Parts of Speech Tagging for Kannada

Swaroop L R, Rakshit Gowda G S, Shriram Hegde and Sourabh U
Department of Information Science and Engineering, Ramaiah Institute of Technology
{swarooplr13, rakshugs, usourabh2011, sshegde66}@gmail.com

Abstract

Parts of speech (POS) tagging is the process of assigning the part of speech tag to each and every word in a sentence. In this paper, we have presented POS tagger for Kannada, a low resource south Asian language, using Condition Random Fields. POS tagger developed in the work uses novel features native to Kannada language. The novel features include Sandhi splitting, where a compound word is broken down into two or more meaningful constituent words. The proposed model is trained and tested on the tagged dataset which contains 21 thousand sentences and achieves a highest accuracy of 94.56%.

1 Introduction

Kannada, an Asian language spoken in southern part of India, is highly agglutinative and rich in derivational morphology. The language has about 2000 years of history and is one of the top 40 most spoken languages of the world. Kannada has clear standards characterized for each part of its structure. Even though Kannada is a Dravidian Language, with time Kannada has been influenced significantly by Sanskrit.

In Kannada, Sandhi is the process where two or more words join based on certain Sandhi rules to form a compound word. During the process of Sandhi, formation changes occur at the word boundaries. For example,

ನಾವು (navu) + ಎಲ್ಲ (yella) = ನಾವೆಲ್ಲ (navella)

ಗಾಳಿ (gaali) + ಅನ್ನು (annu) = ಗಾಳಿಯನ್ನು (gaaliyannu)

Kannada adopts all the Sandhi rules defined in Sanskrit and has three additional Sandhi rules. A Sandhi splitter isolates the constituent words of a Sandhi word utilizing an extensive lexicon and Sandhi rules. Sandhi splitting of a compound word into its component words gives valuable information about its morphology and parts of speech of the compound word.

2 Literature Survey

POS tagging for Indian languages and especially for Dravidian Languages is a difficult task due to the unavailability of annotated data for these languages. Various techniques have been applied for POS tagging in Indian languages.

Gadde et al. (2018) used morphological features with TNT HMM tagger Brants et al (2000) and obtained 92.36% for Hindi and 91.23 % in Telugu. The Hindi POS tagger used Hindi Treebank of size 450K. Ekbal et al. (2008) used SVM for POS tagging in Bengali obtaining 86% accuracy. A semi-supervised pattern-based bootstrapping technique was implemented by Ganesh et al. (2014) to build a Tamil POS Tagger. Their system scored 87.74% accuracy on 20000 documents containing 271K unique words.

Very little work has been done on Kannada because of scarcity of quality annotated data. Antony et al. (2010) was the initial paper which presented part-of-speech tagger for Kannada. They have proposed a tag set consisting of 30 tags. The tag set comprises 5 tags for nouns, 1 tag for pronoun, 8 tags for verbs, 3 for punctuation, two for numbers and 1 each for adjective, adverb, conjunction, echo, reduplication, intensifier, postposition, emphasize, determiner, complementizer, and question word. The researchers have used Support Vector Classification (SVC) a variation of Support Vector Machine (SVM) used for classification problems and tested on 56,000 words for which they obtained an accuracy of 86%.

Proceedings of the Student Research Workshop associated with RANLP-2019, pages 28–31,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/issn.2603-2821.2019_005

The POS tagger tool using Hidden Markov Model (HMM) for Telugu is developed and tested on Kannada Corpus by Siva Reddy et al. (2011). The model gave the F-measure of 77.63 and 77.66 for cross-language and mono-lingual taggers respectively.

Shambhavi et al. (2012) worked on POS tagging for Kannada using Maximum Entropy approach. For training the POS tagger, 51267 words were tagged manually with the help of the tagset. The tagset consisted of 25 tags and the words were collected from EMILLE corpus. Also, Shambhavi et al. (2012) reported 79.9% and 84.58% accuracy using second order HMM and CRF.A POS Tagger for Kannada Sentence Translation is done by Mallamma et al. (2012). Decision trees are used to tag the words.

Prathyusha et al. (2016) a rule based Agama Sandhi splitter has been presented. Agama Sandhi is one of the 7 Sandhis in Kannada language. M. R. Shree et al. (2016) adopted a CRF model for Sandhi splitting. The output of the model is a character level split of the word, hence constituent meaningful base words of the compound (Sandhi) word can't be identified.
AN Akshatha et al. (2017) developed a rule based Sandhi splitter to extract component words from a compound (Sandhi) word.

3 Methodologies

This section gives a description of the dataset used, the features utilized to train the conditional random fields model.

3.1 Dataset

We use the Kannada Treebank project dataset to train our POS tagger. The Kannada Treebank contains three corpora divided based on topic as General, Conversational, and Tourism. The data set is available on the website[1].

[1]*https://ltrc.iiit.ac.in/showfile.php?filename=downloads/kolhi/*

Topic	Tokens	Sentences
General	218,530	17,175
Tourism	26,521	1,883
Conversational	26,521	2,260

Table 1: Corpus information

The corpora were tagged using the Unified Parts of Speech (POS) Standard in Indian Languages drafted by the Department of Information Technology, Govt. of India.

3.2 Models

Two different CRF models were developed in this work. The features used in the first model [Model 1] are:
1. **Context:** The word to be tagged, its preceding three words and succeeding three words
2. **Length:** A binary feature with a value of 0 if the word is shorter than three characters, value of 1 otherwise
3. **Ending characters** (suffix): Last three characters of word.
4. **Is Punctuation**: A binary feature with value 1 if the token contains a non-alphanumeric character and zero otherwise
5. **Is Digit**: A binary feature with a value of 1 if the token contained a digit and 0 otherwise.
6. **POS of first Sandi word**: It is a novel feature where a compound (Sandhi) word is split into its component words, the parts of speech of the first component word is provided as feature value. In case the word is not a Sandhi word i.e. a non compound word the POS tag of word in the word unigram model is provided as feature, if the POS tag is unavailable a none identifier is provide as feature value. For example the compound word ನಾವೆಲ್ಲ (navella meaning "all of us") is split into ನಾವು ("navu" meaning "us") and ಎಲ್ಲ ("yella" meaning "all") the POS tag of "us" i.e. pronoun is the feature value.

A rule based Sandhi splitter described in AN Akshatha et al. (2017) was used to extract component words from a compound (Sandhi) word. The Sandhi word given as an input is

scanned from the left to right to find the longest prefix. This longest prefix will be referred to as expected prefix. This expected prefix is removed from the Sandhi word leaving behind the Sandhi letters and expected suffix, referred to as remainder word. The last letter of the expected prefix is then removed from the expected prefix and added to the beginning of the remainder word. The first one or two letters of the remainder word is most likely to be containing the resultant Sandhi letters. These letters are looked up in the Sandhi rules to identify the Sandhi. Using the reverse Sandhi rule base, the Sandhi letters are replaced with the prefix's ending letter and suffix's beginning letter according to the Sandhi rules. The expected prefix is then added to the remainder word and the words are split. The prefix and suffix thus generated are looked up in the dictionary containing root Kannada words . If both prefix and suffix are found, the Sandhi rule which was applied to split the words is the required Sandhi and the process is terminated as the Sandhi, prefix and suffix words are identified successfully. If the Sandhi is not determined, the second longest prefix is assigned as the expected prefix and the process is continued until the Sandhi is determined or the expected prefix is null.

7. **Last component word**: The last component word of compound (Sandhi) word is used as feature, a none identifier is provided in case of a non-Sandhi word. For example, for the compound word ಹಣದಾಸೆ ("hanadase" meaning "desire for money") is split into ಹಣ ("hana" meaning "money") and ಆಸೆ ("aase" meaning "desire") the component "aase" is the feature value.

The second model [Model 2] uses all the above listed features along with word embedding feature.

8. **Word embedding**: The word embeddings are obtained by training the text corpus using the FastText tool Bojanowski et al., A. Joulin et al., E. Grave et al. Each word is represented by a vector of size 30. Word embeddings represent the current token at a higher level abstraction that helps to recognize the semantics of the token that are not observed in the training set.

4 Results

Table 2 and 3 summarized the results achieved using both the models. Each corpus (General, Tourism and Conversational) is split with a 70-30 ratio for training and testing the POS tagger. In addition all the three corpora were combined to obtain a mixed corpus, the sentences from the three corpora were randomly jumbled and divided into training and testing data.

Accuracy = (No of correctly tagged words) / (Total no of words);

	General	Tourism	Convers-ational	Combined
Model 1	93.42	93.11	91.61	92.69
Model 2	95.84	94.96	93.47	94.56

Table 2: Accuracies of each model

	Precision	Recall	F1-Score
Model 1	91.8	92	91.6
Model 2	93.78	93.21	93.4

Table 3: Detailed result for combined corpus

The accuracy for Model 2 (with word embedding feature) is on the higher side compared to model 1, likewise the cost of training in terms of time and processing for model 2 is higher compared to model 1. The lower accuracies of conversational corpus are a result of higher frequency of colloquial words which makes Sandhi splitting harder. The General corpus being the largest corpus achieves the highest accuracy. The result for combined corpus is almost equal to the three individual corpora; this asserts the models are nearly domain independent. Table [4] gives detailed scores for individual parts of speech tag for Model 1. In this work the Bureau of Indian Standards (BIS) Part Of Speech (POS) tagset prepared for Indian Languages by the POS Tag Standardization Committee of Department of Information Technology has been followed.

POS Tags (BIS)	Precision	Recall	F1-score
N__NN	0.905	0.967	0.935
N__NNP	0.887	0.557	0.684
QT__QTC	0.978	0.858	0.914
DM__DMD	0.981	0.966	0.974
V__VM__VF	0.962	0.969	0.966
RD__PUNC	0.991	0.999	0.995
PR__PRP	0.955	0.962	0.958
V__VM__VNF	0.881	0.890	0.886
JJ	0.780	0.829	0.804
RB	0.739	0.653	0.693
CC__CCD	0.909	0.969	0.938
V__VM__VINF	0.918	0.816	0.864
PSP	0.875	0.920	0.897
CC__CCS	0.860	0.805	0.831
RP__RPD	0.858	0.724	0.786
RD__SYM	0.987	0.924	0.954
QT__QTF	0.706	0.572	0.632
DM__DMQ	0.833	0.714	0.769
DM_DMI	0.776	0.864	0.817
RP__INTF	0.622	0.505	0.557
N__NST	0.835	0.773	0.803
PR__PRQ	0.795	0.837	0.815
RP_INTF	0.687	0.201	0.300
V__VM__VNG	0.853	0.615	0.708
DM__DMI	0.560	0.467	0.509
V__VM	0.000	0.000	0.000
N_NNV	0.667	0.050	0.093
QT__QTO	0.974	0.521	0.679
RP__NEG	0.818	0.818	0.818
V__VAUX	0.738	0.413	0.479
RP__INJ	1.000	0.567	0.723
PR__PRF	0.934	0.966	0.950
CC__CCS__UT	0.000	0.000	0.000
PR__PRI	1.000	0.278	0.435
NULL	0.000	0.000	0.000
PR_PRI	0.667	0.194	0.300
CC_CCS	0.000	0.000	0.000
RD__ECH	1.000	0.333	0.500
N_NN	0.000	0.000	0.000
PR__PRC	1.000	1.000	1.000

Table 4: Result for each POS Tags of Model1

References

Gadde P and Yeleti M.V. 2008. *Improving statistical pos tagging using linguistic feature for hindi and telugu*. Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing, ICON.

Brants T. 2000. *A statistical part-of-speech tagger*. Proceedings of the sixth conference on Applied natural language processing.

Ekbal A. and Bandyopadhyay S. 2008. *Part of speech tagging in bengali using support vector machine*. International Conference on Information Technology.

Ganesh, J, Parthasarathi R, Geetha T, Balaji J. 2014. *Pattern based bootstrapping technique for Tamil pos tagging*. Mining Intelligence and Knowledge Exploration.

Antony P.J and Soman K.P. 2010. *Kernel based Part of Speech Tagger for Kannada*. Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao.

Siva Reddy and Serge Sharoff. 2011. *Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources.* Proceedings of IJCNLP workshop on Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies.

Shambhavi B R, RamakanthKumar P and Revanth. 2012. *Maximum Entropy Approach to Kannada Part Of Speech Tagging*. International Journal of Computer Applications.

Shambhavi B R and RamakanthKumar. 2012. *Kannada Part-Of-Speech Tagging with Probabilistic Classifiers*. International Journal of Computer Applications.

Mallamma V Reddy and Dr. M. Hanumanthappa. 2012. *POS Tagger for Kannada Sentence Translation*. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS).

H. L. Shashirekha and K. S. Vanishree. 2016. *Rule based Kannada Agama Sandhi splitter*. International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur.

M. R. Shree, S. Lakshmi and Shambhavi B.R. 2016. *A novel approach to Sandhi splitting at character level for Kannada language*. International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore.

S. R. Murthy, A. N. Akshatha, C. G. Upadhyaya and P. R. Kumar. 2017. *Kannada spell checker with sandhi splitter*. International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017.

P. Bojanowski*, E. Grave*, A. Joulin, T. Mikolov, *Enriching Word Vectors with Subword Information*.

A. Joulin, E. Grave, P. Bojanowski, T. Mikolov. *Bag of Tricks for Efficient Text Classification.*

A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, T. Mikolov. *FastText.zip: Compressing text classification models*.