Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation Based Learning

Avinesh.PVS, Karthik G
Dept. of Computer Science
IIIT - Hyderabad
{avinesh, karthikg}@students.iiit.ac.in

Abstract

In this paper we describe Part Of Speech (POS) tagging and Chunking using Conditional Random Fields (CRFs) and Transformation Based Learning (TBL) for Telugu, Hindi and Bengali. We show here how to train CRFs to achieve good performance over any other ML techniques. Improved training methods based on the morphological information, contextual and the lexical rules (developed using TBL) were critical in achieving good results. The CRF and TBL based POS tagger has an accuracy of about 77.37%, 78.66%, and 76.08% for Telugu, Hindi and Bengali, and the chunker performs at 79.15%, 80.97% and 82.74% for Telugu, Hindi and Bengali respectively.

1. Introduction:

POS-tagging is the process of assigning the part of speech tags to the natural language text based on both its definition and its context. Identifying the POS-tags in a given text is an important aspect of any Natural Language Application.

POS tagging has been developed using the statistical implementations, linguistic rules and sometimes both. Some of the statistical models are the Hidden Markov Models (HMMs) (Cutting 1992), Maximum Entropy Models (MEMMs) (Adwait Raatnaparakhi [2] 1999), CRFs (Fei Sha and Fernando Pereira [1] 2002) and TBL (Eric Brill [7] 1992). These taggers don't work well when small amount of tagged data is used to estimate the parameters of the tagger. So we need to add some extra information like morphological roots and the possible tags for the words in the corpus to improve the performance of the tagger.

Chunking or shallow parsing is the task of identifying and segmenting the text into syntactically correlated word groups. It is considered as an intermediate step towards full parsing.

This paper presents the use of CRFs with the help of morphological information and the transformation rules in POS tagging and Chunking of Indian languages. In Indian languages, the availability of the tagged corpus is very less and so most of the techniques suffer due to data sparseness problem. For the current task the training and the test data is provided by SPSAL workshop at IJCAI 2007.

2. Similar Works

Most of the previous work used two main machine-learning approaches for sequence labeling. The first approach lies on k-order generative probabilistic models of paired input sequences, for instance HMM (Frieda and McCallum [1] 2000) or multilevel Markov Models (Bikel et al. 1999).The second approach views the sequence labeling problem as a sequence of a classification problem, one for each of the labels in the sequence.

CRFs bring together the best of generative and classification models. Like classification models, they can accommodate many statistically correlated features of the inputs, and they are trained discriminatively. But like generative models they can trade off decisions at different sequence positions to obtain a globally optimal labeling. Lafferty [5] (2001) showed that CRFs beat related classification models as well as HMMs on synthetic data and on POS-tagging task.

Among the text chunking techniques Fei Sha and Fernando Pereira [1] (2000) proposed a Conditional Random Field based approach; Lance A. Ramshaw (1995) proposed a Transformation-Based Learning approach. There are also other approaches based on Maximum entropy (Rob Koeling), memory- based etc.

3.1 Conditional Random Fields

Conditional random field is a probabilistic framework for labeling and segmenting data. It is a form of undirected graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence. CRFs define conditional probability distributions $P(\mathbf{Y}|\mathbf{X})$ of label sequences given input sequences. Lafferty et al. defines the probability of a particular label sequence Y given observation sequence X to be a normalized product of potential functions each of the form

$$\exp(\sum \lambda_j t_j (Y_{i-1}, Y_i, X, i) + \sum \mu_k s_k (Y_i, X, i))$$

where $t_j(Y_{i-1}, Y_i, X, i)$ is a transition feature function of the entire observation sequence and the labels at positions *i* and *i-1* in the label sequence; $s_k (Y_i, X, i)$ is a state feature function of the label at position I and the observation sequence; and λ_j and μ_k are parameters to be estimated from training data.

$$F_j(Y,X)= \sum f_j (Y_{i-1}, Y_i, X, i)$$

where each $f_j (Y_{i-1}, Y_i, X, i)$ is either a state function $s(Y_{i-1}, Y_i, X, i)$ or a transition function $t(Y_{i-1}, Y_i, X, i)$. This allows the probability of a label sequence **Y** given an observation sequence **X** to be written as

$$P(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\lambda}) = (1/Z(\mathbf{X})) \exp(\sum \lambda_j F_j(\boldsymbol{Y}, \boldsymbol{X}))$$

$Z(\mathbf{X})$ is a normalization factor.

3.2 Transformation Based Learning

Transformation-based learning starts with a supervised training corpus that specifies the correct values for some linguistic feature of interest, a baseline heuristics for predicting the values for that feature, and a set of rule templates that determine a space of possible features in the neighborhood surrounding a word, and their action is to change the system's current guess as to the feature for the word. To learn a model, one first applies the baseline heuristic to produce initial hypotheses of the training corpus. Where this baseline prediction is not correct, the templates are then used to form the instantiated candidate rules. This process eventually identifies all the rules candidates generated by that template set that would have a positive effect on the current tag assignments anywhere in the corpus. Those candidate rules are then tested against the rest of corpus, to identify the negative changes. This entire process is then repeated on the transformed corpus deriving candidate rules, scoring them, and selecting one with the maximal positive effect.

This way the lexical and the contextual rules are generated from the training corpus.

4. Approach:

4.1 POS Tagging

The approach we used for POS tagging is as follows: CRF model (CRF++) is used to perform the initial tagging and then a set of transformation rules is applied to correct the errors produced by CRFs.

Initially we used basic features in CRFs, later added the morphological information like the root word, all possible pos tags for the words in the corpus, the suffix and prefix information. This information is added to the training corpus and then it is trained using these features.

To measure the performance of CRFs against other ML approaches we carried out various experiments using Brant's TnT [3] for HMMs, Maxent for MEMMs. Interestingly HMMs performed as high as CRFs with basic features. We preferred CRFs over HMMs as addition of features like the root words were much easier in CRFs.

4.2.Chunking

For chunking first we tried out HMM's to mark the chunk labels. Later the system is trained on the feature templates for predicting the chunk boundary names using CRFs. Finally the chunk labels and the chunk boundary names are merged to obtain the chunk tag. It is basically HMM+ CRF model for chunking.

5. Experiments

5.1 POS Tagging

Initially we trained the CRF on baseline template i.e. over local words of the current word with a window size of 2, 4, and 6; and tried all possible combinations of features. It was observed that CRFs gave better results with a window size of 4 and the combinations of previous words and the current word. Using the basic template the accuracy was 73.47%.

As Telugu is an agglutinative language i.e. the words are joined together to form new words and the postfixes are often attached to the word, so we used the suffix information for each word. The last two letters, last three, and last four letters of the word are added as suffixes in the training corpus.

Later we added the root information and the probable tags of the word from the morphological analyzer. The combination of the word and its root marked an increment of 1% in the performance of the system.

The size of the word also played a major role in assigning the POS tags. The threshold considered was 3 i.e. words whose length is less than three belonged to one class and the rest to the other. This marked an improvement of 1% in the performance. This is because the average length of non-functional words in Indian languages is around 3.

Transformation rules produced by TBL are then used to change the incorrect tags produced by the CRFs. Interestingly it gave an increase of 0.6% for Hindi where as for Telugu initially the accuracy decreased. This is due to the agglutinative nature of Telugu. Due to this the rules had a negative effect some times.

These errors are further reduced by the deleting few of the transformation rules which induced negative effect. This gave an improvement of 1% for all the three languages.

The same model is used for Telugu, Hindi and Bengali except for few differences in the window size i.e. for Hindi, Bengali and Telugu we used a window size of 6, 6 and 4 respectively.

Language	Training Data	Test Data	Results (%)
Telugu	21425	5193	77.37
Bengali	20397	5225	76.08
Hindi	21470	4924	78.66

Fig.1 POS Tagging Results and Data size

5.2 Chunking

Initially we tried out HMM's to mark the chunk boundary and then some rules to identify the boundary names which gave a precision of 76.59% (Telugu test data).

Then we used CRF model with basic features such as words, POS tags and the combination of the both; which improved the performance of the system to 79.15 % (Telugu test data).The Fig.2 shows the result of chunking (Telugu test data) using the POS tags provided with the test set.

Basically the same HMM+CRF model is used for Telugu, Hindi and Bengali chunking. And the features and the combination of features used in the CRF model are also the same.

Chunk Labels	Precision (%)	Recall (%)	$F_{\beta=1}$
B-CCP	79.15	67.21	72.97
B-JJP	50.00	10.00	16.67
B-NP	78.17	90.27	83.79
B-RBP	44.83	27.08	33.77
B-VG	76.50	79.76	78.09
I-CCP	42.86	37.50	40.00
I-JJP	100.00	16.67	28.57
I-NP	82.45	71.19	76.41
I-RBP	38.46	27.78	32.26
I-VG	83.93	80.13	81.98
Overall	79.15	79.15	79.15

Fig2. Chunking (Telugu) with reference POS tags

Language	Results (%)
Telugu	79.15
Bengali	82.74
Hindi	80.97

Fig 3. Chunking Results

6. Error Analysis

Actual tag	Assigned tag	Counts
NN	NNP	218
NN	JJ	208
NN	RB	85
PREP	NLOC	82
NN	PREP	61
VRB	VFM	58
JJ	NN	50
NN	QFNUM	46
VFM	NEG	24
PRP	NN	10

Fig 4.Error Analysis for POS-tagging

7. Conclusion

The overall results obtained for POS tagging is 77.37% and for chunking it is 79.17% (Telugu). The accuracy of the Telugu POS Tagging seemed to be low compared to other Indian Languages due to agglutinative nature of the language.

One more interesting thing to observe is that in some of the cases the sandhi is splited and in some other cases it is not splited.

Eg:

1: *pAxaprohAlace* (NN) = *pAxaprahArAliiu* (NN) + *ce* (PREP)

2: *vAllumtAru*(V) = *vAlylyu*(NN) + *uMtAru*(V)

We demonstrated the use of CRFs and TBL for POS tagging for Indian Languages which gave us good results. This could be future looked into to improve the performance of the tagger.

8. Acknowledgments

We would like to express our gratitude to Dr. Dipti Mishra and Prof. Sangal for their guidance and support. We would also like to thank Prashanth and Himanshu for their valuable suggestions.

References

[1] Fei Sha and Fernando Pereira 2003, Shallow Parsing with Conditional Random Fields.*In the Proceedings of HLT-NAACL.*

[2] Adwait Ratnaparkhi , 1998, Maximum Entropy Model For Natural Language Ambiguity Resolution, Dissertation in Computer and Information Science, University Of Pennslyvania, 1998.

[3] Thorsten Brants, 2000. TnT – a Statistical Part-of-Speech Tagger. *Proceeding of the sixth conference on Applied Natural Language Processing (2000) pg 224-231*.

[4] Charles Sutton, An Introduction to Conditional Random Fields for Relational Learning

[5] John Lafferty, Andrew McCallum and Fernando Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.

[6] Himanshu Agarwal and Anirudh Mani, Part of Speech Tagging and Chunking with Conditional Random Fields**.** *In the Proceedings of NWAI workshop 2006.*

[7] Eric Brill. 1995 Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging**.** *Computational Linguistics.*

[8] CRF++: Yet Another Toolkit

http://chasen.org/~taku/software/CRF++