A Constraint Grammar POS-Tagger for Tibetan

Edward Garrett
SOAS, University of London
eg15@soas.ac.uk

Nathan W. Hill
SOAS, University of London
nh36@soas.ac.uk

Abstract

This paper describes a rule-based part-of-speech tagger for Tibetan, implemented in Constraint Grammar and with rules operating over sequences of syllables rather than words.

1 A POS-tagger for Tibetan

In earlier work, we described a rule-based tagger for Classical Tibetan, implemented using regular expressions (Garrett et al., 2014). Since then, the rule tagger has kept moving: our grammatical understanding of Tibetan has evolved by reading and hand tagging 236,167 Tibetan words.[1]

The primary purpose of the rule-based tagger has been to speed up the hand tagging of texts. The output of a lexical tagger, assigning to each word all of its possible tags, is fed into the rule tagger, which then removes only those analyses precluded by the context. The result of this process is highly ambiguous, with an average of 1.3936 remaining tags per word. However, it is also highly accurate, with 99.8 percent of words receiving the correct tag (van Halteren, 1999). In consequence, the human can focus their efforts on determining the correct tag for words that remain ambiguous, without needing to worry about words that the rule tagger is sure about.

In the development of the tagger, no attempt was made to divide the corpus into separate training and test sets. To do so would have been counterproductive, as it would have required us to read and tag texts without learning from them. To the contrary, we have seized every chance to develop and further refine the rule set. It is a pleasing result and some measure of success that the tagger performs well when evaluated against the materials that inspired it.

Despite its initial promise and usefulness, the regular expressions tagger has been deprecated. It soon became evident that maintaining the rule set required a regular expressions wizard with a keen eye for slashes. Being both difficult to read and difficult to maintain, the rule set seemed an unlikely candidate for linguists to build on and continue to use in the future. An additional purely hypothetical concern was that the tagger might soon require rules that would exceed the expressive capacity of regular expressions.

These concerns, combined with the fortuitous discovery of a new framework, led us to translate the entire rule set into CG-3 (Bick & Didriksen, 2015), the latest version of Constraint Grammar. We translated our regex rules into CG rules with operators such as SELECT and REMOVE.

To give an example, in (1) we show the input to the tagger as a sequence of cohorts in CG-3 format. Each cohort consists of a surface form, shown within brackets inside quotes, followed by one or more readings. Each reading includes a lemma followed by one or more tags. Rules then apply to the input to remove impossible readings.

(1)
```
"<ཨ་ནེ་>"
        "ཨ་ནེ་" n.count
"<དང་>"
        "དང་" case.ass
        "དང་" cv.ass
        "དང་" v.invar
"<ཨ་ཁུ་>"
        "ཨ་ཁུ་" n.count
```

The word དང་ has three possible readings. In (1), དང་ is being used to coordinate two nouns, and so the correct reading is associative case or [case.ass]. Two separate rules remove [v.invar] and [cv.ass] as possible readings. A simplified version of the former rule is shown in (2).

(2)
```
REMOVE v.xxx (-1C n.xxx)
(0 ("<དང་>")) (1C n.xxx) ;
```

This rule removes [v.invar] from དང་ when it is sandwiched between two unambiguous (signified by C) nominals (n.xxx).

[1] Numbers in this section reflect a snapshot of the Classical Tibetan corpus as of 15 June, 2015.

The exercise of translating the rules into CG had no effect on the overall performance of the system, since it introduced no new rules or rule types. However, the exercise did put the rule tagger on a more secure footing for the future. Translated into CG, most of the rules can now be deciphered by linguists. Moreover, the general readability of CG means that linguists are now willing to take a stab at creating and modifying rules without the help of a technician.

2 Segmentation as syllable tagging

Tibetan orthography does not use whitespace or other means to mark the boundaries between words. However, the intersyllabic tsheg character (U+0F0B), resembling a dot, is used to mark the boundaries between orthographic syllables.[2] For example, consider the following sentence:

(3a) ངཡིས་མི་མང་པོ་བསད་
I killed many people.

(3b) ང་|p.pers
ཡིས་|case.agn
མི་|n.count
མང་པོ་|adj
བསད་|v.past

The correct segmentation for (3a) is shown in (3b), where each word appears on its own line, and the pipe character separates a word from its POS-tag. Only one word in this example consists of more than a single syllable.

Following Xue's (2003) general approach to Chinese word segmentation, Liu et al. (2011) propose that Tibetan word segmentation be recast as a syllable tagging problem. The task is then to tag each syllable according to its position in the word. We adopt their 6+2 tag set, which they say yields the best results given the average length of Tibetan words. We analyze (1) as:

(3c) ང་|S
ཡིས་|S
མི་|S
མང་|X
པོ་|E
བསད་|S

The S tag is given to syllables forming words on their own, while X and E mark the first and last syllables of disyllabic words. Longer words are marked with X-Y-E (trisyllabic), X-Y-Z-E (quadrisyllabic), and X-Y-Z-M*-E, with any number of M, for words of 5 or more syllables.

Two additional complex tags, SS and ES, are needed for the class of "abbreviated" syllables. These are situations of orthographic fusion where no intersyllabic tsheg separates the end of a word from the case marker or converb that follows it. Since such fusion only affects phonologically open syllables, whereas the same grammatical functions are indicated with different markers after closed syllables, we achieve consistency and avoid the unnecessary proliferation of POS-tags by treating such markers as their own words. (4) shows a form of the first-person pronoun with fused agentive case; because the syllable must be treated as two words, it is assigned the tag SS. Similarly in (5), meaning "of the many", because the genitive case marker must be pulled off from the word that precedes it to form its own word, the syllable is tagged ES instead of E.

(4) བས་ > ང|p.pers ས་|case.agn
བས་|SS

(5) མང་པོའི་ > མང་པོ་|adj འི་|case.gen
མང་|X
པོའི་|ES

It is important to remember that while the genitive marker shown in (5) only ever occurs in abbreviated syllables (and so only ever occurs in syllables tagged SS or ES), other abbreviated case markers and converbs look the same as natural word endings. For example, there are many possible analyses of the syllable མར་. It may continue an existing word (not shown), or be the beginning of a new word. The final ར་ may be the natural ending of the word, as in (6a), meaning "butter", or a case marker, as in (6b), meaning "down there".

(6a) མར་ > མར་|n.mass
མར་|S
(6b) མར་ > མ|d.dem ར་|case.term
མར་|SS

In summary, we use syllable tags to represent words as a sequence of tagged syllable tokens, as an alternative to joining syllables together into a sequence of word tokens. Special care must be taken when dealing with abbreviated syllables. As illustrated in (5), while such case markers and

[2] In this paper, henceforth, we refer to orthographic syllables with the term "syllable" . By doing so, we are not committing to analysing these units as syllables in the sense of phonological theory.

converbs are always counted as their own tokens in word-based tagging, they are fused with the preceding token in syllable-based tagging.

3 A syllable-based POS-tagger

In the next phase of the rule-based tagger, we modify the CG rules to operate over sequences of syllables rather than words. To do so, we change both the input and the rules themselves. Cohorts now become syllables instead of words. Syllable cohorts that belong to unambiguous words have only one reading, with one tag drawn from the 6+2 tagset and another drawn from the POS-tagset. Syllables belonging to ambiguous words will receive multiple readings, with each reading receiving the same 6+2 tag but a different POS-tag:[3]

(7)
```
"<ཨ་>"
        "ཨ་" X n.count
"<ནེ་>"
        "ནེ་" E n.count
"<དང་>"
        "དང་" S case.ass
        "དང་" S cv.ass
        "དང་" S v.invar
"<ཨ་>"
        "ཨ་" X n.count
"<ཀུ་>"
        "ཀུ་" E n.count
```

We recast rule (2) in syllabic terms, with only one difference from the original.

(8)
```
REMOVE v.xxx (-1C n.xxx)
(0 ("<དང་>") LINK T:IsWord)
(1C n.xxx) ;
```

Since the context word དང་ is monosyllabic, and since the POS-tag of a word is marked on all of its syllables, we can pretend that positions -1 and 1 are occupied by the words before and after དང་. In this and many other rules, we add a condition for monosyllabic wordhood.

(9)
```
TEMPLATE IsWord = 0C (S) ;
```

The template in (9) tests whether a syllable is an unambiguous monosyllabic word (and so tagged S) as opposed to being part of a word with the preceding or following syllable. Many rules, for example, target the syllable ས་, which is either negation [neg] or the noun "mother" [n.count]. Obviously, such rules should not apply to ས་ when it is part of a larger word such as བླ་ས་ "lama" [n.count].

Syllable-based rules become more interesting when they must manipulate multisyllabic words. For example, disyllabic nominals may be either verbal nouns or count nouns, but verbal nouns may not be followed by determiners. Simplifying somewhat by ignoring an exception to the rule, here is the original word-based rule:

(10)
```
REMOVE n.v.xxx (0 (n.count))
    (1C (d.plural)) ;
```

The syllable-based rule adds an extra condition when scanning for the determiner:

(11)
```
REMOVE n.v.xxx (0 (n.count))
    (T:NextInitial LINK 0C (d.plural)) ;
```

The template T:NextInitial ensures that the rule will correctly remove the impossible reading from both syllables of the nominal.

(12)
```
LIST Initial = S X SS ;
    TEMPLATE NextInitial=*1C Initial ;
```

Initials are defined as those syllables that can begin words. The template scans forward to the next syllable that can be an initial, and proceeds if that syllable is an initial on all of its readings. Since the syllables of a disyllabic nominal will be tagged X and E, the next initial for both is the first syllable of the word that follows.

A similar template, T:PrevFinal, scans left to grab the final syllable of the preceding word. Not unlike (11), (13) draws on the template in (14) to remove the tag [n.count] from both syllables of a verbal noun when it follows [case.term].

(13)
```
REMOVE (n.count) (T:PrevFinal
    LINK 0C (case.term)) (0 n.v.xxx) ;
```

(14)
```
LIST Final = S E SS ES ;
    TEMPLATE PrevFinal=*-1C Final ;
```

By linking conditions, we can scan more than one word in either direction. For example, the rule below selects the tag [d.det] for རང་ when it occurs in the context [n.count] [adj] རང་ མི་ འདུག་. Since Tibetan adjectives can be multisyllabic, it is necessary to seek past all syllables of the adjective to the final syllable of the preceding noun.

[3] This approach brings to mind Ng and Low's (2004) all-at-once character-based POS-tagger and segmenter for Chinese.

(15) SELECT (d.det) (-1C (adj) LINK T:PreviousFinal LINK 0C (n.count)) (0 ("<རང>") LINK T:IsWord) (1 ("<མི>") LINK T:IsWord) (2 ("<འདུག|?>"r) LINK T:IsWord) ;

As noted in the previous section, special care must be taken with abbreviated syllables. The genitive case marker, for instance, manifests as the abbreviated syllable འི, as in (5) above, or as one of several standalone syllables. Therefore, a rule such as the following which specifies ལས་ as a noun rather than ablative case if preceded by a genitive must accommodate both standalone (16) and abbreviated case (17).

(16) SELECT (n.count) (-1 ("<(གི|ཀྱི|གྱི)>"r) LINK T:IsWord) (0 ("<ལས>") LINK T:IsWord) ;

(17) SELECT (n.count) (T:PrevAbGen) (0 ("<ལས>") LINK T:IsWord) ;

(18) TEMPLATE IsAbGen = 0 ("<.+འི>"r) ; TEMPLATE PrevAbGen = T:PrevFinal LINK T:IsAbGen ;

In other cases, syllable-based tagging obviates the need for specific rules relating to abbreviated syllables. For example, (19) removes noun tags from ས་, provided that it is not preceded by end of sentence punctuation or by an intersyllabic tsheg. In addition to being a freestanding word meaning "earth", ས་ also marks agentive case when attached to open syllables, as in (4) above.

(19) REMOVE n.xxx (-1 tshegless) (0 (case.agn) LINK 0 ("<ས་?>"r)) ;

(20) SET shad = ("<[།༎]+>"r) ; SET tshegless = ("<.*[^་]>"r) - shad ;

No such rule is needed in the syllable-based rule tagger. If ས་ is attached to the preceding syllable, then that syllable will be tagged SS or ES, and the hypothesis that ས་ means "earth" will simply not arise.

4 Future directions

The syllable-based tagger adds complexity to the word-based tagger without improving its overall performance. So why use it?

In a traditional pipeline approach, tokenization or segmentation precedes part-of-speech tagging, with the output of the former process feeding the latter. This has the disadvantage that errors made at earlier stages in the pipeline propagate to later stages. The success of the pipeline is effectively limited by the quality of its initial component. So a Tibetan POS-tagger can only be as good as the word segmentation system that precedes it.

Another common limit of traditional pipelines is that different components often require that the data be represented in different ways. In practice, this can be an obstacle to component interaction. By bringing the data requirements of the word segmenter and the POS-tagger in line with each other, we hope to facilitate the development of more cooperative NLP components, including a joint approach to segmentation and tagging.[4]

References

Eckhard Bick and Tino Didriksen. 2015. CG-3 - Beyond classical constraint grammar. *Proceedings of the 20th Nordic Conference on Computational Linguistics (NODALIDA 2015)*, pages 31-39.

Edward Garrett, Nathan Hill and Abel Zadoks. 2014. A rule-based part-of-speech tagger for Classical Tibetan. *Himalayan Linguistics*, 13(1):9–57.

Hans van Halteren. 1999. Performance of taggers. In Hans van Halteren (ed.), *Syntactic Wordclass Tagging.* Springer: Netherlands, 81-94.

Huidan Liu, Minghua Nuo, Longlong Ma, Jian Wu, and Yeping He. 2011. Tibetan word segmentation as syllable tagging using conditional random field. *25th Pacific Asia Conference on Language, Information and Computation,* 168–177.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? *Proceedings of EMNLP.* Barcelona, Spain.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29-48.

[4] The Classical Tibetan corpus, as well as the various rule taggers described in this paper, are available on the following GitHub site: https://github.com/tibetan-nlp