Part of Speech Tagging for a Resource Poor Language : Sindhi in Devanagari Script using HMM and CRF

Bharti Nathani
Department of computer Science
Banasthali Vidyapith
Banasthali
bhartinathani@rediffmail.com

Nisheeth Joshi
Department of Computer Science
Banasthali Vidyapith
Banasthali
nisheeth.joshi@rediffmail.com

Abstract

Part of speech tagging is a pre-processing step of various NLP applications. Mainly it is used in Machine Translation. This research proposes two POS taggers, i.e., an HMM-based and CRF based tagger. To develop this tagger, the corpus of manually annotated 30,000 sentences has been prepared with the help of language experts. In this paper, we have developed POS taggers for Sindhi Language (in Devanagari Script), a resource poor language, using HMM (Hidden Markov Model) and Conditional Random Field (CRF).Evaluation results demonstrated the accuracies of 76.60714% and 88.79% in the HMM, and CRF, respectively.

1 Introduction

The main aim of NLP is to create suitable algorithms for computers to understand human language. These language technology tools help the users in translation and understanding of languages. For example, with the help of machine translation, a person can communicate or share information in their native language. Although plenty of different language processing tools are available for some languages, still some of the languages have not been able to attract the attention of the research community. The Sindhi language is one of them. There are 22 official languages of India. Sindhi is one of them. In 1967 it was recognized as an official language of India. The Eighth Schedules of the constitution of India includes the languages which are resource-poor and need to preserved and developed. Sindhi was included in this schedule in the 21stAmendment. Although in India, Sindhi is officially recognized, but it is not an official language of any of the states in India.

The "International Education System" puts a big problem for Sindhi people. The upcoming Sindhi generation is unable to write and speak the Sindhi language, as their parents do not communicate with them in the Sindhi language. This may lead to the extinction of the Sindhi language. For language preservation, our prime need is to save its speakers by developing some language processing tools which help them to learn Sindhi. In the last few years, some work has been done in the Sindhi language in Arabic script. Sindhi Devanagari is more resource-poor than Sindhi Arabic. In this research work we have developed a SLP (Sindhi Language Processing) tool i.e., POS Part of Speech Tagger in Devanagari Script.

The words can be classified in various lexical categories, such as nouns, verbs, etc. These categories are also known as Parts of Speech. Parts of Speech define their morphological and syntactical behavior. POS Tagging is a task of classifying each word in a corpus to a given syntactic class such as noun, verb, etc.

A word can belong to more than one lexical category, depending on its use in a sentence. The main objective of the POS tagging process is to remove this ambiguity. Tagger uses the contextual information to assign the tag. Part of Speech tagging is used in various applications of NLP, such as Machine Translation, Information retrieval, information extraction, spelling correction, and word sense disambiguation. This paper presents the development of two automatic taggers using Conditional (CRF) and Hidden Markov Model(HMM) .

2 Related work

POS tagging is assigning the syntactic or lexical category to a word in a sentence. POS tagging is a

Proceedings of the 18th International Conference on Natural Language Processing, pages 611–618
Silchar, India. December 16 - 19, 2021. ©2021 NLP Association of India (NLPAI)

fundamental task of NLP. It is an important pre-processing task of various Natural Language Processing (NLP) applications such as in IR, Text summarization, machine translation etc. In this section, we will discuss the related work done in this area.

2.2.1 Other Languages

The work on automatic POS tagging was started in the early 1960s. Ekbal, A. et al. (2007) developed a POS tagger for Bengali language using condition random field approach with a tag set of 26 POS tag. For training they used 72,341 words and 20 thousand words for testing. They got the accuracy of 90.3%.

Hasan, M. F. et al. (2007) applied few stochastic approaches such as unigram, bigram HMM and Brills POS tagging on Hindi, Bangla and Telugu with different size of the corpus. They found that Brill's transformation-based tagger's performance is good in comparison to other approaches.

Singh, T. D. et al. (2008) developed a POS tagger for Manipuri text using an unsupervised learning approach CRF. The system gave the Recall of 70% precision of 77.78% and F-measure of 73.68%.

Sharma et al. (2011) used an HMM algorithm to improve the accuracy of existing Punjabi POS Tagger. This Bi-gram tagger resolves the problem of ambiguity for complex and compound sentences. They have taken the training corpus of 20,000 tokens and a test corpus of 26 479 tokens. They achieved 90.11% accuracy.

Garrette, D et al. (2013) discussed the various aspects of semi-supervised Learning of POS taggers. They work for Kinyarwanda and Malagasy two resource-poor languages and study the effect of various kind of data on POS- tagger.

Singh, J., Joshi, N., & Mathur, I. (2013) used Statistical approach to develop Marathi POS tagger, i.e. Unigram, Bigram, Trigram and HMM. They achieved 77.38% accuracy for Unigram approach, 90.30% for Bigram, 91.46% for Trigram and 93.82% for HMM.

Sunitha, C. (2015) research work proposed a hybrid approach for POS tagging (CRF and rule-based approach) of Malayalam language. They used the tag set developed by IIIT Hyderabad. They got 94% accuracy.

Pakray, P. et al. (2015) developed various resources for Mizo language (an official language of Mizoram State) such as Mizo-to-English dictionary; tag set consist of 24 items and POS tagger.

Buys et al. (2016) proposed a model, which uses a Wsabie, a discriminative embedding model train a morphological tagger. They evaluated this on 11 languages and concluded that this model performs very well when used for closely related languages.

A CRF based approach was used by Sarkar, K. (2016) for developing POS Taggers for three language pairs, i.e. Bengali-English, Telugu-English and Hindi- English. They have got an average of 79.99 F1 scores.

For Odia Language, a CRF++ based POS tagger was developed by Behera, P. (2017). For this, they manually prepared POS annotated, the corpus of 600thousand tokens, using BIS tag set. The tagger is trained on 2,36,793 tokens and tested with 1,28,646 tokens. They got 94.39% accuracy for the known data and 88.87% accuracy for unknown data.

Mishra, P., Mujadia, V., & Sharma, D. M. (2018) presented an approach for POS tagging of resource poor language. This approach requires only the bilingual corpora of sentences. They have transferred the features of the resource-rich language to resource-poor language, for this they have used word alignment algorithm using Giza ++.

2.2.2 Sindhi language

Maher and Memon (2010 A) developed a POS Tagger using Word Net approach. This tagger was tested on lexicon containing 26,366 tagged words, and the accuracy was 97.14%. This Tagger gave higher accuracy on the past and presented tense sentences, but on future tense sentences, it gave lesser accuracy.

Maher and Memon (2010 B) developed the first POS tagging system for Sindhi (Perso-Arabic Script). They used a rule-based approach. The size of the lexicon was 26,366 and useda tag set of size 67. They tested this tagger on 1,500 sentences, which consisted of 6,783 words and obtained 96.28% accuracy.

Motlani et al. (2015) built a POS Tagger for Devanagari Script of Sindhi language using Conditional Random Fields. They tested and trained the tagger using 10-fold cross-validation. They used BIS tag set, and the accuracy of tagger was 92.6%.

3 The Approach

POS Tagging approaches are broadly classified into rule-based, stochastic, and hybrid approaches. In the rule-based approach, handwritten disambiguation linguistic rules are used for tagging. Stochastic is also known as a data driven approach, which requires pre tagged corpus for training. Hybrid is a combination of rule-based and data driven. To develop a POS tagger, we have chosen the stochastic approach. This is a data driven approach. Rule based approach is time consuming and needs language expertise to write the rule. For morphologically rich language, it is impossible to write all the rules

Stochastic approach is a probability-based approach. We have used two standard algorithms HMM (Hidden Markov Model) and CRF (Conditional Random Field) for POS tagging. HMM is an example of a Generative model, whereas CRF is an example of Discriminative model (Sutton, C., & McCallum, A. (2012)). For a particular data set, we cannot predict in advance which model will give the correct results. Each type of model is having its own limitations and delimitation.

4 Corpus Annotation

The stochastic approach is data driven. This requires the manually annotated corpus for training. The stochastic approach gives better results when the manually annotated corpus is used for training. In this sequence, we have manually annotated the corpus of 30000 sentences by using the guidelines described by Lata et al. (2012). For tagging, we have used the IL tag set. A tag set is a collection of tags used by a tagger. The tag set is described in the following tables:

S. No.	Tag	Description	Example
1.	NN	Common Nouns	किताबु,माणहू
2.	NNP	Proper Nouns (Name of Person)	भारत, देहरादून
3.	NST	Noun Denotating Spatial and Temporal Expression	अगियां(आगे), पुठियां(पीछे)
4.	PRP	Proper Noun	अव्हांजे(आपकी), असांजे(हमारे)
5.	VM	Verb Main	थी(हो), रखण(रखना)
6.	VAUX	Verb Auxiliary	आहे(है), सघंदा(सकते)
7.	JJ	Adjective (Modifier of Noun)	कमज़ोर(कमजोर),तेज़(तेज)
8.	RB	Adverb	धीरे, जल्दी
9.	PSP	Post Position	खां(से),जे(के)
10	RPD	Particles	बि(भी),त (तो)
11	QTF	Quantifiers	घटि (कम), रुग॒ो(केवल)
12	QTC	Cardinals	हिक(एक), ब(दो)
13	CCD	Conjunctions	पर(बल्कि), ऐं(और)

14	INTF	Intensifier	वधीक(अत्यधिक), तमाम(बहुत)
15	NEG	Negative	ननथा(नहीं)
16	SYM	Symbol	$, &, *, (,)
17	ECH	Echo Words	हलको - फुलको(हलका/JJ फुलका/ECH)
18	QO(QTO)	Ordinals	पहिरियों(पहला),बियनि (दूसरे) टिएं(तीसरे)
19	DMI	Demonstrative (Indefinite)	बियनि(किसी),कंहिं(किसी)
20	CCS	Subordinator	यदि(अगर),याने(अर्थात)
21	PRF	Pronoun (Reflexive)	ख़ुदि(खुद),पंहिंजी(अपनी)
22	DMD	Demonstrative (Deictic)	इहो, ही (यह),इनजो(इसका)

Table 1: Tag set for Sindhi Devanagari.

5 POS Tagging using HMM

For a given input sentence, we can calculate the best tag sequence using the following formula:

$$T' = \text{argmax}_T P(W/T)^* P(T) \qquad (1)$$

Where P(T) is a prior probability of tag sequence (i.e., tag transition probability), and P(W/T) is emission probability. P(T) is calculated by using following formula:

$$P(T) = P(t_1)^* P(t_2/t_1)^* P(t_3/t_1t_2) \ldots *(t_n/t_1 \ldots t_{n-1}) \quad (2)$$

According to bigram assumption:

$$P(t_i/t_i - 1) = \frac{c(t_{i-1,}t_i)}{c(t_{i-1})} \qquad (3)$$

Where c(t_{i-1},t_i) is the counting of how many times tag $t_{i,}$ comes after tag t_{i-1} (Previous tag). To calculate the emission probability:

$$P(W/T) = P(w1/t1) * P(w2/t2) \ldots P(wi/ti) * \ldots P(Wn/tn) \qquad (4)$$

$$\prod_{i=1}^{n} p(w_i t_i) \qquad (5)$$

$$P(w_i / t_i) = \frac{c(w_i, t_i)}{c(t_i)} \qquad (6)$$

Where c(w_i/ t_i) calculate the probabilities , that a given tag t_i is associated with given word w_i.HMM algorithm chooses the most likely tag sequence with the help of a decoding algorithm, i.e. Viterbi algorithm.

5.1 Experiments

We have implemented the above algorithm in Python. Following table shows the Statistics of Training and Testing Data set:

Data Set	Number of Sentences
Training Data Set	30000
Testing Data Set	1000

Table 2: Data set for Experiment

5.2 Evaluation

Evaluation of output text is done by using the following formula:

$$\text{Accuracy} = \frac{\text{Total Number of correct POS tag generated by POS Tagger}}{\text{Total Number of POS Tag}}$$

We have tested this POS Tagger with 1000 sentences, which consists of 15680 tokens and we found 12012 matches. So, the overall accuracy obtained is 76.60714%.

6 POS Tagging using CRF

CRF was introduced by Lafferty et al., 2001.CRF is a discriminative undirected graphical model that belongs to the family of condition distribution. CRF is the most popular method used for structured prediction in the NLP task. In the discriminative approach, for a given input x and output y, the probability is calculated directly p(y|x) whereas in the generative model joint probability p(x,y) is generated. Structured means that the output of an algorithm is a structured object such as a tree or a sequence.

CRF is used for the POS tagging task. This discriminative model x for a given observation sequence O=<o1, o2, o3…. oT> where observation is the sequence of tokens, and State sequence S=<s1, s2, s3……sT> is the POS tag. Conditional probabilities are calculated as:

$$P\wedge(s/o)=\frac{1}{Z_{0}}\exp(\sum_{t=1}^{T}\sum_{k}\lambda kfk(st-1,st,0,t) \qquad (7)$$

In the above equation, f_k is a transition feature function, which is learned via input or observation sequence. The weight of this function is k which defines the weight which is learned in training.

The normalization factor Z_0 is calculated using the following formula:

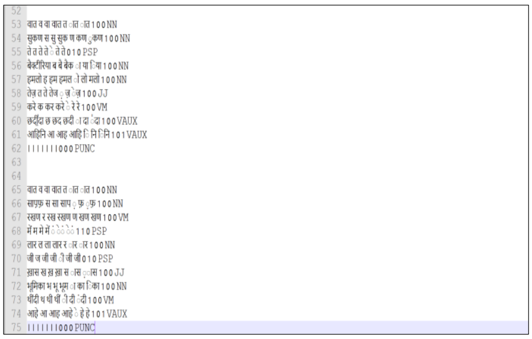
$$Z_o=\sum_{s}\exp(\sum_{t=1}\sum_{k}kfk(s_t-1,s_t,o,t)) \qquad (8)$$

This makes all conditional probabilities sum equal to 0.

We have used CRF++ [1]for training and testing our tagger. Three files are required for CRF implementation.1 Training file 2. Testing File 3. Template File. The complete corpus is divided into 80-20 ratio, where 80% is used for training data, and 20 % is used for testing data. CRF model is developed in three steps:

6.1 Creation of Training and Testing File

All the words or token of a sentence must be represented using one token per line format.



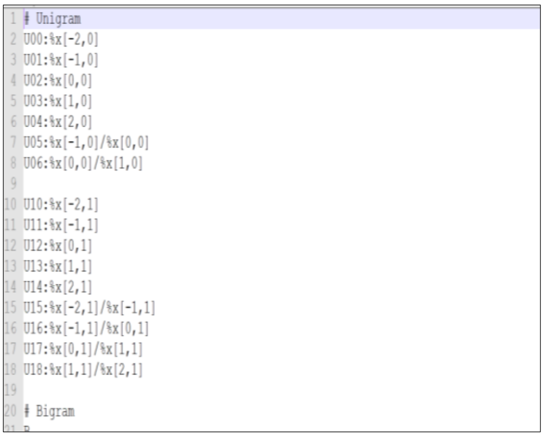
```
वात व वा वात त ात ात 1 0 0 NN
सुकण स सु सुक ण कण ुकण 1 0 0 NN
ते त ते ते े ते ते 0 1 0 PSP
बेक्टीरिया ब बे बेक ा या िया 1 0 0 NN
हमलो ह हम हमल ो लो मलो 1 0 0 NN
तेज़ त ते तेज ़ ज़ ेज़ 1 0 0 JJ
करे क कर करे े रे रे 1 0 0 VM
छदींदा छ छद छदी ा दा ंदा 1 0 0 VAUX
आहिनि आ आह आहि ि नि िनि 1 0 1 VAUX
। । । । । । । 0 0 0 PUNC


वात व वा वात त ात ात 1 0 0 NN
साप़फ़ स सा साप ़ फ़ ़फ़ 1 0 0 NN
रखण र रख रखण ण खण खण 1 0 0 VM
में म मे में ं ें ें 1 1 0 PSP
लार ल ला लार र ार ार 1 0 0 NN
जी ज जी जी ी जी जी 0 1 0 PSP
ख़ास ख ख़ ख़ा स ास ़ास 1 0 0 JJ
भूमिका भ भू भूम ा का िका 1 0 0 NN
थींदी थ थी थीं ी दी ंदी 1 0 0 VM
आहे आ आह आहे े हे हे 1 0 1 VAUX
। । । । । । । 0 0 0 PUNC
```

Figure 1: Sample Training File

Sentence boundary is identified by putting an extra blank line. Each token is represented along with its features in fixed columns, which are separated by space. First Column represents the word or token, and the last column represents the output on which we train CRF. The remaining columns represent the value of the features we have used in CRF. The sample training and the testing file are shown in the figure 1.

6.2 Creation of Feature Template

The template file defines the features used in CRF. Each line in a file represents one template. A macro used in a template will specify the token in input data, r specifies the row number from current token and c specifies the absolute position of the column. The sample of template is shown in the following figure.



```
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

U10:%x[-2,1]
U11:%x[-1,1]
U12:%x[0,1]
U13:%x[1,1]
U14:%x[2,1]
U15:%x[-2,1]/%x[-1,1]
U16:%x[-1,1]/%x[0,1]
U17:%x[0,1]/%x[1,1]
U18:%x[1,1]/%x[2,1]

# Bigram
B
```

Figure 2: Sample Template File

6.3 Training and Testing of CRF Model

For training the command is:

[1] http://taku910.github.io/crfpp/

crf_learntemplate_filetrain_filemodel_file

Where the template file and train file which we have created in the previous step. For testing the command is:

% crf_test -m model_filetest_files ...

6.4 CRF Model Feature

We have created a CRF model which includes the following features:

- **Word length**: All the inflected words belong to open category that includes verbs and nouns. Inflection makes them lengthy. We included this feature as a binary feature. We have taken a word length of 3. If the word length exceeds 3 characters, we set the value 1 other wise 0.

- **Contextual information**: The task of the POS tagger is to assign a correct syntactical category to a word. Some words are ambiguous in the corpus. For example, the word "Book" can be used as a Noun or Verb. To resolve this ambiguity, the POS tagger will use the context, i.e. the preceding word and the following word. We have used the context window of size 5, i.e. current word and two previous and two following words.

- **Auxiliary verbs**: Auxiliary verb belongs to a closed category. We have prepared the list of 30 most frequently used auxiliary verbs. This feature is included as a binary feature if the token belongs to this list set the value of this feature 1 otherwise 0. Following are the few examples of Auxiliary verbs:

घुरिजे	चाहिए
वेंदियूं	जाती
आहे	है
वेंदा	जाते

- **Postposition**: Post positions are the most frequently used token in the sentence. They also belong to a closed category. We have identified the 11 most frequently used post position. This feature is also used as a binary feature. Following are the few examples of Post Position.

वटां(पाससे), वांगुर(तरह)

ते (पर), तां (ऊपरसे)

- **Affix, i.e. prefix and suffix:** Sindhi is a morphologically rich language, i.e. various forms of words are present. We can make various word forms, using affixation, i.e. by adding suffix and prefix. We have taken the length of the prefix 3 characters. It is proved that the length of 3 characters gives the best results (Motlani et al. (2015)). We have used a different combination of prefix and suffix length to train the tagger for morphology.

- **Postposition**: Post positions are the most frequently used token in the sentence. They also belong to a closed category. We have identified the 11 most frequently used post position. This feature is also used as a binary feature. Following are the few examples of Post Position.

वटां(पाससे), वांगुर(तरह)

ते (पर), तां (ऊपरसे)

6.5 Results

We have developed six CRF models. These models are evaluated using the aforementioned formulas. The following table shows the overall accuracy of various CRF model for each Tag. A final CRF model CRF_M7 is developed with all features and got 88.79% accuracy.

CRF Model	Features	Accuracy (%)
CRF_M0	No Feature	82.92
CRF_M1	Context	85.53
CRF_M2	Post Position	85.85

CRF_M3	Auxiliary Verb	85.96
CRF_M4	Length	86.34
CRF_M5	3 Prefix	96.93
CRF_M6	3 Suffix	97.05
CRF_M7	All Above Features	88.79

Table 3: Overall accuracy of CRF Model

7 Conclusion and Future Work

POS tagging is an important prerequisite task for any NLP research. In this research work we have developed two taggers for a resource poor language Sindhi in Devanagari script, using stochastic approach. For this we have used 1.HMM and 2.CRF. First we have manually annotated the corpus of 30000 sentences. We have got accuracy of 76.60714% and 88.79% for HMM and CRF respectively.

Sindhi is a morphologically, rich language. The various morphological features such as prefix, postfix and word length will help in defining the POS of a word. CRF can incorporate all these features in the model, and this is one of the main advantages of CRF over HMM. In future to handle the exceptional cases, we could merge the rule-based tagger with the statistical approach. We could use various other machine learning approaches such as SVM to develop the POS tagger. In addition we can develop other tools such as Named Entity Recognizer (NER) which will increase the accuracy of tagger. A robust tagger could be developed using an ensemble approach.

Acknowledgments

We would like to acknowledge lot of people for their help and support. Dr. Haso Dadlani (Ret. Head Department of Sindhi Govt. college Ajmer) for helping us in creating and annotating the data. Dr. Murlidhar Jetley for providing all the necessary Sindhi Grammar books.

References

Behera, P. (2017). An Experiment with the CRF++ Parts of Speech (POS) Tagger for Odia. Language in India, 17(1).

Ekbal, A., Haque, R., & Bandyopadhyay, S. (2007, December). Bengali part of speech tagging using conditional random field. In Proceedings of Seventh International Symposium on Natural Language Processing (SNLP2007) (pp. 131-136).

Garrette, D., Mielens, J., & Baldridge, J. (2013). Real-world semi-supervised learning of POS-taggers for low-resource languages. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 583-592).

Hasan, M. F., UzZaman, N., & Khan, M. (2007). Comparison of Unigram, Bigram, HMM and Brill's POS tagging approaches for some South Asian languages.

I. I. Ayogu, A. O. Adetunmbi, B. A. Ojokoh and S. A. Oluwadare, "A comparative study of hidden Markov model and conditional random fields on a Yorùba part-of-speech tagging task," 2017 International Conference on Computing Networking and Informatics (ICCNI), 2017, pp. 1-6, doi: 10.1109/ICCNI.2017.8123784.

Lata, S., Chandra, S., Verma, P. and Arora, S. (2012). Standardization of POS Tag Set for Indian Languages Based on XML Internationalization Best Practices Guidelines. Proceedings of LREC (WILDRE) First Workshop on Indian Language Data: Resources and Evaluation. Istanbul, Turkey. 01-17.

Mahar, J. A., &Memon, G. Q. (2010 A). Sindhi Part of Speech Tagging System using WordNet. International Journal of Computer Theory and Engineering, 2(4), 538.

Mahar, J. A., &Memon, G. Q. (2010, B). Rule Based Part of Speech Tagging of Sindhi Language. In Signal Acquisition and Processing, 2010. ICSAP'10. International Conference on (pp. 101-106). IEEE.

Mishra, P., Mujadia, V., & Sharma, D. M. (2018). POS Tagging For Resource Poor Indian Languages Through Feature Projection.

Motlani, R., Lalwani, H., Shrivastava, M., & Sharma, D. M. (2015). Developing Part-of-Speech Tagger for a Resource-Poor Language: Sindhi. In Proceedings of the 7th Language and Technology Conference (LTC 2015), Poznan, Poland.

Pakray, P., Pal, A., Majumder, G., & Gelbukh, A. (2015, October). Resource building and parts-of-speech (POS) tagging for the Mizo language. In 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI) (pp. 3-7). IEEE.

Sutton, C., & McCallum, A. (2012). An introduction to conditional random fields. Foundations and Trends® in Machine Learning, 4(4), 267-373.

Sharma, S. K., &Lehal, G. S. (2011, June). Using hidden markov model to improve the accuracy of Punjabi pos tagger. In 2011 IEEE International Conference on Computer Science and Automation Engineering (Vol. 2, pp. 697-701). IEEE.

Singh, J., Joshi, N., & Mathur, I. (2013, August). Development of Marathi part of speech tagger using statistical approach. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1554-1559). IEEE.

Singh, T. D., Ekbal, A., & Bandyopadhyay, S. (2008). Manipuri POS tagging using CRF and SVM: A language independent approach. In proceeding of 6th International conference on Natural Language Processing (ICON-2008) (pp. 240-245).

Sunitha, C. (2015, August). A hybrid Parts of Speech tagger for Malayalam language. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1502-1507). IEEE