Genetic Algorithm (GA) Implementation for Feature Selection in Manipuri POS Tagging

Kishorjit Nongmeikapam
Department of Computer Sc. & Engg.
Indian Institute of Information Technology (IIIT)
Manipur, India
kishorjit@iiitmanipur.ac.in

Sivaji Bandyopadhyay
Department of Computer Sc. & Engg.
Jadavpur University
Kolkata, India
sivaji_cse_ju@yahoo.com

Abstract

Feature selection is a major hurdle for the CRF and SVM based POS tagging. The features are of course listed which will have a very good impact with the identification of POS. Among the listed features, the feature selection is purely a manual effort with hit and trail methods among them. The best way for better output is to design a system where the system itself identifies the best combination of features. A Genetic Algorithm (GA) system is design so that the best possible combination can be sort out instead of a manual hit and trail method in feature selection. The system shows a Recall of **80.00%**, Precision (**P**) of **90.43%** and F-score (**F**) of **84.90%**.

1 Introduction

The Part of Speech (POS) tagging labels each word or token in a sentence with its appropriate syntactic category of a human language called part of speech. POS plays an important role in the language syntactic structure. It tells us the basic elements of the syntactic governing rules of a language that is the grammar of a human language. In general the POS tagging identifies the types of noun, pronoun, verb, adjective, determiner, etc.

The Manipuri language is one type of Tibeto-Burman language. Tibeto-Burman languages are generally agglutinative in nature. The Manipuri language is not simply agglutinative but one can observe that it is highly agglutinative like Turkish. Manipuri language or otherwise known as the Meeteilon by the locals is a lingua franca language spoken in the state of Manipur, which is in the North Eastern part of India.

This paper is organized with the related works in Section 2, concepts of CRF in Section 3 followed by the concepts of GA in Section 4, the experimental design, experiment and evaluation is discussed in Section 5 and Section 6 respectively. The last but not the least Section 8 draws the conclusion of this work.

2 Related works

POS tagging forms one of the basic Natural Language Processing (NLP) work. Several works are reported for different languages. To list some among them, the POS tagger for English is reported with a Simple Rule-based based in (Brill, 1992). Also a transformation-based error-driven learning based POS tagger in (Brill, 1995), maximum entropy methods based POS tagger in (Ratnaparakhi, 1996) and Hidden Markov Model (HMM) based POS tagger in (Kupiec, 1992). The works of Chinese language are also reported which ranges from rule based (Lin et al., 1992), HMM (Chang et al., 1993)to Genetic Algorithms (Lua, 1996). For Indian languages like Bengali works are reported in (Ekbal et al., 2007a), (Ekbal et al., 2007b) (Ekbal et al., 2008c), (Anthony et al., 2010) for Malayalam and for Hindi in (Smriti et al., 2006).

CRF based Manipuri POS tagging is reported in (Kishorjit et al., 2012a) and also reported that an improvement is done using reduplicated multiword expression in (Kishorjit et al., 2012b). Manipuri

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 267–274,
Varanasi, India. December 2016. ©2016 NLP Association of India (NLPAI)

being resource poor language it is reported in (Kishorjit et al., 2012a) that a transliterated model is adopted.

3 Conditional Random Field (CRF)

Conditional Random Field is a very popular topic for the researcher to classify the classes through probabilistic model. Conditional Random Field (Lafferty et al., 2001) 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 :

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

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:

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

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.

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

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,

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

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

4 Genetic Algorithm (GA)

The use of genetic algorithm in the world of Computer science and engineering become very popular. John Holland in 1975 developed the idea of Genetic Algorithm, which is a probabilistic search method. Genetic Algorithm implements the idea of the real world for natural selection, mutation, crossover and production of the new offspring. The basic steps or algorithm in GA which is also mention in (Kishorjit et al., 2011) are as follows.

```
Algorithm: Genetic Algorithm

Step 1: Initialize a population of chromosomes.

Step 2: Evaluate each chromosome in the population or chromosome pool.

Step 3: Create offspring or new chromosomes by mutation and crossover from the pool.

Step 4: Evaluate the new chromosomes by a fitness test and insert them in the population.

Step 5: Check for stopping criteria, if satisfies return the best chromosome else continue from step 3.

Step 6: End
```

The flowchart in Figure 1 explains the above algorithm. The Genetic Algorithm have five basic components, they are:

1. Chromosome representations for the feasible solutions to the optimization problem.
2. Initial population of the feasible solutions.
3. A fitness function that evaluate each solution.
4. A genetic operator that produce a new population from the existing population.
5. Control parameter like population size, probability of generic operators, number of generations etc.

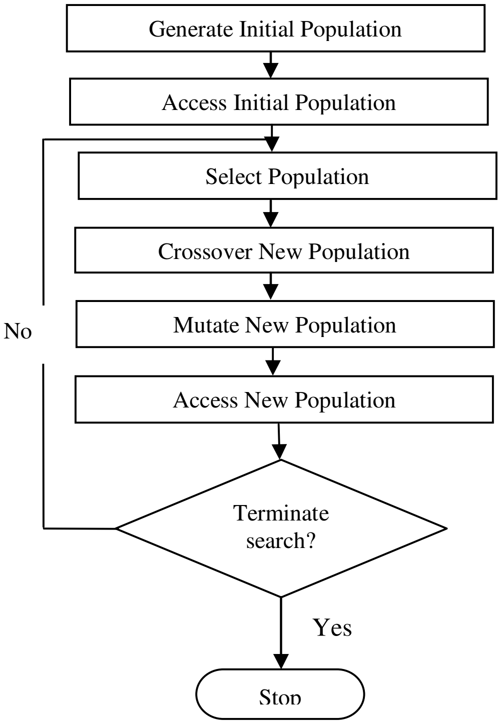

Figure 1. Flow chart of Genetic Algorithm.

5 System Design

The backbone of this system is the CRF model. The CRF based POS tagging model requires to pay much attention in the listing of the feature candidates. In the next step a careful selection of the combinations of the features is required. In much of the cases it is observed that the selection of the feature combination is a manual effort. In each run of the system, one has to select or drop a feature candidate in order to get the best output. In short it's a hit and trail method for the best combinations of the candidate features. One stops the experiment when it reached a good output but it may not be the best.

The work in this paper solves the problem of manual selection of candidate features for the best combination. This is achieved with the use of the Genetic Algorithm technique. So, the system design here is a combination of CRF and supported by the GA in the feature selection.

5.1 The CRF Model

One of the most popular CRF run toolkit for the POS tagging is used. A readily available C++ based CRF++ 0.53 package[1] which is as open source for segmenting or labeling sequential data is used. The CRF model for Manipuri POS tagging (Figure 2) consists of mainly data training and data testing. Cleaning of corpus is important to yield a better result. By cleaning it means to make the corpus error free from spelling, grammar, etc. Linguistic experts are employee for manual tagging. This manually POS tag data is used for the experimental purpose. Also the corpus is a domain based one. This work is done in the newspaper based corpus domain.

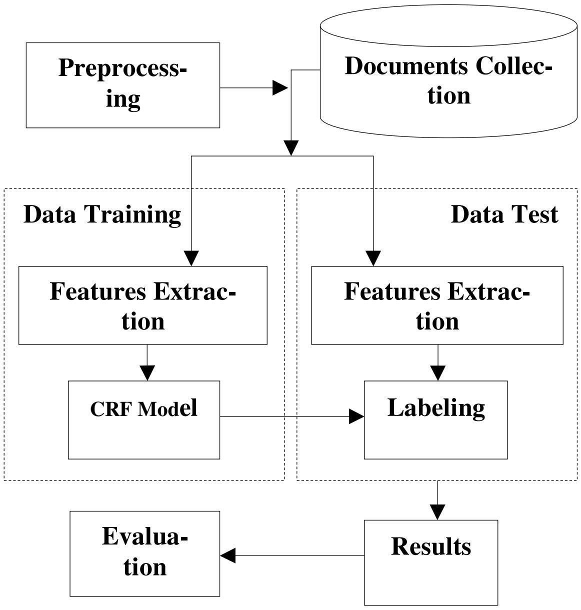

Figure 2. CRF Model of POS tagging

5.2 The Candidate Features

While dealing with the concept of CRF in Section 3 it is mentioned that there is a feature function. Various carefully selected candidate features are to be listed. This candidate features are listed with an aim that it will directly or indirectly influence the tagging of the POS. The candidate features details that have been applied for Manipuri POS tagging are as follows:

Current and adjoining words: The word and the adjoining words may play a crucial role in deciding the POS of the word. The grammatical rule shows an influence among the adjacent words thus current and adjoining words are considered as a feature.

[1] http://crfpp.sourceforge.net/

Root word: In order to get the stem word stemming is done as mentioned in (Kishorjit et al., 2011). Root word is considered as another feature. The root word itself sometimes tells the POS specially the noun and verb.

Adjoining root words: The preceding and following root words of a particular word sometimes influence the present word in case of POS.

Acceptable suffixes: As mention in (Nonigopal 1981) 61 suffixes have been manually identified in Manipuri and the list of suffixes is used as one feature. The language is highly agglutinative so ten columns separated by space for each word to store each suffix present in the word is adopted. A "0" notation is being used in those columns when the word consists of less or no acceptable suffixes.

Acceptable prefixes as feature: Also based on (Nonigopal 1981) 11 prefixes have been manually identified in Manipuri and the list of prefixes is used as a feature. For every word the prefix is identified and a column is created mentioning the prefix if it is present, otherwise the "0" notation is used.

Binary notation for suffix(es) present: The suffixes play an important role in Manipuri since it is a highly agglutinative language. For every word if suffix(es) is/are present during stemming a binary notation '1' is used, otherwise a '0' is stored.

Number of acceptable suffixes as feature: For every word the number of suffixes is identified during stemming, if any and the number of suffixes is used as a feature.

Binary notation for prefix(es) present: The prefixes play an important role in Manipuri since it is a highly agglutinative language. For every word if prefix(es) is/are present during stemming a binary notation '1' is used, otherwise a '0' is stored.

Binary Notation of general salutations/preceding word of Name Entity: In order to identify the NE salutations like Mr., Miss, Mrs, Shri, Lt., Captain, Rs., St., Date etc. that precede the Name Entity are considered as a feature. A binary notation of '1' if used, else a '0' is used.

Binary notation of general follow up words of Name Entity: Name Entities are generally MWEs. The following word of the current word can also be considered as a feature since a name may have ended up with clan name or surname or words like 'organization', 'Lup' etc for organization, words like 'Leikai', 'City' etc for places and so on. A binary notation of '1' if used else a '0' is used.

Digit features: Date, currency, weight, time etc are generally digits. Thus the digit feature is an important feature. A binary notation of '1' is used if the word consists of a digit else a '0' is used.

Length of the word: Length of the word is set to 1 if it is greater than 3 characters. Otherwise, it is set to 0. Very short words are generally pronouns.

Word and surrounding word frequency: A range of frequencies for words in the training corpus are identified: those words with frequency <100 occurrences are set to the value 0, those words which occurs >=100 times but less than 400 times are set to 1. The determiners, conjunctions and pronouns frequently use.

Surrounding POS tag: The POS of the surrounding words are considered as an important feature since the POS of the surrounding words influence the current word POS.

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.

Multiword Expression (MWE): In order to avoid ambiguity or misinterpretation the MWE as a feature is important.

Reduplicated Multiword Expression (RMWE): (RMWE) are also considered as a feature since Manipuri is rich of RMWE. Identification is done using (Kishorjit et al, 2010).

5.3 Corpus preparation

A Manipuri newspaper corpus text document is used as an input file. The training and test files consist of multiple tokens. In addition, each token consists of multiple (but fixed number) columns where the columns are used by a template file. The template file gives the complete idea about the feature selection. Each token must be represented in one line, with the columns separated by white spaces (spaces or tabular characters). A sequence of tokens becomes a **sentence**. Before undergoing training and testing in the CRF, the input document is converted into a multiple token file with fixed columns and the template file allows the feature combination and selection.

An example of the template file which consists of feature details for two example stem words before the word, two stem words after the word, current word, the suffixes (upto a maximum of 10 suffixes), binary notation if suffix is present, number of suffixes, the prefix, binary notation of prefix is present, binary notation if digit is present, binary notation if general list of salutation or preceding word is present, binary notation if general list of follow up word is present, frequency of the word, word length, POS of the current word, POS of the prior two word, POS of the following two word details is as follows:

```
 য়ুনিভর্সিটিশিংদা য়ুনিভর্সিটি দা শিং 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 1 0 NLOC
লুরল্কবা লুরল্ক বা 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 VN
মতাঙদা মতাঙ দা 0 0 0 0 0 0 0 0 0 1 1 ম 1 0 0 0 1 1 RB
মমালোন্দা মমালোন্ দা 0 0 0 0 0 0 0 0 0 1 1 ম 1 0 0 0 1 0 NN
তাঙ্করিবা তাঙ্করি বা 0 0 0 0 0 0 0 0 0 1 1 ত 1 0 0 0 1 0 VN
অসিনা অ না সি 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 1 3 PR
```

Figure 3. Example Sample Sentence in the Training and the Testing File

To run the CRF generally two standard files of multiple tokens with fixed columns are created: one for training and another one for testing. In the training file the last column is manually tagged with all those identified POS tag. In the test file we can either use the same tagging for comparisons and evaluation or only 'O'.

5.4 The Chromosome and Features

The representation of chromosome is discussed in this section. The chromosome pool or population is developed as mentioned in Section 4. Each chromosome consists of genes, which is binary valued. When the gene value is '1' then the feature is selected and when it is '0' the feature is not selected. Figure 4 demonstrates the feature representation as a chromosome. In figure 4, $\mathbf{F_n}$ represents the features.

Feature:

$\mathbf{F_1}$	$\mathbf{F_2}$	$\mathbf{F_3}$	**…**	$\mathbf{F_{n-2}}$	$\mathbf{F_{n-1}}$	$\mathbf{F_n}$

Chromosome:

1	**0**	**1**	**…**	**0**	**1**	**0**

Selected Feature subset = {F1, F3,..., Fn-1}

Figure 4. Example Feature encoding using GA

From the Chromosome pool or initial population, first of all initial selection of chromosome is done. A randomly selected crossover point is marked and **crossover** is executed. During crossover, the sub parts of genes are exchanged from the two chromosomes at the crossover point.

The **mutation** is also done in random so that the chromosomes are not repeated. The objective of mutation is restoring the lost and exploring variety of data. The bit value is changed at a randomly selected point in the process of mutation.

Three fold cross validation technique is used as a **Fitness function**. By three fold cross validation we mean dividing the corpus into 3 nearly equal parts for doing 3-fold cross-validation (use 2 parts for training and the remaining part for testing and do this 3 times with a different part for testing each time).

After fitness test the chromosomes which are fit are placed in the pool and the rest of the chromosomes are deleted to create space for those fit ones.

5.5 Model File after training

In order to obtain a model file we train the CRF using the training file. This model file is a ready-made file by the CRF tool for use in the testing process. In other words the model file is the learnt file after the training of CRF. We do not need to use the template file and training file again since the model file consists of the detail information of the template file and training file

5.6 Testing

The test file is created with feature listed for each word. The last column is tag with the respective POS otherwise 'O' is assigned for those words which are not MWEs. This file has to be created in the same format as that of the training file, i.e., fixed number of columns with the same fields as that of training file.

The output of the testing process is a new file with an extra column which is POS tagged. The new column created by the CRF is the experimental output of the system.

6 Experiment and Evaluation

The experiment is to identify the best combination of features among the feature list. Feature selection is purely based on the Genetic Algorithm (GA) where the features are considered as chromosomes. In each experimental run the selected chromosome is feed as the feature for the CRF system. The out is verified with a three-fold cross validation.

Manipuri corpus are collected from a newspaper domain which is freely available on internet. The correction and filtration of the errors are done by linguistic experts. In the corpus some words are written in English, such words are rewritten into Manipuri in order to avoid confusion or error in the output. The corpus we have collected includes 45,000 tokens which are of Gold standard created by the linguistic experts.

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

Recall,

$$\mathbf{R} = \frac{No\ of\ correct\ ans\ given\ by\ the\ system}{No\ of\ correct\ ans\ in\ the\ text}$$

Precision,

$$\mathbf{P} = \frac{No\ of\ correct\ ans\ given\ by\ the\ system}{No\ of\ ans\ given\ by\ the\ system}$$

F-score,

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

Where β is one, precision and recall are given equal weight.

A number of problems have been faced while doing the experiment due to typical nature of the Manipuri language. The Manipuri language is a tonal language so sometimes its ambiguity creates lots of problem in annotating the POS by the linguist. In Manipuri, word category is not so distinct. The verbs are also under bound category. Another problem is to classify basic root forms according to the word class. Although the distinction between the noun class and verb classes is relatively clear; the distinction between nouns and adjectives is often vague. Distinction between a noun and an adverb becomes unclear because structurally a word may be a noun but contextually it is adverb. Further a part of root may also be a prefix, which leads to wrong tagging. The verb morphology is more complex than that of noun. Sometimes two words get fused to form a complete word.

6.2 Experiment for selection of best feature

A 3-fold cross validation is adopted in this experiment, the corpus is divided into 3 nearly equal parts for doing 3-fold cross-validation (use 2 parts for training and the remaining part for testing). A total of 45,000 words are divided into 3 parts, each of 15000 words.

The features are selected using the GA and experiments are performed in order to identify the best feature. The best features are those which gave the best accurate POS tag in a given text. The experiment is stopped with 60 generations since the output doesn’t show significant change in the F-score.

In each run of the CRF tool the feature template are changed according to the chromosome selected. Figure 5 shows the chart in terms of Recall (**R**), Precision (**P**) and F-score (**F**).

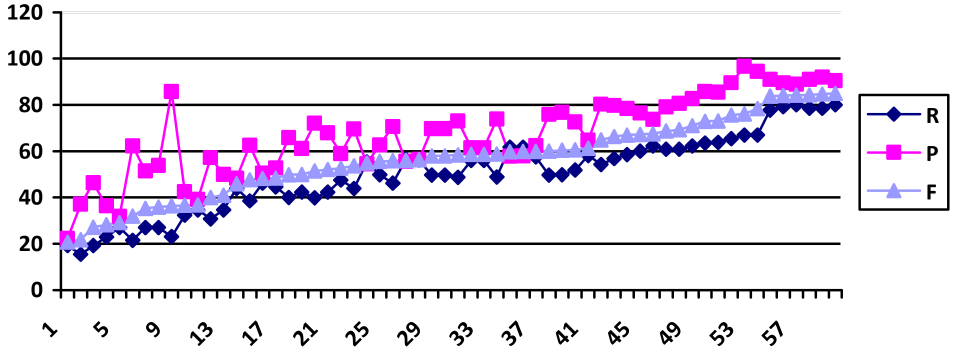

Figure 5. Chart of the performance in 60 generations

The System stops running when the F-Score shows no change in the F-score, i.e., which shows the output is exhausted.

6.3 The Best Feature

The best result for the Manipuri POS so far reported in (Kishorjit et al. 2012) is the Recall (**R**) of **80.20%**, Precision (**P**) of **74.31%** and F-score (**F**) of **77.14%**.

The work here shows the best result for the CRF based POS tagging with the Recall (**R**) of **80.00%**, Precision (**P**) of **90.43%** and F-score (**F**) of **84.90%**. This shows that the F-score is improved from the previous claim of 77.24% to 84.90%.

The implementation of GA could identify the best combination of features. This happens with the following feature set:

F= { W_{i-1}, W_i, W_{i+1}, W_{i+2}, RW_{i-1}, RW_i, RW_{i+1}, RW_{i-2}, no. of acceptable standard suffixes, no. of acceptable standard prefixes, acceptable suffixes present in the word, NE salutations, NE, POS_{i-1}, POS_i, POS_{i+1}word length, word frequency, digit feature, symbol feature, MWE, reduplicated MWE}

In the above feature list **W_i** represents current and surrounding words, **RW_i** represents root words, NE represents the name entity, POS represents the part of speech and MWE represents the multiword expression.

The use of GA for the feature selection stretches the best combination among the listed features. Apart from the best combination it also helps in improving the overall F-score from 77.24% to 84.90%. This means there is best combinations which manual selection missed.

The selection of the feature is done with the application of GA and the F-score can be improved by some 7.66%.

7 Conclusion

Among the works of CRF based POS tagging, this papers comes up with a new method of feature selection through genetic algorithm (GA). This approach reduces the manual effort of hit and trial. The tonal nature of the language sometimes create ambiguity so may be in future it may have to frame some rule or design a system to overcome the ambiguity.

The work is done on a newspaper based corpus, so it has scope to check in other domain and it may differ in other domain.

References

Antony, P.J.: Mohan, S.P.: Soman, K.P.:SVM Based Part of Speech Tagger for Malayalam, In the Proc. of International Conference on Recent Trends in Information, Telecommunication and Computing (ITC), pp. 339 – 341, Kochi, Kerala, India (2010).

Brill, Eric.: A Simple Rule-based Part of Speech Tagger. In the Proceedings of Third International Conference on Applied NLP, ACL, Trento, Italy (1992).

Brill, Eric.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in POS Tagging, Computational Linguistics, Vol. 21(4), pp543-545, (1995).

Chang, C. H.& Chen, C. D.: HMM-based Part-of-Speech Tagging for Chinese Corpora, In the Proc.

of the Workshop on Very Large Corpora, Columbus, Ohio, pp40-47(1993).

Ekbal, Asif, Mondal, S & Sivaji Bandyopadhyay: POS Tagging using HMM and Rule-based Chunking, In the Proceedings of SPSAL2007, IJCAI, pp25-28, India (2007).

Ekbal, Asif, R. Haque & & Sivaji Bandyopadhyay: Bengali Part of Speech Tagging using Conditional Random Field, In the Proceedings 7th SNLP, Thailand (2007).

Ekbal, Asif, Haque, R. & & Sivaji Bandyopadhyay: Maximum Entropy based Bengali Part of Speech Tagging, In A. Gelbukh *(Ed.)*, Advances in Natural Language Processing and Applications, Research in Computing Science (RCS) Journal, Vol.(33), pp67-78 (2008).

Ekbal, Asif, Mondal, S & Sivaji Bandyopadhyay: Part of Speech Tagging in Bengali Using SVM, In Proceedings of International Conference on Information Technology(ICIT), pp. 106 – 111, Bhubaneswar, India (2008)

Ratnaparakhi, A. :A maximum entropy Parts- of-Speech Tagger, In the Proceedings EMNLP 1, ACL, pp133-142(1996).

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

Kishorjit, N., Nonglenjao, L., Nirmal .Y, and Sivaji B., “Improvement of CRF Based Manipuri POS Tagger by Using Reduplicated MWE (RMWE)”, In: Proceedings of First International Conference on Information Technology Convergence and Services (ITCS 2012), Bangalore, India, pp. 55-69, January 4, 2012.

Kishorjit, N., Nonglenjao, L., Nirmal Y., and Sivaji B., “Reduplicated MWE (RMWE) Helps In Improving The CRF Based Manipuri POS Tagger”, International Journal of Information Technology and Computer Science (IJITCS) Vol 2, No 1, ISSN: 2231-1939, DOI: 10.5121/ijitcs.2012.2106, pp. 45-59, Feb 2012.

Kishorjit, N. and Sivaji B., “A Transliterated CRF Based Manipuri POS Tagging”, In: Proceedings of 2nd International Conference on Communication, Computing & Security (ICCCS 2012), Elsevier Publication, NIT Rourkela, India

Kishorjit, N., & Sivaji Bandyopadhyay. Identification of Reduplicated MWEs in Manipuri: A Rule based Approached. In the Proc. of *23rd ICCPOL-2010*, San Francisco; 2010, p. 49-54.

Kishorjit, N. and Sivaji, B.,“ Genetic Algorithm (GA) in Feature Selection for CRF Based Manipuri Multiword Expression (MWE) Identification”, International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 5, ISSN : 0975 – 3826, pp. 53-66, Oct 2011.

Kupiec, R.: Part-of-speech tagging using a Hidden Markov Model, In Computer Speech and Language, Vol 6, No 3, pp225-242(1992).

Lin, Y.C., Chiang, T.H. & Su, K.Y.: Discrimination oriented probabilistic tagging, In the Proceedings of ROCLING V, pp87-96(1992).

Lua, K. T.: Part of Speech Tagging of Chinese Sentences Using Genetic Algorithm, In the Proceedings of ICCC96, National University of Singapore, pp45-49 (1996).

Nonigopal Singh, N: A Meitei Grammar of Roots and Affixes, A Thesis, Unpublish, Manipur University, Imphal (1987)

Smriti Singh, Kuhoo Gupta, Manish Shrivastava, & Pushpak Bhattacharya: Morphological Richness offsets Resource Demand –Experiences in constructing a POS tagger for Hindi, In the Proceedings of COLING- ACL, Sydney, Australia (2006).