CHUNKING IN MANIPURI USING CRF

Kishorjit Nongmeikapam[1] ,Chiranjiv Chingangbam[1], Nepoleon Keisham[1], Biakchungnunga Varte[1], Sivaji Bandopadhyay[2]

[1]Department of Computer Science & Engineering, Manipur Institute of Technology, Manipur University, Imphal, India
[2]Department of Computer Science & Engineering, Jadavpur University, West Bengal, India

ABSTRACT

This paper deals about the chunking of the Manipuri language, which is very highly agglutinative in Nature. The system works in such a way that the Manipuri text is clean upto the gold standard. The text is processed for Part of Speech (POS) tagging using Conditional Random Field (CRF). The output file is treated as an input file for the CRF based Chunking system. The final output is a completely chunk tag Manipuri text. The system shows a recall of 71.30%, a precision of 77.36% and a F-measure of 74.21%.

KEYWORDS

CRF; POS; Chunk; Manipuri

1. INTRODUCTION

The Manipuri Language has its origin in the north-eastern parts of India, widely spoken in the state Manipur, and some in the countries of Myanmar and Bangladesh. The Manipuri Language belongs to a high agglutinative class of language. The Conditional Random Fields (CRFs) serve as a powerful model for predicting structured labeling.

Chunking is the process of identifying and labeling the simple phrases (it may be a Noun Phrase or a Verb Phrase) from the tagged output, of which the utterance of words for a given phrase forms as a chunk for this language. A POS tagged sequence output might also form as a base input for the CRF-based chunking.

We synthesized a full scale Manipuri chunked file as the output. The procedure that we follow is that the input file is passed onto a CRF based POS tagger, and then this output from the tagger serve as the input for the CRF based Chunking, which duly generates the output chunked file.

The paper is arranged in such a way that the related works is listed in Section II. Section III describes the concept of Conditional Random Field (CRF) which is followed by the System design at IV. The experiment and evaluation is discussed at Section V and the conclusion is drawn at Section VI.

2. RELATED WORKS

Until now, no works in the area of CRF based chunking has ever been performed on the Manipuri language. Most of the previous works for other languages on this area make use of two machine-learning approaches for sequence labeling, namely HMM in [1] and the second approach as the

sequence labeling problem as a sequence of a classification problem, one for each of the labels in the sequence.

Apart from the above two approaches, the CRF based chunking utilizes and gives the best of the generative and classification models. It resembles the classical model, in a way that they can accommodate many statistically correlated features of the inputs. And consecutively, it resembles the generative model, they have the ability to trade-off decisions at different sequence positions, and consequently it obtains a globally optimal labeling. It is shown in [2] that CRFs are better than related classification models. Parsing by chunks is discussed in [3]. Dynamic programming for parsing and estimation of stochastic unication-based grammars is mentioned in [4] and other related works are found in [5]-[7].

And on the field of text chunking, [1] proposed a Conditional Random Field based approach. The works on chunking can be observe applying both rule based and the probabilistic or statistical methods.

3. CONCEPT OF CONDITION RANDOM FIELD

The concept of Conditional Random Field [8] is developed in order to calculate the conditional probabilities of values on other designated input nodes of undirected graphical models. CRF encodes a conditional probability distribution with a given set of features. It is an unsupervised approach where the system learns by giving some training and can be used for testing other texts.

The conditional probability of a state sequence $X=(x_1, x_2,..x_T)$ given an observation sequence $Y=(y_1, y_2,..y_T)$ is calculated as :

$$P(Y|X) = \frac{1}{Z_X}\exp\left(\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(y_{t-1}, y_t, X, t)\right) \quad ---(1)$$

where, $f_k(y_{t-1}, y_t, X, t)$ is a feature function whose weight λ_k is a learnt weight associated with f_k and to be learned via training. The values of the feature functions may range between $-\infty \ldots +\infty$, but typically they are binary. Z_X is the normalization factor:

$$Z_X = \sum_{y}\exp\sum_{t=1}^{T}\sum_{k}\lambda_k f_k(y_{t-1}, y_t, X, t) \quad ---(2)$$

which is calculated in order to make the probability of all state sequences sum to 1. This is calculated as in Hidden Markov Model (HMM) and can be obtained efficiently by dynamic programming. Since CRF defines the conditional probability P(Y|X), the appropriate objective for parameter learning is to maximize the conditional likelihood of the state sequence or training data.

$$\sum_{i=1}^{N}\log P(y^i|x^i) \quad ---(3)$$

where, $\{(x^i, y^i)\}$ is the labeled training data.
Gaussian prior on the λ's is used to regularize the training (i.e., smoothing). If $\lambda \sim N(0,\rho^2)$, the objective function becomes,

$$\sum_{i=1}^{N}\log P(y^i|x^i) - \sum_{k}\frac{\lambda_k^2}{2\rho^2} \quad ---(4)$$

The objective function is concave, so the λ*'s* have a unique set of optimal values.

4. SYSTEM DESIGN

The system works with the application of CRF in two layers. The first layer is meant for the POS tagging of the Manipuri text file using certain features as mention in [9]. In the second layer the output file of the CRF based POS tagging is used as an input file of the CRF based chunking. Fig.1 explains the System block diagram.

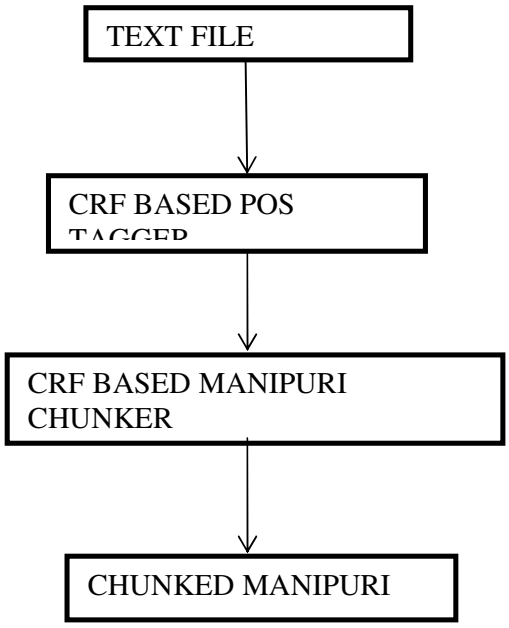
The chunking tag is the I-O-B tagging. That is as follows:

TABLE I. IOB TAGGING

B-X	Beginning of the chunk word X
I-X	Intermediate or non beginning chunk word X
O	Word outside of the chunk text

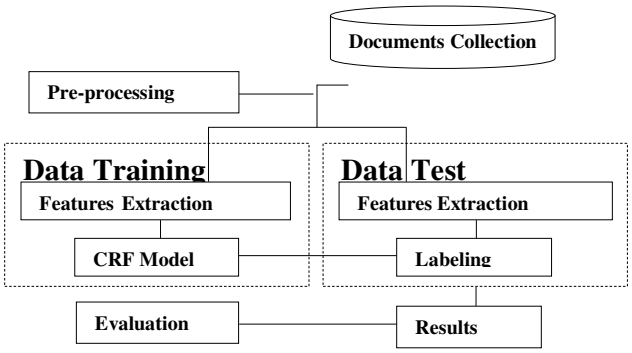
The processing and running of the CRF is shown on Fig. 2.

Figure 1. System Block diagram



The input file for the first time is a training file which gives and output of a model file and in the second run the input file is a testing file. The output file of the CRF is a labeled file.

Figure 2. CRF based POS tagging



The working of CRF is mainly based on the feature selection. The feature listed for the POS tagging is as follows:

F= { W_{i-m}, … ,W_{i-1}, W_i, W_{i+1}, ..., W_{i+n}, SW_{i-m}, …, SW_{i-1}, SW_i, SW_{i+1},... , SW_{i-n} , number of acceptable standard suffixes, number of acceptable standard prefixes, acceptable suffixes present in the word, acceptable prefixes present in the word, word length, word frequency, digit feature, symbol feature, RMWE}

The details of the set of features that have been applied for POS tagging in Manipuri are as follows:

The details of the set of features that have been applied for POS tagging in Manipuri are as follows:

1. Surrounding words as feature: Preceeding word(s) or the successive word(s) are important in POS tagging because these words play an important role in determining the POS of the present word.

2. Surrounding Stem words as feature: The Stemming algorithm mentioned in [10] is used. The preceding and the following stemmed words of a particular word can be used as features. It is because the preceding and the following words influence the present word POS tagging.

3. Number of acceptable standard suffixes as feature: As mention in [10], Manipuri being an agglutinative language the suffixes plays an important in determining the POS of a word. For every word the number of suffixes are identified during stemming and the number of suffixes is used as a feature.

4. Number of acceptable standard prefixes as feature: Prefixes plays an important role for Manipuri language. Prefixes are identified during stemming and the prefixes are used as a feature.

5. Acceptable suffixes present as feature: The standard 61 suffixes of Manipuri which are identified is used as one feature. The maximum number of appended suffixes is reported as ten. So taking into account of such cases, for every word ten columns separated by a space are created for every suffix present in the word. A "0" notation is being used in those columns when the word consists of no acceptable suffixes.

6. Acceptable prefixes present as feature: 11 prefixes have been manually identified in Manipuri and the list of prefixes is used as one feature. For every word if the prefix is present then a column is created mentioning the prefix, otherwise the "0" notation is used.

7. Length of the word: Length of the word is set to 1 if it is greater than 3 otherwise, it is set to 0. Very short words are generally pronouns and rarely proper nouns.

8. Word frequency: A range of frequency for words in the training corpus is set: those words with frequency <100 occurrences are set the value 0, those words which occurs >=100 are set to 1. It is considered as one feature since occurrence of determiners, conjunctions and pronouns are abundant.

9. Digit features: Quantity measurement, date and monetary values are generally digits. Thus the digit feature is an important feature. A binary notation of '1' is used if the word consist of a digit else '0'.
10. Symbol feature: Symbols like $,% etc. are meaningful in textual use, so the feature is set to 1 if it is found in the token, otherwise 0. This helps to recognize Symbols and Quantifier number tags.
11. Reduplicated Multiword Expression (RMWE): (RMWE) are also considered as a feature since Manipuri is rich of RMWE. The work of RMWE is used in [11].

5. EXPERIMENT AND EVALUATION

The text document file is cleaned for processing where the error and grammatical mistakes are minutely checked by an expert. For the POS tagging the expert also mark each word with the POS using a tag set. The POS marked texts are used for both training and testing.

Once the text document are tagged with the POS the same text with POS and the previous features are used to run the CRF based chunking. In other word the POS tag are used as the other features for the chunking. The C++ based CRF++ 0.53 package[1] is used in this work and it is readily available as open source for segmenting or labeling sequential data.

In total to train and test the system 30000 words corpus is used. This corpus is considered as gold standard since an expert manually identifies the POS and the chunk words. Fig.3 shows the sample of POS and chunking which are marked by the expert.

.....
.....
চাওবনা NN B-X
অঙাংবা JJ B-X
লৈ NC I-X
অমা QT I-X
পায় VFC B-X
। SYM O
.....
.....

Figure 3. Smaple of the words with POS and BOI chunking

Of the 30000 words 20000 words are considered for the training and the rest of the 10000 are used for the testing.

Evaluation is done with the parameter of Recall, Precision and F-score as follows:

$$\text{Recall, } \mathbf{R} = \frac{\text{No. of correct tags generated by the system}}{\text{No. of correct tags in the test}}$$

$$\text{Precision, } \mathbf{P} = \frac{\text{No. of correct tags generated by the system}}{\text{No. of tags generated by the system}}$$

1 http://crfpp.sourceforge.net/

F-score, $\mathbf{F} = \frac{(\beta^2+1)PR}{\beta^2 P+R}$

Where is one, precision and recall are given equal weight.
Different combinations of the features are tried for the chunking of the Manipuri text document. Among the combinations the best features are found to be as follows:

F= { W_{i-2}, W_{i-1}, W_i, W_{i+1}, SW_{i-1}, SW_i, SW_{i+1}, number of acceptable standard suffixes, number of acceptable standard prefixes, acceptable suffixes present in the word, acceptable prefixes present in the word, word length, word frequency, digit feature, symbol feature, reduplicated MWE, POS}

The Table II shows the recall, precision and f-measure of the system.

TABLE I. BEST RESULT

Model	Recall	Precision	F-Score
CRF	**71.30**	**77.36**	**74.21**

3. CONCLUSIONS

So far, the chunking work on Manipuri is not reported and this work can be a starting point for the future. Other algorithms for the improvement of the score can also be worked on. The main handicap with this language is its highly agglutinative nature. The system shows a recall of 71.30%, a precision of 77.36% and a F-measure of 74.21% which has lot of rooms for improvement.

REFERENCES

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

[2] John Lafferty, Andrew McCallum and Fernando Pereira, Conditional Random Fields: Probabilistic Models for Segment-ing and Labeling Sequence Data.

[3] S. Abney. Parsing by chunks. In R. Berwick, S. Abney, and C. Tenny, editors, Principle-based Parsing. Kluwer Academic Publishers, 1991.

[4] S. Geman and M. Johnson. Dynamic programming for parsing and estimation of stochastic uni_cation-based grammars. In Proc. 40th ACL, 2002.

[5] A. Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. In C. Cardie and R. Weischedel, editors, EMNLP-2. ACL, 1997.

[6] E. F. T. K. Sang. Memory-based shallow parsing. Journal of Machine Learning Research, 2:559.594, 2002.

[7] T. Zhang, F. Damerau, and D. Johnson. Text chunking based on a generalization of winnow. Journal of Machine Learning Research, 2:615.637, 2002.

[8] Lafferty, J., McCallum, A., Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, In the Procceedings of the 18th ICML01, Williamstown, MA, USA., 2001, p. 282-289.

[9] Kishorjit, N. and Sivaji, B., "A Transliteration of CRF Based Manipuri POS Tagging", In the Proceedings of 2nd International Conference on Communication, Computing & Security (ICCCS-2012), Elsevier Ltd, 2012

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

[11] Kishorjit Nongmeikapam, Nonglenjaoba L., Nirmal Y. & Sivaji Bandhyopadhyay, Reduplicated MWE (RMWE) Helps in Improving the CRF Based Manipuri POS Tagger, International Journal of Information Technology Convergence and Services (IJITCS) Vol.2, No.1, DOI : 10.5121/ijitcs.2012.2106, 2012, p.45-59.

Authors

Kishorjit Nongmeikapam is working as Asst. Professor at Department of Computer Science and Engineering, MIT, Manipur University, India. He has completed his BE from PSG college of Tech., Coimbatore and has completed his ME from Jadavpur University, Kolkata, India. He is presently doing research in the area of Multiword Expression and its applications. He has so far published 30 papers and presently handling a Transliteration project funded by DST, Govt. of Manipur, India. He is the author of the Book, "See the C Programming Language".

Chiranjiv Chingangbam is presently a student of Manipur Institute Of Technology. He is pursuing his B.E. in Dept. of Computer Science and Engineering. His area of interest is NLP.

Nepoleon Keisham is presently a student of Manipur Institute Of Technology. He is pursuing his B.E. in Dept. of Computer Science and Engineering. His area of interest is NLP.

Biakchnungnunga Varte is presently a student of Manipur Institute Of Technology. He is pursuing his B.E. in Dept. of Computer Science and Engineering. His area of interest is NLP.

Sivaji Bandyopadhyay is working as a Professor since 2001 in the Computer Science and Engineering Department at Jadavpur University, Kolkata, India. His research interests include machine translation, sentiment analysis, textual entailment, question answering systems and information retrieval among others. He is currently supervising six national and international level projects in various areas of language technology. He has published a large number of journal and conference publications.