PAN Localization Project

Project No:
Ref. No: PANL10n/KH/Report POS

Khmer Part-of-Speech Tagger

20 September 2008

Cambodia Country Component
PAN Localization Project
PAN Localization Cambodia (PLC) of IDRC

Table of Contents

1 Introduction

1.1 POS overview

Every language has its parts of speech such as verb, noun, adjective...etc. POS tagging (Part of Speech tagging) is a process of automatic assigning the POS for the word. It is based on both its definition, as well as its context. It can be used for text parsing, information extraction, text summarization, grammar checker and machine translation [4].

There are many approach models to develop the automatic POS tagger such as Hidden Markov Model (HMM), Transformation Base and Decision Trees [4]. We have chosen Tree-tagger application, which is based on Decision Tree model, for semi-automatic tagging of Khmer language. There are two reasons to choose this application. First, it is freely available for research purpose. Secondly, it has been successfully used to tag many languages such as German, English, French and a few others [2].

1.2 Khmer language overview

Khmer, also known as Cambodian is the official language of the Kingdom of Cambodia. There are so many issues involved in Khmer Grammar rule. Because of the limitation of time we will not be able to describe all the problems in Khmer Grammar rule, however, we are going to summarize some of them which we faced during our research on Khmer POS Tagging.

Khmer words are written continuously without word delimiter. Due to the limitation of resources for Khmer language, there is only CHOUN NATH Dictionary which is recognized as the official dictionary in Cambodia [4]. Khmer language has a lot of ambiguities i.e. it has different POS dependent on the context of the sentence. Beside of this, some words can be joined together and have another meaning and POS.

The problems which we mentioned above are just a subset of the problem we faced during our research on Khmer POS Tag Set. We have defined 21 tags by grabbing the rules from a Khmer grammar book, Khmer dictionary and some documents related to Khmer grammar to make the POS tag set for Khmer language.

2 Khmer POS Tagger

2.1 Khmer POS tagset

The tag-set we defined contains only major tags based on three main materials listed in the references section [3] [8] [9]. We do not include every sub category POS (POS that is the division of the major POS) defined by other authors [4] because we find it hard to consider them as standardized ones due to some debates.

There are 21 tags defined and documented according to a set of rules discussed in the materials used. The following is a table of all the 21 tags.

Category	POS Tag ID No.	POS Name	POS Tag
Noun	1	Noun	NN
	2	Proper Noun	NNP
Pronoun	3	Pronoun	PRP
	4	Relative Pronoun	RPN
Verb	5	Verb	V
	6	Auxiliary	AUX
Adverb	7	Adverb	RB
Adjective	8	Adjective	JJ
	9	Quantitative adjective	QAD
	10	Possessive adjective	PAD
	11	Determiner adjective	DAD
Ordinal Number	12	Ordinal Number	ON
Preposition	13	Preposition	IN
Interjection	14	Interjection	UH
Conjunction	15	Conjunction	CC
Foreign words	16	Foreign words	FW
Abbreviation	17	Abbreviation	AB
Symbol	18	Symbol	SYM
Mark	19	Mark	M
Tense expression words	20	Tense expression words	TXW
Additional Word	21	Additional Word	AW

Table 1: Khmer tagset

We are not going to explain all the tags, but the last one. Additional word is a category tag name given by our team. Khmer Language has a POS called NiBatSap ( ). We have no English word for this POS, so we decided to use Additional Word as a name for this POS in our POS Tag Set.

Additional word is a word used to clarify the meaning of a noun or verb in a sentence. In current study, we did not identify the exact rule for this POS (Additional word). Although we defined 21 tags for Khmer POS, some of them are still missing which may require further research and development.

2.2 Semi-automatic tagging

2.2.1 Treetagger

The Treetgger is a set of applications developed for annotating text with POS and lemma information. It has been developed within the TC project at the Institute for Computational Linguistics of the University of Stuttgart [5]. The Treetagger consists of two programs: train-tree-tagger for training (2.2.1 step 1) and tree-tagger for tagging (2.2.1 step 2).

To do the semi-automatic tagging we need two main steps given below.

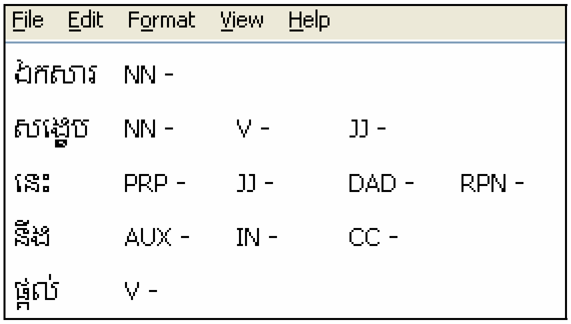
Step 1: Training

Following files are required to complete the training process: lexicon, open tags, tagged data and it produces a parameter file that contains the rules for training.

- **Lexicon:** a file contains words with their POS. The part of speech for each word can be more than one, but the word in the lexicon file cannot be duplicated. The lexicon file produced is based on the tagged data.

    - **File format:**

[Word]([tab][space][Lemma])$^+$



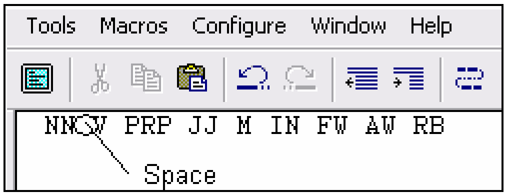
```
ឯកសារ   NN -
សម្ដេច    NN -    V -     JJ -
នេះ     PRP -   JJ -    DAD -   RPN -
នឹង     AUX -   IN -    CC -
ផ្ដល់    V -
```

Figure 1: Lexicon's file format

- **Open Class tags:** a file which contains a list of open class tags i.e. possible tags for unknown word. This information is needed to estimate possible tags for the unknown words. We decided following tags to be the open class tags, NN, V, JJ, M, IN, FW, AW, RB (see 2.1).

    - **File format:**

      ([tag][space])$^+$

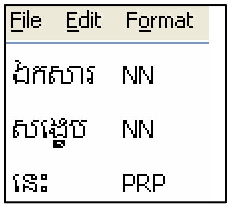


```
NN V PRP JJ M IN FW AW RB
```

Figure 2: Open class tag's file format

- **Tagged data:** a file which contains tagged training data. The data must be in one word per line format. It means that each line contains one token and one tag in the order, separated by a tab.

    - **File format:**



```
ឯកសារ   NN
សម្ដេច    NN
នេះ     PRP
```

Figure 3: Tagged data's file format

- **Output file:** A file in which the resulting tagger parameters are stored.

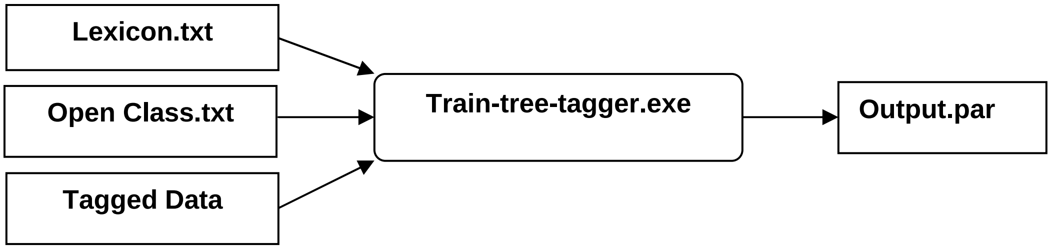
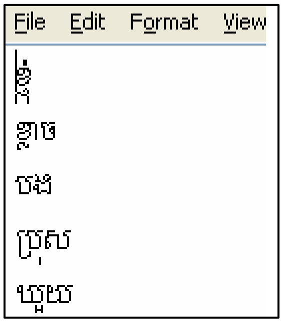


Figure 4: The process model of train-tree-tagger application

Step 2: Tagging

The application requires the parameter file, untagged file, and it produces a tagged file.

- **Parameter file:** an output file we got from train-tree-tagger in step 1.
- **Input file:** The data in this file must be in one word per line format.
  - **File format:**



ខ្ញុំ
ខ្លាច
បង
ប្រុស
ឃ្មុយ

Figure 5: Input file's format

- **Output file:** a tagged data after the execution of the tree-tagger.

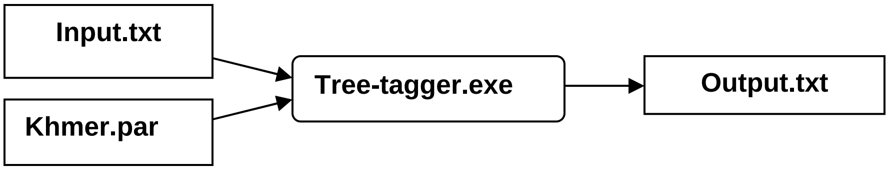


Figure 6: The process model of tree-tagger application

2.2.2 Corpus' data

There are two different sources, the LICADHO's report [5] and the Khmer Rouge Trial website [6], from where we have taken the text. The content of the documents are both written in official and daily speaking language. There are some preprocess on the text before they can be used as the data for the tagger application. Those preprocess are Conversion, word segmentation.

- **Conversion:** the documents we retrieve from both sources are written in none-Unicode, because of this we need to convert those documents in to Unicode.

- **Word Segmentation:** As we mentioned in Khmer Language Overview section Khmer words are written continuously without delimiter. The input data for tagger application requires all the word separate from each other by new line. That is the reason we need to split the sentence into words.

Finally, the untagged text is ready for tagging. We needed manually tagged corpus about 20,000 words for the semi-automatic tagging.

2.2.3 Training and results

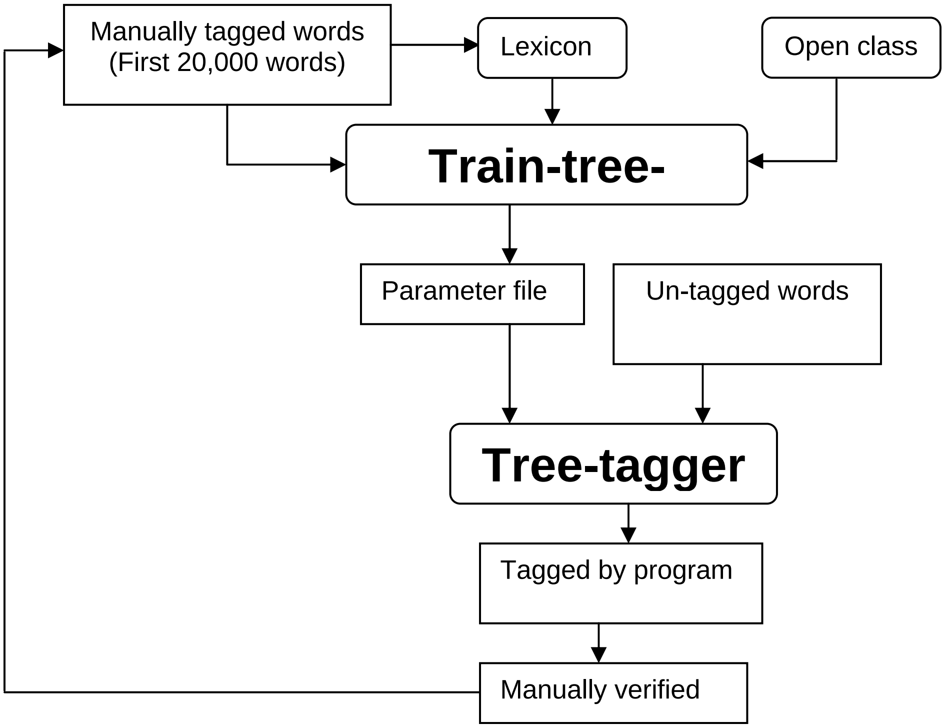


Figure 7: The process model of training corpus

We started from the first 20,000 tagged words to generate a Lexicon file. We got a parameter file provided by train-tree-tagger application. We used the parameter file to tag next 10, 000 untagged words by using Tree-tagger application. We got a 10,000 tagged words after the execution of the tree-tagger. Then we corrected those 10,000 tagged words manually and checked the accuracy. The table below is a result from our experiment.

Results

Target data	Tagged Data	Lexicon	Words to tag	Accuracy
31,230	20,414 (manually tagged)	2,531	10,816	90.39%
41,413	31,230	3,362	10,183	94.52%
51,835	41,413	3,854	10,422	96.61%
62,522	51,835	4,259	10,687	86.26%
73,206	62,522	5,233	10,684	92.50%
78,492	73,206	5,934	5,286	98.80%

Table 2: Result of semi-automatic tagging

3 Conclusion

To obtain higher accuracy, large trained data is required. In addition to that, data should be taken from different domains. For our research, we have used the Decision Trees model to tag the data. The application can not tag the raw data which contains the text written continuously without delimiter (Word marker), every word must be separated by space character or new line. That concludes that an update of Khmer word segmentation is required with higher accuracy results. Beside that new words have been created every day such as name of people, name of organization, or word translated from foreign languages etc. This is another reason to expand trained data to get higher accuracy and we plan to train the corpus up to 150,000 words.

References

[1]. Saleh Yousef Al-Hudail. A Hidden Markov Model Based POS Tagger for Arabic.
[2]. TreeTagger - a language independent part-of-speech tagger.

http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html, 02/09/2008.

[3]. Dictionnaire Cambodgien, (5th ed.). 1967. Institute of Buddhist. Cambodia Printhouse ( ), 165.

[4]. Chena NOU. Khmer Part-of-Speech Tagging. Global Information and Telecommunication Studies, WASEDA UNIVERSITY.

[5]. http://www.smo.uhi.ac.uk/~oduibhin/oideasra/interfaces /winttinterface.htm, 02/09/2008.

[6]. http://www.licadho.org, 02/09/2008.

[7]. http://www.krtrial.info, 02/09/2008.

[8]. Khin Sok. 2004. Khmer Language Grammar ( ). Royal Academy of Cambodia.

[9]. Khmer Grammar Study Program for Freshmen. 2005. Institute of Foreign Languages (IFL), Department of English.