Constraint Grammars for Tibetan Language Processing

Christian Faggionato
SOAS, University of London
10 Thornhaugh St, Bloomsbury
WC1H 0XG
cf36@soas.ac.uk

Edward Garrett
SOAS, University of London
10 Thornhaugh St, Bloomsbury
WC1H 0XG
eg15@soas.ac.uk

Abstract

This paper describes the diverse and distinctive ways that Constraint Grammar has been used within a Tibetan verb lexicon project. We present three CG3 grammars and how they fit into our workflow, along with the practical problems they were designed to solve.[*]

1 Introduction

The aim of our work is to develop a corpus-based verb lexicon of Tibetan covering the three major periods in the history of the language: Old, Classical and Modern Tibetan. The starting point for this work is a manually annotated corpus of Tibetan texts. This is obtained by importing part-of-speech tagged Tibetan texts into the BRAT annotation tool, where human annotators then draw labeled dependency arcs between verbs and their arguments. Here's an example:

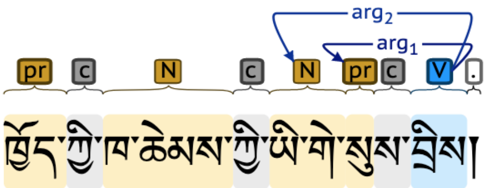

Figure 1. Verb-argument annotation.

In Figure 1, translated as "Who wrote the text of your testament?", the verb "write" is linked to its two arguments, the writer (*arg1*) and the thing written (*arg2*).

Tibetan word-order is relatively flexible, and Tibetan case-marking does not provide a failsafe way of identifying verb arguments. Therefore, we hand-annotate these relations, but narrowly so, resulting in texts whose syntactic dependency structure is only partially annotated, as Figure 1 makes clear with its many unlinked words. This creates an opportunity for automated annotation methods to fill in the gaps. Section 2 of this paper describes a CG3 grammar that does just that.

Section 3 of the paper turns to the challenge of incorporating Old Tibetan materials into our workflow. Our Classical Tibetan annotators had the luxury of working with previously POS-tagged texts. However, no manually POS-tagged texts exist for Old Tibetan. We present some of the orthographic differences between Classical and Old Tibetan, and then describe the CG3 grammar we developed to normalize Old Tibetan texts into Classical Tibetan. By first applying this grammar, we can POS-tag our Old Tibetan texts using a tagger trained on Classical Tibetan materials.

In Section 4, we describe ongoing work on a third CG3 grammar, which has broader aims than the first two grammars. We wish to draw examples for our verb lexicon not just from manually annotated texts, but also from a wide range of additional Tibetan texts. To do so, we must automatically annotate these further texts. The grammar described in Section 4 does just this, taking a POS-tagged text as input and outputting a text enhanced with the dependency relations we find essential.

We conclude the paper in Section 5. Because of the practical role these grammars currently play in our project, it would be both premature and improper to carry out a formal evaluation at this time. Instead, we make some concluding remarks and discuss the future direction of our work.

2 Other Dependencies

As illustrated in Figure 1, our project's manual annotations do not come close to providing full dependency parses for Tibetan sentences. In light of the project's primary goals, such complete parses may be unnecessary. Annotating certain relations, however, is essential. For example, in Figure 1, *arg1* is སུ ‘who’. The fact that it is followed by the ergative case marker suffix ས་ is important to us, because understanding how a Tibetan verb is used includes knowing how its arguments are case-marked.

We assert that it is possible to establish this particular case-marking relation, among other dependency relations, using automated methods. This is where Constraint Grammar 1 (G1) fits in. G1 consists of around a hundred hand-crafted rules which ensure that most words of a sentence have a non-root parent. G1 links modifiers such as adjectives, determiners and demonstratives to nouns, and converbs, punctuation and adverbs to verbs. We rely on Tibetan's relatively strict noun-phrase internal word-order. Unsurprisingly, G1 consists largely of SETPARENT and MAP rules. Here is an example:

```
#genitive + pron:
SETPARENT (Case=Gen) (NONE p
ALLPOS) TO (-1 (PRON));
MAP (@case) TARGET (ADP) -
TAGS (p Head_NOUN OR (ADV));
```

In Tibetan, if a genitive case marker follows a pronoun, then it must depend on that pronoun via the case relation. The SETPARENT rule establishes this dependency, and the MAP rule assigns the tag @case to the genitive adposition.

Other examples are more complex, but in the end, the rules of G1 combine together to assign a near complete dependency parse for a Tibetan sentence, provided the starting point is text which has been manually annotated for verb-argument structure. Figure 2 shows the result of applying G1 to the sentence in Figure 1.

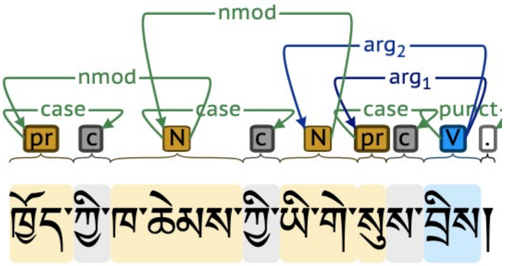

Figure 2. G1 applied to Figure 1.

The first two words of this sentence are the second person pronoun ཁྱོད ‘you’ and the genitive case marker ཀྱི་. The dependency relations between them are established by the rules just mentioned.

Refinements and further additions to G1 may in time make it possible for us to offer a version of our hand-annotated texts that incorporates automatically inserted additional relations which fill in the gaps for a complete dependency parse. However, some relations are likely to require human adjudication, and so for now we content ourselves with lesser aspirations for G1.

3 Old Tibetan Normalization

Old Tibetan, which includes accounts of the Tibetan Empire and extends from the 7th-10th centuries, presents unique challenges. Although its vocabulary and grammar are strikingly similar to Classical Tibetan, it has many differences in spelling and orthography (Dotson & Helman-Ważny, 2016). For example, the Classical Tibetan genitive case marker ཀྱི་ may be written in Old Tibetan as ཀྱྀ་. Instead of the standard *gigu* vowel we get a reverse *gigu*. In other cases, Old Tibetan words have characters that do not occur in their Classical Tibetan equivalents. Not only does the Old Tibetan form མྱྀ་ ‘person’ have a reverse *gigu*, but it also has a *ya-btags*: in Classical Tibetan the word would be much more simply མི་.

Our second CG3 grammar was developed to deal with these and other differences between Old and Classical Tibetan. We call it the Old Tibetan Normalization Grammar (G2), and its purpose is to make Old Tibetan look like Classical Tibetan. The differences just described can be characterized at the syllable level. It is possible in such cases to define simple regular-expression based SUBSTITUTE rules like the following rule. (Note that *gigu* has been referenced using its Unicode escape value since superscript vowels display awkwardly when not attached to a base character.)

```
#Replace the ‶reverse gigu″
with gigu everywhere
SUBSTITUTE ("([^<]*)\\
u0F80(.*)"r) ("$1ི$2"v)
TARGET (σ);
```

SUBSTITUTE rules do not always suffice to capture the differences between Old and Classical Tibetan. Traditionally in Classical Tibetan, syllables are separated by a *tsheg* (the dot seen in the above examples). In Old Tibetan texts, syllable margins are not so clear and often a syllable (verb, noun and so on) is merged together with the following case marker or converb: སྟགི་ > སྟག་གི་, དུསུ་ > དུས་སུ་, བཀུམོ་ > བཀུམ་མོ་. To handle these cases, we came up with a cascading series of SPLITCOHORT rules, where initial rules split specific complex syllables into separate syllables, and later rules apply generically to syllables of a particular type. Here is an example of a specific SPLITCOHORT rule:

```
SPLITCOHORT (
  "<མཆི>" "མཆིས་" σ
  "<སྣ$1>"v "ན$1"v σ
) ("<མཆིསྣ(་?)>"r);
```

And here is an example of a more general and therefore less readable rule that applies to cases like གཅལྟོ་ > གཅལད་ཏོ་:

```
SPLITCOHORT (
  "<$1>"v "$1$3ད་"v σ
  "<$2>"v "ཏ$4"v σ
) ("<(.+)((.)\\u0F9F([\\
u0F7C\\u0F7A]་?))>"r);
```

The SPLITCOHORT rules reveal the form of the input that is passed to G2. Instead of passing word tokens, the grammar is passed syllable tokens. Any syllable which G2 normalizes is added in its original form as a new reading with the tag ↑OT by the following rule near the end of the grammar.

```
APPEND ("$1"v ↑OT)
("<(.*)>"r) (NOT 0 ("$1"v));
```

G2 concludes with a choice between two rules, depending on whether the user wants to select Old Tibetan "readings" (i.e. Old Tibetan syllables and syllables that didn't require normalization), or Classical Tibetan "readings" (i.e. Classical Tibetan normalizations as well as syllables not requiring normalization).

```
#Uncomment one rule:
#SELECT (↑OT);
#SELECT (σ);
```

Thus, each syllable of the input is treated as a CG3 cohort, whose different readings are the different syllable forms (Old or Classical) that it can take. Syllable readings are then joined together in their Classical Tibetan form in order to make a Classical Tibetan normalization.

The approach we are taking has three merits. First, we have carefully characterized the orthographic differences between Old Tibetan and Classical Tibetan, which is valuable in itself. Second, we can apply Meelen and Hill's (2017) tagger to the Classical Tibetan normalizations, rather than struggle with tagging Old Tibetan texts. And third, preserving a record of which syllables have been transformed enables us to reverse the process and denormalize back to Old Tibetan, after our Old Tibetan texts have been hand-annotated. After all, Tibetan scholars do not in general want Old Tibetan texts to look like Classical Tibetan.

4 Verb-Argument Annotation

So far we have described a workflow that consists of the following steps, which may not all be necessary for a given text:

```
text normalization → POS-
tagging → BRAT import →
```

```
manual verb-argument
annotation → automated other
dependency annotation
```

We would prefer to create a verb lexicon that is informed by and draws examples from Tibetan texts that have not been manually annotated, and not just those that have. To this end, we are pursuing various strategies for automatically annotating verb-argument structure.

The Verb Argument Dependencies grammar (G3) attempts to solve this problem with CG3. The input to G3 is a POS-tagged text without dependency annotations. G3 starts by inserting "helper" tags, such as tags which identify candidate constituent junctures. For example, the following rule tags those words which, if they occur to the left of a word, could not be part of a noun phrase with that word.

```
SET LEFT_NP_BOUNDARY =
(VERB) OR (ADP) OR (PUNCT)
OR (SCONJ) OR (PART);
```

The grammar then proceeds through dozens of SETPARENT and MAP rules, which set and label the verb-argument dependencies. These rules are rather intricate and will not be exemplified here.

G3 concludes with a series of "fixing rules" which SUBSTITUTE or remove mistaken tags. For example, if a noun preceding a genitive case marker has been marked as an argument, this cannot be correct, since a word to the right of the genitive would always be the argument.

```
SUBSTITUTE (@arg2) (*)
TARGET Head_NOUN + (@arg2)
(1 (Case=Gen)) (p (VERB));
```

In other cases, the fixing rules relate to specific verbs or verb classes that behave differently from the norm. For example, verbs of movement cannot take *arg2*:

```
SUBSTITUTE (@arg2) (@arg1)
TARGET Head_NOUN + (@arg2)
(p (VERB) + VMOVE - ("མཆི་"));
```

In general, G3 has worked very well with transitive verbs, where *arg1* is marked with ergative case. The main challenge has been to detect the argument structure of verbs with multiple arguments lacking case-marking.

5 Conclusion

In this paper we described three grammars that have proved helpful to our Tibetan verb lexicon project. By automating predictable dependency annotations, G1 has allowed our annotators to focus narrowly on verb-argument annotation. G2's treatment of Tibetan syllables as CG3 tokens has replaced haphazard search and replace with an accountable and reversible approach to text normalization. Finally, G3 is tackling the challenging task of automating verb-argument annotation. This remains a work in progress, subject to further improvement and comparison with alternative methods.

In future, we hope to address some missing elements of the work presented here. As regards G1, it will always be valuable to reduce the number of words outside the dependency structure. In addition, it may be worth evaluating the correctness of those dependencies which are not obvious against a reference set of hand-annotated examples. In terms of G2, the software processing pipeline including denormalization remains to be released. Finally, the status of G3 in our pipeline needs to be clarified; from there, evaluative metrics may well follow.

The texts and grammars discussed in this paper are freely available for anybody to examine and use. For further details, see our "Tibetan NLP" page on GitHub, in particular the tibcg3 repository.

References

Brandon Dotson and Agnieszka Helman-Ważny. 2016. *Codicology, Paleography, and Orthography of Early Tibetan Documents: Methods and a Case Study*. Wiener Studien Zur Tibetologie und Buddhismuskunde, Vol. 89. University of Vienna: Vienna.

Marieke Meelen and Nathan Hill. 2017. Segmenting and POS tagging Classical Tibetan using a memory-based tagger. *Himalayan Linguistics* 16(2).