Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Parts of Speech Tagger for Morphological Analysis of Maithili Language using Instance Based Learning

Prabhat Kumar Singh, Dr. Harish Patidar

Department of Computer Science and Engineering, Mandsaur University, Mandsaur, India
Email: prabhatprobable@gmail.com

Maithili is the member of an Indo-Aryan language family, mainly spoken in Indian states Bihar, Jharkhand and most part of Nepal. It is recognized as the scheduled language of the Indian constitution.It is resource poor language,little work has been done towards design and development of NLP tools.There is a need of developing different NLP applications for the Maithili language.As creating rules for the Maithili language is a tedious task due to ambiguity at word level and sentence level,we require a model which does not require more linguistic information for processing of words.Statistical and Machine learning approaches does not require more linguistic knowledge and human effort for writing rules.We have chosen machine learning algorithm to design POS tagger for Maithili language.As in our previous work ,found that Instance Based learning approach gives best performance as compared to used other machine learning algorithm,,we have used Instance based learning based approach for designing POS tagger for the Maithili language.Proposed model trained with own created tagged corpus for the language.Accuracy obtained can be improved by train large corpus of the Maithili language.
Keywords: Maithili, Morphological Analyzer, POS, Instance Based Learning, NLP.

1. Introduction

Maithili has some similarities with other Indo-Aryan languages like Hindi, Bengali, and Assamese but also has its own unique features. It has a complex phonological system, with a variety of vowels and consonants, along with a rich morphology that involves detailed inflectional and derivational patterns. Maithili literature includes a wide range of genres, including poetry, prose, drama, and folk songs. Well known literary figures such as Vidyapati,

the prominent Maithili poet of the 14th century, had made a great contribution to the language's rich literary tradition. Maithili has great cultural and historical value, but it faces challenges in the modern world, such as standardization, support from institutions, and use in digital spaces. To protect Maithili, various efforts are being made like creating language tools, opening schools, and encouraging its use in official settings.In recent years, Maithili has been valued more as a key part of India's language heritage. Efforts like digitizing books, creating tools, and making apps aim to bring Maithili into the digital world. Overall, the Maithili is a rich blend of culture, history, and linguistic diversity, with a centuries-old legacy. Despite challenges, Maithili has a long, proud history and is spoken by millions around the world.

Parts of Speech Tagging: Part of Speech tagging in Natural Language Processing is very important in understanding and generation of Natural language. The procedure of POS tagging is very hard in Natural Language Processing applications, as it requires assigning part of speech labels to each word in a sentence of the dictionary. Each word of the sentence is required to be tagged with a particular POS like noun, pronoun, verb, adjective etc.

Part of speech tags for the Maithili sentence are shown in the following figure.

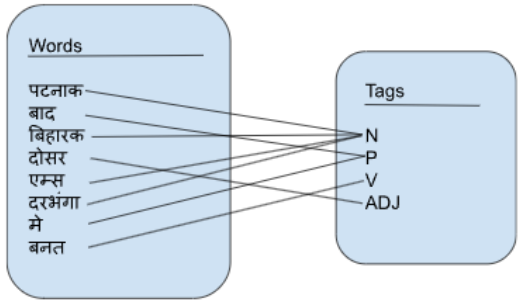

Figure 1.1: Part of speech tags for the Maithili sentence

2. APPROACHES FOR POS TAGGING

There are many approaches for part of speech (POS) tagging. Stochastic approaches, Machine learning approaches and Rule based approaches are mainly used to develop parts of speech tagger.Parts of speech can be classified as supervised and unsupervised POS tagging.

1. Stochastic POS Tagging: Frequency,probability and statistics is included in stochastic approaches.Most frequently used tag for a particular word in the tagged training corpus and uses this information to tag that word in the unannotated text in the stochastic approaches .The main challenges with this approach is tag sequences for sentence are not as per the grammatical rules of the language.

2. Rule based POS Tagging: Rule-based POS (Part of Speech) tagging methods uses hand written rules and considering contextual information for tagging parts of speech to individual words in the sentence of the language.These written rules are based on contexts, so called as context frame rules.Example for the context frame rule in Maithili is as follows :

“If a word ends in "आ" (aa) and is followed by a Noun, tag it as an Adjective.”

Let us take an instance for this rule , In "बड़का लड़का" (baṛakā laṛkā), which means “Big Boy”,the word "बड़का" ends with “आ” and followed by a Noun “लड़का”,rule say that “बड़का” is an Adjective.

Another rule we can write : “If a word ends with "इ" (i) and is followed by a pronoun, it is likely a verb.”

In "देखि ओ" (dekhi o), which means “saw him”, the word "देखि” ends with "इ", which is followed by Pronoun “ओ” , rule says "देखि" is a Verb.

3. Machine Learning-Based Approach: In Machine learning methods word structures use algorithms for understanding, in this case no need to write rules manually. Raw texts or labeled data is used to train models in machine learning approaches. Supervised learning needs annotated texts while no need of annotated or labeled text in case of unsupervised learning, it can work on unprocessed text. It’s like training the system to analyze words and morphemes on its own. In supervised learning techniques such as Conditional Random Fields (CRFs),analyzing word structures is done as a task of sequence labeling. It helps in tagging of morphemes or predicting boundaries of morphemes in a word.

3. RELATED WORK

Martine Z (2012) presented POS tagging using different approaches. POS tagging is very important for text preprocessing in the field of NLP. The main goal is to achieve 100% accuracy without or with minimal human intervention. There are two factors due to which we are not able to get 100% accuracy, the first one is the presence of ambiguous words, and the next is unknown words, i.e. the words POS tagger has not encountered in its training corpus. The author presented different methods for tagging like rule-based, transformation-based learning, Marker Model taggers, Maximum entropy methods, etc. The author also discussed other approaches like neural network finite state transducer, Decision tree, Support Vector Machine, etc. As compared to rule-based methods, stochastic methods give better accuracy from 96 to 97 % as they use annotated corpus for POS tagging.[3]. Adinarayanan et. al. (2015) reviewed research on Sanskrit language part of speech tagger using different approaches. Researchers highlighted morphological richness in the Sanskrit language and its grammar. Authors also discussed different natural language processing techniques to develop tools for the language. Different models for POS tagging have been discussed including statistical, rule based and hybrid models. Importance of preparing a root words database has been highlighted. Root words table is very important for proper tagging of different POS classes ,specially in rule based approach. The researchers also discussed word analysis, syntax analysis, morphology, semantic analysis and phonology in the language. Authors compared different approaches for POS tagging and found that hybrid models (stochastic and rule based) give better performance as compared to using a particular single approach. Authors found that a combination of rule based approach and stochastic approach gives accuracy of 90 % which is higher than using a single model. Researchers concluded that more work is needed to handle complex morphology and challenges to create NLP tools in the Sanskrit language.[4]. Kumar

et al. (2012) presented a performance comparison of POS tagging for the Magahi language, which is resource-poor and it was the first attempt to develop an NLP tool for the language. The authors tested four parts of speech taggers - TnT, mxpost, SVMTool, and MBT. mxpost tool, which is based on maximum entropy, gives the best result for the annotation of POS. All four taggers are trained for approximately 50,000 Magahi words and use 33 different tags. Total of 13,000 words of which known words were 86% and unknown words 14%. Tested for all four tools. MXPOST tool gives around 90% performance which is the best among all used tools for the work. SNMT00l gives the worst result around 41% as a large number of tags are required to be classified.[5]. Part of speech tagger has been developed by Samir Amri et. al. (2019) using machine learning models for Amazigh language spoken in North Africa .Researchers used three different machine learning models-Support Vector Machine(SVM), Conditional Random Field (CRF) and TreeTagger. The researchers manually created a database of 85000 Amazigh words from available text with 28 tags. Accuracy achieved from CRF, TreeTagger and SVM are 90.08%,92.06% and 89.38 % respectively. TreeTagger gives highest accuracy among the three algorithms used and SVM model gives lowest accuracy.[6]. Li, H. et. al. (2022) proposed POS tagger using a combination of rule based and transformer model. Proposed model is developed in two steps, in the first step researchers write rules to reduce the possible number of tags and then apply a transformer model to predict the remaining tags. Groningen Meaning Bank (GMB) dataset was used for testing of the proposed system.Accuracy achieved for words and sentences are 98.06 and 76.04 respectively.[7]. Part of Speech tagger for Amharic language using deep neural network has been developed by HirpassaS et. al. (2023). Accuracy obtained for the system is 97.23 %,which is the highest frequency for Amharic language POS tagger till date. Precision for known words and unknown words evaluated as 0.970 and 0.772 respectively.Overall recall for the proposed system recorded as 0.948.[8]. Pooja M Bhatt et. al (2021) presented part of speech tagger for the language Gujarati using a combination of machine learning and optimal feature selection techniques. The proposed tagger gives an accuracy of 82%. precision and recall obtained for the proposed model are 0.830 and 0.840. Authors also evaluated other metrics like f-measure which is 0.870.[9]. Somnath Banerjee et. al. (2014) proposed a part of speech tagger for Bangla language using hybrid approach. The accuracy obtained for the system is 90.50%. The precision for the POS tagger is 0.899 and recall for the proposed system is 0.935. The researchers proposed two transliteration systems to convert Bangla words into Bangla script with accuracy of 0.062.[10]. The Indian language Bengali POS tagger was proposed using Hidden Markov Model (HMM) by Sandipan et. al. (2007). Developed POS tagger gives accuracy of 88% for Bengali, 68% for Telugu and 83% for Hindi language. Proposed chunker gives an accuracy of 84% for Bengali, 80% for Hindi, and 66% for Telugu.[11]. Part of speech tagger for hindi language was proposed by Dalal et. al. (2006) using statistical methods. Proposed work usages maximum entropy model and different features like dictionary features, words features, context based features, and corpus based features. Researchers also presented a chunker for the language. Proposed POS taggers have obtained accuracy of 82.22% whereas chunker accuracy is 82.4%.[12]. The design and development of morphological analyzers for a variety of languages has been the subject of numerous studies employing various methodologies. Hindi morphological analyzer has been described by Ankit et al.(2014) which handles inflectional morphology.[13]. Indian language Marathi morphological analyzer has been presented by Mugdha Bapat et al.(2010) using finite state method which works for

inflectional paradigms and accuracy of analyzer found high.[14]. As there are no existing morphological analyzers for Bishnupriya Manipuri language, Nayan Jyoti Kalita et al.(2014) developed a finite state transducer based morphological analyzer for the language and used XFST tool for the same. Although the language had very few resources , it obtained a good precision for the work.[15]. Bhuvaneshwari et al. (2011) presented morphological generators for the Indian language Kannada using multiple finite state machine.[16]. T. N Vikram et al. (2007 developed morphological analyzer for the south Indian language Kannada using finite state transducer which can work as a POS tagger, spell checker and stemmer.[17]. Ankur Priyadarshi et al.(2019) proposed the first Maithili part of speech tagger using Conditional Random Fields based classifier and accuracy found was 82.67%.[18]. Bharati et al. (2020) presented an unsupervised learning algorithm for the morphological analysis and generation of Hindi.[19]. Xuri Tang(2006) proposed an algorithm for English morphological analysis using machine learning approach which can solve the problem of manual work and rule inconsistency.[20]

4. PROPOSED METHODOLOGIES

Following are the proposed solutions for objective 2 shown in Figure 3.2.

3.4.2.1 Algorithm

1. Load the Dataset: scan the dataset from a CSV file called 'your_dataset.csv'.

2. Define the IBLAClassifier Class: construct a class named IBLAClassifier with three methods:

. __init__() to initialize the classifier.

a. fit() to train the classifier using the training data.

b. predict() to make predictions using the test data.

0. Initialize Variables: Create variables for the features (I), labels (J), number of folds for cross-validation (num_folds), size of each fold (fold_size), and a list to store accuracy values (accuracies).

0. Perform 10-Fold Cross-Validation: Repeat the following steps for each fold in the range of num_folds:

. Find the start and end indices for the test data.

a. Split the data into training and test sets using these indices.

0. Train and Test the Classifier:

. Create an instance of the IBLAClassifier with k=5.

a. Train the classifier with the training data (I_train and J_train).

b. Use the classifier to make predictions on the test data (I_test).

0. Calculate Accuracy:

. Calculate the accuracy for the current fold by comparing the predictions with the actual test labels (J_test).

a. Add the accuracy value to the accuracy list.

0. Compute Average Accuracy: Calculate the average accuracy by taking the mean of all accuracy values in the accuracy list.

0. Print the Average Accuracy: Display the average accuracy with a message like "Average Accuracy: avg_accuracy".

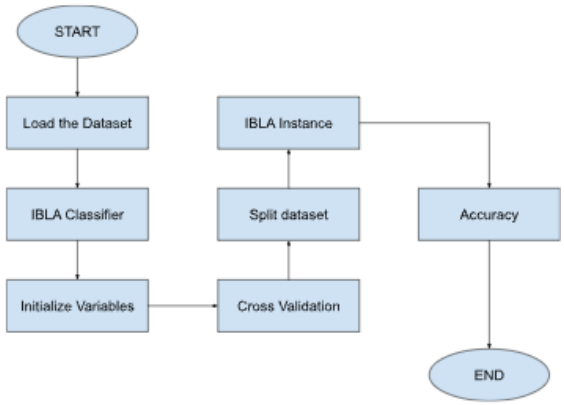

Figure 1.2 : Instance-Based Learning Algorithm (IBLA) Flow Chart

For the purpose of initialization, training and prediction, We would load the dataset from a CSV file named 'your_dataset.csv' and define a class called IBLAClassifier. Variables for features, labels and other necessary parameters will also be initialized. To ensure correct results, we will perform 10-fold cross-validation in which the dataset will be splitted into training and test sets, we will train the classifier, and make predictions. Accuracy for each of the fold will be calculated and these values will be stored. Afterwards, we will compute and print the average accuracy to understand the classifier's overall effectiveness.

5. RESULT ANALYSIS

An Instance-Based learning approach is best for designing a morphological analyzer for the Maithili Language.[2]. We have classified the parts of speech for our own created dataset of Maithili Words. We have implemented our algorithm in Python 3.0,and tagged POS for different grammatical attributes such as Noun,Pronoun,VerbAdjective,Conjunction, and Postposition .We have computed different metrics for separate classes to evaluate the performance of the model, we have obtained a true positive rate of 0.967 and a false positive rate of 0.345 for the Noun class .Precision and recall obtained for the model are 0.57 and 0.967 respectively as shown in table 1.1, the recall value is up to the benchmark value, but precision is low.

Table 1.1: Metrics for POS Class

Class	TP Rate	FP Rate	Precision	Recall
Noun	0.967	0.345	0.57	0.967
Pronoun	0.671	0.011	0.885	0.671
Adjective	0.45	0.009	0.939	0.45
Verb	0.644	0.014	0.919	0.644
Conjunction	0.844	0.016	0.73	0.844
Postposition	0.368	0.004	0.583	0.368
Adverb	0.586	0.013	0.784	0.586

We have also evaluated the F-measure to get a combined result score of precision and recall, we have found an F-measure score of 0.717 for the Noun class. The obtained ROC Area for nouns is 0.902 which is an impressive value shown in table 1.2 and figure 1.3. Evaluation Metrics like F-measure, ROC Area, and PRC are more important if the dataset is not balanced as we have manually prepared the dataset these parameters are comparatively more important and the values obtained are good.TP rate and FP Rate for pronoun class were found as 0.671 and 0.011 respectively here FP rate is good but the TP rate is not impressive due to the imbalanced dataset.

Table 1.2 : Metrics for POS Class

Class	F-Measure	MCC	ROC Area	PRC Area
Noun	0.717	0.583	0.902	0.816
Pronoun	0.763	0.747	0.931	0.759
Adjective	0.608	0.592	0.892	0.752
Verb	0.757	0.726	0.919	0.814
Conjunction	0.783	0.773	0.959	0.779
Postposition	0.452	0.458	0.819	0.359
Adverb	0.671	0.656	0.934	0.756

We have obtained precision, recall, and F-measure for pronoun class as 0.885, 0.671, and 0.673 respectively. ROC area for pronoun class was obtained as 0.931 which is an acceptable value for the model. We have also calculated metrics for other parts of speech like verbs, adverbs, adjectives, conjunctions, and prepositions. Verb class has a precision of 0.919 whereas the recall value of 0.644. The ROC area obtained for the verb class is 0.919. accuracy of the proposed model is 70.71% which is low but after preparing a big dataset it will be improved.

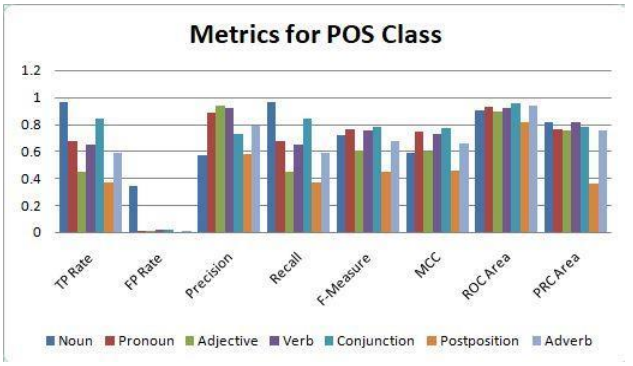

Figure 1.3: Metrics for POS class

We have also created a confusion matrix for POS classes in the following Table.

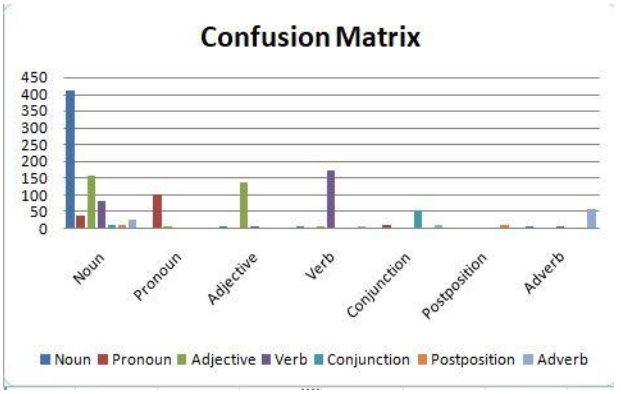

Figure 1.4: Confusion Matrix

Finally, we have calculated different errors in the proposed model. The mean absolute error and root mean squared error obtained are 0.128 and 0.23 respectively in the following Table 4.7.

Table 1.3: Accuracy and Error

Correctly Classified Instances (Accuracy)	939	70.71%
Incorrectly Classified Instances	389	29.29%
Kappa statistic	0.6123	
Mean absolute error	0.128	
Root mean squared error	0.2348	
Relative absolute error	57.17%	
Root relative squared error	70.18%	
Total Number of Instances	1328	

6. CONCLUSION

After getting the best algorithm for the morphological analyzer we have designed the proposed algorithm based on the Instanced Based Learning Approach. We have implemented the proposed algorithm in Python 3.0. We have manually prepared a dataset with the help of a Maithili speaker containing 1328 Maithili words with the help of the LDCIL corpus. The proposed model works for 7 parts of speech classes - Nouns, Pronouns, verbs, adverbs, adjectives, postpositions, and conjunctions. We have calculated 8 different evaluation metrics like TP Rate, FP Rate, F-measure, Recall, Precision, ROC Area, MCC, and PRC Area for individual POS classes. The accuracy obtained for the proposed model is 70.71% which is low but as the dataset is not balanced, other metrics like F-measure and ROC Area are more important. We have obtained the ROC Area for all individual classes more than 0.9, which is impressive. The proposed model deals with inflectional as well as derivational morphology. As we have prepared tagged corpus manually, the size of the corpus is small. In future accuracy can be improved by creating a large dataset for training. In future other NLP tools such as Parser, and Machine translation systems can be developed for Maithili language.

References

1. G. Cardona, "Indo-Aryan Languages," in The Cambridge Encyclopedia of Language Sciences, Cambridge: Cambridge University Press, 2010, pp. 111–113.
2. P. K. Singh and H. Patidar, "Performance Comparison of Machine Learning Algorithms for Morphological Analysis of Maithili Language," 2024 IEEE International Conference on Contemporary Computing and Communications (InC4), Bangalore, India, 2024, pp. 1-6.
3. Martinez, Angel. (2012). Part-of-speech tagging. Wiley Interdisciplinary Reviews: Computational Statistics. 4. 10.1002/wics.195.
4. Adinarayanan, Sharadha & Ranjanie, Sri & Naren.J,. (2015). Part-of Speech Tagger For Sanskrit: A State of Art Survey. International Journal of Applied Engineering Research. 10. 24173-24178.
5. Singh, P., Kore, A., Sugandhi, R., Arya, G., & Jadhav, S. (2013). Hindi Morphological Analysis and Inflection Generator for English to Hindi Translation.
6. Samir Amri, Lahbib Zenkouar, and Reda Benkhouya. 2019. A Comparative Study on the Efficiency of POS Tagging Techniques on Amazigh Corpus. In Proceedings of the 2nd International Conference on Networking, Information Systems & Security (NISS '19).
7. Li, H., Mao, H., & Wang, J. (2022). Part-of-Speech Tagging with Rule-Based Data Preprocessing and Transformer. Electronics, 11(1), 56.
8. HirpassaS, LehalG. Improving part-of-speech tagging in Amharic language using deep neural network. Heliyon. 2023.
9. Pooja M Bhatt , Dr. Amit Ganatra, POS-HOML: POS Tagging Technique For Gujarati Language Using Hybrid Optimal And MachinTamile Learning Approaches International Journal of Engineering Trends and Technology Volume 69 Issue 11, 256-262, November, 2021 ISSN: 2231 – 5381.
10. Somnath Banerjee, Alapan Kuila, Aniruddha Roy, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2014. A Hybrid Approach for Transliterated Word-Level Language Identification: CRF with Post-Processing Heuristics. In Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation (FIRE '14).
11. Dalal Aniket, Kumar Nagaraj, Uma Sawant and Sandeep Shelke (2006), "Hindi Part-of-Speech Tagging and Chunking: A Maximum Entropy Approach", Proceedings of NLPAI-2006, Machine Learning Workshop on Part Of Speech and Chunking for Indian Languages.

12. Sandipan Dandapat (2007), “Part Of Speech Tagging and Chunking with Maximum Entropy Model”, Proceedings of IJCAI Workshop on Shallow Parsing for South Asian Languages.
13. P. Mishra, "Demographics of Maithili Speakers in India and Nepal," Census of India Report, Government of India, 2011.
14. N. Sharma, "The Decline of Maithili: A Sociolinguistic Perspective," Nepalese Linguistics, vol. 37, no. 1, pp. 45–59, 2020.
15. S. A. Khan, W. Anwar, U. J. Bajwa, X. Wang, “A Light Weight Stemmer for Urdu Language: A Scarce Resourced Language”, 3rd Workshop on South and Southeast Asian NLP, Pp. 69–78, 2012.
16. Kumar, Deepak, Manjeet Singh, and Seema Shukla. "Fst based morphological analyzer for Hindi language." arXiv preprint arXiv:1207.5409(2012).
17. Melinamath, Bhuvaneshwari C., and A. G. Mallikarjunmath. "A morphological generator for Kannada based on finite state transducers." Electronics Computer Technology (ICECT), 2011 3rd International Conference on. Vol. 1. IEEE, 2011.
18. Sahoo, Kalyanamalini. "Oriya nominal forms: a finite state processing."TENCON2003. Conference on Convergent Technologies for the Asia-Pacific Region. Vol. 2. IEEE, 2003.
19. Bapat, Mugdha, Harshada Gune, and Pushpak Bhattacharyya. "A paradigm-based finite state morphological analyzer for Marathi." Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP). 2010.
20. Kalita, Nayan Jyoti, Navanath Saharia, and Smriti Kumar Sinha. "Morphological Analysis of the Bishnupriya Manipuri Language Using Finite State Transducers." International Conference on Intelligent Text Processing and Computational Linguistics. Springer Berlin Heidelberg, 2014.