Comparative Study of Naïve Bayesian Classifier and Transformation-Based Learning for Myanmar Function Tagging

Win Win Thant, Tin Myat Htwe and Ni Lar Thein

Abstract— This paper describes the use of two machine learning techniques, Naive Bayesian classifier (NB) and transformation-based learning (TBL), to address the task of assigning function tags to Myanmar sentences. Function tagging is a process of assigning syntactic categories like subject, object, time and location to each word in the text document. It is an important step in Natural Language Processing. Function tags can help to improve the performance of Myanmar to English machine translation system. In this paper, we present a comparison of two methods in our experiments. The results showed that TBL was better and outperformed NB and there was a slight difference between the results.

Keywords— Naïve Bayesian, transformation-based learning, function tagging, Myanmar sentences

I. Introduction

FUNCTION tagging is one of the essential steps in Myanmar to English machine translation system. The labels such as subject, object, time, etc. are named as function tags. By function, it is meant that action or state which a sentence describes. The system operates at word-level with the assumption that input sentences are pre-segmented, pos-tagged and chunked. Function tags of Myanmar language is defined because these tags are useful for any application trying to follow the thread of the text –they find the 'who does what' of each clause, which can be useful to gain information about the situation or to learn more about the behavior of words in the sentence [2].

Myanmar is a morphologically rich language with agglutinative nature. Being agglutinative language most of the words are postpositionally inflected with various grammatical features. The language text is challenging because of the problems like ambiguity and inefficiency. Also the interpretation of natural language text depends on context based techniques. It is considered to be an important intermediate stage for semantic analysis in natural language processing (NLP) application such as information retrieval (IR), information extraction (IE) and question answering (QA).

In this paper, we compare NB and TBL for function tagging of Myanmar language [8] [9]. The manually annotated tagged corpus of about 3000 sentences is used as training and testing data. Naive Bayesian classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. This assumption is called class conditional independence. An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the variable values necessary for classification.

Transformation-based learning is one of the most successful rule-based machine learning algorithms. It is a flexible and powerful method which is easily extended to various tasks and domains, and it has been applied to a wide variety of NLP tasks, including part- of- speech tagging, parsing and phrase chunking. .

II. Myanmar Language

The Myanmar language, Burmese, belongs to the Tibeto-Myanmar language group of the Sino-Tibetan family. Like Thai, Vietnamese, and Mandarin, Myanmar (Burmese) is a tonal language. It is the official language of Myanmar (Burma).

A. Structures of Myanmar Language

Unlike English language Myanmar is syntax of relatively free word order language [5]. This can be easily illustrated with the example "သူသည် စာအုပ်ကို စားပွဲပေါ်တွင် ထားသည်။" (He places the book on the table) as shown in table 1.

TABLE 1
WORD ORDER IN MYANMAR LANGUAGE

Case	Myanmar Sentences	Word order
1	သူ စာအုပ်ကို စားပွဲပေါ်တွင် ထားသည်။	(Subj-Obj-Pla-Verb)
2	သူ စားပွဲပေါ်တွင် စာအုပ်ကို ထားသည်။	(Subj-Pla-Obj-Verb)
3	စာအုပ်ကို စားပွဲပေါ်တွင် သူ ထားသည်။	(Obj-Pla-Subj-Verb)
4	စာအုပ်ကို သူ စားပွဲပေါ်တွင် ထားသည်။	(Obj-Subj-Pla-Verb)
5	စားပွဲပေါ်တွင် သူ စာအုပ်ကို ထားသည်။	(Pla-Subj-Obj-Verb)
6	စားပွဲပေါ်တွင် စာအုပ်ကို သူ ထားသည်။	(Pla-Obj-Subj-Verb)

In all the cases, subject is သူ (He), object is စာအုပ်ကို (the book), place is စားပွဲပေါ်တွင် (on the table) and verb is ထားသည်

W. W. Thant is with Natural Language Processing Laboratory, University of Computer Studies, Yangon, Myanmar (e-mail:winwinthant@gmail.com).

T. M. Htwe is with Natural Language Processing Laboratory, University of Computer Studies, Yangon, Myanmar (e-mail: tinmyathtwe@gmail.com).

N. L. Thein is with Natural Language Processing Laboratory, University of Computer Studies, Yangon, Myanmar.

(places). From the above example, it is clear that phrase order does not determine the functional structure in Myanmar language and permits scrambling. Myanmar language follows Subject-Object-Verb orders in contradiction with English language.

B. Complexity and Ambiguity

The highly agglutinative languages like Myanmar, words get inflected. Many times we need to depend on syntactic function or context to decide upon whether the particular word is a noun or adjective or adverb or postposition. This leads to the complexity in Myanmar function tagging. A postposition may be categorized as subject, direction or simile.

The postpositional marker (PPM) of subject phrases or object phrases or time phrase or place phrase or direction phrase can be omitted. For example: She goes to school.

သူမ - ကျောင်း - **သို့** - သွားသည်။

She - school - **to** - goes

(or)

သူမ - ကျောင်း - သွားသည်။

She - school - goes

The word သို့ (to) can be omitted and it is valid in Myanmar sentence. The omission of PPM can affect the accuracy of function tagging process.

Natural languages give rise to functional ambiguity that words may have in different positions, i.e. one word is in general connected with different tags in the lexicon. In other words, words match more than one functional category depending on the context that they appear in sentences. For example, if it is considered the word ကလေး 'baby' in the following two sentences,

ကလေးသည် ချစ်ဖို့ကောင်းသည်။(The baby is cute.)

ကျွန်ုပ်သည် **ကလေး**ကို ချစ်သည်။ (I love the baby.)

In the first sentence, ကလေး 'the baby' takes the position of subject. But in the second sentence, it is an object.

Besides ambiguity of words, inflection and derivation of the language are other reasons that make natural language understanding very complex. For instance, တွင် 'in' contains the following inflection in Myanmar language.

သူ မနက်**တွင်** ဈေးသို့ သွားသည်။(He goes to the market in the morning).

သူ ရန်ကုန်**တွင်** နေသည်။(He lives in Yangon.)

In the first sentence, တွင် 'in' takes the postpositional marker (PPM) of time. But in the second sentence, it is the postpositional marker (PPM) of place.

III. Overview Of Function Tagging

In the task of function tagging, we use the output of morphological analyzer which tags the function of Myanmar sentences with correct segmentation, POS (part-of-speech) tagging [6] and chunking information [7].

There are many chunks in a sentence such as NC (noun chunk), PPC (postpositional chunk), JC (adjectival chunk), RC (adverbial chunk), CC (conjunctional chunk), SFC (sentence final chunk) and VC (verb chunk).

A chunk contains Myanmar words and POS tags with features. There are 21 POS tags and over 80 features.

For example: သူသည် စာအုပ်ကို ဆရာ့အား ပေးသည် (Myanmar)

He gives the book to the teacher. (English)

The input segmented sentence with POS tag and chunk for the above Myanmar sentence is "NC[သူ/PRN.Possessive]%PPC[သည်/PPM.Subj]%NC[စာအုပ်/NN.Object]%PPC[ကို/PPM.Obj]%NC[ဆရာ့/NN.Possessive]%PPC[အား/PPM.Accept]%VC[ပေး/VB.Common]%SFC[သည်/SF.Declarative]".

The proposed function tagset for Myanmar language has 56 tags [9]. The tags in the proposed tagset are described by the below table 2 with 8 different categories.

TABLE 1I
CATEGORIES OF FUNCTION TAGS

Category	Function Tags
Normal	Subj,Obj,Tim,Pla,Dir
Postposition	PSubj,PObj,PIobj,PPla,PDir,PLea,PArr, PConPla,PExt,PTim,PConTim,PTimSta, PTimEnd,PSim,PCom,POwn,PPcomplO ,PUse,PCau,PAim
Postpositional Marker	SubjP,ObjP,IobjP,PlaP,DirP,LeaP,ArrP, ConPlaP,ExtP,TimP,ConTimP,TimStaP, TimEndP,SimP,ComP,OwnP,PcomplOP ,UseP,CauP,AimP
Conjunction	CCS,CCM,CCC,CCP,CCA
Complement	PcomplS,PcomplO
Verb	Active
Sentence's final marker	Dec,Int
Adjective	Ada

The output Myanmar function tagged sentence is "PSubj[သူ]%SubjP[သည်]%PObj[စာA္ပ်]%ObjP[ကို]%PIobj[ဆရာ့]%IobjP[Aား]%Active[ပေး]%Dec[သည်]".

TABLE 1II
FUNCTION TAGGING

Myanmar	English	Chunk	POS and its features	Function tags
သူ	He	NC	PRN.Person	PSubj
သည်	-	PPC	PPM.Subj	SubjP
စာA္ပ်	the book	NC	NN.Object	PObj
ကို	-	PPC	PPM.Obj	ObjP
ဆရာ့	the teacher	NC	NN.Possessive	PIobj
Aား	to	PPC	PPM.Accept	IobjP
ပေး	gives	VC	VB.Common	Active
သည်	-	SFC	SF.Declarative	Dec

The system architecture is shown in the following figure1.

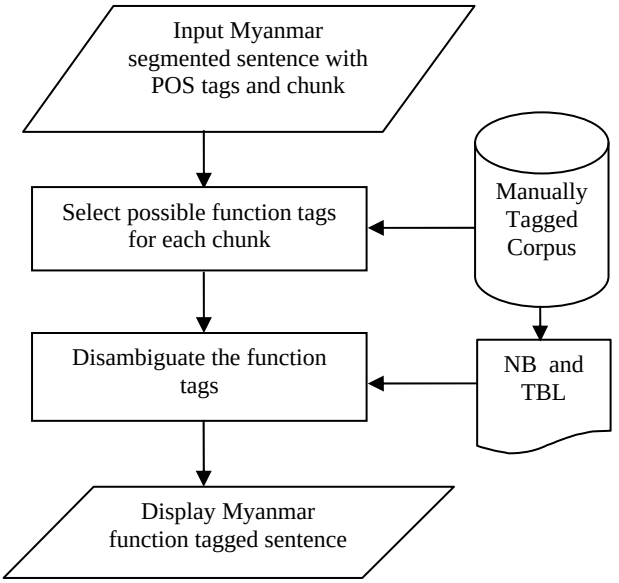

Fig. 1: System Architecture

IV. Naïve Bayesian Classifier

The Naïve Bayes classifier is a classification method that is used for categorical data based on applying Bayes' theorem. By the classical Bayes approach, for a record to be classified, the categories of the predictor variables are noted and the record is classified according to the most frequent class among the same values of those predictor variables in the training set. A rigorous application of the Bayes theorem would require availability of all possible combinations of the values of the predictor variables:

$$p(G, F_1, F_2,\ldots,F_n) = p(G)\, p(F_1, F_2,\ldots,F_n|G) = p(G)p(F_1|G)p(F_2,\ldots\ldots,F_n|G, F_1) = p(G)p(F_1|G)p(F_2|G, F_1)p(F_3|G, F_1, F_2) \ldots p(F_n|C, F_1, F_2, \ldots F_{n-1})$$

When the number of variables is large enough, this requires a training set of unrealistically large size.

The Naïve Bayes method overcomes this practical limitation of the rigorous Bayes approach to classification. The major idea of it is to use the assumption that predictor variables are independent random variables. This makes it possible to compute probabilities required by the Bayes formula from a relatively small training set:

$$p(G, F_1, F_2, \ldots, F_n) = p(G)p(F_1|G)p(F_2|G) \ldots p(F_n|G) = p(G)p(F_i|G)$$

So, Naïve Bayes assumption could be presented by formula.

$$p(G|F_1, F_2, \ldots F_n)=1/Z\ p(G) \prod_{i=1}^{n} p(F_i|G)$$

where Z is a scaling factor dependent only on F1, F2, . . . , Fn , i.e., a constant if the values of the feature variables are known.

To sum up, we can say that in spite of its naive design and over-simplified assumption, Naïve Bayes classifiers often work much better in many complex real-world situations than one might expect [4].

w	a word
c	category of a word
pc	POS tag word with category
$t_1, t_2 \ldots t_k$	possible tags of the word with category
$n_1, n_2 \ldots n_j$	possible tags of the next word with category
$C(t_k,c)$	the number of occurrences of t_k followed by c
C(c)	the number of occurrences of c in the training set
$C(n_j,t_k)$	the number of occurrences of n_j followed by t_k
$C(t_k)$	the number of occurrences of t_k in the training set

Fig 2: Notational conventions for function tagging

comment: Training
for a pc of w **do**
  for all tags t_k of pc **do**
    $P(t_k|pc) = C(t_k,pc)/C(pc)$
  end
end
for all tags t_k of wc **do**
  for all tags n_j of wc **do**
    $P(n_j|t_k) = C(n_j,t_k)/C(t_k)$
  end
end
comment: Disambiguation
for all tags t_k of wc **do**
  score (t_k)=log $P(t_k)$
  for all tags n_j in the next wc **do**
    score (t_k)=score(t_k)+log $P(n_j|t_k)$
  end
end
choose t=arg max t_k score (t_k)

Fig 3: Naïve Bayesian classification for function tags disambiguation

V. Transformation-Based Learning

TBL is developed by Brill [1995] for POS tagging [3]. It is also used for other NLP areas, such as text chunking, prepositional phrase attachment, parsing, dialogue act tagging and named entity recognition. Figure 4 illustrates the learning process. First, un-annotated text is passed through an initial-state annotator. The initial-state annotator can range in complexity from assigning random structure to assigning the output of a sophisticated manually created annotator. Once text has been passed through the initial-state annotator, it is then compared to the truth as specified in a manually annotated corpus, and transformations are learned that can be applied to the output of the initial state annotator to make it better resemble the truth. To define a specific application of transformation-based learning, one must specify the following:

1. The initial state annotator.
2. The space of transformations the learner is allowed to examine.
3. The scoring function for comparing the corpus to the truth and choosing a transformation.

Once an ordered list of transformations is learned, new text can be annotated by first applying the initial state annotator to it and then applying each of the learned transformations, in order.

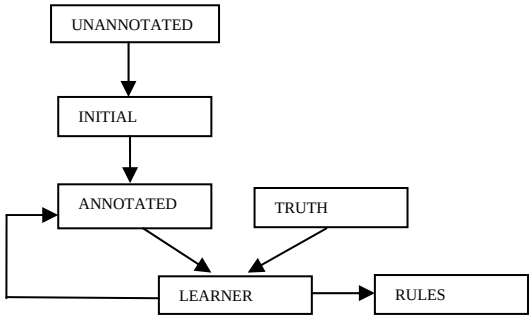

Fig. 4: Transformation-based learning

According to the TBL function tagging technique, each word is labeled with its most likely (frequent) tag. Frequency information is stored in a lexicon, which has been constructed during the training phase and contains all the words in the training corpus associated with their most frequent function tag, as was measured from the training corpus.

Once the assignment of initial function tags has been completed for all the words in the corpus, an ordered sequence of lexical rules is applied to the corpus. Each one of these lexical rules operates only on a single word, and its preconditions consider only morphological cues. For example, a typical lexical rule has the following form:

IF (the lexical item of its previous word is "ကို" the lexical item of its next word is "လုပ်သည်")
THEN Classify current word as an object complement

For example: U Hla makes the gold a ring.
ဦးလှ -သည် -ရွှေ -ကို -လက်စွပ် -လုပ် -သည်။
PSubj-SubjP-PObj -ObjP -PcomplO-Active-Dec

TABLE IV.
TRANSFORMATION WITH LEXICAL RULES

Source Tag	Target Tag	Transformation
Subj	PcomplS	The sentence contains "သည်" and the lexical item of its next word is "နက်သည်" (or) "ရှည်သည်" (or) "မြင့်သည်"
PIobj	PcomplO	the lexical item of its previous word is "ကို" the lexical item of its next word is "တင်မြောက် သည်" (or) "ချက်သည်" (or) "ထင်သည်"
PcomplS	Obj	the lexical item of its previous word is "ကို" the lexical item of its next word is "ပေးသည်" (or) "ပြောသည်"

After the application of lexical rules has been completed, an ordered list of contextual rules is applied to the corpus. Each of these rules can change the tag assigned to a word according to the context in which the word appears.

All of the resources needed (the lexicon, the lexical and the contextual rule set) are created during the training phase. It involves two training stages. In the first stage, rules are learned from the training corpus to assign function tags. These rules operate on word types. The tag chosen for each word holds for all occurrences of the word in the corpus. The output of this phase is a lexicon, containing every word in the training corpus associated with its most frequent tag, and an ordered list of lexical transformation rules that are based on morphological information. A lexical rule template describes all the possible forms of the rules that can be produced. In the second training phase, rules are learned to use contextual cues to improve tagging accuracy. Example of such rule is the following:

IF (current word tagged CCC AND the word two after tagged as SubjP)
THEN Tag previous and following word as PSubj

For example: Ma Ma and Hla Hla are clever.
မမ -နှင့် -လှလှ -သည် -လိမ္မာ -သည်
PSubj -CCC -PSubj -SubjP -Ada -Dec

TABLE V
TRANSFORMATION WITH CONTEXTUAL RULES

Source Tag	Target Tag	Transformation
PUse	POwn	the next tag is OwnP
PSubj	PObj	the second tag is CCC and the fourth tag is ObjP
Obj	PcomplS	the second tag is CCC, the third tag is PcomplS and the fourth tag is Active
Obj	Subj	the second tag is CCC and the fourth tag is CCC and the fifth tag is Active

These rules operate on individual word tokens. The output of this phase is also an ordered list of transformation rules that are based on contextual information such as the current tags of the surrounding words or the surrounding words themselves. A contextual rule template describes all possible contextual rules that can be derived in this training phase. There are 192 rules (including lexical and contextual rules) [9].

VI. CROSS VALIDATION EXPERIMENTS

In order to derive a robust and unbiased estimate of the method's performance, we used 10-fold cross validation at each individual experiment. In 10-fold cross-validation, 90% of the data is used for training and the remaining 10% for testing, and this procedure is repeated ten times, using a different 10% for testing each time. There are nearly 3000 training sentences. The overall accuracy was calculated by taking the mean of the ten measures for both NB classifier and TBL and shown in table 6. After conducting the k- fold cross validation process with NB and TBL, ten results were achieved.

TABLE VI
10-FOLD CROSS VALIDATION RESULTS FOR NB AND TBL

	Naïve Bayesian (NB)	Transformation-based learning (TBL)
Average Accuracy (%)	89.13	93.52

As can be seen in table above NB was correct 89.13% while NB was correct 93.52% accuracy.TBL performs better on languages with very large structural and morphological differences.

VII. Measures of Performance

The performance of the two methods used in this study were evaluated using the following accuracy, sensitivity, specificity measures and as follows,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

where TP , TN , FP and FN denotes true positives, true negatives, false positives and false negatives respectively.

TABLE VII
THE ACCURACY, SENSITIVITY AND SPECIFICITY FOR NB AND TBL

	Naïve Bayesian (NB)	Transformation-based learning (TBL)
Accuracy (%)	89.13	93.52
Sensitivity (%)	82.2	95.3
Specificity (%)	94.8	90.41

As can be seen from the table above, TBL outperformed the other method. Some conclusions can be made from this experiment:

1. In NB function tagging, we found that the lack of postpositional makers in the input sentences affect on the performance. As we increase the number of postpositional markers that are used in the evaluation, the accuracy will be increased.
2. In TBL function tagging, the existing rule templates, are adequate in coping with the Myanmar language morphology. But the computational cost can be high.

VIII. Error Analysis

The errors can be occurred by POS tagging results and data sparseness. The following table shows some erroneous tags produced by the function tagging process of two methods along with the correct tag and its error rate.

TABLE VIII
CONFUSION OF FUNCTION TAGS

Output tag	**Correct tag**	**Error rate (%)**	
		NB	**TBL**
Subj	PcomplS	52	17
Obj	PcomplS	11	6
PPla	Pla	9	0
Tim	PTim	4	0
Obj	Subj	13	21

If the sentences are written grammatically, the accuracy is nearly the same for two methods. However, some postpositional markers (PPM) are optional in Myanmar language. If the sentence is constructed with optional PPM, TBL function tagging has lower error rate than NB tagging. It can be observed that most errors of NB method come from training corpus coverage problems and conditional independence assumption may not be reasonable.

IX. Conclusion

The main objective of this paper is to investigate the performance of function tagging by comparing its results with two methods namely the Naïve Bayes classifier and transformation-based learning. The comparisons were made using the corpus of about 3000 sentences. The result of the comparison between the two methods showed that TBL outperforms NB in function tagging. In this paper 10-fold cross validation was used to minimize the bias associated with random sampling of training and test data. The comparison results showed that Naïve Bayes Classifier has limited usefulness and TBL is better and more useful in function tagging than NB. No previous comparison between these two methods has been previously published for Myanmar language.

References

[1] D. Blaheta, " Function tagging ". Ph.D. Dissertation, Brown University. Advisor-Eugene Charniak. 2003

[2] E. Charniak, " A maximum-entropy inspired parser ". Technical Report CS-99-12, Brown University, August,1999.

[3] E. Brill, "A Simple Rule-Based Part of Speech Tagger", In Proceedings of the Third Conference on Applied Computational Linguistics (ACL), Trento, Italy, 1992.

[4] L. Versteegen, " The Simple Bayesian Classifier as a Classification Algorithm".1999

[5] Myanmar Thudda, vol. 1 to 5 in Bur-Myan, Text-book Committee, Basic Edu., Min. of Edu., Myanmar, ca. 1986.

[6] P.H. Myint, "Lexicalized HMM-based Part-of-Speech tagger for Myanmar Language". In Proceedings of the tenth International Conference on Computer Applications, Yangon, Myanmar, 2012.

[7] P.H. Myint, "Chunk Tagged Corpus Creation for Myanmar Language". In Proceedings of the ninth International Conference on Computer Applications, Yangon, Myanmar, 2011.

[8] W.W. Thant, T. M. Htwe, and N. L. Thein, "Function Tagging for Myanmar Language".International Journal of Computer Applications (IJCA), Volume 26, Number 2, July 2011.

[9] W.W. Thant, T. M. Htwe, and N. L. Thein, "Resolving Function Tagging Ambiguity in the Myanmar Language using Transformation-based Learning". In Proceedings of the tenth International Conference on Computer Applications, Yangon, Myanmar, 2012.