Parts of Speech Tagger for Kannada

Adarsh Prakash
Department of Computer Science
University at Buffalo
adarshpr@buffalo.edu | Person #: 5020 8760

Abstract

Kannada is one of 30 most spoken languages in the world. It's a dravidian language spoken predominantly in the state of Karnataka, India. Despite the large usage base, like other Indian languages, there exist minimal linguistic resources for computing and processing. It's rich morphology and agglutinative nature pose a great challenge to even the most basic of natural language processing tasks such as lemmatization, parsing, part-of-speech tagging, chunking etc., In this paper, a supervised learning approach has been proposed to perform **Part-of-Speech** tagging for Kannada by utilizing **linear chain Conditional Random Fields**. A hand-tagged corpus of 7000 words has been used for this task and feature extraction has been performed by leveraging word vectors generated from a trained **Word2Vec** model.

1 Introduction

Part-of-Speech tagging is one of the most basic natural language processing tasks where each word of a sentence is tagged with appropriate syntactic label such as noun, pronoun, verb, adjective and so on. A simple illustration as applied to Kannada is shown below:

ವಿಶ್ವ	ವಿಖ್ಯಾತ	ಕೃಷ್ಣ	ಮಂದಿರ	ಉಡುಪಿಯಲ್ಲಿಯೇ	ಇರುವುದು
vishwa	vikhyAtha	kriShNa	maMdira	uDupiyalliyE	iruvudu
noun	**adj**	**noun**	**noun**	**noun**	**verb**

With a literary history of over 1500 years, grammar, vocabulary and script of Kannada have been influenced both by Sanskrit and Prakrit. The result is a morphologically rich and agglutinative language which poses many challenges to natural language processing. A tagset of 14 tags has been used to perform the part-of-speech tagging. These details have been discussed in further sections.

The corpus was obtained from Indian Institute of Science (IISC) and contains over 7000 words each of which have been assigned a part-of-speech tag by hand. Significant preprocessing has been performed on this corpus to accurately group the words into individual sentences (necessary for CRF) and to correct the human error in the initial part-of-speech tagging process.

All the tasks involved in implementation have been performed in **Python**. Specifically, **gensim** module was used to build the Word2Vec model for feature extraction and **tensorflow** module was used to build the linear chain CRF model. Care has been taken to verify at each stage of implementation that no UTF-8 characters have been misinterpreted or lost.

2 Related Work

Many of the parts of speech taggers have been developed predominantly for English and other European languages. Relatively fewer attempts have been made to develop POS taggers for Indian languages. Nisheeth et al. [1] proposed a POS tagger for Hindi based on Hidden Markov Model that achieves an accuracy of 92% using Indian Language (IL) POS tagset. A Tamil POS tagger developed by Jaybal et al. [2] which achieves an accuracy of 87.74% is one of the first attempts for a Dravidian language.

Few literatures have accomplished notable performance in POS tagging for Kannada. A POS tagger for Kannada using Support Vector Machine (SVM) was proposed by Antony et al. [3]. Their proposal achieves an accuracy of precision of 86% using a hierarchical tagset. Siva Reddy et al. [4] developed a POS tagger for Kannada using Telugu resources with a Hidden Markov Model (HMM). They argue that morphological similarity between these two languages allows for Cross-Language POS tagger. Although, their tagger achieves 77.7% precision they rely heavily on the quality of lemmatization for which very few accurate resources exist. Work by Shambhavi et al. [5] uses EMILLE corpus to compare the performance between Maximum Entropy (Maxent), Hidden Markov Model (HMM) and Conditional Random Fields (CRF). Their CRF model outperformed Maxent and HMM models with an accuracy of 84.6%.

Most recent and notable of these works is the Kannpos POS tagger for Kannada developed by Pallavi et al. [6]. They utilized a large annotated corpus of 80,000 words from TDIL and achieved an accuracy of 92.94% using n-gram CRF model. It can also be noted that, apart from the Cross-Language POS tagger from Siva Reddy et al. [4], implementation details of none of the other POS taggers for Kannada are publicly available.

3 Challenges

As introduced in section 1, Kannada language poses many challenges to even the most basic of language processing tasks. These challenges also limit the extent to which conventions of common spoken languages such as English and other European languages can be utilized for part-of-speech tagging for Kannada. These hurdles have been briefly outlined in the sections that follow.

3.1 Agglutinative Nature

In contrast to Inflectional languages which use isolated elements, Kannada is Agglutinative in nature. A single word may constitute two or more morphemes chained together (sometimes even beyond 10 or 20 morphemes in classical literature!) to yield a single definitive meaning.

Illust. 1 :

ಗಳಿಸಿಕೊಳ್ಳುವೆನೆಂಬಾಲೋಚನೆಯಿಂದ
ಗಳಿಸಿ + ಕೊಳ್ಳುವೆನು + ಎಂಬ + ಆಲೋಚನೆ + ಇಂದ
gaLisikoLLuveneMbAlochaneyiMda

3.2 Tokenization and Lemmatization

Most important challenge to tokenization stems from the effect of agglutinative nature of words. Once we identify the boundary between two tokens, mere separation of these tokens will result in loss of meaning. The two tokens usually require some form of morphing to exist separately and still retain the original meaning. This 'morphing' is governed by complex grammatical rules outlined in later section.

Illust. 2 :
ಸವಿಗನ್ನಡ $\neq$ ಸವಿ + ಗನ್ನಡ
ಸವಿಗನ್ನಡ $=$ ಸವಿ + ಕನ್ನಡ

savigannaDa ≠ savi + gannaDa
savigannaDa = savi + kannaDa

Lemmatization is also riddled with similar challenges. Especially, since stemming will result in loss of meaning for majority words (mostly verbs). In fact, stemming morphs verbs into nouns.

Illust. 3 :
ಗಳಿಸು ≠ ಗಳಿಕೆ

gaLisu ≠ gaLike
(to earn) ≠ (income)

Hence stemming the word **gaLisu** from Illustration 1 to **gaLi** will result in ambiguity and eventually loss in meaning.

3.3 Grammar

In addition to the previous challenges, certain complex grammatical rules are also involved in the use of kannada root words. Conventions and tools available to interpret grammar of English or other European languages fall short for such complex rules.

Sandhi - Rules that govern the decomposition of agglutinized words into atomic words.

Vibhakti - Rules that govern how suffixes to a word change the tone of reference to the subject. There are seven Vibhaktis that are followed even in simple sentences.

Chandassu - Rules that govern the construction of structured literature. This works by counting the time required to pronounce a certain syllable. These rules then guide the construction of sentences or poetry based on a specific number of time units (amsha and maatre) assigned to each numbered line.

4 Feature Extraction

Because of all the challenges presented in the previous section, extracting features from Kannada words comes down to the UTF-8 range of the characters within the words. Instead of this traditional approach, this paper attempts to map each of the words onto a neutral vector space. These vector representations of individual words will be used as features for further supervised learning tasks. These vectors are derived by constructing a Word2Vec model.

4.1 Word2Vec

Natural Language Processing systems conventionally treat words as discrete atomic symbols. These encodings are arbitrary and provide no useful information to the higher level statistical model regarding the relationships that may exist between individual symbols. For example, words 'bike' and 'car' are just treated as string of characters (or encodings - UTF8, UTF16, ASCII ..) and this means that statistical models, say HMMs, that use these encodings can not leverage the contextual information such as - both are modes of transport or objects used by the subject in the sentence.

These obstacles can be overcome by using vector representation of words instead of arbitrary encodings. **Vector Space Model** (VSM) represents words in a continuous vector space where semantically similar words are mapped to nearby points. This idea is leveraged by the Word2Vec model. **Word2Vec** is an efficient *neural probabilistic predictive model* for learning word embeddings (vectors) from raw text. Internally, Word2Vec can be implemented by using mainly Continuous Bag-of-Words model or Skip-Gram model. Feature extraction for the current work was done using bag-of-words (CBOW) model because of its statistical effect of smoothing over a distribution.

4.2 Context Window

As introduced earlier, the task of POS tagging is performed by training a linear chain CRF model. This model requires contextual information of a word and its neighborhood. Hence, feature set of any word, which is a vector derived from the Word2Vec model, must also be accompanied by the information of it's neighboring words. So, context window of 5 words is defined and feature set of any word includes features of its neighboring words as illustrated below.

$$\underbrace{\textbf{(W-2)} \quad \textbf{(W-1)} \quad \textbf{W} \quad \textbf{(W+1)} \quad \textbf{(W+2)}}_{\text{Context Window for word } \mathbf{W}}$$

5 Tagset

A tagset of 14 part-of-speech tags modeled after Universal POS Tagset has been used in the tagging process. Table 1 illustrates the use of each tag.

Table 1: Description of tagset

Tag	Description	Example
CC	Conjunction	ಮತ್ತು, ಹಾಗೂ
DEM	Demonstrative	ಈ, ಆ
DET	Determiner	ಅಲ್ಲಲ್ಲಿ, ಹೆಚ್ಚು
JJ	Adjective	ಒಳ್ಳೆಯ, ದುಷ್ಟ
NN	Noun (common and proper)	ಭಾಷೆ, ಕರ್ನಾಟಕ
NUM	Numeric	೧೯೭೭, ೬೮
PRP	Pronoun	ನಾನು, ಇವರ
PSP	Postposition	ಜೋತೆ, ಮೇಲೆ
QC	Quantifier/Cardinal	ಒಂದು, ಬಹಳ
RB	Adverb	ವೇಗವಾಗಿ, ಪೂರ್ತಿಯಾಗಿ
SYM	Symbol	!, ?
UT	Quotative	ಎಂದು, ಹೇಳುವಂತೆ
VM	Verb	ಬರೆದನು, ನಡೆಯುತ್ತ
WQ	Question word	ಯಾರು, ಯಾವ

6 Implementation

A simple **Linear Chain Conditional Random Fields** model (log-linear) was used to train and predict the part-of-speech tags of the words in the corpus.

$$P(Y|X) = \frac{1}{Z(X)} \quad exp\{\Sigma_{t=1}^{T}\Sigma_{k=1}^{K}\lambda_k f_k(Y_t, Y_{t-1}, X_t)\}$$

where

$\mathbf{Z(X)}$ = $\Sigma_Y\ exp\{\Sigma_{t=1}^{T}\Sigma_{k=1}^{K}\lambda_k f_k(Y_t, Y_{t-1}, X_t)\}$, Partition function

$\mathbf{Y}$ - POS tag of the current word

$\mathbf{X}$ - feature set of the current word

T - sequence length

K - number of features

$f_k()$ - feature function k of the current word

7 Results

The trained CRF model was evaluated against a test set of 900 words and the the following results were observed.

Table 2: Classification Report

Class	Precision	Recall	F1-Score	Support
Conjuncts	0.90	0.60	0.72	15
Demonstrative	0.85	0.65	0.73	17
Adjective	0.14	0.08	0.10	49
Noun	0.60	0.78	0.68	306
Pronoun	0.74	0.58	0.65	110
Adposition	0.80	0.29	0.42	14
Quantifier	0.69	0.45	0.55	20
Adverb	0.74	0.30	0.43	46
Symbol	0.99	0.99	0.99	83
Quotative	1.00	0.89	0.94	9
Verb	0.64	0.72	0.67	190

8 Scope for Future Work

Certain interesting observations were made during course of implementation. Firstly, classification accuracy on the training set showed a clear increase correlated to a proportional increase in the training data. Second, increase in the number of features during Word2Vec construction led to an increase in the classification accuracy of CRF model. Finally, training the Word2Vec model beyond 1000 iterations resulted in an overfit CRF model with 95% training accuracy but a decrease classification accuracy on the test set.

Based on these observations, the following conclusions can be drawn in regards of future work that can be done this area.

1. Obtaining a large corpus would definitely increase the performance of both Word2Vec and linear chain CRF models.
2. Combining the contextual information from Word2Vec with the actual character encodings by applying efficient lemmatization techniques that can overcome the challenges presented section 3.2 of this document.

References

[1] Joshi, N., Darbari, H., Mathur, I.: HMM based POS tagger for Hindi. In: Proceeding of 2013 International Conference on Artificial Intelligence, Soft Computing (AISC-2013) (2013)

[2] Ganesh, J., Ranjani Parthasarathi, T. V. Geetha, and J. Balaji. "Pattern Based Bootstrapping Technique for Tamil POS Tagging." In Mining Intelligence and Knowledge Exploration, pp. 256–267. Springer International Publishing, 2014

[3] Antony, P.J., Soman, K.P.: Kernel based part of speech tagger for Kannada. In: International Conference on Machine Learning and Cybernetics (ICMLC), vol. 4, pp. 2139–2144, IEEE (2010)

[4] Reddy, S., Serge S.: Cross language POS taggers (and other tools) for Indian languages: an experiment with Kannada using Telugu resources. Cross Ling. Inf. Access 11 (2011)

[5] Shambhavi, B.R., Ramakanth, K.P., Revanth, G.: A maximum entropy approach to Kannada part of speech tagging. Int. J. Comput. Appl. 41(13), 9–12 (2012)

[6] Pallavi., Pillai, A.S.: Parts Of Speech (POS) Tagger for Kannada using conditional random fields (CRFs). In: National Conference on Indian Language Computing (NCILC 2014) 1st to 2nd Feb 2014