CRF based Part of Speech Tagger for domain specific Hindi Corpus

Vaishali Gupta
Department of Computer Science, Apaji Institute, Banasthali University
Rajasthan, India

Nisheeth Joshi, PhD
Department of Computer Science, Apaji Institute, Banasthali University
Rajasthan, India

Iti Mathur, PhD
Department of Computer Science, Apaji Institute, Banasthali University
Rajasthan, India

ABSTRACT

Natural language processing (NLP) is a field of artificial intelligence and computational linguistics which is concerned with the interactions between human (natural) languages and computers. As known, NLP is related to the area of human–computer interaction. There are various phases involves in Natural language processing. POS Tagging is one of the necessary phases in NLP.

Part of Speech Tagger is an important tool that is used to develop language translator and information extraction. The problem of tagging in natural language processing is to find a way to tag (annotate) each and every word in a sentence. This study presents a part of speech tagger (POS Tagger) for domain specific Hindi Language. The evaluation of the system is done on the Agricultural domain of Hindi Corpus using Conditional Random Field model.

General Terms

NLP, Machine Translation, Machine Learning

Keywords

POS Tagger, Corpus, CRF, Model File, AI, Agriculture Domain.

1. INTRODUCTION

Natural Language Processing is a branch of artificial intelligence (AI) that deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages. POS tagging is the process of assigning a part of speech like noun, pronoun, verb, adverb, preposition, adjective or other lexical class marker to each word in a given sentence. Part of Speech tagger is a most basic application of natural language processing.

In linguistic corpus, part-of-speech tagging (POST or POS tagging) also called grammatical tagging or word-category disambiguation. It is the process of marking up a word in a corpus as respective to a particular part of speech, based on its context and definition both i.e., its relationship with related and adjusted words in a phrase, sentence, or paragraph. A simplified form of this tagging is commonly taught to school-age children, in the identification of words as nouns, pronouns, verbs, adjectives etc.

Now a days, POS tagging is done in the context of computational linguistics using some algorithms in accordance with a set of descriptive tags. POS-tagging algorithms fall into three distinctive groups: rule-based, statistical and Hybrid based tagger. Rule-based tagger use linguistic rules to assign the correct tags to the words in the sentence or file. E. Brill's tagger, one of the first and most widely used English POS-taggers, applied rule-based algorithms.

Statistical Part of Speech tagger is based on the probabilities of occurrences of words for a given particular tag through HMM and CRF approach. Hybrid based Part of Speech tagger is combination of Rule based approach and Statistical approach. Part of Speech tagging is an important tool of natural language processing. It is used in several Natural Languages processing based software implementation. Accuracy of all NLP tasks like grammar checker, phrase chunker, machine translation etc. depends upon the accuracy of the Part of Speech tagger. Tagger plays an important role in speech recognition, natural language parsing and information retrieval. In this project, follows the Statistical based approach for POS Tagging, more specifically CRF Model of statistical approach.

1.1 Analysis of Existing System

The present system i.e. the present translators do not yield accurate outputs in all cases. Part-of-speech tagging is harder than just having a list of words and their parts of speech, because some words can represent more than one part of speech at different times, and because some parts of speech are complex or unspoken. This is not rare in natural languages, a large percentage of word-forms are ambiguous.

1.2 Identify the problem

Generally, In Schools, students are taught that there are 9 parts of speech in English language: noun, verb, article, adjective, preposition, pronoun, adverb, conjunction, and interjection. However, there are many more categories and sub-categories. For nouns, the singular, plural and possessive forms can be distinguished. In many languages, words are also marked for their "case" (role as object, subject etc.), grammatical gender (masculine/faminine), and so on; while verbs are marked for tense, aspect, and other things. Linguists define parts of speech to various finer levels, which reflect a chosen "tagging system".

This kind of problem occurs due to:

i. Word Sense Disambiguation

ii. Name Entity Disambiguation

Examples:

a. Source Sentence:

I want to ask you out on a date.

Google Translated:

मैं एक तारीख को बाहर पूछना चाहता हूँ।

Human Translated: मै तुम्हे अपने साथ बाहर ले जाना चाहता हूँ।

b. Source Sentence: Bro, let's hang out.

Google Translated: भाई चलो बाहर लटका|

Human Translated: भाई चल साथ में वक़्त गुजरते है|

c. Source Sentence: Are you feeling down?

Google Translated: आप नीचे महसूस कर रहे हैं?

Human Translated: क्या तुम्हे अच्छा नही लग रहा?

1.3 Possible Solution

One solution to address the ambiguity problem is to choose and work upon a very specific domain to limit the usage of words and their meanings so that the ambiguity problem is limited.

For this purpose, only chosen Agricultural domain to resolve ambiguity choosing agriculture as a domain provides some added advantages enlisted as below.

- Agriculture is one of the most overlooked areas in terms of knowledge.
- Most of the farmers are illiterate or incapable of understanding English language.
- Translating knowledgebase into understandable language from English, can help farmers immensely.

By preparing and using corpus of agriculture related sentences, accuracy of the process can be improved. Accuracy can also be further improved by applying certain methods and algorithms such as Conditional Random Field Model (CRF) and Hidden Markov Model (HMM) for annotation of words. Here CRF++ used for learning and testing of tag annotation.

2. LITERATURE SURVEY

Joshi et al [3] (2013) have proposed a Part of Speech tagger for Hindi on the basis of HMM approach. To implement this tagger, they have used IL POS tagset which are provided by IIIT Hyderabad. On the basis of this tagset, they have annotated the corpus of 15,200 sentences (3,58,288 words) from tourism domain to train their system and they also disambiguated correct word-tag combinations using the contextual information available in the text. They obtained the accuracy of 92.13% on test data. Akilan and Naganathan [4] (2012) developed a rule based POS tagset for classical tamil texts. Tamil tagset is divided into two basic parts: noun and verb form and tags are allocated to words on the basis of plural marker, case marker, postposition, verb, adjective, adverb, particle and numerals. This model is designed on the basis of form agreement method, in which form of noun is ‘type’ pattern and form of verb is ‘token’ pattern. The developed POS tagger is used for training and tagging task. In the training task, corpus is validated with tagset. In tagging task, firstly generated the root word from the corpus and then tagged that root word. Jyoti et al [5] (2013) have developed a POS tagger for ‘Marathi’ language using supervised learning approach. In this approach, they used statistical Hidden Markov model. Basic idea behind HMM model is to calculate the probability of best word sequence of tags and then automatically assign the exact tag to particular Marathi word. To check the accuracy of tagged resultant data, authors have evaluated their whole system on the basis of some evaluation methodology like recall, precision and f-score. Finally they have obtained 93.82% accuracy of the proposed POS tagger. Shrivastava & Bhattacharyya [6] (2008) have developed a POS tagger using simple Hidden Markov Model with naïve stemming approach. For the Stemming they have required list of suffixes of Hindi language to remove the longest suffix matched. Naïve Stemmer is used to increase performance of tagger with 93.12% accuracy. Complete performance of this tagger is better than the simple stochastic tagger. Jyoti et al [7] (2013) developed a Marathi part of speech tagger by using statistical approach. In this statistical approach, authors have used some statistical methods like unigram, bigram, trigram and HMM methods. They also presented a tagset for tagging Marathi text. Along with this, compared the unigram, bigram, trigram and HMM methods to check the correctness of tagger’s output and obtained the accuracy of 77.38%, 90.30%,91.46% and 93.82% respectively. Aniket et al [8] (2007) have presented a POS tagger for morphologically rich language Hindi. To design this tagger, authors used maximum entropy markov model based statistical approach. They have also used tiny dictionary of Hindi and stemmer. Through the developed tagger, they have easily captured the lexical and morphological characteristics of words and also generate tagged output. To evaluate this system, corpus of 15,562 words was tested and obtained the 94.89% best accuracy and 94.38% average accuracy. Fareena et al [9] (2012) presented a Part of Speech tagger for Urdu language on the basis of data driven approach. This approach was most efficient and also called as Brill’s Transformation Based Learning (TBL). In this approach, initially tagged every word by guessing and then search the errors. To resolve these errors, they used supervised approach, by which they can correctly tagged the data at training time. To check the accuracy of this system, they have used 36 tags to tagged the corpus of 1,23,775 tokens and finally they achieved 84% correct results. Toutanova and Manning [10] (2000) have designed a maximum entropy approach based POS tagger. Performance of the automatically built tagger can be further increased by the co-ordination of various features: (i) expansion of knowledge sources which is accessible to the tagger (ii) to give special attention to unseen words (iii) identify the features for disambiguation of verb forms. To develop their own tagger, they acquired a maximum entropy approach which is based on probability distribution. At last, they obtained 96.86% overall accuracy on the pentree bank and 86.91% accuracy on unseen words. Gimpel et al [11] (2011) focused on the social media text of twitter for POS tagging. Initially, they have designed various features for tagging, a tagset and annotated the data. After this, they have developed English POS tagger for twitter data and manually assigned the tags on 1827 tweets. To evaluate this tagger, they performed various experiments and obtained approx 90% accuracy. This tagger was also compared with Stanford tagger and they found that reduced the 25% relative errors.

3. PROPOSED METHODOLOGY

3.1 Corpus Creation

To create an agriculture based corpus, 1000 sentences in English language were chosen based on agriculture. These English sentences were manually translated into Hindi and each of the words of the sentence were tagged according to their respective Parts-of-Speech, this process is called annotation. The aforementioned corpus was then used for further processing.

3.2 Tagset for Annotation

To annotate whole corpus, own tagset has been created with the help of IL POS tagset proposed by Bharti et. al.[2]

Table.1 : Tagset for annotation

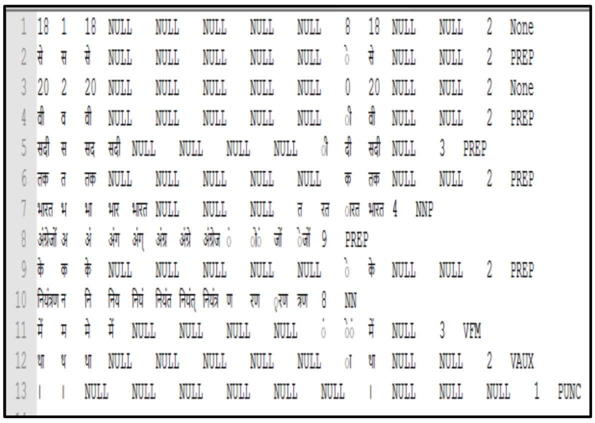
S.N.	Tag	Description (Tag Used for)
1.	NN	Nouns
2.	NNP	Proper Nouns
3.	NNC	Common Nouns
4.	PRP	Pronoun
5.	PREP	Preposition
6.	VM	Verb Main (Finite or Non-Finite)
7.	VAUX	Verb Auxiliary
8.	JJ	Adjective (Modifier of Noun)
9.	RB	Adverb (Modifier of Verb)
10.	RP	Particles
11.	QFNUM	Quantifiers
12.	CC	Conjuncts
13.	QW	Question Words
14.	INTF	Intensifier
15.	NEG	Negative
16.	SYM	Symbol
17.	NONE	Forigen Words

3.3 Create Training File

Here, 13 features applied on the corpus for creating a training file for CRF:

- 7 of which are related to prefix.
- 4 of the subsequent features are related to suffix.
- The next feature is related to the length of the word.

Last feature of the corpus is the name of the tag assigned to the word.



```
18	1	18	NULL	NULL	NULL	NULL	NULL	8	18	NULL	NULL	2	None
से	स	से	NULL	NULL	NULL	NULL	NULL	े	से	NULL	NULL	2	PREP
20	2	20	NULL	NULL	NULL	NULL	NULL	0	20	NULL	NULL	2	None
वी	व	वी	NULL	NULL	NULL	NULL	NULL	ी	वी	NULL	NULL	2	PREP
सदी	स	सद	सदी	NULL	NULL	NULL	NULL	ी	दी	सदी	NULL	3	PREP
तक	त	तक	NULL	NULL	NULL	NULL	NULL	क	तक	NULL	NULL	2	PREP
भारत	भ	भा	भार	भारत	NULL	NULL	NULL	त	रत	ारत	भारत	4	NNP
अंग्रेजों	अ	अं	अंग	अंग्	अंग्र	अंग्रे	अंग्रेज	ं	ों	जों	ेजों	9	PREP
के	क	के	NULL	NULL	NULL	NULL	NULL	े	के	NULL	NULL	2	PREP
नियंत्रण	न	नि	निय	नियं	नियंत	नियंत्	नियंत्र	ण	रण	्रण	त्रण	8	NN
में	म	मे	में	NULL	NULL	NULL	NULL	ं	ें	में	NULL	3	VFM
था	थ	था	NULL	NULL	NULL	NULL	NULL	ा	था	NULL	NULL	2	VAUX
।	।	NULL	NULL	NULL	NULL	NULL	NULL	।	NULL	NULL	NULL	1	PUNC
```

Fig.1 : Corpus after applying features

3.4 Training Performed by CRF++

Conditional random fields (CRFs) are a class of statistical modeling method often applied in machine learning and pattern recognition, where they are used for prediction of structure. Whereas a traditional classifier anticipate a label for a unique sample without regard to "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF popular in natural language processing, which guess a sequences of labels for sequences of input.

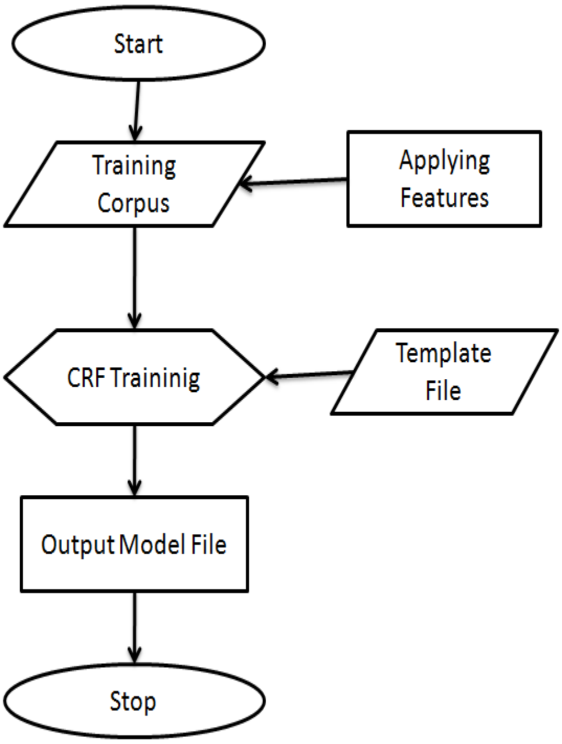


Fig.2 : Training of Tags using CRF++

Use *crf_learn* command:

% crf_learn template_file train_file model_file

where *template_file* and *train_file* are the files you need to prepare in advance. *crf_learn* generates the trained model file in *model_file*.

3.5 Testing Performed by CRF++

Use *crf_test* command:

% crf_test -m model_file test_files ...

where *model_file* is the file *crf_learn*creates. In the testing, you don't need to specify the template file, because the model file has the same information for the template. *test_file* is the test data you want to assign sequential tags. This file has to be written in the same format as training file.

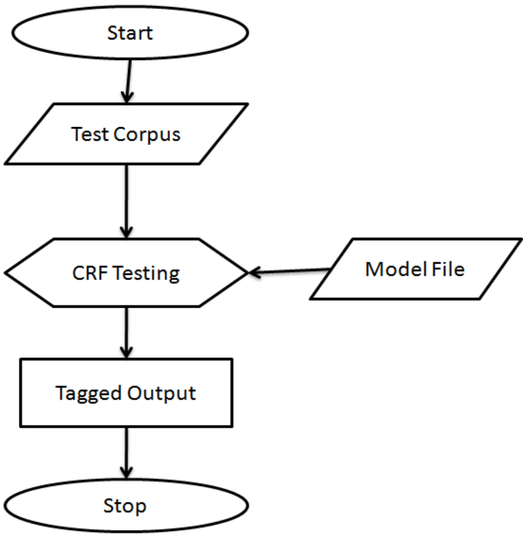
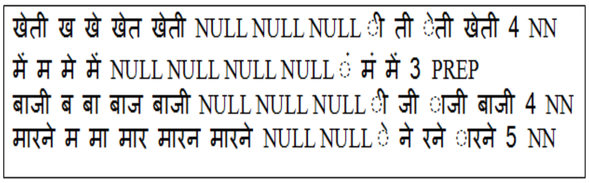


Fig.3 : Testing of Tags using CRF++



```
खेती ख खे खेत खेती NULL NULL NULL ी ती ेती खेती 4 NN
में म मे में NULL NULL NULL NULL ं मं में 3 PREP
बाजी ब बा बाज बाजी NULL NULL NULL ी जी ाजी बाजी 4 NN
मारने म मा मार मारन मारने NULL NULL े ने रने ारने 5 NN
```

Fig.4 : Output given by CRF++

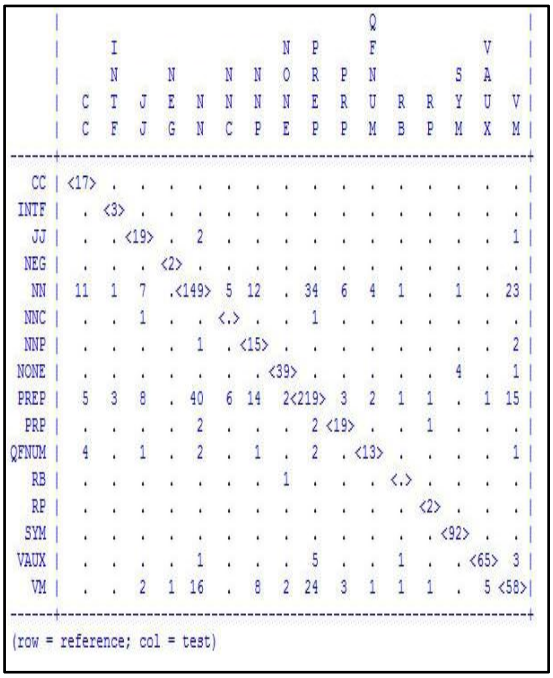
4. EVALUATION SYSTEM

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an evaluation matrix or error matrix. It is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning. Each column of the matrix represents the instances in an anticipated class while each row represents the instances in an actual class or vice-versa. The name stems from the fact that it makes it easy to see if the system is confusing two classes i.e. commonly mislabeling one as another.

It is a distinctive type of contingency table, with two different dimensions ("predicted" and "actual"), and identical sets of "classes" in both dimensions (each consolidation of dimension and class is a variable in the contingency table).

True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), are the four distinctive possible outcomes of a single prediction for a two-class case with classes "1" ("yes") and "0" ("no"). A false positive is when the outcome is incorrectly classified as "yes" (or "positive"), when it is in fact "no" (or "negative"). A false negative is when the outcome is incorrectly classified as negative when it is in fact positive. True positives and true negatives are obviously correct categorizations. Keeping track of all these possible outcomes is such an error-prone activity, that they are usually display in what is called a confusion matrix.

The following diagram is a 16x16 confusion matrix where, the rows specify reference tags (manually annotated tags) and the columns specify tags that are predicted by CRF++.



```
      |                                           Q                     |
      |       I                       N   P       F               V     |
      |       N       N       N   N   O   R   P   N           S   A     |
      |   C   T   J   E   N   N   N   N   E   R   U   R   R   Y   U   V |
      |   C   F   J   G   N   C   P   E   P   P   M   B   P   M   X   M |
------+-----------------------------------------------------------------+
   CC | <17>  .   .   .   .   .   .   .   .   .   .   .   .   .   .   . |
 INTF |   .  <3>  .   .   .   .   .   .   .   .   .   .   .   .   .   . |
   JJ |   .   . <19>  .   2   .   .   .   .   .   .   .   .   .   .   1 |
  NEG |   .   .   .  <2>  .   .   .   .   .   .   .   .   .   .   .   . |
   NN |  11   1   7   .<149>  5  12   .  34   6   4   1   .   1   .  23 |
  NNC |   .   .   1   .   .  <.>  .   .   1   .   .   .   .   .   .   . |
  NNP |   .   .   .   .   1   . <15>  .   .   .   .   .   .   .   .   2 |
 NONE |   .   .   .   .   .   .   . <39>  .   .   .   .   .   4   .   1 |
 PREP |   5   3   8   .  40   6  14   2<219>  3   2   1   1   .   1  15 |
  PRP |   .   .   .   .   2   .   .   .   2 <19>  .   .   1   .   .   . |
QFNUM |   4   .   1   .   2   .   1   .   2   . <13>  .   .   .   .   1 |
   RB |   .   .   .   .   .   .   .   1   .   .   .  <.>  .   .   .   . |
   RP |   .   .   .   .   .   .   .   .   .   .   .   .  <2>  .   .   . |
  SYM |   .   .   .   .   .   .   .   .   .   .   .   .   . <92>  .   . |
 VAUX |   .   .   .   .   1   .   .   .   5   .   .   1   .   . <65>  3 |
   VM |   .   .   2   1  16   .   8   2  24   3   1   1   1   .   5 <58>|
------+-----------------------------------------------------------------+
(row = reference; col = test)
```

Fig.5 : Results of Confusion Matrix

Table.2 : Evaluation of all tags

Tags	Precision	Recall	F-Score	Accuracy
CC	45.94	100	62.96	45.94
INTF	42.85	100	60	42.85
JJ	50	86.36	63.33	46.34
NEG	66.66	100	80	66.66
NN	69.95	58.66	63.81	46.85
NNP	30	83.33	44.11	28.30
NONE	88.63	88.63	88.63	79.59
PREP	76.30	68.43	72.15	56.44
PRP	61.29	79.16	69.09	52.78
QFNUM	65	54.16	59.09	41.93
RP	40	100	57.14	40
SYM	94.84	100	97.35	94.84
VAUX	91.54	86.67	89.04	80.24
VM	55.76	47.54	51.32	34.52
Overall System	62.77	82.35	68.43	54.09

5. CONCLUSION

The primary goal of the ongoing research in NLP has been to remove the challenges that are faced by the current NLP technologies, the biggest of which is Ambiguity. To abate the ambiguity in proposed project, only utilised concerning single domain corpus, i.e. Agriculture.

Here, POS tagger has been created. By which successfully test the 100 sentences using CRF++. Testing is done cause of trained the system using a corpus of 1000 agriculture specific sentences. The accuracy of overall system achieved 54.16% which was obtained by confusion matrix.

Table.3 : Results of overall System

Evaluation Parameter	Results in Percentage
Precision	63%
Recall	82%
F-Score	68%
Accuracy	54%

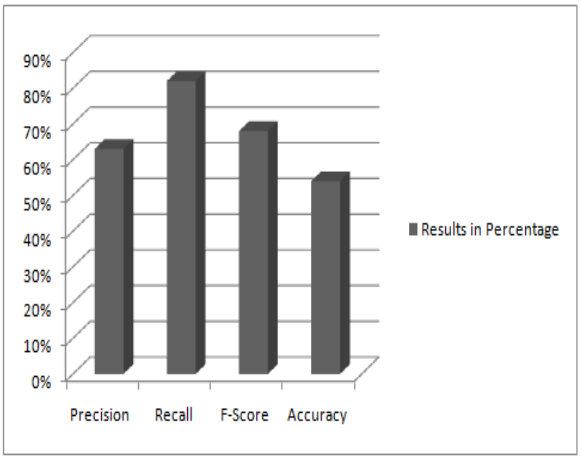


Fig.6 : Final Results of this tagger System

6. REFERENCES

[1] Megyesi, Beáta. "Brill's rule-based PoS tagger".

[2] Akshar Bharati Bharati, A., Chaitanya V., Sangal R., (1995) "Natural Language Processing – A Paninian Perspective". Prentice-Hall India, New Delhi (1995).

[3] Joshi Nisheeth, Hemant Darbari, and Iti Mathur. 2013. HMM based POS tagger for Hindi. Proceeding of 2013 International Conference on Artificial Intelligence, Soft Computing (AISC-2013).

[4] R. Akilan, E.R.Naganathan. 2012. POS Tagging for Classical Tamil Texts. International Journal of business Intelligent. Volume 01. No.01. June 2012.pp 27-30.

[5] Jyoti Singh, Nisheeth Joshi, Iti Mathur. 2013. Marathi Part of Speech Tagger Using Supervised Learning. In proceeding of International Conference on Advanced Computing, Networking and Informatics, India. June 2013. pp 251-257.

[6] Shrivastava, Manish, and Pushpak Bhattacharyya. 2008. Hindi POS tagger using naive stemming: harnessing morphological information without extensive Linguistic knowledge. In International Conference on NLP (ICON08), Pune, India.

[7] Singh, J., Joshi, N., & Mathur, I. 2013. Development of Marathi part of speech tagger using statistical approach. In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on (pp. 1554-1559). IEEE.

[8] Dalal, Aniket, Kumar Nagaraj, U. Swant, Sandeep Shelke, and Pushpak Bhattacharyya. 2007. Building feature rich pos tagger for morphologically rich languages: Experience in Hindi. ICON (2007).

[9] Naz, Fareena, Waqas Anwar, Usama Ijaz Bajwa, and Ehsan Ullah Munir. 2012. Urdu part of speech tagging using transformation based error driven learning. World Applied Sciences Journal 16, no. 3: 437-448.

[10] Toutanova, Kristina, and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13, pp. 63-70. Association for Computational Linguistics.

[11] Gimpel, Kevin, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments." In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, pp. 42-47. Association for Computational Linguistics.

[12] Gupta, Vaishali, Nisheeth Joshi, and Iti Mathur. "POS tagger for Urdu using Stochastic approaches." Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. ACM, 2016.