Resolving Function Tagging Ambiguity in the Myanmar Language Using Transformation-Based Learning

Win Win Thant, Tin Myat Htwe, Ni Lar Thein
University of Computer Studies, Yangon
winwinthant@gmail.com

Abstract

This article investigates the use of transformation-based learning for resolving function tagging ambiguity in the Myanmar language. Function tagger plays an important role in natural language applications like speech recognition, natural language parsing, information retrieval and information extraction. In this paper, the function tagger [12] learns rules to correct its mistakes. A set of rule templates is used to create specific rules. At initial stage of function tagging for Myanmar, it is trained with a very limited resource of annotated corpus. The performance can be maximized with a substantial amount of annotated corpus. The function tagset has been developed for training and testing the function tagger. The present tagset consists of 56 tags. A corpus size of about three thousand sentences is used for training and testing the accuracy of the function tagger. The tagger learned 192 rules (including lexical and contextual rules) and achieved 93% accuracy.

1. Introduction

Transformation-based learning (TBL) 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.

The aim of the work presented in this paper is to study the performance of transformation based learning for resolving function tagging ambiguity in the Myanmar language. Function tagging aims to assign unambiguous tags to words in electronic documents, according to the function in which they belong, i.e., subject, object, time, verb etc [1]. 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.

To handle such complexities and use computers to understand and manipulate natural language text, there are various research attempts under investigation. Most of these researches have been developed for popular languages like English. However, there are few studies for Myanmar language. So, the study presents the resolving of function tagging ambiguity for Myanmar language. The function tagger implementation [12] is used for the experiments.

2. Related Work

Pascale Fung et al., [8] described a character-based statistical parser, which gave the best performance to-date on the Chinese treebank data. They augmented an existing maximum entropy parser with transformation-based learning, creating a parser that can operate at the character level. Since the word segmenter was the first component in the parser, its poor performance creates a big problem for their parser. The solution to the problem was to leverage another successful machine-learning algorithm, transformation-based learning (TBL). In their approach, the task of word segmentation could be easily mapped to a tagging problem in a similar way to that pioneered by Ramshaw and Marcus [10] for English text-chunking and the task of the tagger was to use lexical and syntactic features of the word to determine the most likely tag for that particular use of the word in the given sentence. An intermediate step between POS tagging and full parsing was text-chunking, which was dividing a sentence into syntactically correlated segments called chunks, or base phrases. The parse tree was constructed recursively in a bottom-up fashion from left to right, until the root has been reached. They presented experiments that show that their parser achieved results that were close to those achievable under perfect word segmentation conditions. They also explained that their system also outperformed a parser that is based on maximum entropy only.

Eric Brill and spoken language systems group [2] described a number of extensions to the rule-based tagger. They explained that no relationships between words were directly captured in stochastic taggers and many useful relationships, such as that between a word and the previous word, or between a tag and the following word, were not directly captured by Markov-model based taggers. To remedy the problem, the transformation-based tagger was extended by adding contextual transformations that could make reference to words as well as part of speech tags. Next, they showed a rule-based approach to tagging unknown words. To try to improve upon unknown word tagging accuracy, they built a transformation-based learner to learn rules for more accurately guessing the most likely tag for words not seen in the training corpus. Finally, they showed how the tagger can be extended into a k-best tagger, where multiple tags can be assigned to words in some cases of uncertainty.

3. Function Tagset for Myanmar

The proposed tagset for Myanmar language has 56 tags where there are 20 tags for postpositions, 20 tags for postpositional markers, 5 tags for normal, 5 tags for conjunctions, 1 for verb, 2 for sentence's final marker, and 1 for each subject complement , object complement and adjective [9]. The tags in the proposed tagset are described by the below table 1 with 8 different types.

Table 1. Function tagset

Type	Tag	Description
Normal	Subj	Subject
	Obj	Object
	Tim	Time
	Pla	Place
	Dir	Direction
Postposition	PSubj	Subject
	PObj	Object
	PIobj	Indirect Object
	PPla	Place
	PDir	Direction
	PLea	Leave
	PArr	Arrive
	PConPla	Continuous Place
	PTim	Time
	PConTim	Continuous Time
	PTimSta	Time Start
	PTimEnd	Time End
	PExt	Extract
	PSim	Similie
	PCom	Compare
	POwn	Own
	PPcomplO	Object Complement
	PUse	Use
	PCau	Cause
	PAim	Aim
Postpositional Marker	SubjP	PPM of Subject
	ObjP	PPM of Object
	IobjP	PPM of Indirect Object
	PlaP	PPM of Place
	DirP	PPM of Direction
	LeaP	PPM of Leave
	ArrP	PPM of Arrive
	ConPlaP	PPM of Continuous Place
	TimP	PPM of Time
	ConTimP	PPM of Continuous Time
	TimStaP	PPM of Time Start
	TimEndP	PPM of Time End
	ExtP	PPM of Extract
	SimP	PPM of Similie
	ComP	PPM of Compare
	OwnP	PPM of Own
	PcomplOP	PPM of Object Complement
	UseP	PPM of Use
	CauP	PPM of Cause
	AimP	PPM of Aim
Conjunction	CCS	Join the sentences
	CCM	Join the meanings
	CCC	Join the words
	CCP	Join with particles
	CCA	Join as an adjective
Complement	PcomplS	Subject Complement
	PcomplO	Object Complement
Verb	Active	Verb
Sentence's final marker	Dec	Declarative
	Int	Interrogative
Adjective	Ada	Adjective

4. Corpus Training

Supervised learning is used for most function tagging process, meaning that function tagger need to be trained on a manually tagged corpus. It is started by tagging a small part of the corpus completely by hand, and trained a first version of the function tagger on this material. The tagger is run on a different part of the corpus, manually corrected the tagger output, and added the corrected material to the training corpus.

The function tagger is re-trained on the bigger training corpus, so that it could be applied to yet another part of the corpus, yielding slightly better results than it did the first time. By repeating this process, an increasingly better tagger is obtained simultaneously. An increasingly larger manually corrected corpus now consists of nearly about 3000 sentences; it made from the domains described in table 3. Obviously, this is not a balanced corpus; it was indeed chosen because of its easy accessibility. This corpus is used to train the function tagger and to test them.

Table 2. Textual domains in the corpus

Domain	Share (%)
Literature	60%
History	35%
Sports	10%
Culture	5%

5. 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 [10], prepositional phrase attachment [4], parsing [5], dialogue act tagging [11] and named entity recognition [7]. Figure 1 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[6]. 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.

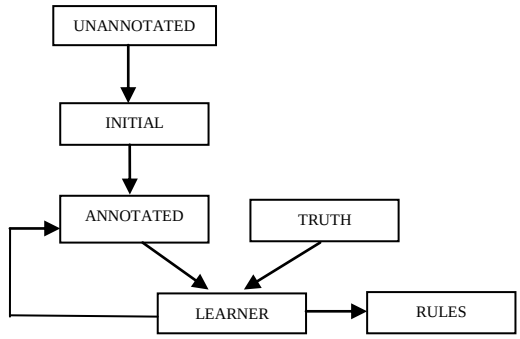

Figure 1. Transformation-based learning

6. TBL Function Tagging

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 sentence contains မှာ and the lexical item of its next word is "ရှိသည်") THEN
Classify current word as a subject complement

For example: He has a car.
သူ -မှာ -ကားတစ်စီး -ရှိ -သည်။
PSubj -SubjP -PcomplS -Active -Dec

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 3. 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. The environment used for changing a word tag consists of the words and tags within a window of four words, including the word under examination.

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. Examples of such rule are the followings:

IF (the next tag is SimP) THEN
Tag current word as PSim

For example: He is brave as a lion.
သူ -ခြင်္သေ့ -ကဲ့သို့ -ရဲရင့် -သည်။
Subj -PSim -SimP -Ada -Dec

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 4. 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 also describes all possible contextual rules that can be derived in this training phase.

7. Cross-Validation Experiments

A training corpus where each chunk is accompanied by its manually disambiguated function tag and a test corpus consisting only of chunk are required. Since the development of the training set continues, N-fold Cross Validation (CV) process is used for determining overall accuracy, where N=10. Processing of each fold consists of the following steps:

1. From randomized main tagged corpus select 1/N-th number of sentences. Remove all function tags to create the Independent Set (IS).
2. Use the remaining data as the training set.
3. Train the function tagger on this data.
4. Upon completion use the IS for tests to determine automatic function tagging accuracy.

The size of the complete corpus is about 3000 sentences and is organized in a single file, where each line corresponds to a single sentence. The average length of words in each simple sentence is 7 and the average length of words in each complex sentence is 12. There are about 35000 words in the corpus. Each word of this corpus has been tagged using an extremely rich, full-featured tagset for the Myanmar language. A new version of the original corpus was created, where each function tag was mapped onto the tagset. Then, the sentences of the newly created corpus were shuffled using a randomizer. The reason for doing so is the structure of the corpus, which is composed of small sentence groups that belong to the same domain.

In this test case, the function tagger is evaluated using 10-fold cross validation over different corpus sizes. The results are shown in Figure 2. The error bar corresponds to the standard deviation of the average accuracy over the six runs of the 10-fold cross validation. As expected, function tagging accuracy increases as the corpus size increases. Accuracy seems to stabilize around 93%. This is mainly due to the tagging difficulties for the Myanmar language, such as morphological complexity and lack of postpositional marker.

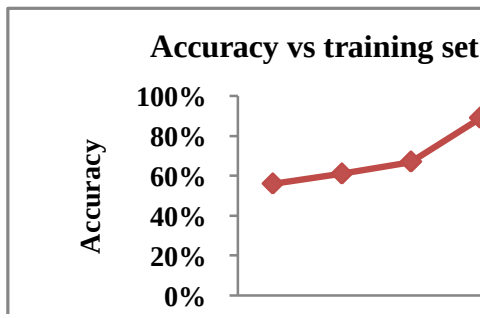

Figure 2. Function tagging accuracy versus corpus size. All results are from the 10-CV tests.

Another point of interest is the examination of the number of learned rules. Usually, high performance of a learning task when combined with a small rule set size indicates robustness of the learning task. Large numbers of learned rules often indicate problems in the learning task. As the task of training the function tagger involves two different learning sub-tasks (learning lexical rules and learning contextual rules), we had the opportunity to examine them separately. As was explained previously, the lexical rules correspond to the morphology and the contextual rules to the grammatical and syntactic features of the language. Thus, by examining the sizes of lexical and contextual rule sets separately, any potential problems to the morphologic or the grammatical/syntactic properties of the language can be isolated.

The size of the learned rule set against corpus size is shown in Figure 3. The number of both

types of rule (lexical and contextual) increases almost linearly with the corpus size. The numbers of the lexical rules are much larger than the number of the contextual rules.

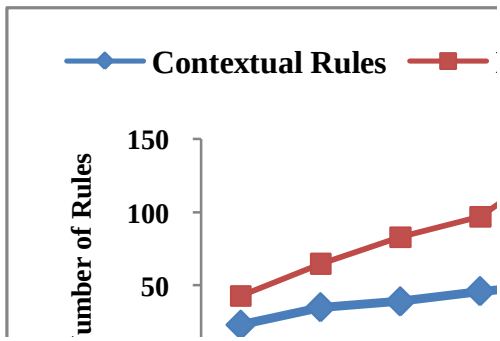

Figure 3. Number of lexical and contextual rules versus corpus size

8. Problematic Words and Tags

There are many problematic words and tags in Myanmar sentences. One of the problematic words is အိမ် 'house' (or) 'home', which may be subject, object, place, direction, or subject complement, making it hard to determine its correct category, as can be seen in table 5:

Table 5. Function tags for 'အိမ်'

Function tags	English	Myanmar
PcomplS	He has a **house**.	အိမ်
PDir	He returns **home** from the trip.	အိမ်
PPla	He is at **home**.	အိမ်
PSubj	My **house** is near the school.	အိမ်
PObj	He buys a **house**.	အိမ်

This is not rare in natural languages, and a large percentage of functions of words are ambiguous. The size of imbalanced corpus affects the tagging accuracy. For example, the word မနေ့က 'yesterday' is included many times in the corpus as the Tim function tag. Sometimes, this word may be other function tag depends on the sentence structure. For example:

Yesterday was Friday.
မနေ့က -သောကြာနေ့ -ဖြစ်ခဲ့ -သည်။
Subj -PcomplS -Active -Dec

He went to school yesterday.
သူ -မနေ့က -ကျောင်း -သွား ခဲ့ -သည်။
Subj -Tim -Dir -Active -Dec

Table 6 lists the words that are most often mistagged by the function tagger, along with the proportion of the total number of errors that these errors constitute.

Table 6. The five words that are most commonly mistagged in the cross- validation experiments

Word	Error (%)
အိမ်	13.1
မနေ့က	3.6
သူ	5.9
ခဲတံ	2.6
ရန်ကုန်	4.8

Increasing training corpus size is important to correct the function tagging errors. However, the errors can be reduced by using the balanced corpus. Table 7 lists the most common tag confusions made by the function tagger. The table shows the erroneous tag produced by the function tagger along with the correct tag and its error rate.

Table 7. The most common tag confusions made by the function tagger

Output tag	Correct tag	Error rate (%)
Subj	PcomplS	52
Obj	PcomplS	11
PPla	Pla	9
Tim	PTim	4
Obj	Subj	13

9. Conclusion

In the work presented here a popular machine learning technique, the transformation-based learning, has applied to the task of function tagging in the context of the Myanmar language. The function tagger is trained over relatively small-sized annotated corpus. The proposed system is still in its early stages and the training set is still expanding. Although a significant improvement have been noticed, the results show that the learning process discussed here is likely to converge slowly. Therefore, a large training set may be necessary for obtaining the high accuracy. Tests made on the training set show that the upper limit for the top test accuracy ranges between 92-93%. Although humans use more complex interpretation mechanisms the information available in text alone will not be sufficient to reach 100% accuracy. It would be very interesting, but also quite difficult, to find out the accuracy limit for humans on the same task. If the answer is significantly higher than 93% then adding more knowledge-based information to the tagging process should enable to remove some of the errors.

References

[1] Blaheta, D., and Johnson, M., "Assigning function tags to parsed text", In Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics, 234–240, 2000.

[2] Brill, E., "A Report of Recent Progress in Transformation-Based Error-Driven Learning", Proceedings of the 12th National Conference on Artificial Intelligence (AAAI), 722–727.

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

[4] Brill, E., "Automatic Grammar Induction and Parsing Free Text: A Transformation-Based Approach", In Proceedings of the 31st Meeting of the Association of Computational Linguistics, 1993.

[5] Brill, E., "A Corpus-Based Approach to Language Learning", PhD Dissertation, Department of Computer and Information Science, University of Pennsylvania, 1993.

[6] Brill, E., "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case study in Part of Speech Tagging", in Computational linguistics, vol. 21, no 4, p. 543-565, 1995.

[7] Day, D. et al., "Mixed-Initiative Development of Language Processing Systems", In Fifth Conference on Applied Natural Language Processing, pp.48–355, 1997.Association for Computational Linguistics. Online (access date: 2003-02-01)

[8] Fung, P. and Ngai, G. et al., "A Maximum-Entropy Chinese Parser Augmented by Transformation-Based Learning", In ACM Transactions on Asian Language. Processing, 3(2), 159-168, 2004.

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

[10] Ramshaw, L. and Marcus, M., "Text chunking using transformation-based learning", In Proceedings of the 3rd ACL Workshop on Very Large Corpora (Cambridge, MA, 1995).

[11] Samuel, K., "Lazy Transformation-Based Learning", In Proceedings of the 11th International Florida Artificial Intelligence Research Symposium Conference. Online (access date: 2003-01-27).

[12] Thant, W. W., Htwe, T. M. and Thein, N. L., "Function Tagging for Myanmar Language", International Journal of Computer Applications (IJCA), vol. 26, no. 2, p. 34-41, 2011.