Sanskrit Tag-sets and Part-Of-Speech Tagging Methods
- A Survey

Sulabh Bhatt\textsuperscript{a}, Krunal Parmar\textsuperscript{a} and Miral Patel\textsuperscript{b}
\textsuperscript{a}Department Of Computer Science, Gujarat University, Ahmedabad, India
\textsuperscript{b}Department Of I.T., G H Patel College of Engineering and Technology, Vallabh Vidhyanagar, India

ABSTRACT:
This paper presents some great features of Sanskrit- one of the oldest language of the world from Natural Language Processing (NLP) perspective. Part Of Speech (POS) tagging is the most initial step for developing any NLP application. POS tagger assigns a tag like noun, verb, pronoun, adjective or other category that best suits to the word and also the context of the sentence to which it belongs. Moreover, this paper also provides brief introduction to various approaches and the working of two most famous statistical methods used for POS tagging: Hidden Markov Model (HMM) and Conditional Random Fields (CRF).

Keywords: NLP, POS tagger, tagset, Sanskrit, Stochastic Tagging, n-gram, HMM, CRF

I. INTRODUCTION

Natural Language Processing (NLP): The aim of Natural Language Processing is to make the computers understand how humans naturally speak, write, and communicate. The main goal of NLP is to bridge the human-computer gap by reducing extra efforts needed by a human to communicate with a computer. Major applications of NLP are Machine translation, Named entity recognition (NER), Part of speech (POS) tagging, Question answering, etc.

Most of the research have been done for English language, but English has ambiguous grammar so we need a strong and unambiguous grammar which is provided by maharishi Panini in the form of Ashtadhyayi [4]. Sanskrit grammar is the most simplest and ancient grammar in the world. Unlike other languages of the world, there has never been any kind of change in the structure of Sanskrit grammar.

One of the most important and initial activities in processing a natural language is POS tagging. A POS Tagger assigns a POS tag to each word of a sentence. The design of the annotated Sanskrit corpus is rapidly going on in India [5]. Natural languages are basically very complicated and Sanskrit also falls in the same group.

In the development of a POS Tagger, the quality of the POS annotation in a corpus is very much important. A tagged corpus can be used for many activities. A good tagger can serve as a base for many NLP related activities. It can be also used for linguistic research. Moreover, it can be used further for stemming in information retrieval (IR), since knowing a word’s part of speech can help in telling us which morphological affixes it can take.

Automatic POS taggers can be used for developing automatic word-sense disambiguating algorithms; POS tagging can also be used for developing text to speech related activities.

II. DISCUSSION OF SANSKRIT TAG-SETS

Based on some general principles for tagset design strategy, many POS tagsets have been developed by different organizations. This section provides a survey of some tagsets developed especially for Sanskrit or for Indian natural languages and then their evaluation with respect to Sanskrit.

Tags are shown in figures in their respective sections.

A. JNU Sanskrit tagset(JPOS)

This POS tagset is developed by Dr. R. Chandrashekar as a part of his Ph.D. thesis ‘Part-of-Speech Tagging for Sanskrit’ (2007) [19]. This tagset very strictly follows Paninian grammatical rules. This tagset is fine-grained and captures linguistic features as per the traditional grammar of Sanskrit language.

This tagset can be classified as per the morphological structure of Sanskrit words. This tagset contains three kinds of tags. Word class main tags, feature sub-tags and punctuation tags [5].

This tagset is having 134 tags. 65 word class tags, 43 feature sub-tags and 25 punctuation tags and one tag AJ to tag unknown words [18].
B. IL-POSTS Sanskrit Tagset

This tagset can be derived from IL-POSTS, which is a standard framework for Indian languages developed by Microsoft Research India Lab (MSRI) [5].

This tagset encodes data at three levels: Categories, Types, and Attributes. As far as we know, tagsets for four languages have been derived from this framework: Hindi, Bangla, Tamil and Sanskrit [5].
C. *Sanskrit Consortium Tagset (CPOS)*

This tagset is developed for tagging sandhi-free Sanskrit corpora. This tagset is based on traditional Sanskrit grammatical categorization.

This tagset has 28 tags. This is intuitively a flat tagset and allows the tagging of major categories only. It seems that, most of the categories have been adapted from the JPOS tagset [5].

III. POS TAGGING TECHNIQUES

Various methods exist for POS Tagging. The tagging models can be classified into two types: Supervised and Unsupervised. Both of these techniques differ in regarding the degree of automation of the training process and the tagging process.

The supervised models require a pre-tagged corpus which can be used for training to explore information regarding the tagset, tag sequence probabilities, word-tag frequencies and/or rules, etc. Various taggers are existing based on supervised models.
The unsupervised models do not require previously tagged corpora. Instead, they use advanced computational techniques to automatically generate tagsets, transformation rules, etc. Using this information, they either generate the contextual rules needed by rule based or transformation based taggers or calculate the probabilistic information required by the stochastic POS taggers.

Both the supervised and unsupervised POS taggers can be further classified into three types:

A. **Rule based Tagger**

Rule-based taggers use rules, which may be hand-written by experienced linguistic people or derived from a tagged corpus. Rules can be defined from previous experiences and they help in distinguishing the tag ambiguity.

E.g., Brill tagger (Brill, 1995) [11] is a rule based tagger. It includes lexical rules, used for initialization purpose and contextual rules, defined to correct the wrong tags.

B. **Stochastic Tagger**

Stochastic taggers are based on statistics i.e., probability or frequency to tag the input words. The simplest stochastic taggers solve the issue of ambiguity of words based on the probability that the given word occurs with a particular tag. In the testing data, the ambiguous instance of a word is assigned the tag that is encountered most frequently in the training set for the same word. The disadvantage of this method is that it might assign a correct tag to a given word but it could also yield invalid sequences of tags. One alternative way to the word frequency approach is to calculate the probability of a given sequence of tags taking place together. This is referred to as n-gram technique, which states that the best suitable tag to the given word can be determined by the probability that it occurs with the preceding n-1 tags. The stochastic model is based on various models such as n-grams, Maximum Likelihood Estimation, HMM.

1) **HMM Based Tagger**: HMM-based methods require to evaluate argmax formula. It can be very expensive because, finding the sequence which maximizes the probability, all possible tag sequences must be checked. Therefore, a dynamic programming technique known as the Viterbi Algorithm can be used to find optimal tag sequence. Also, there have been several studies to utilise unsupervised method for training a HMM for POS Tagging. Baum-Welch algorithm [Baum, 1972] is the most widely known for this, that can be used to train HMM from un-annotated corpus. A POS tagger based on HMM assigns the best tag to a word by calculating the forward and backward probabilities of tags along with the sequence provided as an input. The following equation explains this phenomenon [12].

\[
P(t_i|w_i) = P(t_i|t_{i-1}) \cdot P(t_{i+1}|t_i) \cdot P(w_i|t_i)
\]

Here, \( P(t_i|t_{i-1}) \) is the probability of a current tag given the previous tag and \( P(t_{i+1}|t_i) \) is the probability of the future tag given the current tag. This captures the transition between the tags. These probabilities are computed using following equation [12].

\[
P(t_i|t_{i-1}) = \frac{\text{freq}(t_{i-1}, t_i)}{\text{freq}(t_{i-1})}
\]

**Figure 6. Tag transition probability [12]**

Each tag transition probability can be computed by calculating the frequency count of two tags seen together in the corpora divided by the frequency count of the previous tag seen independently in the corpus. This is done because we know that it is more likely for some tags to precede the other tags.

2) **CRF Based Tagger**: Charles Sutton et al. (Sutton et al., 2005) formulated CRFs as follows. Let G be a factor graph over Y. Then \( p(y|x) \) is a conditional random field if for any fixed x, the distribution \( p(y|x) \) factorizes according to G. Thus, every conditional distribution \( p(y|x) \) is a CRF for some, perhaps trivial, factor graph. If \( F = \{A\} \) is the set of actors in G, and each factor takes the exponential family form, then the conditional distribution can be written as [14]:

\[
\]
Figure 7. CRF working formula [14]

\[ p(y|x) = \frac{1}{Z(x)} \prod_{i \in G} \exp \left\{ \sum_{k=1}^{R(A)} \lambda_{jk} f_{jk}(y_{i-1}, y_i, x_i) + \sum_{k=1}^{S(A)} \mu_{ksk}(y_i, x_i) \right\} . \]

X here is a random variable over data sequences to be labeled, and Y is a random variable over corresponding label sequences. All components Yi of Y are assumed to range over a finite label alphabet Y. For example, X might range over natural language sentences and Y range over part-of-speech tagging of those sentences, with Y the set of possible part-of-speech tags. The random variables X and Y are jointly distributed, but in a discriminative framework we construct a conditional model \( p(Y|X) \) from paired observation and label sequences, and do not explicitly model the marginal \( p(X) \).

CRFs define conditional probability distributions \( p(Y|X) \) of label sequences given input sequences. Lafferty et al. defines the probability of a particular label sequence Y given observation sequence X to be a normalized product of potential functions each of the form [14]:

\[ \exp(\sum \lambda_{jk} f_{jk}(Y_{i-1}, Y_i, X, i) + \sum \mu_{ksk}(Y_i, X, i)) \]

Where \( t_j(Y_{i-1}, Y_i, X, i) \) is a transition feature function of the entire observation sequence and the labels at positions \( i \) and \( i-1 \) in the label sequence; \( s_k(Y_i, X, i) \) is a state feature function of the label at position \( i \) and the observation sequence; and \( \lambda_j \) and \( \mu_k \) are parameters to be estimated from training data.

\[ F_j(Y, X) = \sum f_j(Y_{i-1}, Y_i, X, i) \]

Where each \( f_j(Y_{i-1}, Y_i, X, i) \) is either a state function \( s(Y_{i-1}, Y_i, X, i) \) or a transition function \( t(Y_{i-1}, Y_i, X, i) \). This allows the probability of a label sequence Y given an observation sequence X to be written as:

\[ P(Y|X, \lambda) = \frac{1}{Z(X)} \exp(\sum \lambda_j F_j(Y, X)) \]

Where \( Z(X) \) is a normalization factor.

C. Neural Tagger

Neural taggers are based on neural networks. They learn the parameters of POS tagger from a training data set. The performance of Neural Taggers can be better than stochastic taggers.

IV. CONCLUSION

Even though Sanskrit is considered as a mother of all Indo-European languages, very less work is done using stochastic approach for Sanskrit. So, developing NLP applications for Sanskrit using stochastic methods is still an open area for researchers.

Development of a highly accurate POS tagger for Indian languages is an active research area of NLP. The obstacle in developing POS tagger for Indian languages including Sanskrit is the none or less availability of large corpora. The annotated corpora will help researchers to explore the utilization of statistical methods to enhance the existing models for data analysis.

For a tagger to function as a practical component in a language processing system, a tagger must be Robust, Efficient, accurate, tunable and reusable. The accuracy of any NLP application depends on the accuracy of a POS tagger.

ACKNOWLEDGMENT

We are extremely grateful to Mr. Hardik Joshi for their invaluable suggestions and encouragement.

REFERENCES


