**PART-OF-SPEECH TAGGING**

**FOR**

**UNDER-RESOURCED LANGUAGES**

*B. Tech Seminar Report*

*by*

**M. Sri Krishna Shanmukh**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, WARANGAL

2012-2016

**TABLE OF CONTENTS**

[1 INTRODUCTION 4](#_Toc447548194)

[1.1 What is POS Tagging? 4](#_Toc447548195)

[2 DIFFERENT APPROACHES TO BUILDING POS TAGGER 6](#_Toc447548196)

[2.1 Rule Based/Grammar Based Approach 6](#_Toc447548197)

[2.2 Maximum Entropy Model (MEM) 6](#_Toc447548198)

[2.3 Brill Tagging 6](#_Toc447548199)

[2.4 Hidden Markov Model (HMM) 6](#_Toc447548200)

[2.5 Memory Based Learning 7](#_Toc447548201)

[2.6 Genetic Tagging 7](#_Toc447548202)

[2.7 Hybrid 7](#_Toc447548203)

[3 OUR APPROACH 8](#_Toc447548204)

[3.1 Mapping POS Tagging To Association Rule Mining 8](#_Toc447548205)

[3.2 Context Based Association Rule 8](#_Toc447548206)

[3.3 POS Tagging Challenges 10](#_Toc447548207)

[4 Building Classifier Model 12](#_Toc447548208)

[4.1 Finding Association Rule for a Context Based List 12](#_Toc447548209)

[4.2 B. Handling Class Imbalance Problem 12](#_Toc447548210)

[4.3 Handling Phrase Boundary Problem 13](#_Toc447548211)

[4.4 D. POS Tag-Wise Grouping of Association Rules 13](#_Toc447548212)

[5 POS Tagging of Test Data 14](#_Toc447548213)

[6 Performance Analysis 16](#_Toc447548214)

[7 Conclusion 17](#_Toc447548215)

[8 References 18](#_Toc447548216)

**Abstract**

This report presents a data mining approach for part-of-speech (POS) tagging, an important Natural language processing (NLP) classification task. A semi-supervised associative classification method for POS tagging is proposed. Existing methods for building POS taggers require extensive domain and linguistic knowledge and resources. Our method uses a combination of a small POS tagged corpus and untagged text data as training data to build the classifier model using association rules. Our tagger works well with very little training data also. The use of semi-supervised learning provides the advantage of not requiring a large high quality tagged corpus. These properties make it especially suitable for resource poor languages. The experiments on various resource-rich, resource-moderate and resource-poor languages show good performance without using any language specific linguistic information. It is to be noted that inclusion of such features in our method may further improve the performance. Results also show that for smaller training data sizes our tagger performs better than state-of-the-art CRF tagger using same features as our tagger.

# INTRODUCTION

## What is POS Tagging?

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

Once performed by hand, POS tagging is now done in context of computational linguistics, using algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags. POS-tagging algorithms fall into two distinctive groups: rule-based and stochastic.

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 (as opposed to many artificial languages), a large percentage of word-forms are ambiguous. For example, even "dogs", which is usually thought of as just a plural noun, can also be a verb:

**The sailor dogs the hatch.**

Correct grammatical tagging will reflect that "dogs" is here used as a verb, not as the more common plural noun. Grammatical context is one way to determine this; semantic analysis can also be used to infer that "sailor" and "hatch" implicate "dogs" as 1) in the nautical context and 2) an action applied to the object "hatch" (in this context, "dogs" is a nautical term meaning "fastens (a watertight door) securely").

POS tagging (henceforth referred to as just tagging) is an important NLP classification task that takes a word or a sentence as input, assigns a POS tag or other lexical class marker to a word or to each word in the sentence, and produces the tagged text as output. For this task several rule based, stochastic supervised, and unsupervised methods are available for a number of languages. All of these (including the state-of-the-art taggers) require training data and linguistic resources like dictionaries in large quantities.

These taggers do not perform well for *resource poor languages*, which do not have much resources and training data. So, there is a need to develop generic semi-supervised tagging methods which take advantage of untagged corpus and require less or no lexical resources. In order to perform well, these techniques require a large untagged corpus. Unfortunately, for many resource poor languages, even obtaining this is hard.

# DIFFERENT APPROACHES TO BUILDING POS TAGGER

## Rule Based/Grammar Based Approach

This approach mainly consists of two parts namely: a **dictionary** and **rules**. The Dictionary contains words and their possible part of speech tags while the rules help to disambiguate where more than one part of speech is assigned to a word. The rules may include things to do with morphology etc. and usually they are of two forms: **lexical** rules and **contextual** rules. Though the approach provides high accuracy, it requires a lot of man power to create the rule and one need to be aware of the linguistic feature of the language.

## Maximum Entropy Model (MEM)

Make use of several observations from the inputs but one is taken at a time, then extraction of useful features from the single observation of a word is done, Finally based on the extracted features you classify the word to the tag set with the highest probability.

## Brill Tagging

Introduced by Brill in 1994 and also called transformation based learning, It uses rules which are generated from data (corpus) by machine learning techniques automatically. The rules are learned by the following stages

* Words in corpus are assigned most suitable tag sets.
* Select the maximum tagging based on all possible transformation.
* Using established rule, the data is re-tagged again.
* Then you repeat the last two until no more improvement.

## Hidden Markov Model (HMM)

Probability is the key engine of this method, usually uses Bayesian inference model. Therefore, the part of speech tagging is treated as a classification problem. The tokenized words are given as a sequence of observations to the classifier, then prediction of the class of the tag set. The prediction of the class uses the product of prior probability of the tag set and maximum likelihood of the tag set.

## Memory Based Learning

This is a concept similar to case based reasoning, where training is done to a classifier and the results of the tag set are stored in the memory. Then by use of machine learning algorithm similarity is done between the new inputs and what is stored in the memory and based on the similarity metrics, classification to the right tag set can be done. This is a data based approach.

## Genetic Tagging

This is an evolutionary model of approach which searches optimal solution of the tag set by use of heuristic means. Makes use of three operators: fitness operator, mutation operator and cross operator to make the classification.

## Hybrid

The model tends to mix rule based approach and data based approach. Rule based approaches achieves high precision while data based approaches achieves high coverage. The aim is to try to strike a balance between precision and coverage.

# OUR APPROACH

## Mapping POS Tagging To Association Rule Mining

The sense of a word in a document is effectively determined by its context. A context can occur in multiple places in the text. We refer to this list of occurrences of a context as its ***context based list***. We use this idea for building our tagger. In our method, we mine context based association rules from training data containing both tagged and untagged text. Our method works as follows:

* We collect all possible words occurring in the same context from the untagged data into a list called *context based list* (formally defined later). In this way we are able to find groups of words of *similar categories* from the untagged data.
* Using the annotated set, we find association rules for the context based lists. Each rule maps a context based list to a suitable POS tag. These association rules work as the context based classification rules.
* Lastly, we group these context based association rules according to their POS tags to form clusters. This set of clusters is used as the classifier model to tag words using the method described later.

By experimenting with two varieties of bi-gram (one with preceding word and the other with succeeding word) and trigram as possible contexts it is found that trigram works best for our method. For a word instance Wi , we fix its context as a trigram containing Wi in the middle and we use this context to find the *context based list*.

## Context Based Association Rule

For each *context based list L*, our approach finds association rule that maps the context based list L to a POS tag T with *support* and *confidence* parameters defined below. Since each list is obtained from a unique context word pair, so each association rule uniquely associates a context to a POS tag and works as the context based tagging rule.

In the following definitions and formulae we develop the intuition and the method to compute the interestingness measures of the significant association rules. The complexity in defining support is due to the presence of untagged training data required for semi-supervised learning. The support is the count of occurrences of the context in the dataset.

Context based lists are made from untagged data D and we are interested in those words of these lists for which we know the tag in annotated set AS. Hence, we define support of a context as follows

### AllTagContextSupport

Number of unique words of a context based list L whose tags are available (in annotated set *AS)* is denoted as AllTagContextSupport(L). This measure gives the number of tagged words of L.

### ContextSupport

For a list of words L in which duplicates may be present, *ContextSupport(L)* is defined as the set of unique words present in L.

### Coverage

For a *context based list* L, this measure represents the confidence that enough number of tagged samples are present in L.

### ContextTagSupport

Number of unique words of a context *based list* L present in annotated set AS with a particular tag T is denoted as *ContextTagSupport*(L,T).

### Confidence

For a *context based list* L and tag T, this measure represents the confidence that considerable number of words in list L have a particular tag T and leads to rules.

### WordTagSupport

Frequency of tag T for a word W in the annotated set AS is denoted as *WordTagSupport(*T,W).

### WordTagScore

For a word W and tag T, this represents how good the tag fits the word on a scale of 0 to 1.

### ListTagScore

For a tag T in *context based list* L, where, **AS** is the annotated set. This formula represents the average frequency of tag T in *context based list* L. Intuitively, it represents how good the tag fits the list. Unfortunately, this is not always indicative of the correct tag for the list. For example, if a tag is overall very frequent, it can bias this score. Therefore, we compare this with the following score.

### BackgroundTagScore

For a tag T in annotated set AS, this represents the average frequency of tag T in annotated set AS.

## POS Tagging Challenges

POS tagging, especially for resource poor languages, involves three major challenges listed below. In our approach we handle each of them explicitly.

### Data Sparsity Problem

Some POS tag classes are present in the annotated set with very few representations. This is not enough to derive statistical information about them. In our approach, the use of untagged data reduces this problem.

### Class Imbalance Problem

POS tag classes are highly imbalanced in their occurrence frequency. While selecting a tag this may lead to biasing towards the most frequent tags. Existing solutions of class imbalance problem typically favor rare classes. However, while tagging the *context based lists*, we need to find POS tags for them in such a way that we neither favor frequent tags nor rare tags. We tackle this problem using a novel *Minmax* approach to find the best preferred POS tag instead of the most frequent one.

### Phrase Boundary Problem

Some lists are formed at phrase boundaries where the context comes from two different phrases. We need to filter out these *context based lists* which do not contain words of similar categories. In this case, the context of a word instance need not represent strong context and so the context based list may contain unrelated words. We use suitable parameters to handle this problem.

# Building Classifier Model

## Finding Association Rule for a Context Based List

The first step in our classifier model building method is to compute *context based lists* from an untagged training corpus D. It may be noted that a context based list can store multiple instances of a word. We use a sliding window of size three to collect the context based lists from D, in a single iteration, taking care of sentence boundaries.

In the next step we find association rules for all the context based lists. *BackgroundTagScore* of all the POS tags present in the annotated set *AS* are computed first. Then for a context based list satisfying the threshold values of *Coverage* and *Confidence*, function we find the best preferred tag from the set of tags with maximum *ContextTagSupport*.

For a context based list L present as antecedent in association rule, tag T returned by this algorithm becomes the consequent. This algorithm outputs best preferred tags for all the context based lists and hence finds association rules for all of them.

## Handling Class Imbalance Problem

We handle the *class imbalance problem* by using a novel *Minmax* approach in the function *FindBestPrefTag* and parameters BackgroundTagScore and *ListTagScore*. In *Minmax* approach the preferred tag Ti for *context based list L*, is the one which has maximum *ContextTagSupport*(L,Ti) but minimum *WordTagSupport*(Ti,W) among those words of list L which have tag Ti as the best tag in *AS*. This takes care that the selected tag is supported by majority of the words in the list and is not biased by annotated set's most frequent tag.

To find the best preferred tag in function *FindBestPrefTag*, from the set of all the tags with maximum *ContextTagSupport*, at first we found those tags which were best tags (having maximum *WordTagSupport* value) for the words of list L in *AS*. Next, from this set of preferred tags we find the tag with minimum *WordTagSupport* value. Then criteria *ListTagScore*(L,Ti) >= BackgroundTagScore(Ti) ensures that the selected tag has above average support in the annotated set and the context based list, both. If none of the tags satisfy this criterion, then we tag the list as “NOTVALIST”.

## Handling Phrase Boundary Problem

To filter out context based lists with the *phrase boundary problem* we use two suitable threshold values for parameters *Confidence* and *Coverage. Coverage* takes care of the fact that a context based list has considerable number of words to map it to a tag and *Confidence* ensures that the tag found for the list is the one which is supported by majority of the words in the list.

If context based list L has *Coverage* and *Confidence* values less than the corresponding threshold values *MinCoverage* and *MinConfidence*, we tag L as “NOTVALIST”. If L satisfies both of the threshold values then only we find the set of all the tags which have maximum *ContextTagSupport* (L,Ti) value and use this set to find the best preferred tag for the list.

## POS Tag-Wise Grouping of Association Rules

In the last step, we group context based lists according to their POS tags to get clusters of context based lists as classifier model. We exclude context based lists with tag “NOTVALIST” from the grouping process. Then we process these clusters to store word frequencies, corresponding context word pairs and their frequencies in each cluster. We represent the set of clusters as *Clustset*.

Since we are highly confident about the tags of the words present in the annotated set *AS* so, to improve cluster quality we remove those words from each cluster which do not have a matching cluster tag in *AS*. Finally, we get a set of clusters in which each cluster has words, their counts and associated context word pairs with their counts. Each cluster has a unique POS tag. These clusters are overlapping in nature and words can belong to multiple clusters.

# POS Tagging of Test Data

To tag the words of a test sentence we make use of the test word's context word pair, preceding word and the word frequency in a cluster to decide the tag of the word. When a test word is found in only one cluster then we output the cluster tag as the tag of the test word. But when a test word is found in many clusters, then to select the suitable clusters following priority order is followed:

1. **Criteria 1:** Highest priority is given to the presence of matching context word pair of the test word in the clusters.

2. **Criteria 2:** Second highest priority is given to the presence of matching preceding word of the test word as first word of the context word pairs in clusters.

3. **Criteria 3:** Last priority is given to the frequency of the test word in the clusters.

For test words not present in any cluster we use criterion 1 and 2 to select appropriate clusters. Based on the priority order, only one of the criteria is used to select the suitable clusters. If we are not able to find any suitable cluster then we return “NOTAG” as the tag of the test word.

Even when we find suitable clusters, to increase precision, our method finds POS tags only for those cases where it is confident. It avoids to wrongly classify non confident cases and returns “NOTAG” for them. This is especially useful when the cost of misclassifying (false positive) is high. This also gives opportunity to integrate other language/domain specific POS taggers as they can be used for the non-confident cases.

After selecting the suitable clusters we need to make sure that we have enough confidence in the highest probability tag obtained from the clusters. To ensure this we use the parameter *TagProbDif*, which gives the fractional difference between the highest and the second highest cluster tag probabilities and is defined as follows:

Where, Cmax is the cluster with highest TagProb(Ci) value and Csecmax is the cluster with second highest TagProb(Ci) value. *TagProb(*Ci) of a cluster is defined as follows:

Where, X is set as follows: If the test word is present in cluster Ci then X= test word. For test word not present in any cluster, if the clusters are selected based on the presence of the context word pair of the test word then X= context word pair. If the clusters are selected based on the presence of the preceding word of the test word as first word of the context word pairs in clusters then X= preceding word of the test word. In this way we are able to tag some *unseen/unknown* words also which are not present in the training data. This, in a way, acts as an alternative of smoothing technique for them.

After selecting the clusters (based on priority order) we compute their *TagProb* values using (7) and then compute *TagProbDif* using (6). For *TagProbDif* value above a suitable threshold value *Minprobdif* we output the tag of cluster with highest *TagProb*value as the tag of the test word, otherwise we return “NOTAG’.

# Performance Analysis

Average accuracy values for all languages for various annotated training set sizes(< 25000 words)

In above Table, we compare our results with a supervised CRF tagger. This tagger uses words, their POS tag and context word pair information from annotated data, while our tagger uses words and their context word pair information from untagged data and POS tag information from annotated data. We observe that for annotated data size ≤ *25K* words, our tagger gives better *AverageAccuracy* than CRF tagger. Our tagger also gives better POS tag precisions and better tagging accuracies than CRF tagger for unknown words and performance improves by increasing the untagged data size up to a certain size. This shows that our tagger can be a better choice for resource poor languages. Also, as an additional benefit the model made by our tagger is more human understandable than that made by CRF tagger.

# Conclusion

In this work we developed a semi-supervised associative classification method for POS tagging. We used the concept of context based list and context based association rule mining. We also developed a method to find interestingness measures required to find the association rules in a semi-supervised manner from a training set of tagged and untagged data combined. We showed that our tagger gives good performance for resource rich as well as resource poor languages without using extensive linguistic knowledge. It works well even with less tagged and untagged training data. It can also tag unknown words. These advantages make it very suitable for resource poor languages and can be used as an initial POS tagger while developing linguistic resources for them.

# References

* IEEE paper: A semi supervised association classification method for POS tagging. Issue Date: Oct 30,2014-Nov 1,2014, Written by: Rani P, Pudi V, Sharma D.M
* International Journal on Natural Language Computing (IJNLC) Vol. 4, No.2,April 2015, Benson N Kituku, Musumba George and Peter Wagacha
* A Hybrid Approach to the Development of Part-of-Speech Tagger for Kafi-noonoo Text, Zelalem Mekuria, Yaregal Assabie, 2014
* Computational Linguistics and Intelligent Text Processing, 15th International Conference, CICLing 2014, Kathmandu, Nepal