INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X

(Volume2, Issue3)

Available online at: www.Ijariit.com

STATISTICAL ANALYSIS OF PART OF SPEECH (POS) TAGGING ALGORITHMS FOR ENGLISH CORPUS

Swati Tyagi[1], Gouri Shankar Mishra[2]
CSE Department[1], Software Engineering Department[2]
Sharda University, Greater Noida, UP, India

Abstract— ***Part of speech (POS) Tagging is the procedure of allocating the portion of speech tag or supplementary philological class signal to every single and every single word in a sentence. In countless Usual Speech Processing presentations such as word intellect disambiguation, data recovery, data grasping, analyzing, interrogating, and contraption clarification, POS tagging is imitated as the one of the frank obligatory tool. Categorizing the uncertainties in speech philological items is the mystifying goal in the procedure of growing an effectual and correct POS Tagger. In this paper we difference the presentation of a insufficient POS tagging methods for Bangla speech, e.g. statistical way (n-gram, HMM) and perceptron established approach. A supervised POS tagging way needs a colossal number of annotated training corpus to tag properly. At this early period of POS-tagging for English. In this work we craft a earth truth set that encompasses tagged words from sampled corpus. We additionally investigated the presentation of POS taggers for disparate kinds of words.***

Keyboards: Part-of-speech tagging, HMM, Unigram, Perceptron.

I. PART-OF-SPEECH TAGGING

Part-of-speech tagging mentions to a procedure of allocating part-of-speech labels to words in a corpus. Frank part-of-speech labels contain such word classes as nouns, verbs and adjectives but the labelling procedure usually goes beyond that. Features such as singular/plural forms, grammatical gender and case are additionally considered. A part-of-speech tagger begins the tagging procedure by consulting a machine-readable lexicon to ascertain whether the word it has encountered is present in the lexicon. If it is present next a catalog of part-of-speech tags that corresponds alongside the word is returned. A word could have countless probable labels, but in most cases it just has one. There are additionally periods after the tagger won't be able to find the word in the lexicon. If that's the case the tagger will normally tolerate by dispatching the word to the morphological analyzer. There it will endeavor to decompose the word and ascertain if it encompasses a morpheme ending. If it does the rest of the word will be matched after once more alongside the lexicon. If that match fails too next the final tagging will be reliant on the disambiguation procedure that follows next. The disambiguation procedure is vital both for words that have countless probable parts-of-speech in the lexicon as well as those that haven't been discovered in the lexicon at all. The disambiguation procedure is completed contrarily reliant on that kind of tagging algorithm that you use. There are three dominant kinds of tagging algorithms inside part-of-speech tagging, namely stochastic tagging, rule-based tagging and transformation-based tagging. The last merges features of the two preceding ones.

II. CLASSIFICATION OF POS TAGGER

A Part-Of-Speech Tagger (POS Tagger) is described as a portion of multimedia that assigns portions of speech to every single word of a speech that it reads. The ways of POS tagging can be tear into three categories; law established tagging, statistical tagging and hybrid tagging. A set of hand composed laws are requested alongside alongside it the contextual data is utilized to allocate POS tags to words in the law established POS. The disadvantage of this arrangement is that it doesn't work after the text is not known. The setback being that it cannot forecast the appropriate text. Therefore in order to accomplish higher efficiency and accuracy in this arrangement, exhaustive set of hand coded laws ought to be used. Frequency and probability are encompassed in the statistical approach. The frank statistical way works on the basis of the most oftentimes utilized tag for a specific word in the annotated training data and additionally this data is utilized to tag that word in the unannotated text. But the disadvantage of this arrangement is that a little sequences of tags can come up for sentences that are not correct according to the syntax laws of a precise language. One more way is additionally there that is recognized as the hybrid approach. It could even present larger than statistical or law established approaches. Early of all the probabilistic features of the statistical method are used and next the set of hand coded speech specific laws are requested in the hybrid approach. There are disparate kinds of statistical tagging ways debated in this paper that are- Unigram, Bigram and Trigram. Alongside alongside this the studies completed on the basis of comparisons and evaluation are additionally shown.

POS tagging works on disparate approaches. The disparate models of POS tagging are shown in the pursuing figure.

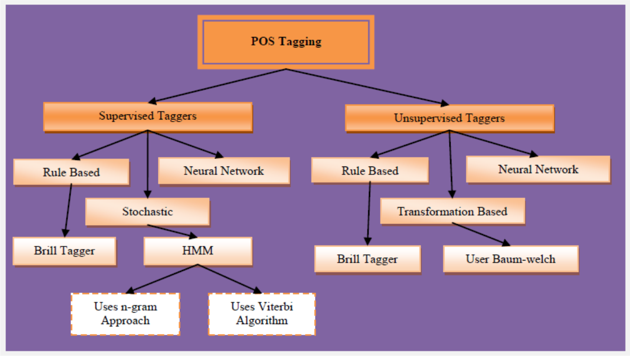


Figure 1POS Classification

1 Supervised POS Tagging

Frequency or probability is the fundamentals utilized the Statistical taggers to tag the text. With the simplest Statistical tagger the setback of ambiguity of words established on the probability that word occurs alongside a particular tag can be resolved. The most public spans in that these tags are oftentimes utilized are the training set and are the one allocated to an unclear instance of that word in the assessing data. Pre-tagged models are needed by the supervised POS tagging models as they are utilized to discover data concerning the tag-set, word-tag frequencies, law sets etc for training. Rise in the size of corpora usually increases the presentation of the models.

This way is termed as the n-gram way that mentions to the fact that the tag that is the best for a given word is ambitious by the probability that occurs alongside the n-1 preceding tags. The drawback of this method is that it can of sequence reclaim a correct tag for a given word but alongside alongside this it can additionally from time to time reclaim invalid sequences of tags. The stochastic ideal is established on assorted models such as Hidden Markov Ideal (HMM), Maximum Likelihood Estimation, Decision Trees, N-grams, Maximum Entropy, Prop Vector Mechanisms and Conditional Random Fields.

Law Instituted Approaches

The oldest part-of-speech tagging arrangement was the one that utilized law established approach. A set of hand composed laws were requested and additionally contextual data was utilized in order to allocate POS tags to words in the law established POS tagging. These laws are usually recognized as context construction rules. Two-stage design was requested in the first algorithms for automatically allocating part-of-speech. Firstly in the early period a lexicon is utilized in order to allocate every single and every single word a catalog of possible portions of speech. Later this in the subsequent period utilized colossal catalogs of hand-written disambiguation laws are utilized alongside the intention to lessen down this catalog to just a solitary part-of-speech for every single word.

Supervised training is needed normally in the law established tagging models that is pre-annotated corpora. The main disadvantages of the law established arrangements are the necessity of a linguistic background and manually constructing the rules.

1.3.1.2 Stochastic

The frequency, probability or statistics are encompassed in the stochastic approach. But the disadvantage of this way can be that from time to time those sequence of tags can come that are not correct as each the syntax laws of a language. An way that is recognized as the n-gram way that computes the probability of a given sequence of tags can be utilized as an alternative to the word frequency approach. The best tag can be ambitious by it for a word by discovering out the probability that it occurs alongside the n preceding tags, whereas the worth of n is set to 1, 2 or 3 for useful purposes. These models are termed as Unigram, Bigram and Trigram [1]. Viterbi algorithm, that is a find algorithm that avoids the polynomial development of a breadth early find by trimming the find tree at every single level employing the best m Maximum Likelihood Estimates (MLE).

2 Unsupervised POS Tagging

The unsupervised POS tagging models is not like supervised models as they do not need pre-tagged corpora. Rather than this, they use elevated computational methods such as the Baum-Welch algorithm so as to automatically instigate tag sets, makeover laws etc.
There are basically two classes in that most of the tagging algorithms fall: rule-based taggers and stochastic taggers. The supervised ways cannot be usefully completed facilely to make them work in applicative settings but they grasp the best presentation in countless NLP tasks]. Not merely this, the supervised arrangements ought to be trained on a colossal number of annotations that are manually provided.

Makeover Instituted Discovering (TBL)

Brill delineated an arrangement that learns a set of correction laws that helps to circumvent linguistic laws that are manual. A set of laws is obtained by instantiating every single law template that has data from the corpus, alongside the aid of predetermined law template. This is completed afterward the initialization process. The words that are tagged incorrectly are requested alongside every single law temporarily and hence the law that reduces the maximum number of errors is recognized and believed to be the best. Nowadays this law is added to the leaned laws and on the new corpus industrialized this procedure iterates by seizing the presently added law, because alongside the aid of staying laws, the reduction of error rate less than a predetermined threshold cannot be possible. Both the makeover established way and the law established way are comparable as they depend on a set of laws for tagging. Initially, the tags to words are allocated established on a stochastic method. For example- for a particular word, the tag that has the higher frequency is assigned. Next to become the final consequence, the set of laws are requested to the primarily tagged data.

III. Applications

We have utilized the tagger in a number of applications. WC delineate three requests here: phrase recognition; word sense disambiguation; and grammatical purpose assignment. These undertakings are portion of a scutiny power to use shallow scutiny methods to remove content from unrestricted text.

1 Phrase Recognition

We have crafted an arrangement that knows easy phrases after given as input the sequence of tags for a sentence. There are recognizers for noun phrases, verb clusters adverbial phrases, and prepositional phrases. Every single of these phrases embodies a contiguous sequence of tags that gratifies an easy grammar. For example, a noun phrase can be a unary sequence encompassing a pronoun tag or an arbitrarily long sequence of noun and adjective tags, perhaps pre. Yielded by a determiner tag and perhaps alongside an embedded possessive marker. The most extended probable sequence is fount (e.g., "the plan committee" but not "the program") Conjunctions are not understood as portion of each phrase; for example, in the fragment "the cats and dogs," "the cats" and "dogs" will be understood as two noun phrases. Prepositional phrase attachment is not gave at this period of processing. This way to phrase credit in a little cases arrests merely portions of a little phrases; though, our way minimizes fake positives, so that we can rely on the recognizers' results.

2 Word Sense Disambiguation

Part-of-speech tagging in and of itself is a functional instrument in lexical disambiguation; for example, knowing that "dig" is being utilized as a noun rather than as a verb indicates the word's appropriate meaning. But countless words have several meanings even as inhabiting the alike portion of speech. To this conclude, the tagger has been utilized in the implementation of an experimental noun homograph disambiguation algorithm [Hearst, 1991]. The algorithm (known as Catch- Word) performs supervised training above a colossal text corpus, meeting lexical, orthographic, and easy syntactic facts for every single sense of the unclear noun. Later a era of training, Catchword classifies new instances of the noun by checking its context opposing that of beforehand noted instances and selecting the sense for that the most facts is found. Because the sense distinctions made are crude, the disambiguation can be accomplished lacking the price of vision centers or inference mechanisms. Early examinations arose in accuracies of concerning 90% for nouns alongside powerfully different senses.
This algorithm uses the tagger in two ways:

(i) To determine the part of speech of the target word (filtering out the non-noun usages) and
(ii) As a step in the phrase recognition analysis of the context surrounding the noun.

3 Grammatical Function Assignment

The phrase recognizers additionally furnish input to an arrangement, Sopa [Sibun] that knows nominal arguments of verbs, specifically, Subject, Object, and Predicative Arguments. Sopa does not rely on data (such as arity or voice) specific to the particular verbs involved. The early pace in allocating grammatical purposes is to partition the tag sequence of every single sentence into phrases. The phrase kinds contain those remarked, supplementary kinds to report for conjunctions, complementizers, and indicators of sentence borders, and an "unknown" type. Later a sentence has been partitioned, every single easy noun phrase is examined in the context of the phrase to its left and the phrase to its right. On the basis of this innate context and a set of laws, the noun phrase is marked as a syntactic Subject, Object, Predicative, or is not marked at all. A label of Predicative is allocated merely if it can be ambitious that the administrating verb cluster is a form of a predicating verb (e.g., a form of "be"). Because this cannot always be ambitious, a little Predicative are labeled Objects. If a noun phrase is labeled, it is additionally annotated as to whether the administrating verb is the closest verb cluster to the right or to the left. The algorithm has an accuracy of concerning 800"/o in allocating grammatical purposes.

IV. POS Taggers

1.) **Unigram Tagger:** A unigram POS tagger plainly assigns to a word, the tag that is most probable for the word, established on a tagged corpus (i.e., training corpus). For instance, it assigns “verb” to the word “fix” if “fix” is extra frequently tagged as a “verb” in the training corpus. To find the most probable tag for every single word, a unigram POS tagger computes and stores the frequency of tags utilized for every single word established on the tagged training corpus. For those words that do not materialize in the training corpus, a unigram tagger assigns “noun” to them as the default tag. The unigram POS tagger is easy and fast, and achieves a satisfactory accuracy if the training corpus is colossal enough. Though, it ignores the encircling (i.e., context) of a word after it assigns a POS tag to it.

2.) **Hidden Markov Model (HMM) Based Tagger:** Similar to the unigram tagger, a hidden Markov model (HMM) based tagger assigns POS tags by searching for the most likely tag for each word in a sentence. However, a HMM based tagger finds a tag sequence for a sentence as a whole, rather than finding a tag for each word separately. Given a sentence $w_1,\ldots,w_n$, a HMM based tagger chooses a tag sequence $t_1,\ldots\ldots,t_n$ that maximizes the following joint probability:

$$P(t_1 \ldots t_n, w_1 \ldots\ldots w_n) = P(t_1 \ldots t_n)P(w_1 \ldots w_n | t_1 \ldots t_n)$$

In practice, it is often impractical to compute P (t1: :: tn). Therefore many different taggers have been proposed to simplify this probability computation. TnT, one of the most commonly used HMM based tagger, uses second order Markov models to simplify the computation; it assumes that the tag of a word is determined by the POS tags of the previous two words [6]. Tree tagger is another popular HMM based tagger, which leverages decision trees to get more reliable estimates of parameters in Markov models

3.) **Maximum Entropy Based Tagger:** The unigram and HMM based taggers are easy to build, however given the nature of their probability models, it is hard to incorporate more complex features into them. The maximum entropy (ME) based tagger is introduced to provide a principled way of incorporating complex features into probability models [16]. Given a sentence $w_1,\ldots\ldots,w_n$, an ME based tagger models the conditional probability of a tag sequence $t_1,\ldots.,t_n$ as:

$$P(w_1 \ldots w_n | t_1 \ldots t_n) \approx \prod_{i=1}^{N} P(t_i | C_i)$$

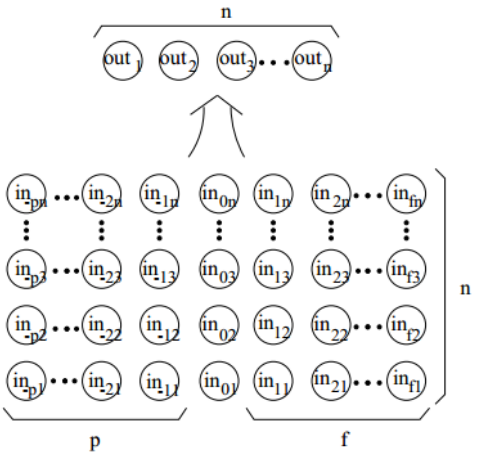
Where $C_1,\ldots\ldots,C_n$ are the corresponding contexts for each word appearing in the sentence. The context C of a word w includes the previous assigned tags before w.

An ME based tagger introduces the concept of features which encode elements of a context C useful for predicting the tag t of a word w. Features are binary valued functions that represent constraints. An ME based tagger will use the features to compute$P(t_i | C_i)$. It will learn the weights of the features that can maximize the entropy of the probability model using the training corpus. Different ME based POS taggers that make use of different features have been proposed. The NLTK natural language toolkit contains a ME based POS tagger that is implemented by Loper and Chichkov (referred to as NLTK tagger in this paper). Stanford POS tagger is another popular ME based tagger that improves the original maximum entropy based tagger by considering two more types of features related to the writing style of letters (e.g., whether the first letter is capitalized or not) .

4.) **Transformation based Tagger:** A transformation based POS tagger assigns POS tags to words based on linguistic knowledge, expressed as rules that are automatically learned from a training corpus. In particular, the transformation based tagger first uses a simple stochastic based tagger to get an initial tag for a word and then go back to fix the error if the word is wrongly tagged. In this way, the rules that could turn a badly tagged text into a good one are automatically learned. TBT tagger is the POS tagger proposed by Brill, who first introduced the idea of transformation based tagger. The TBT tagger uses a unigram tagger to get the initial tags. Annie tagger is another popular transformation based tagger

Perceptron Tagger

The Net-Tagger consists of a MLP-network and a lexicon (see fig. below). Structure of the Net-Tagger without hidden layer; the arrow symbolizes the connections between the layers.



In the output layer of the MLP network, each unit corresponds to one of the tags in the tagset. The network learns during the training to activate that output unit which represents the correct tag and to deactivate all other output units. Hence, in the trained network, the output unit with the highest activation indicates, which tag should be attached to the word that is currently processed. The input of the network comprises all the information which the system has about the parts of speech of the current word, the p preceding words and the *f* following words. More precisely, for each part-of-speech tag posj and each of the p + 1 + *f* words in the context, there is an input unit whose activation in represents the probability that word has part of speech pos_j.

For the word which is being tagged and the following words, the lexical part-of-speech probability P (pos_j|$word_i$) is all we know about the part of speech. This probability does not take into account any contextual influences. So, we get the following in-put representation for the currently tagged word and the following words:

$$in_{ij} = P(pos_j|word_i), if\ i \geq 0$$

For the preceding words, there is more information available, because they have already been tagged. The activation values of the output units at the time of processing are here used instead of the lexical part-of- speech probabilities:

$$in_{ij}(t) = out_j(t+1), if\ i \geq 0$$

Copying output activations of the network into the input units introduces recurrence into the network. This complicates the training process, because the output of the network is not correct, when the training starts and therefore, it cannot be fed back directly, when the training starts. Instead a weighted average of the actual output and the target output is used. It resembles more the output of the trained network which is similar (or at least should be similar) to the target output. At the beginning of the training, the weighting of the target output is high. It falls to zero during the training.

V. RELATED WORK

Michele, Banko et. al. (2004) [1] in this paper, we present a new HMM tagger that exploits context on both sides of a word to be tagged, and evaluate it in both the unsupervised and supervised case. Along the way, we present the first comprehensive comparison of unsupervised methods for part-of-speech tagging, noting that published results to date have not been comparable across corpora or lexicons. Observing that the quality of the lexicon greatly impacts the accuracy that can be achieved by the algorithms, we present a method of HMM training that improves accuracy when training of lexical probabilities is unstable. Finally, we show how this new tagger achieves state-of-the-art results in a supervised, non-training intensive framework.

Fahim Muhammad, Hasan et. al. (2007) [2] In this paper, there are different approaches to the problem of assigning each word of a text with a parts-of-speech tag, which is known as Part-Of-Speech (POS) tagging. In this paper we compare the performance of a few POS tagging techniques for Bangla language, e.g. statistical approach (n-gram, HMM) and transformation based approach (Brill's tagger). A supervised POS tagging approach requires a large amount of annotated training corpus to tag properly. At this initial stage of POS-tagging for Bangla, we have very limited resource of annotated corpus. We tried to see which technique maximizes the performance with this limited resource. We also checked the performance for English and tried to conclude how these techniques might perform if we can manage a substantial amount of annotated corpus.

Helmut, Schmid et. al. (2008) [3] In this paper, we present a HMM part-of-speech tagging method which is particularly suited for POS tagsets with a large number of fine-grained tags. It is based on three ideas:

(1) splitting of the POS tags into attribute vectors and decomposition of the contextual POS probabilities of the HMM into a product of attribute probabilities,
(2) estimation of the contextual probabilities with decision trees, and
(3) Use of high-order HMMs. In experiments on German and Czech data, our tagger outperformed state of- the-art POS taggers.

Benjamin, Snyder et. al. (2008) [4] In this paper, we demonstrate the effectiveness of multilingual learning for unsupervised part-of-speech tagging. The key hypothesis of multilingual learning is that by combining cues from multiple languages, the structure of each becomes more apparent. We formulate a hierarchical Bayesian model for jointly predicting bilingual streams of part-of-speech tags. The model learns language-specific features while capturing cross-lingual patterns in tag distribution for aligned words. Once the parameters of our model have been learned on bilingual parallel data, we evaluate its performance on a held-out monolingual test set. Our evaluation on six pairs of languages shows consistent and significant performance gains over a state-of-the-art monolingual baseline. For one language pair, we observe a relative reduction in error of 53%.

Stefan Block et. al. (2009) [5] In this paper, it presents a comparison of three part-of-speech taggers, µ-TBL, TnT and HunPOS. Their differing tagging approaches are discussed and their performances given identical training- and test data are evaluated. Strengths and weaknesses of the three taggers are also analyzed.

The taggers are all trained and tested on the Penn Treebank, a fully tagged corpus for the English language. The best accuracy results were 95.99% for TnT, 95.97% for HunPOS and 93.1 for µ-TBL. TnT and HunPOS perform better with most part-of-speech classes but there are exceptions.

Marco Brunello et. al. (2009) [6] In this paper, we have presented a method for extracting domain specific terminologies by crawling and processing Web documents. To yield a high-quality terminology directly from raw Web data, we combined different noise-reduction techniques. Near duplicate detection was implemented to prevent obtaining distorted term frequencies. To meet industrialscale requirements, we modified the original algorithm for fast online de-duplication.

By applying a discriminant function based on term statistics of two corpora, we filtered domain relevant terms. We also examined the use of bigram statistics to filter out irrelevant multi-word phrases. We successfully applied the methodology for generating a German health terminology. To extract terminologies for different target domains, only the set of Web sites that are used as seeds for the crawler have to be changed. Also, a different classification model has to be trained. The extra work required for most domains will be minimal compared to the effort of creating domain specific terminologies manually.

Alex, Cheng et. al. (2010) [7] In this paper, we conduct a series of Part-of-Speech (POS) Tagging experiments using Expectation Maximization (EM), Vibrational Bayes (VB) and Gibbs Sampling (GS) against the Chinese Penn Treebank. We want to first establish a baseline for unsupervised POS tagging in Chinese, which will facilitate future research in this area. Secondly, by comparing and analyzing the results between Chinese and English, we highlight some of the strengths and weaknesses of each of the algorithms in POS tagging task and attempt to explain the differences based on some preliminary linguistics analysis. Comparing to English, we find that all algorithms perform rather poorly in Chinese in 1-to-1 accuracy result but are more competitive in many-to-1 accuracy. We attribute one possible explanation of this to the algorithms' inability to correctly produce tags that match the desired tag count distribution.

YoongKeok, Lee et. al. (2010) [8] In this paper, part-of-speech (POS) tag distributions are known to exhibit sparsity — a word is likely to take a single predominant tag in a corpus. Recent research has demonstrated that incorporating this sparsity constraint improves tagging accuracy. However, in existing systems, this expansion come with a steep increase in model complexity. This paper proposes a simple and effective tagging method that directly models tag sparsity and other distributional properties of valid POS tag assignments. In addition, this formulation results in a dramatic reduction in the number of model parameters thereby, enabling unusually rapid training. Our experiments consistently demonstrate that this model architecture yields substantial performance gains over more complex tagging counterparts. On several languages, we report performance exceeding that of more complex state-of-the art systems.

Shen, Li, Joao V. Graça et. al. (2012) [9] In this paper, despite significant recent work, purely unsupervised techniques for part-of-speech (POS) tagging have not achieved useful accuracies required by many language processing tasks. Use of parallel text between resource-rich and resource-poor languages is one source of weak supervision that significantly improves accuracy. However, parallel text is not always available and techniques for using it require multiple complex algorithmic steps. In this paper we show that we can build POS-taggers exceeding state-of-the-art bilingual methods by using simple hidden Markov models and a freely available and naturally growing resource, the Wiktionary. Across eight languages for which we have labeled data to evaluate results, we achieve accuracy that significantly exceeds best unsupervised and parallel text methods. We achieve highest accuracy reported for several languages and show that our approach yields better out-of-domain taggers than those trained using fully supervised Penn Treebank.

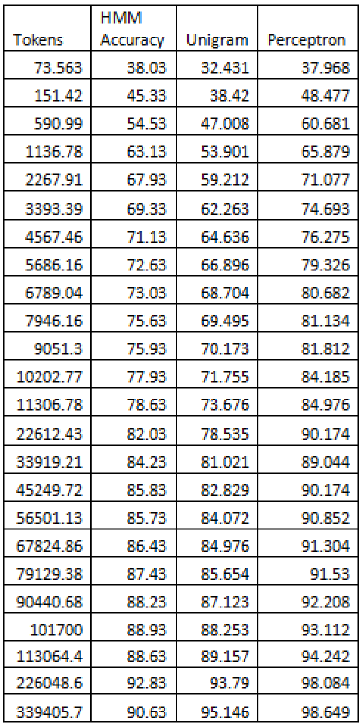
Deepika, Kumawat et. al. (2015) [10] In this paper, Part of speech (POS) cataloguing is the process of allocating the part of speech tag or other philological class sign to each and every word in a sentence. In many Natural Language Processing presentations such as word intellect disambiguation, information recovery, information handling, analyzing, interrogating, and machine interpretation, POS tagging is reflected as the one of the basic obligatory tool. Categorizing the uncertainties in language philological items is the puzzling objective in the procedure of emerging an effectual and correct POS Tagger. Works survey displays that, for Indian lingoes, POS taggers were established only in Hindi, Punjabi, Bengali and Dravidian languages. Some POS taggers were also established generic to the Hindi, Telugu and Bengali tongues. All scheduled POS taggers were grounded on diverse Tag-set, established by diverse organization and individuals.

This paper speaks the various developments in POS-taggers and POS-tag-set for Indian language, which is very essential computational verbal tool needed for many natural language processing (NLP) presentation.

Yuan, Tian, et. al. (2015) [11] In this paper, many software artifacts are written in natural language or contain substantial amount of natural language contents. Thus these artifacts could be analyzed using text analysis techniques from the natural language processing (NLP) community, e.g., the part-of-speech (POS) tagging technique that assigns POS tags (e.g., verb, noun, etc.) to words in a sentence. In the literature, several studies have already applied POS tagging technique on software artifacts to recover important words in them, which are then used for automating various tasks, e.g., locating buggy files for a given bug report, etc. There are many POS tagging techniques proposed and they are trained and evaluated on non software engineering corpus (documents). Thus it is unknown whether they can correctly identify the POS of a word in a software artifact and which of them performs the best. To fill this gap, in this work, we investigate the effectiveness of seven POS taggers on corpus. We randomly sample 100 corpus from Eclipse and Mozilla project and create a text corpus that contains 21,713 words. We manually assign POS tags to these words and use them to evaluate the studied POS taggers. Our comparative study shows that the state-of-the-art POS taggers achieve an accuracy of 83.6%-90.5% on corpus and the Stanford POS tagger and the Tree Tagger achieve the highest accuracy on the sampled corpus. Our findings show that researchers could use these POS taggers to analyze software artifacts, if an accuracy of 80-90% is acceptable for their specific needs, and we recommend using the Stanford POS tagger or the Tree Tagger.

VI. PERFORMANCE OF POS TAGGERS ON CORPUS

We use accuracy (i.e., the ratio of the number of accurately tagged words to the finished number of tagged words) to compute the presentation of a POS tagger on a text corpus. Table below displays the accuracy of every single POS tagger on our crafted earth truth set and their accuracy on a usual English corpus, i.e., the BROWN corpus. Note that the accuracy on the usual English corpus is computed by training a POS tagger on a portion of the BROWN corpus and requesting it on a disparate portion of the BROWN corpus. we discern that the learned POS taggers might accomplish an accuracy of 76% to 95% on the sampled corpus. The Stanford POS tagger performs the best pursued by the Tree Tagger, as the unigram POS tagger performs the worst. The taggers presentation on the corpus are all inferior to their presentation on the usual English corpus. Even though the unigram tagger performs the worst, the difference amid its presentation on corpus and that on the usual English corpus is small. For the supplementary POS taggers, the accuracy of the HMM-Based and Unigram taggers cut the most after they are requested on corpus.



Tokens	HMM Accuracy	Unigram	Perceptron
73.563	38.03	32.431	37.968
151.42	45.33	38.42	48.477
590.99	54.53	47.008	60.681
1136.78	63.13	53.901	65.879
2267.91	67.93	59.212	71.077
3393.39	69.33	62.263	74.693
4567.46	71.13	64.636	76.275
5686.16	72.63	66.896	79.326
6789.04	73.03	68.704	80.682
7946.16	75.63	69.495	81.134
9051.3	75.93	70.173	81.812
10202.77	77.93	71.755	84.185
11306.78	78.63	73.676	84.976
22612.43	82.03	78.535	90.174
33919.21	84.23	81.021	89.044
45249.72	85.83	82.829	90.174
56501.13	85.73	84.072	90.852
67824.86	86.43	84.976	91.304
79129.38	87.43	85.654	91.53
90440.68	88.23	87.123	92.208
101700	88.93	88.253	93.112
113064.4	88.63	89.157	94.242
226048.6	92.83	93.79	98.084
339405.7	90.63	95.146	98.649

Table 1Performance of POS Taggers for English

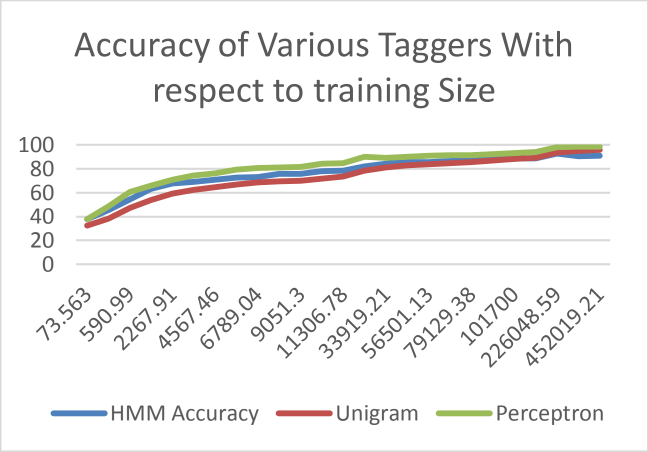
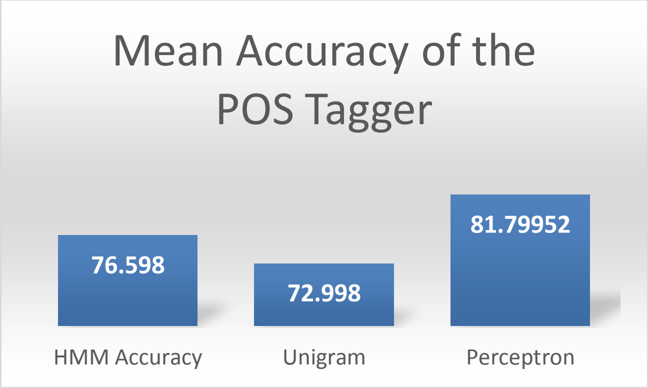


Fig 3



VII. CONCLUSION AND FUTURE WORK

Previous textual scutiny work on corpus and program identifiers display the possible manipulation of part-of-speech tagging in upholding multimedia engineering tasks, such as plan comprehension and bug assignment. In this work, we difference the effectiveness of three state-of-the-art POS taggers on corpus. We craft a earth truth set that encompasses tagged words from sampled corpus. Our preliminary examination aftermath display that the state of- the-art POS taggers might accomplish a reasonable accuracy on corpus (83.6%-90.5%), even though inferior than its accuracy on a usual English corpus (97%, for most taggers). We additionally examine the presentation of POS taggers for disparate kinds of words. Generally, the Stanford and Tree Tagger present the best. In the upcoming, it should be interesting to rise the number of tagged corpus and train POS taggers employing tagged corpus, rather than employing usual English corpus. It should additionally be interesting to present a qualitative discover on the cases in that disparate kinds of taggers produce wrong tags. This could aid us craft an enhanced POS tagging method that performs larger than off-the-shelf POS taggers after apply on corpus.

VIII. REFERENCES

[1] Michele, Banko, and Robert C. Moore. "Part of speech tagging in context." In Proceedings of the 20th international conference on Computational Linguistics, p. 556. Association for Computational Linguistics, 2004.

[2] Fahim Muhammad, Hasan, NaushadUzZaman, and Mumit Khan. "Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill’s tagger) for Bangla." In Advances and Innovations in Systems, Computing Sciences and Software Engineering, pp. 121-126. Springer Netherlands, 2007.

[3] Helmut, Schmid, and Florian Laws. "Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging." In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pp. 777-784. Association for Computational Linguistics, 2008.

[4] Benjamin, Snyder, TahiraNaseem, Jacob Eisenstein, and Regina Barzilay. "Unsupervised multilingual learning for POS tagging." In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1041-1050. Association for Computational Linguistics, 2008.

[5] Stefan Block, "A Comparison of three part-of-speech taggers." PhD diss., Master’s thesis, Uppsala Universitet, 2009.

[6] Marco Brunello. "The creation of free linguistic corpora from the web." In Web as Corpus Workshop (WAC5), p. 9. 2009.

[7] Alex, Cheng, Fei Xia, and Jianfeng Gao. "A comparison of unsupervised methods for Part-of-Speech Tagging in Chinese." In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 135-143. Association for Computational Linguistics, 2010.

[8] YoongKeok, Lee, Aria Haghighi, and Regina Barzilay. "Simple type-level unsupervised POS tagging." In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 853-861. Association for Computational Linguistics, 2010.

[9] Shen, Li, Joao V. Graça, and Ben Taskar. "Wiki-ly supervised part-of-speech tagging." In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1389-1398. Association for Computational Linguistics, 2012.

[10] Deepika, Kumawat, and Vinesh Jain. "POS Tagging Approaches: A Comparison." International Journal of Computer Applications 118, no. 6 (2015).

[11] Yuan, Tian, and David Lo. "A comparative study on the effectiveness of part-of-speech tagging techniques on corpus." In Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on, pp. 570-574. IEEE, 2015.