International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
1st International Conference on Advent Trends in Engineering, Science and Technology
"ICATEST 2015", 08 March 2015

Survey: Part-Of-Speech Tagging in NLP

Nidhi Adhvaryu[1], Prem Balani[2]
[1]*ME Student, Information Technology Department, GCET, GTU affiliated, V.V. Nagar, Gujarat, India, nidhi.adhvaryu12@gmail.com*
[2]*Assistant Professor, Information Technology Department, GCET, GTU affiliated, V.V. Nagar, Gujarat, India, prembalani@gcet.ac.in*

Abstract— Part of Speech tagging (POST) is used to assign different tags to each word of the sentence. POST having major two methods: Supervised tagging and unsupervised tagging. These methods are further divided into two categories: Rule based, Statistical method and Transformation based technique. In rule based method, rules are generated manually and according to all rules sentence will be tagged. In statistical method, three types of techniques are used: HMM, MEMM and CRF. These are corpus based techniques. Transformation based method is used to learn rules and fed features and tag the unannotated corpus.

Keywords: Natural Language Processing, corpus, Part Of Speech tagging, HMM, CRF.

.

1. INTRODUCTION

Natural language processing (NLP) [12] is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation [12]. The solution for language understanding is Part Of Speech tagging. The basic system is human-computer interaction, which allows user to interact with computer using their everyday languages. NLP is used for information retrieval, information extraction and speech processing.

POS Tagging is the task of assigning appropriate POS tag to each word in a sentence, e.g. verb, adverb, noun, pronoun etc. Most words occurring in text has got ambiguity associated with it in terms of their part of speech [5]. For example "power" can be treated as a noun or a verb. So we have to recognize the phrase and other patterns. POS Tagging is use for information extraction, information retrieval and speech processing.

Process of Part Of Speech Tagging: Read the input sentence. Then tokenize the sentence into words. After tokenization, Suffix analysis and prefix analysis is also used for correctly tag each word of sentence. Then use one of the tagging methods to tag each word of sentence of corpus as noun, verb, conjunction, number tag etc. The output is tagged sentence. Then after evaluate the accuracy of output.

Part Of Speech tagging has got much significance in field of computational linguistics which uses algorithms to associate discrete terms, as well as its hidden parts of speech with respect to a set of descriptive tags [5]. Part of speech tagging is used to introduce the relationship of one word with its previous word as well as its next word.

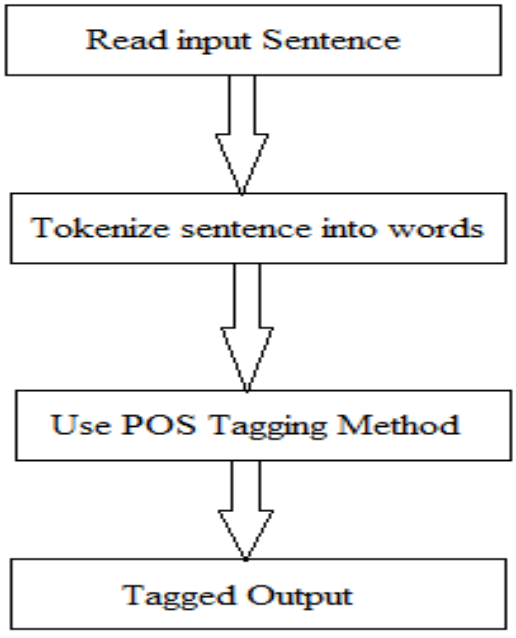

Fig. 1: Process of POS Tagging

2. METHODS OF POS TAGGING

Mainly there are two methods of tagging: supervised tagging and unsupervised tagging. In supervised tagging, already trained (pre-trained) corpus is used and according to that corpus output will be generated. It is easy to tag sentence with supervised tagging. In unsupervised tagging, first task is to train whole corpus and then tag the input sentence. This method is complex compared to supervise tagging.

These methods are also sub-categorized into three types: Rule based Technique, statistical technique and transformation based technique. In rule-based technique hand written rues are used. Statistical technique measures the probabilities according to occurrence of word for particular tag [7]. Statistical technique having three types: HMM, MEMM and CRF. Transformation based method is used to learn rules and fed features and tag the unannotated corpus

3. RULE BASED TECHNIQUE

Rule based approach uses a large database of hand-written disambiguation rules considering the morpheme ordering and contextual information [9]. Rule-based tagger use linguistic rules to assign the correct tags to the words in the sentence or file, e.g. verb identification rule, noun identification rule, pronoun identification rule, adjective identification rule [7]. In this approach the rules are generated manually. This method is very complex and also time consuming.

4. STATISTICAL TECHNIQUE

Statistical Part of Speech Technique is based on the probabilities of occurrences of words for a particular tag [7]. Mostly it is used as corpus based technique. This method is based on probability of word occurrence. So it is possible that two words are same but according to the context of sentence, the tag of both words will be different. It is possible that if first word is noun then second word will also be noun, verb or adjective. So it is depends on the probability distribution.

Mainly there are three sub-techniques: Hidden Marcov Model, Maximum Entropy Marcov Model and Conditional Random Fields.

A. Hidden Marcov Model

In the part of speech tagging, the emergence each word in the sentence and its part of speech are seen as a random process, the emergence of word looks as a observable sequence, and part of speech as an implicit process, the process marked by the word (can be observed a) is to observe the part of speech sequence (implied) [1]. HMM assigns the probable tag sequence to each sentence [11].

At the training phase of HMM based POS tagging, observation probability matrix and tag transition probability matrix are generated [2].

The observation probability, $p(w|t)$ of a word is computed using the following equation [2]:

$$p(w \mid t) = (c(w \mid t))/c(t) \quad (1)$$

Here, c(w|t) is correctly tagged words and c(t) is number of tags.

HMM taggers make Markov assumption which states that the probability of a tag is dependent only on a small, fixed number of previous tags [2]. This is for bigram HMM tagger. In Trigram HMM tagger, the current word depends on previous tag as well as its successive tag. So HMM focuses on three words at a time.

HMM is generative model, assigning a joint probability to paired observation and label sequences; the parameters are typically trained to maximize the joint likelihood of training examples [3]. Here, the probability is based on the previous tag as well as the next word of the sentence.

To define a joint probability over observation and label sequences, a generative model needs to enumerate all possible observation sequences, typically requiring a representation in which observations are task-appropriate words of sentence [3].

B. Maximum Entropy Marcov Model

Maximum entropy Markov models (MEMM) [3] are conditional probabilistic sequence models. In MEMMs, each source state1 has an exponential model that takes the observation features as input, and outputs a distribution over possible next states. These exponential models are trained by an appropriate iterative scaling method in the maximum entropy framework.

MEMMs based on next-state classifiers, such as discriminative Markov models, share a weakness

that is the label bias problem [3]: the transitions leaving a given state compete only against each other, rather than against all other transitions in the model. In probabilistic terms, transition scores are the conditional probabilities of possible next states given the current state and the observation sequence. This causes a bias toward states with fewer outgoing transitions. In the extreme case, a state with a single outgoing transition effectively ignores the observation.

MEMMs, as well as non-probabilistic sequence tagging and segmentation models with independently trained next-state classifiers are all potential victims of the label bias problem [3].

Following figure [3] shows the example of label bias problem.

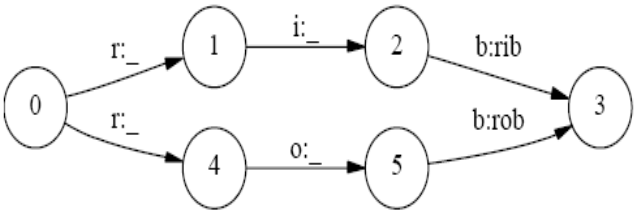

Fig. 2: Label-bias Problem [3]

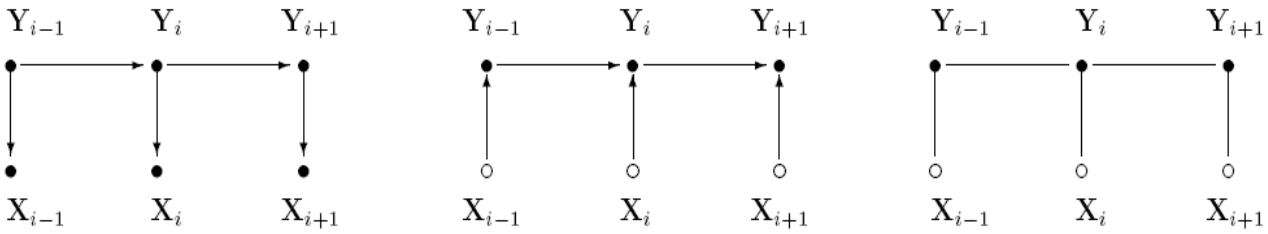

Fig. 3: Graphical structures of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences. An open circle indicates that the variable is not generated by the model [3].

This example [3] represents a simple finite-state model designed to distinguish between the two words rib and rob. Suppose that the observation sequence is r i b. In the first time step, r matches both transitions from the start state, so the probability mass gets distributed roughly equally among those two transitions. Next we observe i. Both states 1 and 4 have only one outgoing transition. State 1 has seen this observation often in training, state 4 has almost never seen this observation; but like state 1, state 4 has no choice but to pass all its mass to its single outgoing transition. States with low-entropy next state distributions will take little notice of observations.

If one of the two words is slightly more common in the training set, the transitions out of the start state will slightly prefer its corresponding transition, and that word's state sequence will always win.

C. Conditional Random Field

Conditional Random Fields [3], a sequence modeling framework that has all the advantages of MEMMs but also solves the label bias problem in a principled way. The critical difference between CRFs and MEMMs is that a MEMM uses per-state exponential models for the conditional probabilities of next states given the current state, while a CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence.

Give a sentence, which consist of word sequence W_1, W2…W_L, assume its corresponding POS sequence is Y_1, Y2…Y_L, Because of the universal ambiguity of POS, it is impossible to have exactly only one corresponding POS sequence, So the task of CRF-based POS tagging is to find a POS sequence Y_1, Y2…Y_L to make the probability P (Y_1, Y2…Y_L | W_1, W2…W_L) maximal [4].

Conditional Random Fields [3] is developed in order to calculate the conditional probabilities of values on other designated input nodes of undirected graphical models and CRF encodes a conditional probability distribution with a given set

of feature. In this approach the system learns by giving some training and can be use for testing In CRF Model [6], the input of CRF based POS tagging is a text file. The training and test files consist of multiple tokens or words with similar features in form of multiple (but fixed number) columns.

Training of CRF [6] system is done in order to get an output as a model file. In the training of the CRF a template file is used whose function is to choose the features from the feature list. Model file is the learnt file by the CRF system for use in the testing process.

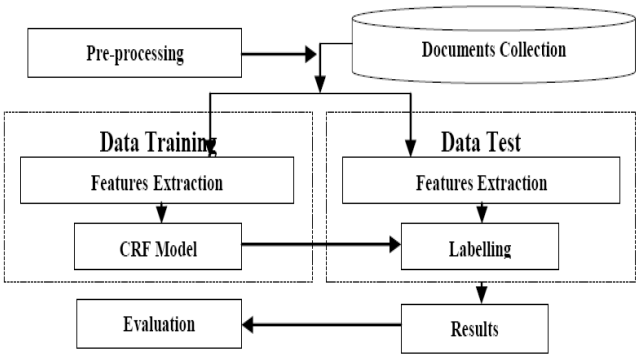

Fig. 4: CRF Model [6]

The testing [6] proceeds by using the model file in the CRF system. The gold standard test file is used for testing. This file is also created in the same format as that of training file, i.e., of fixed number of columns with the same field as that of training file. After testing process the output file will be a new file with an extra column which is tagged with the POS tags.

Some features used by CRF are [10]: Surrounding words as feature, Surrounding Stem words as feature, number of acceptable standard suffixes as feature, number of acceptable standard prefixes as feature, acceptable suffixes present as feature, etc.

In fig. 3, X_i is the current word, X_{i-1} is the previous word and X_{i+1} is the next word. Y_i is the current tagged word, Y_{i-1} is the previous tagged word and Y_{i+1} is the next tagged word.

Fig. 3, shows the chain structure of HMM, MEMM and CRF model. In HMM method one word is depends on the previously tagged word. In MEMM, one word is depends on the previously tagged word as well as next word. CRF model other data.

calculates the joint probability of the sentence according to the feature set.CRF model is overcome the label bias problem, which is occurred in HMM and MEMM. So, CRF method is better than HMM and MEMM.

5. TRANSFORMATION BASED TECHNIQUE

The basic idea of TBL [8] is starting with a temporary solution, then Step-by-step apply transformation rule to improve the tagging problem. The algorithm stops when no more optimal transformation can be applied to make it better. Transformations are extracted from an annotated training corpus. Each transformation includes two parts: a rewrite rule, and a trigging environment. For example we have a transformation with rewrite rule "change tag from verb to noun" and trigging environment "The preceding word is a determiner". It means that the transformation only can be applied to a word next to a determiner. And when we apply the transformation to that word, its tag is changed from verb to noun.

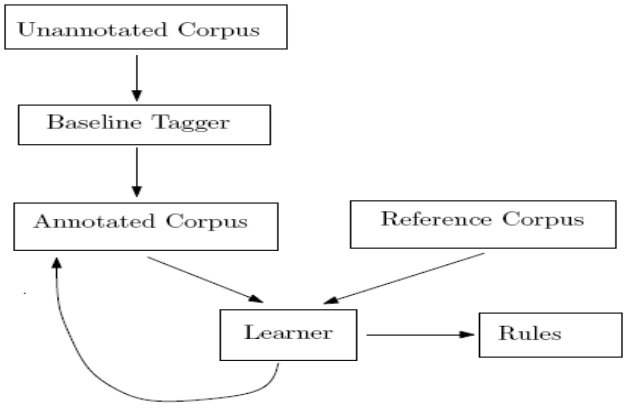

Fig. 5: Transformation Based Learning [8]

The rule learning algorithm includes five steps [8]:
1. Generate all rules that correct at least one error
2. For each rule:
   a. Apply to a copy of the most recent state of the training set
   b. Score the result using the objective function
3. Select the rule with the best score
4. Update the training set by applying the selected rule
5. Stop if the score is smaller than some pre-set threshold T; otherwise repeat from step 1.

Applying TBL to POS tagging can be described briefly in three stages as following (these three stages are repeated until the stopping condition is reached) [8]:

1. Label every word with its most-likely tag, unknown word with most frequent tag
2. Examine every possible transformation and select the one with the most improved tagging
3. Retag the data according to this rule.

In TBL [8], at each round only the best rule is extracted. Then the selected rule is applied to the corpus before starting a new round. So the set of learned rules is an ordered list, the rule standing after in the list can support the one before. In some cases, in order to become truth a sample must be applied step-by-step a sequence of ordered rules. However, this iterative learning process makes the algorithm becomes slow. Furthermore, the forms of rules are restricted by rule templates, and rule templates must be prepared by hand.

A Model for Learning Transformation Rules [8]:

The Model describes to learn transformation rules. Figure 6 illustrates the rule learning process. First, training corpus (unannotated corpus) is passed through a baseline tagger. After that, the current corpus (output of the baseline tagger), reference corpus (annotated corpus), and a set of features are fed into a rule learner. The main part of the rule learner is a feature selection module. The task of this module is selecting the best subset of features in order to generate transformation rules. Instead of using rule templates, we generate transformation rules from a set of features using wrong patterns.

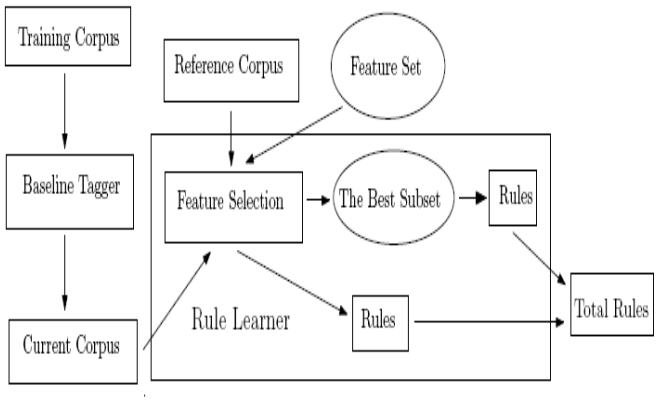

Figure 6. Transformation Rule Learning [8]

A simple solution is that with each subset, firstly we generate all transformation rules, and then apply the rule set to the current corpus and re-calculate the accuracy of the whole system. If we do like that, we will not only lose a lot of time but also cannot compare the quality of two rules both in the same subset and in two different subsets. The later reason is more important [8].

6. CONCLUSION

POS Tagging is used to tag each word of sentence. Mainly two methods are Supervised and unsupervised tagging method. Rule based method is complex because the rules are generated manually. HMM and MEMM having the label bias problem which is solved by CRF method. So CRF method is widely used now a day. Transformation based learning is use the feature selection method to improve the rules of tagging.

REFERENCES

[1] Zhang Youzhi," Research and Implementation of Part-of-Speech Tagging based on Hidden Markov Model", Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications,IEEE-2009 pp. 26-29
[2] Kamal Sarkar and Vivekananda Gayen," A Trigram HMM-Based POS Tagger for Indian Languages", Proceeding of International Conference on Frontiers of Intelligent Computing, 2013, pp. 205–212.
[3] John Lafferty, Andrew McCallum, Fernando C.N. Pereira," Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", 18th International Conference on Machine Learning 2001 (ICML 2001), pp. 282-289.
[4] ZHANG Xiaofei; HUANG Heyan, Zhang Liang, "The Application of CRFs in Part-of-Speech Tagging", International Conference on Intelligent Human-Machine Systems and Cybernetics,IEEE-2009, pp. 347-350
[5] Nidhi Mishra, Amit Mishra, "Part of Speech Tagging for Hindi Corpus", International Conference on Communication Systems and Network Technologies,IEEE-2011, pp. 554-558
[6] Kishorjit Nongmeikapama, Sivaji Bandyopadhyayb, " A Transliteration of CRF Based Manipuri POS Tagging", 2nd International Conference on Communication, Computing & Security (ICCCS),2012, pp. 582-589
[7] Navneet Garg, Vishal Goy, Suman Preet," Rule Based Hindi Part of Speech Tagger",

Proceedings of COLING 2012: Demonstration Papers, pp. 163–174

[8] Ngo Xuan Bach, Le Anh Cuong, Nguyen Viet Ha, Nguyen Ngoc Binh," Transformation Rule Learning without Rule Templates: A Case Study in Part of Speech Tagging", International Conference on Advanced Language Processing and Web Information Technology,IEEE-2008, pp. 9-14

[9] Dinesh Kumar, Gurpreet Singh Josan," Part of Speech Taggers for Morphologically Rich Indian Languages: A Survey", International Journal of Computer Applications Volume 6– No.5, September 2010, pp. 1-9

[10] Kishorjit, N., Bishworjit, S., Romina, M., Mayekleima Chanu, Ng, Sivaji Bandyopadhyay."A Light Weight Manipuri Stemmer", In the Proceedings of National Conference on Indian Language Computing (NCILC), Chochin, India; 2011

[11] Jurafsky, D., Martin, J.H.,"Speech and Language Processing An Introduction to Natural Language Processing", Computational Linguistics and Speech Recognition Pearson Education Series, 2002

[12] Natural Language Processing: http://en.wikipedia.org/wiki/Natural_language_processing