Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 6 (2012) 582 – 589

Procedia Technology

2nd International Conference on Communication, Computing & Security [ICCCS-2012]

A Transliteration of CRF Based Manipuri POS Tagging

Kishorjit Nongmeikapam[a,*], Sivaji Bandyopadhyay[b]

[a]*Deptt. Of Computer Sc.& Engg.,MIT, Manipur University, Imphal 795001, India*
[b]*Deptt. Of Computer Sc.& Engg., Jadavpur University, Kolkata 700032, India*

Abstract

Transliteration is common to all those language which have multiple scripts. Manipuri, which is one of the Schedule Indian Languages, is one of them. This language has two scripts: a borrowed Bengali Script and the original Meitei Mayek (Script). Part of Speech (POS) tagging of the Bengali Script Manipuri text is performed using Conditional Random Field (CRF) which is then followed by the transliteration to Meitei Mayek.

Keywords: CRF; POS; Transliteration; Features; Manipuri;

1. Introduction

Manipuri or the Manipuri Language, which is one of the Scheduled Indian Language usage two scripts: a borrowed Bengali Script and the original Meitei Mayek (Script). This language is a highly agglutinative language with rich in morphemes. The highly agglutinative nature of this language can be proved with this following example word: *"পুশিনহনজরমগদবনিদকো"* ("pusinhənjərəmgədəbənidəko"), which means "(I wish I) myself would have caused to bring in (the article)". Here there are 10 (ten) suffixes being used in a verbal root, they are "*pu*" is the verbal root which means "to carry", "*sin*"(in or inside), "*hən*" (causative), "*jə*" (reflexive), "*rəm*" (perfective), "*gə*" (associative), "*də*" (particle), "*bə*" (infinitive), "*ni*" (copula), "*də*" (particle) and "*ko*" (endearment or wish).

The POS tagging is the task of labelling each word or token in a sentence with its appropriate syntactic category called part of speech. POS tagging has various applications in Natural Language

*Kishorjit Nongmeikapam. Tel.: +91-8974008610.
E-mail address: kishorjit.nongmeikapa@gmail.com

doi:10.1016/j.protcy.2012.10.070

Processing (NLP) systems like IR, Summarization, Machine translation, NER, Multiword Expression (MWE) identification, etc.

The paper is organized with related work in Section 2 followed by the concepts of CRF in Section 3, the transliteration algorithm in section 4, the CRF Model and feature selection in Section 5, and the experiment and its result in Section 6 and the conclusion is drawn.

2. Related works

Several works on POS taggers for different language could be found say for English: a Simple Rule-based based POS tagger is reported in [1], transformation-based error-driven learning based POS tagger in [2], maximum entropy methods based POS tagger in [3] and HMM based POS tagger in [4]. For Chinese, the works are found ranging from rule based, HMM to GA [5]-[7]. For Indian languages like Bengali works are reported in [8]-[10] and for Hindi in [11]. Works of POS tagging using SVM methods is reported in [12]. Manipuri POS tagging is reported in [13]-[17]. The identification of Reduplicated Multiword Expression (RMWE) is reported in [18]-[19]. Web Based Manipuri Corpus for Multiword NER and RMWEs Identification using SVM is reported in [20]. Transliteration work of Manipuri from Bengali Script to Meitei Mayek is reported in [21].

3. Concepts of CRF

The concept of Conditional Random Field [22] 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's an unsupervised approach where the system learns by giving some training and can be use for testing other text.

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 :

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

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 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 makes the probability of all state sequences sum to 1. This is calculated as in HMM and can be obtain 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 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 is concave, so the *λ's* have a unique set of optimal values.

4. The Transliteration Algorithm

The transliteration process is the mapping of a word from a source language script to another target language script. A simple transliteration scheme of Manipuri from Bengali script to Meitei Mayek is described in [21] and the same algorithm is adopted here. In general, the Bengali script which has 52 consonants and 12 vowels is mapped to Meitei Mayek which has 27 alphabets (Iyek Ipee) and its supplements: vowels, Cheitap Iyek, Cheising Iyek and Lonsum Iyek [23] are shown in Tables 1,2,3,4 and 5.

Table 1. Iyek Ipee characters in Meitei Mayek..

Iyek Ipee			
ক->ꯀ (kok)	স(ছ,শ,ষ)->ꯁ (Sam)	ল->ꯂ (Lai)	ম->ꯃ (Mit)
প->ꯄ (Pa)	ন->ꯅ (Na)	চ->ꯆ (Chil)	ত(ট)->ꯇ (Til)
খ->ꯈ (Khou)	ঙ-> ꯉ (Ngou)	থ(ঠ)->ꯊ(Thou)	ব->ꯅ (Wai)
য(য়->ꯌ (Yang)	হ->ꯍ (Huk)	ঊ(ঊ)->ꯎ(Un)	ই(ঈ)->ꯏ(Ee)
ফ->ꯐ (Pham)	অ->ꯑ (Atia)	গ->ꯒ (Gok)	ঝ->ꯓ (Jham)
র->ꯔ (Rai)	ব->ꯕ (Ba)	জ-> ꯖ (Jil)	দ(ড)->ꯗ(Dil)
ঘ->ꯘ (Ghou)	ধ(ঢ)->ꯙ(Dhou)	ভ->ꯚ(Bham)	

Table 2. Vowels of Meitei Mayek.

Vowel letters		
আ->ꯥ(Aa)	এ->ꯦ(Ae)	ঐ->ꯧ(Ei)
ও->ꯣ(o)	ঔ->ꯤ(Ou)	অং->ꯡ(Ang)

Table 3. Cheitap Iyek of Meitei Mayek

Cheitap Iyek			
□->ꯪ (ot nap)	□, □-> ꯫(inap)	□->꯬(aatap)	□-> ꯭(yetnap)
□-> ꯮ (sounap)	□, □->꯰ (unap)	□->꯱(cheinap)	◌ং-> ꯲(nung)

Table 4. Cheising Iyek or numerical figures of Meitei Mayek

Cheising Iyek(Numeral figure)			
১->1(ama)	২->2(ani)	৩->3(ahum)	৪->4(mari)
৫->5(manga)	৬->6(taruk)	৭->7(taret)	৮->8(nipal)
৯->9(mapal)	১০->10(tara)		

Table 5. Lonsum Iyek of Meitei Mayek

Lonsum Iyek			
ক্-> ꯀ (kok lonsum)	ল্-> ꯋ (lai lonsum)	ম্->ꯄ (mit lonsum)	প্-> ꯄ(pa lonsum)
ণ্, ন্-> ꯅ (na lonsum)	ট্,ত্-> ꯇ (til lonsum)	ঙ্->ꯉ(ngou lonsum)	ই, ঈ->ꯏ(ee lonsum)

Alphabets of Meitei Mayek are repeated uses of the same alphabet for different Bengali alphabet like ঘ, ঋ, ঝ, ঘ in Bengali is transliterated to ꯝ in Meitei Mayek.

In Meitei Mayek, Lonsum Iyek (in Table 5) is used when ক্ is transliterated to ꯀ্, ঙ্ transliterate to ꯉ্, চ্ transliterate to ꯆ্ etc. Apart from the above character set Meitei Mayek uses symbols like '꯫' (Cheikhie) for '।' (full stop in Bengali Script). For intonation we use '꯭' (Lum Iyek) and '꯬' (Apun Iyek) for *ligature*. Other symbols are as internationally accepted symbols.
Algorithm use for the transliteration scheme is as follows:

```
Algorithm: transliteration(line, BCC, MMArr[], BArr[])
 1.    line : Bengali line read from document
 2.    BCC : Total number of Bengali Character
 3.    MMArr[] : Bengali Characters List array
 4.    BArr[] : Meitei Mayek Character List array
 5.    len : Length of line
 6.    for m = 0 to len-1 do
 7.     tline=line.substring(m,m+1)
 8.     if tline equals blank space
 9.       Write a white space in the output file
10.     end of if
11.     else
12.      for index=0 to BCC-1
13.       if tline equals BArr[index]
14.        pos = index
15.        break
16.       end of if
17.      end of for
18.      Write the String MMArr[pos] in the output file
19.     end of else
20.   end of for
```

In the algorithm two mapped file for Bengali Characters and corresponding Meitei Mayek Characters which are read and stored in the *BArr* and *MMArr* arrays respectively. A test file is used so that it can compare its *index* of mapping in the Bengali Characters List file which later on used to find the corresponding target transliterated Meitei Mayek Characters Combination. The transliterated Meitei Mayek Character Combination is stored on an output file.

5. CRF Model and feature selection

5.1. CRF working model

CRF based Manipuri POS taggers are found in [15]-[16]. The model used here is the improved work of [16] which is also a CRF based POS tagging. The C++ based CRF++ 0.53 package[a] is used in this work and it is readily available as open source for segmenting or labeling sequential data.

The CRF model for Manipuri POS tagging (Figure 1) consists of mainly data training and data testing. The important processes required in POS tagging using CRF are feature selection and pre-processing. CRF undergoes creation of model file after training and finally the testing with the test corpus.

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

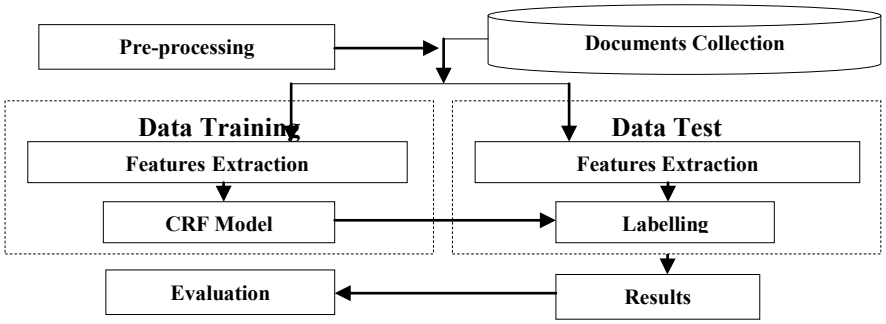

Fig. 1. CRF Model of Manipuri POS Tagging

The input of CRF based POS tagging model is a Bengali Script Manipuri text file. The training and test files consist of multiple tokens or words with similar features in form of multiple (but fixed number) columns. A sequence of tokens and other information in the column becomes a **sentence**. In the training file the last column is manually tagged with all the identified POS tags[b] whereas in the test file we can either use the same tagging for comparisons or only 'O' for all the tokens regardless of POS.

Training of CRF system is done in order to get an ouput 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.

The testing 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.

5.2. Possible feature list

Since the CRF is a technique for pattern classification which has been widely used in many application areas. Feature selection is another factor that impacts classification accuracy. A very careful selection of the feature is important in CRF. Various candidate features are listed. Those candidate features which are listed to run the system 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 [24] 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 [24], 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

---

[b]http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

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.

6. Experiment

The first part of the experiment is the tagging of Bengali script Manipuri using the CRF Model which is followed by the transliteration of the Bengali script Manipuri to Meitei Mayek Manipuri. Section 4 describes about the transliteration scheme.

A total of 30,000 words are divided into two files, one consisting of 24000 words as training file and the second file consisting of 6000 words as testing file. As mention above the sentences are separated into equal numbers of columns representing the different features separated by blank spaces.

In order to evaluate the experiment result for POS tagging, the system used the parameters of Recall, Precision and F-score. These parameters are defined as follows:

$$\text{Recall, } \mathbf{R} = \frac{\textit{No of correct POS tag assigned by the system}}{\textit{No of correct POS tag in the text}}$$

$$\text{Precision, } \mathbf{P} = \frac{\textit{No of correct POS tag assigned by the system}}{\textit{No of POS tag assigned by the system}}$$

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

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

Different combinations are tried with the manually selected of the features. The main target of the POS tagging is the improvement in the F-measure. Among the different experiments with different combinations Table 7 lists some of the best combinations for POS tagging. Table 6 explains the notations used in Table 7.

Table 6. Meaning of the notations

Notation	*Meaning*
W[-*i*,+*j*]	Words spanning from the i^{th} left position to the j^{th} right position
SW[-*i*, +*j*]	Stem words spanning from the i^{th} left to the j^{th} right positions
P[*i*]	The *i* is the number of acceptable prefixes considered
S[*i*]	The *i* is the number of acceptable suffixes considered
L	Word length

F	Word frequency
NS	Number of acceptable suffixes
NP	Number of acceptable prefixes
D	Digit feature (0 or 1)
SF	Symbol feature (0 or 1)
RM	Reduplicated Multiword expression

Table 7. System performance with various feature combinations for POS tagging

Feature	R(in %)	P(in %)	FS(in %)
W[-2,+1], SW[-1,+1], P[1], S[4], L, F, NS, NP, D, SF, RM	**80.20**	**74.31**	**77.14**
W[-2,+2], SW[-2,+1], P[1], S[4], L, F, NS, NP, D, SF, RM	73.53	78.95	76.14
W[-2,+3], SW[-2,+2], P[1], S[4], L, F, NS, NP, D, SF	75.24	69.09	72.04
W[-3,+1], SW[-3,+1], P[1], S[4], L, F, NS, NP, D, SF	72.01	65.45	68.57
W[-3,+3], SW[-3,+2], P[1], S[5], L, F, NS, NP, D	61.76	49.61	55.02
W[-3,+4], SW[-2,+3], P[2], S[5], L, F, NS, SF, RM	39.22	57.14	46.51
W[-4,+1], SW[-4,+1], P[2], S[6], L, NP, D, SF	47.01	37.60	41.78
W[-4,+3], SW[-3,+3], P[3], S[9], L, F, D, SF, RM	25.49	49.06	33.55

The best result for the CRF based POS tagging is the one which shows the best F-measure among the results. This happens with the following feature set:

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}

The experimental result of the above feature combination shows the best result which gives the Recall (**R**) of **80.20%**, Precision (**P**) of **74.31%** and F-measure (**F**) of **77.14%**.

The transliteration algorithm of [21] is efficient with an accuracy of **86.04%.** Implementing this transliteration scheme with the output of the CRF based POS tagging the accuracy dips down a bit that is by **1.21%.** The observation shows wrong transliteration take place for those loan words, which means those words which are not Manipuri originated but borrowed from other.

7. Conclusion

A combine approach of transliteration and the CRF based POS tagger is tried which will be helpful in POS tagging of resource poor Meitei Mayek Manipuri language. Such experiment will prove to be good and useful for other multi script language. Similar approach can be tried in the other applications of Natural language processing. The feature assumption may not be reliable because the features are selected with linguistic knowledge.

References

[1] Brill, Eric. A Simple Rule-based Part of Speech Tagger. In the Proceedings of *Third International Conference on Applied NLP*, ACL, Trento; 1992, Italy .

[2] Brill, Eric. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in POS Tagging, *Computational Linguistics*, Vol. 21(4), 1995, p. 543-545.

[3] Ratnaparakhi, A. A maximum entropy Parts- of- Speech Tagger, In the Proceedings *EMNLP 1, ACL*; 1996, p.133-142.

[4] Kupiec, R.. Part-of-speech tagging using a Hidden Markov Model, In: *Computer Speech and Language*, Vol 6, No 3; 1992, p. 225-242.

[5] Lin, Y.C., Chiang, T.H. & Su, K.Y. Discrimination oriented probabilistic tagging, In the Proceedings of *ROCLING V*; 1992, p. 87-96.

[6] Chang, C. H. & Chen, C. D. HMM-based Part-of-Speech Tagging for Chinese Corpora, In the Proceedings of the *Workshop on Very Large Corpora*, Columbus, Ohio; 1993, p. 40-47.

[7] Lua, K. T. Part of Speech Tagging of Chinese Sentences Using Genetic Algorithm, In the Proceedings of *ICCC96*, National University of Singapore; 1996, p. 45-49.

[8] Ekbal, Asif, Mondal, S.& Sivaji Bandyopadhyay POS Tagging using HMM and Rule-based Chunking, In the Proceedings of SPSAL2007, IJCAI India; 2007,p. 25-28.

[9] Ekbal, Asif, R. Haque & Sivaji Bandyopadhyay. Bengali Part of Speech Tagging using Conditional Random Field, In the Proceedings *7th SNLP*, Thailand 2007.

[10] 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, RCS Journal*, Vol.(33); 2008, p. 67-78.

[11] 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.

[12] 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)*, Kochi, Kerala, India; 2010, p. 339 –341.

[13] Doren Singh, T. & Sivaji Bandyopadhyay. Morphology Driven Manipuri POS Tagger, In the Proceeding of *IJCNLP NLPLPL 2008*, IIIT Hyderabad; 2008 , p. 91-97.

[14] Doren Singh, T. & Sivaji Bandyopadhyay. Morphology Driven Manipuri POS Tagger, In the Proceeding of *IJCNLP NLPLPL 2008*, IIIT Hyderabad; 2008, p. 91-97.

[15] Doren Singh, T., Ekbal, A. & Sivaji Bandyopadhyay. Manipuri POS tagging using CRF and SVM: A language independent approach, In the proceeding of *6th ICON -2008*, Pune, India; 2008, p. 240-245.

[16] 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.

[17] Kishorjit Nongemeikapam , Umananda Sharma A., Martina Devi L., Nepoleon K., Dilip Singh Kh., Sivaji Bandyopadhyay. Will the Identification of Reduplicated Multiword Expression (RMWE) Improve the Performance of SVM Based Manipuri POS Tagging?, A. Gelbukh (Ed.):CICLing 2012, Lecture Notes on Computer Science (LNCS) vol.7181, Part I, ISSN: 0302-9743, DOI 10.1007/978-3-642-28604-9 Berlin, Germany: Springer-Verlag, 2012, p. 117-129.

[18] 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.

[19] Kishorjit, N., Dhiraj, L., Bikramjit Singh, N., Mayekleima Chanu, Ng., Sivaji Bandyopadhyay. Identification of Reduplicated Multiword Expressions Using CRF, A. Gelbukh (Ed.):*CICLing, LNCS* vol.6608, Part I, , Berlin, Germany: Springer-Verlag; 2011, p. 41–51.

[20] Doren Singh, T., Sivaji Bandyopadhyay. Web Based Manipuri Corpus for Multiword NER and Reduplicated MWEs Identification using SVM, In the Proceedings of the *1st WSSANLP (COLING)*, Beijing; 2010 p. 35–42.

[21] Kishorjit, N., Herojit Singh, N., Sonia, Th. and Sivaji, Bandyopadhyay. Manipuri Transliteration from Bengali Script to Meitei Mayek: A Rule Based Approach, C. Singh et al. (Eds.):*ICISIL 2011, CCIS* vol..139, Part 2, , Berlin, Germany: Springer-Verlag; 2011, p. 195–198.

[22] 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.

[23] Kangjia Mangang, Ng. *Revival of a closed account*. Sanamahi Laining & Punsiron Khupham, Imphal; 2003, p. 24-29.

[24] 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.