ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Parts of Speech Tagging For Konkani Language

[1] Diksha N.Prabhu Khorjuvenkar, [2] Megha Ainapurkar, [3] Sufola Chagas
[1] UG Scholar, [2] Assistant Professor
[1][2][3] Department of Information Technology, PCCE, Verna, India

Abstract - **It is remarkable to note that the scope of Natural Language Processing (NLP) is developing and increasing in the area of text mining. Natural Language Processing is a field that covers computer understanding and deals with the manipulation of human language. Human language is an unstructured source of information, and hence to use it, as an input to a computer program, it has to be, first, converted into a structured format [3]. Parts of Speech (POS) tagging is one of the steps which assigns a particular part of speech to a respective word. POS is difficult because most words tend to have more than one parts of speech in different cases and some parts of speech are complex or unspoken. This paper aims at developing part of speech tagging model for Konkani language, using the Konkani corpus.**

Keywords— Part of Speech, Konkani, NLP.

I. INTRODUCTION

Parts of Speech tagging is a significant tool for processing natural languages. It is also known as grammatical tagging or word-category disambiguation. POS taggers are designed with the aim of analyzing corpus data to determine the categories of the words or phrases used in the data[4].It accepts a raw text as input and assigns a particular part of speech like noun, verb, adjective, pronouns, conjunction and their sub-categories.It is a process of mapping from a sequence of words to a sequence of lexical categories.

Part of Speech Tagging has been divided into two classes, Supervised and Unsupervised taggers. Supervised taggers are based on pre-tagged corpora, while unsupervised taggers automatically assigns tag to a given word [5]. Furthermore we have two types: (i)Rule Base Taggers: The rule based POS tagging approach works by creating hand written rules. (ii) Stochastic Taggers: A stochastic approach tries to assign a tag to a particular word depending on probability concepts.

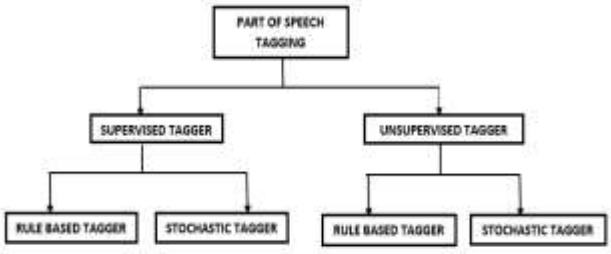

Fig1: Classification of Part of speech tagging

POS Tagging is an initial stage of information extraction, summarization, retrieval, machine translation, speech conversion. In this paper, we are trying to develop a POS tagging model for Konkani corpus by using some algorithm.

II. PROPOSED MODEL

The proposed model is to perform Part of Speech tagging for Konkani language and display the respective tag for unseen data.

1) Konkani language

Konkani is an Indo-European, morphologically rich language. It is one of the twenty two languages incorporated in the Eighth Schedule of the Indian Constitution. Konkani is a language that is spoken in the state of Goa with Devanagari as the officially recognized script. The capital of Goa is Panaji. Apart from Goa, it is also spoken in Maharashtra, Karnataka and Kerala.
Since Konkani is spoken on the Indo-Aryan/Dravidian border, it is significantly influenced by Kannada. Goa was ruled by Kannada for centuries before Portuguese arrival. Therefore old Konkani from the 16th and 17th centuries show major influence of Kannada.

Konkani also uses loan words from Sanskrit, Perso-Arabic, Kannada, and Portuguese. It is also enriched by other languages like Marathi, Malayalam, Hindi and English

2) Model

This model involves three baisc steps. However one can change and modify these steps according to ther needs and with respect to the method they use.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Our proposed model does the POS tagging with the help of the Hidden Markov Model (HMM) and further uses the Viterbi algorithm to find the best tagging for a given word.

The three steps are as follows:

A. Collection of data

This step consists of collecting Konkani text data which is pre-tagged. This data is further used in the training and testing process.

B. Training Phase

The second step is about training the model using our data. Here the pre-tagged data is used to train the model because it helps to learn about the tagsets, frequencies etc which are helpful in tagging process.

During the training phase a pair of words and the respective tag is sent to the machine learning algorithm to generate the appropriate model.

C. Testing Phase

The third step is to test the generated model. In this step,the input consists of Konkani text data(only words). This data is tested and the model generates tags for each of these words.

During the testing process the model tries to predict the tag for unseen data or test data.

III. MODEL DESIGN

As per the three steps described above, in this section we will see how the three steps are executed in detail.

1) Corpus Data

In the first step we try to collect Konkani pre-tagged data. This data is used for training purpose. The testing data consists of the unseen Konkani text.The training data consists of a word along with the associated tag like word1/tag1.

As given in the architecture diagram the POS model is divided into two layers. The first layer is a visible layer corresponding to the input words, and the second layer is a hidden layer learnt by the system with respect to the tags.

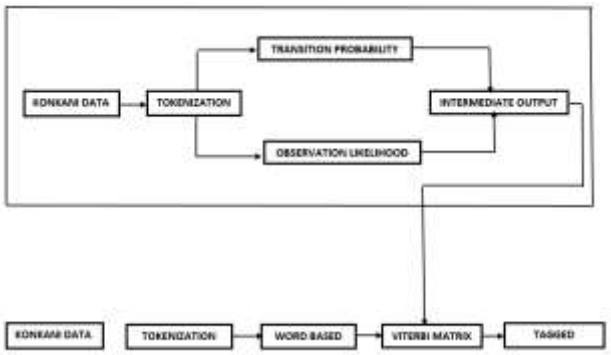

Fig2: Architecture of the model

The main modules in the figure are:

1.Konkani data: This consists of the Konkani pre-tagged corpus.

2.Tokenization: The second module is used to tokenize the data, that is to break down the sentences to its lower form.

3.Word based: In this module we split the sentences according to the delimeter and we get the words with their respective tags.

4.Transition probability: Here we calculate the transition probability of each tag sequence from the tagged corpus.

5.Observation likelihood: In this module we compute the observation probability for each of the words from the tagged corpus.

6.Intermediate output: The intermediate values are stored.

7.Viterbi matrix: It collects all the data from intermediate output module and prepares a state graph in the matrix and computes the transition probability for each transition present in the matrix. It then finds a best tag based on maximum probability.

8.Tagged output: This is the tagged output which is obtained.

2) Training the model

The main part of this model is to train it using our Konkani text data which consists of a word and tag respectively.The model is trainied using the hidden markov model and then the Viterbi algorithm is applied to find the best tag for a word

A. Hidden Markov Model:

Hidden Markov model is appropriate for cases where somethings are hidden and some things are observed. In this case the observed thing is the word and hidden thing is the tag.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

A Hidden Markov Model is defined by the following
•A set of states
•Transition probability between the states
•Observation likelihood which represents the probability of an observation being generated from a hidden state.

HMM tagger chooses the tag sequence which maximizes the probability
P (word | tag) * P(tag | previous n tags)

It chooses a tag sequence for a given sentence rather than individual words. If W be the word sequence and T be the tag sequence then we get an expression as follows:

$P(T|W) = = ar\ gmax_{t \in T} P (T |W)$

After applying Bayes rule we get :

P(T | W) = P(W | T) * P(T) | P(W)

In this expression P(W) can be dropped as it remains same for each sequence. Hence the expresiion for most likey tag sequence is

$P(T|W) = = ar\ gmax_{t \in T} P (T)P (WI T)$

Markov assumption states that probability of the tag sequence is the product of the probability of the constituent n-grams. The two assumptions which can be made are that the words are independent of each other and the probability of a word is only dependent on its tag.

Using these assumptions the equation is

$P(W | T) = P(w_1 | t_1) * P (w_2 | t_2) …….. P(w_n | t_n)$

After this the Viterbi algorithm is applied for finding the best tag sequence depending on the probability Hidden Markov model are used in various other NLP applictions like speech recognition, spell checking, named enitiy recognition etc.

B. Viterbi Algorithm :

This algorithm is used for tagging and is most widely used in other NLP applications.It is a dynamic programming algorithm. It deals wih all the words in a given sentence at the same time and computes a probable tag sequence for the sentence. The main aim of this algorithm is to find the best tag, so to do this it makes a matrix with i denoting the number of tags and j denotes the total number of words in the given sentence.

In the matrix the cell V[i,j] contains the probability of the path with respect to the previous probability of the states.Entries are made column by column. The key point here is that cell V[0,0]=1 and rest cells V[0,b] for all b = 1 to j+2. In this way each cell entry will contain the probability of the most likely path to end in the cell.
Next recursively compute the probability, by maximizing over the probabilities of the preceeding states and finally get the best path from the final state.
This algorithm has its initial application in communication and wireless LAN's. It is now commonly used in speech recognition, bioinformatics, NLP and computational linguistics.

3) Testing the model

The last step of the model is to test it with the unseen or test data. The test data contains only the Konkani words witout the tags.

This data is given as an input to the model. The model generates a tag for each given word,depending on how it is trained in the training phase.

In this phase we get a tagged output containing the words and respective tag and the domain of the test data is displayed.

IV. CONCLUSION

Parts of speech tagging plays a very important role in various NLP applications as well as it is very beneficial in the context of Information Extraction and Question Answering systems [4]. In the above described model,first we generate two sets of data the train data and test data which consists of Konkani text.

Using the train data we train the model and test data is used to test the functioning of the model And hence the tagging for each Konkani word is done. As Konkani is a morphologically rich language, tagging procedure encounters some problems, but they are handled accurately.

Our proposed model does the task of annotating the words occurring in a text with their corresponding, particular part of speech with HMM and Viterbi algorithm. This tagged data can be further used in an appropriate way in the tasks of information retrieval,speech recognition etc

V. FUTURE SCOPE

This POS model for tagging can further be optimized by applying a different method for training process which will eventually change the accuracy and the working of the model.The second addition would be to increase the

amount of training data as well as to use the data which belongs to different domains. When the model is trained on different domains of data, it is sure to give better and efficient result while testing the unseen or test data.

It might also be noted that, along with the tagging process one can try to predict the domain to which the test documents belongs. This will be dependent on the main words present in the test document which depict its domain.

REFERENCES

1)L.R.Rabiner, B.H.Juang, "An introduction to Hidden Markov model,(IEEE),(1986)

2)Mishra,N. and Mishra, A " Part of Speech Tagging for Hindi Corpus", International Conference on Communication Systems and Network Technologies,(ICCSN),(2011)

3)Patheja,P , Waoo,A. and Garg,R, "Analysis of Part of Speech Tagging", Internantional Journal of Computer Applications,(ICISS),(2012)

4)Nisheeth J , Hemant D and Iti M, "HMM based POS tagger for Hindi",Parallel,Distributed Computing Technologies and Applications, (PDCTA),(2013)

5)Panka Patil,H; Patil,A. and Pawar,B "Part-of-Speech Tagger for Marathi Language using Limited Training Corpora", International Journal of Computer Applications (IJCA),(2014