Pune Institute of Computer Technology, Pune

Department of Computer Engineering

A Mini Project Report on

“POS Taggers For Indian Language”

SUBMITTED TO THE UNIVERSITY OF PUNE, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING
Computer Engineering

BY

SAMARTH MALI	**ROLL NO:41146**
SARTHAK NIRGUDE	**ROLL NO:41153**
KEDAR PAWAR	**ROLL NO:41158**

UNDER THE GUIDANCE OF PROF. P. S. JOSHI

DEPARTMENT OF COMPUTER ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE
2024-25

<u>CERTIFICATE</u>

This is to certify that the Project Report entitled

”Patient Readmission Prediction”

Submitted by

SAMARTH MALI	**ROLL NO:41146**
SARTHAK NIRGUDE	**ROLL NO:41153**
KEDAR PAWAR	**ROLL NO:41158**

is a bonafide work carried out under the supervision of Prof. P. S. JOSHI and it is submitted towards the partial fulfillment of the requirement of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering(Computer Engineering).

Prof. P. S. JOSHI

(Guide)

Dr. Geetanjali Kale

(Head of Department

of Computer Engineering)

Dr. S.T. Gandhe

(Principal,Pune Institute of Computer

Technology, Pune)

Internal Examiner

External Examiner

Place: Pune

Date:

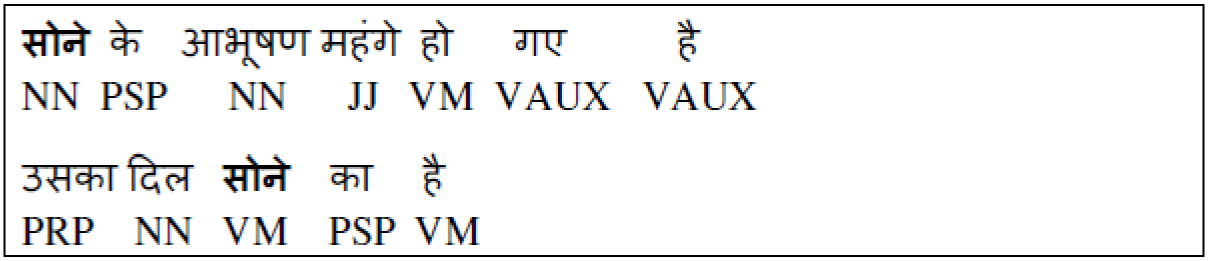
TABLE OF CONTENTS

INTRODUCTION

Part of Speech (POS) Tagging is the first step in the development of any NLP Application. It is a task which assigns POS labels to words supplied in the text. This is the reason why researchers consider this as a sequence labeling task where words are considered as sequences which needs to be labeled. Each word's tag is identified within a context using the previous word/tag combination. POS tagging is used in various applications like parsing where word and their tags are transformed into chunks which can be combined to generate the complete parse of a text.

Taggers are used in Machine Translation (MT) while developing a transfer based MT Engine. Here, we require the text in the source language to be POS tagged and then parsed which can then be transferred to the target side using transfer grammar. Taggers can also be used in Name Entity Recognition (NER) where a word tagged as a noun (either proper or common noun) is further classified as a name of a person, organization, location, time, date etc.

Tagging of text is a complex task as many times we get words which have different tag categories as they are used in different context. This phenomenon is termed as lexical ambiguity. For example, let us consider text in Table 1. The same word 'सोना' given a different label in the two sentences. In the first case it is termed as a common noun as it is referring to an object (Gold Ornament). In the second case it is termed as a verb as it is referring to an experience (feelings) of the speaker. This problem can be resolved by looking at the word/tag combinations of the surrounding words with respect to the ambiguous word (the word which has multiple tags). Over the years, a lot of research has been done on POS tagging. Broadly, all the efforts can be categorized in three directions. They are: rule based approach where a human annotator is required to develop rules for tagging words or statistical approach where we use mathematical formulations and tag words or hybrid approach which is partially rule based and partially statistical. In the context of European languages POS taggers are generally developed using machine learning approach, but in the Indian context, we still do not have a clear good approach. In this paper we discuss the development of a POS tagger for Hindi using Hidden Markov Model (HMM).



सोने	के	आभूषण	महंगे	हो	गए	है
NN	PSP	NN	JJ	VM	VAUX	VAUX
उसका	दिल	**सोने**	का	है		
PRP	NN	VM	PSP	VM		

Table 1. Example of Lexical Ambiguity

IMPLEMENTATION & IMPORTANT MODULES

In this project we are doing POS tagging for Hindi sentences and for that we have used **Python**. For developing a HMM based tagger we were first required to annotate a corpus based on a tagset.

Modules

Downloading dataset

So using our source code we first download a Hindi dataset which has numerous sentence in Hindi.

Preprocessing the downloaded dataset

Our next step is to preprocess the corpus dataset which we have downloaded so as to implement operations on it. This is done by selecting every individual sentence from the dataset for which we want POS tagging.

Stripping words in sentence

Following the previous step, we strip the words from the sentence so that separate operations can be performed. For example, we take a sentence and strip it, we then perform operation on each constituting word to tag it with the most accurate POS tags.

Training POS tagger

This way we achieve the module of training the POS tagger with the results thus obtained for each and every data in the corpus. Now the POS tagger is ready to tag any Hindi sentence.

Tagging new line

.At the end when the POS tagger is trained, it can then be used for tagging new Hindi lines, according to the user’s choice.

Implementing Concepts

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.

$$P(t_i|w_i) = P(t_i|t_{i-1}).P(t_{i+1}|t_i).P(w_i|t_i) \qquad \mathbf{(1)}$$

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 equation 2.

$$P(t_i|t_{i-1}) = \frac{freq\ (t_{i-1}, t_i)}{freq(t_{i-1})} \qquad \mathbf{(2)}$$

Each tag transition probability is computed by calculating the frequency count of two tags seen together in the corpus 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. For example, an adjective (JJ) will be followed by a common noun (NN) and not by a postposition (PSP) or a pronoun (PRP). Figure 1 shows this example

अच्छा लड़का (*) अच्छा के (*) अच्छा तुम

JJ NN JJ PSP JJ PRP

Figure 1. Tag transition probabilities

POS Tags for Hindi sentences

S.No.	Tag	Description (Tag Used for)	Example
1.	NN	Common Nouns	लड़का, लड़के, किताब, पुस्तक
2.	NST	Nourn Denotating Spatial and Temporal Expressions	ऊपर, पहले, बहार, आगे
3.	NNP	Proper Nourns (name of person)	मोहन, राम, सुरेश
4.	PRP	Pronoun	वह, वो, उसे, तुम
5.	DEM	Demonstrative	वह, वो, उस
6.	VM	Verb Main (Finite or Non-Finite)	खाता, सोता, रोता, खाते, सोते, रोते
7.	VAUX	Verb Auxilary (Any verb, present besides main verb shall be marked as auxillary verb)	है, हुए, कर
8.	JJ	Adjective (Modifier of Noun)	सांस्कृतिक, पुरानी, दुपहिया
9.	RB	Adverb (Modifier of Verb)	जल्दी, धीरे, धीमे
10.	PSP	Postposition	में, को, ने
11.	RP	Particles	भी, तो, ही
12.	QF	Quantifiers	बहुत, थोडा, कम

13.	QC	Cardinals	एक, दो, तीन
14.	CC	Conjuncts (Coordinating and Subordinating)	और, की
15.	WQ	Question Words	क्यों, क्या, कहा
16.	QO	Ordinals	पहला, दूसरा, तीसरा
17.	INTF	Intensifier	बहुत, थोडा, कम
18.	INJ	Interjection	अरे, हाय
19.	NEG	Negative	नहीं, ना
20.	SYM	Symbol	? , ; : !
21.	XC	Compounds	केंद्र/XC सरकार/NN रंग/XC बिरंगे/JJ
22.	RDP	Reduplications	धीरे/RB धीरे/RDP