Precision	Recall	F1-score
S-Age	0.93	0.89	0.91
B-Age	1.00	0.97	0.98
I-Age	0.75	0.75	0.75
E-Age	0.82	0.79	0.81
S-Loc	0.88	0.65	0.75
B-Loc	0.93	0.79	0.85
I-Loc	0.94	0.68	0.79
E-Loc	0.81	0.64	0.71
S-Pred	0.96	0.93	0.95
B-Pred	0.92	0.77	0.84
I-Pred	0.95	0.98	0.97
E-Pred	0.94	0.91	0.93
S-Them	0.81	0.73	0.77
B-Them	0.93	0.83	0.88
I-Them	0.93	0.66	0.78
E-Them	0.89	0.62	0.73
S-Time	0.85	0.67	0.75
B-Time	0.80	0.67	0.73
I-Time	1.00	0.80	0.89
E-Time	0.87	0.56	0.68
SYM	1.00	1.00	1.00
O	0.77	0.93	0.84
Average	**0.85**	**0.84**	**0.84**

Conclusion

This paper analyzes the performance of SRL on Marathi language with a Support Vector Machine. Since the word order is free in the Marathi language and no work has been reported on SRL for the Marathi text so far, the task of developing an SRL system for the Marathi language is not straightforward. Also, there was no SRL corpus available; therefore, the task was to develop a new dataset for this research. The system was trained on 38,929 tokens and tested on 9731 tokens. It did well by having an overall recall of 84.00%, precision of 85.00%, and F1-score of 84.00%. It particularly did a good job at Agent and predicate classification. Other methods of classification might be explored to improve SRL for Marathi in the future.

REFERENCES

[1] Bharati, A., Sangal, R., Sharma, D. M., & Bai, L. (2006). Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages. *LTRC-TR31*, 1-38.

[2] Carreras, X., & Màrquez, L. (2005, June). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the ninth conference on computational natural language learning (CoNLL-2005)* (pp. 152-164).

[3] Collobert, R., & Weston, J. (2007, June). Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 560-567).

[4] Das, A., Ghosh, A., & Bandyopadhyay, S. (2010, August). Semantic role labeling for Bengali using 5Ws. In *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010)* (pp. 1-8). IEEE.

[5] FitzGerald, N., Täckström, O., Ganchev, K., & Das, D. (2015, September). Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 960-970)

[6] Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational linguistics*, *28*(3), 245-288.

[7] Hacioglu, K. (2004). Semantic role labeling using dependency trees. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics* (pp. 1273-1276).

[8] Pal, R., & Sharma, D. M. (2019, August). A dataset for semantic role labelling of Hindi-English code-mixed tweets. In *Proceedings of the 13th Linguistic Annotation Workshop* (pp. 178-188).

[9] Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, *31*(1), 71-106.

[10] Palmer, M., Gildea, D., & Xue, N. (2011). *Semantic role labeling*. Morgan & Claypool Publishers.

[11] Pandian, S. L., & Geetha, T. V. (2009). Semantic role labeling for Tamil documents. *International Journal of Recent Trends in Engineering*, *1*(1), 483

[12] Rathod, P. H., Dhore, M. L., & Dhore, R. M. (2013). Hindi and Marathi to English machine transliteration using SVM. *International Journal on Natural Language Computing*, *2*(4), 55-71.

[13] Roth, M., & Woodsend, K. (2014, October). Composition of word representations improves semantic role labeling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 407-413).

[14] Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.