Abstract

This paper describes a rule-based part-of-speech tagger for Tibetan which has been implemented using Constraint Grammar. The tagger resulted from translating an existing Tibetan part-of-speech tagger implemented using regular expressions. The advantages of the CG tagger over the regular expressions tagger advocate for further uses of CG within Tibetan NLP. The paper ends by considering the possible application of CG to the hard problem of Tibetan word segmentation.

1 A Regular Expressions Tagger

Garrett et al. (2014) describe a rule-based tagger for Classical Tibetan, implemented using regular expressions. Their system ingests pages of Tibetan text in horizontal format. In this format, a single space marks the boundary between words, and line breaks separate sentences. Each word consists of a word form followed by a tag, with the pipe character in between. Whitespace is not permitted within words. For example, the first three words of one page are:

མི་|neg བཤིག་|v.past ལ་|cv.loc

A lexical tagger consults a dictionary of word forms and assigns to each word all of the possible tags that it could have, with brackets surrounding possible tags, turning the first word of the above example into:

མི་|n.count|neg

Garrett et al.'s tagger (henceforth, GHZ) consists of more than 300 regex rules, applied one after the other, which assign or remove tags from a word based on its surrounding context. Each rule has an explanatory background, written in English, a precise rule statement, also written in English, and the formal rule itself, which consists of the match pattern and the replace string.

Several of the tagger's rules deal with the problem of distinguishing [neg] from [n.count], since mi can be either negation or the nominal 'person', while ma can be either negation or the nominal 'mother'. For example, the background for Rule #063 says: "Although ma most characteristically negates the past, in the prohibitive construction it negates the present. This fact allows certain examples of ma to be securely analyzed as the negation prefix rather than the noun 'mother.' More precisely, the rule says: "If ma is followed by an unambiguous present verb stem, which in turn is followed by a possible imperative converb (i.e. cig, Žig, šig), then assign [neg] to ma, and remove [d.indef] from cig, Žig, šig." The pattern and replace expressions for this rule are shown below:

**PATTERN:**

((?:^|\s)མ་|\S*\[neg\]|\S*\[v\.pres\]|\S*\[cv\.imp\])(?:\[d\.indef\])?\S*

**REPLACE:**

$1|neg \$2$3

A second example (Rule #057a) illustrates the removal of the tag [n.count] from mi in a specific expression: "In the phrases sems mi dgaḥ-ba and sīn་ mi dgaḥ-ba remove from mi the tag [n.count]."

**PATTERN:**

((?:^|\s)(?:སེམས་|sིང་)\S+\[n\.|\|\S*\[n\.|\|\S*\])

**REPLACE:**

$1$2

The GHZ tagger was evaluated along the two dimensions of accuracy and ambiguity (van Halteren, 1999). 99.1% of words were assigned the correct tag (accuracy), and the average word was left with 1.39 tags (ambiguity).

2 A Constraint Grammar Tagger

In our search for a part-of-speech tagger capable of scaling to tag a freely available digital Tibetan
corpus approaching one million syllables, including texts from various historical periods and genres, we evaluated and soon encountered difficulties using the GHZ tagger. Within short order, it became evident that updating and maintaining GHZ rules would require a regular expressions wizard with a keen eye for slashes. Those with the linguistic subject knowledge to write grammar rules for Tibetan were unlikely to also possess or wish to obtain the technical skills to write and maintain complex regular expressions. We found that while the GHZ rule statements were immediately accessible to linguists, the regular expressions were not.

To address these difficulties, we translated the GHZ tagger into Constraint Grammar, an exercise that brought manifest advantages to our project. Each rule of GHZ was translated into one or more CG rules. For example, Rule #063 above became the following two rules:

SELECT (neg) (0 ma) (1C (v.pres)) (2 (cv.imp)) ;

REMOVE (d.indef) (-2 ma) (-1C (v.pres)) (0 (cv.imp)) ;

The gains in readability and maintainability are immediately clear. Numbering words to the left and right of the context word (0) is intuitively appealing, and the use of transparent commands such as SELECT and REMOVE makes the basic purpose of each rule easy to grasp. While certain features of CG syntax still need to be learned explicitly, by and large the syntax is one that can be acquired by reading and adapting existing examples.

Our CG POS-tagger makes heavy use of the regular expressions flag, as for example in our restatement of GHZ Rule #057a. Using this flag, we were able to adapt without difficulty nearly all of the complex regular expressions found in GHZ.

REMOVE (n.count) (-l ("<ཅིག་( ི་)>"r)) (0 (<"ི་">)) (1 ("<ིདེ་ཉེན?>"r)) ;

We have found that most of the CG rules in our grammar can be deciphered by the linguists on our team, and that these linguists are also willing to create and modify at least a subset of the rules without needing to consult a technical expert.

The complete grammar, as well as the corpora on which it has been tested, is available for free download online.¹

3 Possible Extensions of CG

The practical advantages of CG to our project have led us to look for other applications of the framework within our Tibetan NLP pipeline.

As with Chinese, Tibetan text does not use whitespace or other mechanisms to mark word boundaries; as with Chinese, the automatic determination of word boundaries by computer is a "hard problem". One method has followed Huidan et al's (2011) development of Xue's (2003) character tagging approach to Chinese segmentation, which recasts Tibetan word segmentation as a syllable tagging problem, with each syllable in search of an appropriate word-internal position label. For example, the only syllable of a monosyllabic word is tagged with 'S' for 'single syllable", and the first, middle, and end syllables of multisyllabic words are tagged with "B", "M", and "E", respectively. The machine then applies the syllable tagging patterns it learns from a training corpus to the new texts it is exposed to.

Using Taku Kudo's CRF++ toolkit,² we created a syllable tagging word segmenter for Tibetan. Despite the fact that character tagging approaches to Chinese word segmentation have shown promise, our segmenter's performance was disappointing. This poor performance has unwelcome consequences: since part-of-speech tagging requires a segmented corpus, mistakes in word segmentation feed mistakes in part-of-speech tagging.

To tackle this problem, one strand of our current research into Tibetan word segmentation involves modifying some of our CG POS-tagger's rules for the purposes of syllable tagging. It is sometimes possible to state with certainty the position of a specific syllable within a word, even though the full extent of the word or its neighboring words might not be known. Such certainties can then be used to constrain a statistical model, thereby improving the performance of the model.

In our full paper, we will explore this emerging prospect in addition to presenting our functioning CG POS-tagger.

References


¹ http://larkpie.net/tibetancorpus

² http://taku910.github.io/crfpp/