Optimisation of the Largest Annotated Tibetan Corpus Combining Rule-based, Memory-based, and Deep-learning Methods

MARIEKE MEELEN, University of Cambridge
ÉLIE ROUX, BDRC
NATHAN HILL, SOAS, University of London

This article presents a pipeline that converts collections of Tibetan documents in plain text or XML into a fully segmented and POS-tagged corpus. We apply the pipeline to the large extent collection of the Buddhist Digital Resource Center. The semi-supervised methods presented here not only result in a new and improved version of the largest annotated Tibetan corpus to date, the integration of rule-based, memory-based, and neural-network methods also serves as a good example of how to overcome challenges of under-researched languages. The end-to-end accuracy of our entire automatic pipeline of 91.99% is high enough to make the resulting corpus a useful resource for both linguists and scholars of Tibetan studies.

CCS Concepts: • **Computing methodologies** → **Neural networks**;

Additional Key Words and Phrases: NLP, POS tagging, Tibetan, historical treebanks

ACM Reference format:
Marieke Meelen, Élie Roux, and Nathan Hill. 2021. Optimisation of the Largest Annotated Tibetan Corpus Combining Rule-based, Memory-based, and Deep-learning Methods. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 20, 1, Article 7 (March 2021), 11 pages.
https://doi.org/10.1145/3409488

1 INTRODUCTION

Until a few years ago, Tibetan was considered a very low-resource and under-researched language from the point of view of Natural Language Processing and Corpus Linguistics. Digital resources, both in the form of textual data as well as dictionaries and glossaries, were scarce, although a number of Old and Classical Tibetan manuscripts had been digitised and high-quality scans were available on the website of various libraries and the Buddhist Digital Resource Center (BDRC)[1]

[1] https://www.tbrc.org.

The authors gratefully acknowledge Meelen's British Academy Postdoctoral Fellowship Grant No. pf170063, her time on the ERC Advanced Grant "Open Philology" (Grant No. 741884), and Hill's ERC Synergy grant "Beyond Boundaries: Religion, Region, Language and the State" (ASIA 609823) for making this research possible.
Authors' addresses: M. Meelen, University of Cambridge; email: mm986@cam.ac.uk; É. Roux, BDRC, email: roux.elie@gmail.com; N. Hill, SOAS, University of London; email: nh36@soas.ac.uk.

2375-4699/2021/03-ART7
https://doi.org/10.1145/3409488

and a certain amount of NLP research had been carried out in China, without, however, making data or code available.[2]

In recent years, the BDRC developed further digitisation tools and ontologies.[3] Classical Tibetan text collections such as the Derge edition of the *Kangyur* and *Tengyur* (translated words of the Buddha and their commentaries) are now available in digital format through the work of Esukhia[4] (316 volumes), who also prepared a number of Modern Tibetan corpora[5] and a rule-based tokeniser called *botok*.[6] As a result of the "Tibetan in Digital Communication" project at SOAS, University of London (see also Garrett et al. 2014), there is now furthermore a manually corrected segmented and POS-tagged corpus available as a Gold Standard, the "SOAS corpus," consisting of four Classical Tibetan texts: མཛངས་བླུན་ཞེས་བྱ་བ་མདོ། *mdzangs blun zhes bya ba'i mdo* (sutra of the wise and the fool) མར་པ་ལོ་ཙྪའི་རྣམ་ཐར། *mar pa lo tsA'i rnam thar* (Biography of Marpa the translator), བུ་སྟོན་ཆོས་འབྱུང་། *bu ston chos 'byung* (Buton's History of Buddhism in India) and མི་ལའི་རྣམ་ཐར། *mi la'i rnam thar* (Biography of Milarepa).[7]

Meelen and Hill [2017] took the SOAS corpus as a baseline to train a memory-based segmenter and POS tagger used these tools and all of the texts that were digitised by the BDRC in 2017 to create the first large-scale segmented and POS-tagged corpus known as "ACTib"—the Annotated Corpus of Classical Tibetan, available from Zenodo (see Meelen et al. 2017a and Meelen et al. 2017b). Although the first (2017) version of ACTib is a useful resource, the segmentation and POS labels of the files were never analysed, checked or corrected. Therefore, to make these resources even more useful for both linguists and scholars of Tibetan studies, a number of issues need to be addressed in all stages of the annotation procedure. The first ACTib version, for instance, was minimally preprocessed, resulting in various "messy" outputs related to problems that arose during parsing of non-standard xml-files, uncorrected digital text versions, and so on. In addition, the output of the segmenter was not systematically checked against any Tibetan dictionaries or glossaries, nor was there ever a full error analysis of the POS-tagged output, resulting in various errors that could have been prevented or corrected.

In this article, we present an improved annotation procedure that addresses all issues of preprocessing, segmentation and POS tagging in detail. In Section 2, we first conduct a full error analysis. In Section 3, we investigate ways of optimising the automatic annotation method, both for segmentation and POS tagging. Corrections in the form of rule-based replacements and dictionary lookups are then outlined in Section 4. And in Section 5, we present our results. The semi-supervised methods presented here not only result in a new and improved version of the largest annotated Tibetan

[2]Relevant here is only research on tokenization and POS-tagging. Research on Tibetan tokenization proceeded through three phases: the first phase focused on string matching using dictionary look up [Jiang 2003; Tsering 扎西次仁 1999], the second phase combined NP chunking with dictionary lookup and HMMs [Sun et al. 2009], and the third phase approached tokenization as a problem of POS tagging over individual syllables [Kang et al. 2013; Liu et al. 2015]. All researchers now follow the third technique, albeit with different understandings of Tibetan wordhood. Turning to POS-tagging, there is a dominant POS tag set in the PRC [Gya and Tsering 2010; Tshe Ring Rgyal 才让加 and Mchog Thar Rgyal 吉太加 2005] and taggers that include dictionary lookup [Gya and Tsering 2010], CRF tagging [Wu et al. 2014], maximal entropy [Ma et al. 2016], and BiLSTM+CRF [Wang et al. 2019]. For summaries of the history of Tibetan NLP research see Hill and Jiang [2016], Hackett [2019, 102–104], and on POS-tagging specifically Wang et al. [2019, 490–491].

[3]https://github.com/buda-base.

[4]https://github.com/Esukhia/derge-kangyur and https://github.com/Esukhia/derge-tengyur.

[5]https://github.com/Esukhia/Corpora.

[6]https://github.com/Esukhia/botok.

[7]https://github.com/tibetan-nlp/soas-corpus.

corpus to date,[8] the integration of rule-based, memory-based, and neural-network methods also serves as a good example of how to overcome challenges of under-researched languages.

2 ACTIB ERROR ANALYSIS OF THE 2017 VERSION

Since the ACTib corpus is rather large (~13k XML files each comprising a number of texts or entire collections of texts featuring almost 100k unique tokens), it is impossible to manually check and correct the entire corpus. In every stage of the annotation process, we have therefore conducted a detailed error analysis. Some features of the Tibetan language, furthermore, pose additional challenges, e.g., the use and availability of two representations (Tibetan Unicode and romanisation using the "Extended Wylie Transliteration Scheme,"[9] referred to as Wylie in the rest of the article), the lack of sufficiently accurate tools for Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR), and the general dearth of adequate digitised language resources such as exhaustive dictionaries, and so on. In this section, we discuss the most frequently found errors in each of the stages; the next sections outline solutions to these challenging issues.

2.1 Preprocessing Issues

In 2017, the BDRC employed two different XML models for their source files. Meelen and Hill [2017], however, only optimised their code to preprocess these files for the most common TEI-based model. This meant that some errors crept in the preprocessed files whenever a specific XML tag was not correctly identified and parsed, which was the case for some metadata tags in the XML header. As a result, the files that served as input to the segmenter sometimes contained additional information from the metadata header, such as numbers and dates, which percolated throughout the annotation process. Finally, a number of XML files contained fragments of metadata in English and German, which was erroneously labelled as Tibetan content. These issues have now been addressed by optimising the preprocessing script in such a way that it only takes the Tibetan content and the line and page numbers, regardless of their XML model. As a result, no additional numbers, dates or other information from the metadata headers has accidentally ended up in the segmented and POS-tagged texts.

A more general issue with the digitised input texts was that the original transcription at that time had not been corrected. Since 2017, however, Esukhia has manually corrected one of the most important text collections of the ACTib: the Derge *Kangyur* and a revised version of the *Tengyur*. We can now use these new manually corrected input resources and thereby address a number of transcription errors that, again, percolated through the system in the 2017 version.

2.2 Segmentation Issues

At the segmentation stage it was clear that most errors consisted of instances of case markers and converbs that were still attached to the tokens they modify. These markers, however, should each receive their own tag and thus be separated properly. These markers always appear in the same form and their orthography only ever varies in consistent, predictable ways. To facilitate further downstream tasks like POS tagging and parsing, we therefore wrote a rule-based script to automatically split off these markers from their preceding tokens (see Section 4.1).

2.3 POS Tagging Issues

Meelen and Hill [2017, 79] showed that the POS tagger generally performed well on the test set, but since the amount of training data was rather limited (318k tokens), the results did not always

[8] https://zenodo.org/record/3785071.
[9] http://www.thlib.org/reference/transliteration/#!essay=/thl/ewts/.

generalise to the very large corpus of the BDRC that furthermore consists of texts from different times and genres. In addition, Classical Tibetan has a large number of homonyms that should receive different tags, which did not always happen. Another issue we found concerned a number of tokens that were tagged as verbs but were, in fact, nouns. Since Nathan Hill's lexicon of Tibetan verbs is now digitised [Hill and Garrett 2017],[10] it is possible to look up the tokens tagged as verbs in this exhaustive list. Finally, we found many errors in homophonous case markers and converbs. After checking the verbs, these could be corrected based on their context (see Section 4 below).

3 IMPROVING THE TAGGING PROCESS

The first step in improving the overall results is to attempt to improve the tagging procedures themselves, both in the segmentation as well as in the POS-tagging stage. Although the global accuracies of the syllable and POS taggers were not bad to begin with (Meelen and Hill 2017 report 93% and 95%, respectively), there is still room for improvement when generalising from the relatively small SOAS corpus to all the digitised texts in the BDRC collection. For a corpus of Old Tibetan, Faggionato and Meelen [2019] proposed a number of ways to improve the results of the taggers, testing different tag sets, taggers (memory-based vs. neural-network), changing scripts (Tibetan Unicode vs. Wylie transliteration) and, finally, through optimisation of the training data. In this section, we build on these suggestions for Old Tibetan to create better taggers for our new Classical Tibetan corpus.

3.1 Optimising the Tag Set

Since there are very few syllable classification labels (only eight) to begin with and each of those contain valuable information about the position of the syllable in a word, the segmentation tag set cannot be improved any further. For POS tagging, however, Faggionato and Meelen [2019] report some improvement (from 95% to 96.3% global accuracy) when using the much smaller Universal Dependency (UD) tag set, consisting of only 15 tags. The tag set developed by Garrett et al. 2014 is rather large (79 morpho-syntactic tags) and this causes major issues for the out-of-vocabulary (OOV/unseen) items.

There are various other ways to improve the tag set. First of all, since line and page numbers are kept in their respective positions in the text, it makes sense to develop specific tags for those too, e.g., `line.num` and `page.num`. Further scrutiny of the tag set yields a number of tags that can be problematic in context. A good example of this is the temporal adverb marker "adv.temp," which is used for adverbs that are, in fact, still behaving more like nouns. Since homophonous case markers and converbs are assigned their respective tags based on the context, this leads to problems when adverbial tags, such as `adv.temp`, that are actually used for nouns immediately precede them. The solution here would be to change the `adv.temp` tag to a nominal tag, e.g., `n.temp`.

In practice, for scholars of Tibetan studies, it is very useful to have more morpho-syntactic information than the small UD tag set can offer, even with the additions of genitive and agentive markers suggested by Faggionato and Meelen [2019]. As mentioned above, the results of the POS tagger (trained on the training split of the SOAS corpus as it is the only existing gold standard) cannot automatically be generalised to our much larger BDRC corpus and the larger the tag set, the harder the task. We have therefore decided to make two versions of the new ACTib available: one with the smaller UD tag set and one with the larger, more detailed tag set.

3.2 Unicode vs. Wylie Transliteration

Faggionato and Meelen [2019, 310] test the memory-based tagger on a Wylie transliteration (instead of the Tibetan Unicode script) of the SOAS corpus too. The results for the small UD tag

[10] http://doi.org/10.5281/zenodo.574876.

set are slightly better for the Wylie transliteration (96.5% global accuracy for Wylie vs. 96.3% for Tibetan Unicode); for the large tag set, however, results for Wylie are slightly worse (94.7% vs. 95.0% for Tibetan Unicode). Since they do not provide an analysis of these results, nor do they explain why there may be differences in the first place, in this section, we aim to provide further insights into the origin of these results.

There are various automatic Unicode-to-Wylie converters available, e.g., from the Tibetan and Himalayan Digital Library website,[11] Esukhia,[12] and BDRC.[13] These tools systematically convert Tibetan Unicode into romanised transliteration in accordance with the Wylie conventions (see Hill 2012). In general, every character in Tibetan Unicode corresponds to a single character in the Wylie transliteration. A typical example is བསྒྲུབས་, which is transliterated as *bsgrubs* "achieved." Note, however, that the end-of-syllable marker *tsheg* is converted to a space between two syllables. After syllables are combined to form words at the end of the segmentation process, spaces between syllables are converted to underscores and those at the end of words are deleted, since the tagger treats spaces as token boundaries. In Tibetan Unicode, however, words ending with or without a syllable marker *tsheg*, e.g., བསྒྲུབས་ or བསྒྲུབས, are treated as two different words by the tagger. Furthermore, the *tsheg* counts as a separate character and thus as one of three final characters the tagger takes into account when assigning a tag to unseen tokens. Since there is no difference in meaning or function between words with or without *tsheg*, the tagger using Wylie transliteration has an advantage as there will be fewer unseen tokens and more final characters that could help decide which morpho-syntactic label is most appropriate.

There are two further factors that differentiate the Tibetan Unicode from the Wylie transliteration, affecting the efficiency of the tagger. The Tibetan script was originally a syllabic abugida, in the sense that every sign in isolation represented a syllable consisting of an initial consonant with a default vowel /a/ in the coda, which could be modified to other vowels /i,e,u,o/ through diacritic markers on top or below the consonant sign. This type of abugida, however, is not ideal for a language like Old Tibetan that featured a number of consonant clusters at the beginning and end of words. To represent these clusters, stacked signs and so-called prefixes and suffixes were used, as shown in the aforementioned example བསྒྲུབས་, which is transliterated as *bsgrubs* with a prefixed *b-*, stacked *-sgr-*, diacritic *-u-*, and suffixed *-bs*. At the same time, there are certain case markers and converbs in Tibetan that are attached to their preceding words in the form of suffixes, rather than forming independent syllables on their own. Examples of these are the ergative ས *-s* and terminative ར *-r* when following words ending in vowels or འ. Since the segmenter treats converbs and case markers as separate tokens, in the Wylie transliteration they end up looking like independent syllables *sa/ra* (with default /-a/ vowels) rather than suffixes *-s/-r*, which can make a difference for the tagger. In this case performance in Wylie transliteration could be weaker, because apart from being a potential case marker or converb (depending on context), unlike the actual suffixed forms *-s/-r*, both *sa* and *ra* are independent nouns as well meaning "earth" and "goat" respectively, thus expanding the number of ambiguous homonyms the tagger has to deal with. This then can at least partly explain why results for the Wylie transliteration are slightly worse in tests with the larger tag set.

There are various subtle ways to remedy the aforementioned issues depending on the desired outcome (i.e., Wylie or Tibetan Unicode). The issues with the syllable marker *tsheg* for Tibetan Unicode can be addressed by normalising the text in such a way that word-final *tsheg*s are cut off

[11] http://www.thlib.org/reference/transliteration/wyconverter.php.
[12] https://github.com/Esukhia/pyewts.
[13] https://github.com/buda-base/ewts-converter and https://github.com/buda-base/jsewts.

at the time of POS tagging.[14] The second issue with certain case markers and converbs becoming ambiguous in the Wylie transliteration can be addressed by simply converting the text to Wylie *before* segmentation, and/or by refining the conversion and segmentation replacement rules so that cut-off suffixed case markers and converbs like *-s/-r* are not converted to their syllabic *sa/ra* readings. Finally, results of the Wylie transliteration for the larger tag set can be improved by an optimisation of parameters settings, which was hitherto only done for the Tibetan Unicode.

3.3 Memory-based vs. Neural-Network Tagging

Another way to optimise the tagging process is by using a more state-of-the art method of sequence labelling, i.e., a neural-network-based POS tagger, rather than a memory-based tagger. To do so, we first of all needed to create word embeddings, or vector spaces of vocabulary in context. Faggionato and Meelen [2019] use the first segmented version of ACTib to train word embeddings, because it was the only available segmented corpus of considerable size. Since that version was not optimised, nor corrected, however, the reported results of the Targer neural-network tagger [Chernodub et al. 2019] are with 95.8% Global Accuracy only slightly better than those of the memory-based tagger (95.0%).

Since we now have a number of ways to optimise the results of the segmentation, we can use these optimised files and combine them into a large and much better data set to train the word embeddings. In addition to this, we deliberately left out the segmented files from the BDRC collection that were OCRed. Since the OCRed text nor their XML format was corrected, these files contain numerous mistakes that are impossible to correct even semi-automatically. Since there are only 886 of such badly OCRed files and the entire collection consists of over 13k files, this is only a small decrease in size of input for the word vectorisation process.

We created Classical Tibetan word embeddings using FastText,[15] as, unlike other word vectorisation tools like Word2Vec, FastText takes characters into account as well and therefore may yield more promising results in subsequent Neural Network-based NLP tasks [Bojanowski et al. 2017]. To make our results comparable to Faggionato and Meelen [2019], we then trained the BiLSTM-CNN-CRF tagger[16] on the training split of the SOAS corpus with the exact same hyperparameters and evaluated the results on the held-out test split.[17] These newly created word embeddings yielded an increase in Global Accuracy of almost two percent. Since all hyperparameters were kept stable, the new 97.29% Global Accuracy of this neural-network tagger indirectly signals that segmentation of the overall corpus has improved significantly as well.

Apart from creating better word embeddings, results of neural-network taggers could be improved through feature engineering and/or by adjusting the design and pipeline of the tagging procedure, e.g., by switching from a recurrent to a convolutional neural network or by adding further (Bi)LSTM and Conditional Random Field layers. We will address this in future research.

3.4 Optimising the Training Data

A final way to optimise the tagging procedure itself is by double-checking and optimising the data on which the taggers are trained, removing any mistakes and labelling inconsistencies that may have crept in when developing the existing Gold Standard. Although the SOAS corpus contains

[14] Note that it may be useful to keep the *tsheg* at other times, e.g., when it is of philological importance to give a more accurate rendering of the manuscript.
[15] https://fasttext.cc/.
[16] https://github.com/achernodub/targer.
[17] Following Faggionato and Meelen [2019], we calculated the F1 instead of normal accuracy to make it directly comparable to the results presented by Meelen and Hill [2017]. Actual accuracies are slightly higher than the Global Accuracies presented here.

a variety of texts, there are a number of features found in the BDRC corpus that are not found in the training part of the SOAS corpus. A good example of these are line and page numbers, which are marked by the non-Tibetan characters "l" and "p," respectively, followed by a number corresponding to the location of lines and pages/folios in the manuscript. These line and page obviously occur frequently in the BDRC collection, but they are not recognised by the tagger, since they do not occur in the training data.

In addition, there is a large number of further punctuation markers, such as ྿ or ྅ that are not found in the SOAS corpus. Instead of post-processing the segmented and POS-tagged texts, replacing incorrectly labelled out-of-vocabulary tokens like these, we could add these tokens with their respective tags to the training data so that the taggers can learn and generalise based on that. Since punctuation is usually calculated separately from regular tokens when reporting results of evaluations, this does not necessarily provide evidence for enhanced performance of the taggers as such. It does, however, result in a POS-tagged version of the entire BDRC collection that is better than before these additional features were added to the training data.

Finally, as briefly mentioned above, the training data could be improved by critically re-examining a number of tokens tagged as `adv.temp`. Many of these tokens function as adverbials and were therefore tagged as such. However, most of these are originally, and to a certain extent still syntactically, nouns. The fact that they are syntactically behaving like nouns is clear, because they can be followed by (case) markers, as shown by ནང་པའི་ སྐྱབས་ གནས་, transliterated and POS-tagged in example (1):

(1) nang pa ‘i skyabs gnas
`adv.temp gen.case n.count`
‘The refuge of the following morning’

Although these types of tokens may function as adverbials, it would be better to give them a nominal rather than adverbial morpho-syntactic tag, e.g., `n.temp` instead of `adv.temp`. When doing this, the tag `n.temp` could be collapsed into a simple nominal tag `N` in the smaller UD tag set, which would make more sense in the context and thus enhance tagging performance.

4 RULE-BASED CORRECTIONS AND LOOK-UPS

This section outlines the rule-based corrections and dictionary look-up mechanisms that were employed to address the issues discussed in Section 2. In Section 4.3, we furthermore discuss some final rule-based corrections we executed in the post-processing stage to "clean up" the segmented and POS-tagged corpora and make them ready for distribution.

4.1 Segmentation

Meelen and Hill [2017] recast Tibetan segmentation as a syllable-tagging task, proving "beginning," "middle," and "end" markers to each syllable and then recombining syllables. There were some syllables, e.g., དང་ དེ་ and ཡིག་ that were sometimes found with SS ("two single syllables") or ES ("end + single syllable"). Since these syllables do not end with འོ་ འི་ ས་ འམ་ འང་, or ར་ these SS and ES syllable labels are simply impossible. We therefore changed erroneously tagged instances of དང་ དེ་, and ཡིག་ with a simple replacement rule (SS/ES > S). A similar example is the complex syllable རྣམ་, which must be tagged SS or ES, because no Tibetan word can have འི་ in it. Since SS is a label that would be correct more often than ES, any other labels assigned to རྣམ་ were replaced by SS.

After correcting these incorrectly tagged syllables and combining them into words, we did an additional check for converbs and case markers. These particles form a closed category and have only a limited number of orthographical variants. As such, they could be easily identified and split off their preceding tokens. In addition, we employed `botok` using the 2019 version of the Grand Monlam Dictionary[18] to look up the results of the segmenter. Whenever a multi-syllable segmented form was not found in the dictionary, we used *botok*'s max match algorithm to find a better segmentation.

Finally, to improve the automatic sentence segmentation (following Meelen and Hill [2017], we not only inserted utterance boundaries after every Tibetan punctuation marker ། *shad*) but also merged utterances that were erroneously split due to the occurrence of a so-called "double-*shad*," which marks the end of sentences or other coherent passages. In the 2017 version of ACTib, whenever there was a double *shad*, each *shad* would get a separate utterance boundary, resulting in a large number of utterances that only consisted of the second *shad* of the sequence. In the new version of ACTib, we corrected this at the end of the segmentation stage, so that double *shads* are kept together at the end of a single utterance as intended.

4.2 POS Tagging

The issue of wrongly tagged homophonous case markers and converbs mentioned in Section 2.3 above was also addressed with a simple replacement rule. Take, for example, the sequence ཤཱཀྱ་སེང་གེ/n.prop ལ་/case.all *shakya seng ge* ***la*** "for Shakya Sengge," with a dative/allative case marker *la*. The homophonous converb *la*, however, is found directly following verbs only and this pattern is relatively common: all converbs generally follow verbs and their homophonous case-marking counter parts generally follow nouns. After tokens tagged as verbs were checked in the verb lexicon, we applied a case marker vs. converb check correcting the erroneous tags based on the tag of the immediately preceding token.

Although this captured a number of mistakes, there are some cases where this did not work. Since case markers can follow bare nouns but also entire noun phrases, the immediately preceding token of a case marker could in theory be a determiner, adjective or numeral appearing at the end of the noun phrase as well. There are indeed cases, where a case marker can follow an adverbial marker, e.g., the aforementioned རྡོངས་, meaning "the following morning," which is tagged as `adv.temp`, but was historically a noun. Since converbs can also be found directly after adverbs if they immediately follow a verb, these cases can be ambiguous for a tagger that takes the context into account. These examples are not easy to address with simple context-sensitive replacement rules. Instead, it is worth addressing them in an earlier stage, by having a critical look at the tag set (see Section 3.1 above), POS-tagging guidelines and the original training data to ensure these exceptions can be dealt with in a more efficient way (see Section 3.4 above).

4.3 Post-Processing

Before publishing the new versions of the segmented and POS-tagged corpora on Zenodo, we conducted a final "data sanity" check to see if all files indeed contained all the material we needed, and, more importantly, not more than that. There is a small number of files that the BDRC already identified as "blacklisted" files, because their content is corrupted in various ways.[19] Some of these files could not even be processed and were thus picked up by our segmentation and tagging scripts automatically. Others, however, were processed, but in the post-processing stage removed from the final data set to keep it clean.

[18] http://monlamit.com/.
[19] See a full list on the Buda-Base repository on GitHub.

Finally, some English and German metadata was accidentally included, because it was erroneously contained within the same XML tag as the Tibetan content. This includes some names and contact information from institutes who supported the digitisation of some of the files, some references to images and also certain references to other scripts and characters that are not part of the Tibetan Unicode set, e.g., "Dakinisecretletterhere" or "Musical notesandsigns here." We systematically checked all files and deleted any such non-Tibetan content that should not be part of the corpus.

5 RESULTS, OUTPUTS, AND CONCLUSION

In this final section, we present our results and outputs. We first discuss the test results evaluating the entire pipeline. Then, we give a short overview of all our outputs in the form of open source data and software applications.

5.1 Evaluating the Optimised Pipeline

Since Meelen and Hill [2017] only evaluate the syllable and token labelling algorithms, it remains unclear how good the results are of the segmentation stage (i.e., after recombination of syllables into tokens based on their syllable labels). To provide a full evaluation, we first of all removed the syllable and Part-of-Speech labels, then de-tokenised the segmentation Gold Standard of the SOAS corpus and finally, put it in BDRC XML format. To be able to test the pipeline from beginning to end, we created randomised 80/10/10 splits for training, development and test data and then manually aligned over 12k test tokens comparing the Gold Standard with the test results. Naturally, errors in segmentation always resulted in at least one (but often two) errors in POS tagging, decreasing overall accuracies of the entire pipeline. Since we have already discussed the Global Accuracies of our models above and it is more useful for Tibetan and linguistic scholars to know how effective the overall pipeline and how reliable the resulting corpora are, we limit ourselves to reporting the latter here:

Segmentation: 99.06%
POS tagging (large tag set): 92.34%
Overall pipeline from plain text or XML to POS-tagged corpus: 91.99%

Segmentation errors are now occurring in less than 1% of the tokens, almost all of which are proper nouns. These then immediately result in POS tagging errors. In addition, despite our rule-based correction after POS tagging, some errors in ambiguous homonyms remain. Although most (semi)final particles are unambiguous or can be disambiguated in the context (e.g., after a verb and before a clause-final *shad*), this is not always the case. Finally, mass and count nouns are often mixed up by the POS tagger, though errors like these have a minor impact and do not matter when using the smaller UD tag set only. The overall process yields almost 92% accuracy, which is a major improvement resulting in maximally increased usability of the newly created resources.

5.2 Outputs and Dissemination

We will make both the segmentation and annotation scripts available freely through GitHub[20] for all the above-described options: for plain text or XML source files, with configuration and settings files for both the small and large tag sets as well as various output options (segmentation, POS tagging or both). In principle, these scripts could be used for any form of Tibetan, but it should be noted that the tests are based on the training split of the four Classical Tibetan texts from the SOAS corpus only and results may thus differ for different genres and time periods.

[20] https://github.com/lothelanor/actib.

Our second main output is the new version of the ACTib, both in segmented and POS-tagged form (with the small UD and the large tag sets). These will be made available through Zenodo[21] as plain text files using the same key file-naming conventions as their BDRC XML originals to make the corpus optimally useful for Tibetan scholars in particular.

Finally, we will share our new word embeddings file in vector format as well as the complete binary model created with FastText. This file contains 100-dimensional word vectors for over 90k unique tokens found in the newly segmented ACTib.

5.3 Conclusion

This article presents a detailed error analysis of various annotation methods as well as an evaluation of the entire pipeline from plain text or XML file to segmented and POS-tagged Tibetan outputs. The NLP models as well as the data themselves can be optimised in a number of ways, but we furthermore argue that by studying the types of errors in the various outputs in greater detail, better results can be achieved through rule-based corrections and dictionary lookups after both segmentation and POS tagging. The concrete outputs, both the open-source software and the new version of ACTib, will thus be of great use to linguists and scholars of Tibetan studies. These resources are now of such a high level of accuracy that it is worthwhile to extend them with relevant metadata and phrase-structure to create a historical treebank (see Meelen and Roux 2020). In future work, we aim to create an even better version of the ACTib by improving the neural-network model for POS tagging in particular by optimising the hyperparameters and feature engineering and implementing a small number of highly complex rule-based corrections that were beyond the scope of the present article.

REFERENCES

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* 5 (2017), 135–146.

Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. 2019. TARGER: Neural argument mining at your fingertips. In *Proceedings of the 57th Annual Meeting of the Association of Computational Linguistics (ACL'19)*.

Christian Faggionato and Marieke Meelen. 2019. Developing the old Tibetan treebank. In *Proceedings of Recent Advances in Natural Language Processing*, Angelova, Mitkov, Nikolova, and Temnikova (Eds.). 304–312.

Edward Garrett, Nathan W. Hill, and Abel Zadoks. 2014. A rule-based part-of-speech tagger for Classical Tibetan. *Himal. Linguist.* 13, 2 (2014), 9–57.

Tsering Gya and Dbangphyug Tsering. 2010. Research on a standard for POS tagging of contemporary Tibetan for TIP. In *Proceedings of the 12th Seminar of the International Association for Tibetan Studies*. 1–12.

Paul G. Hackett. 2019. *Digital Encoding, Preservation, Translation, and Research for Tibetan Buddhist Texts*. Walter de Gruyter, 91–110. DOI: https://doi.org/10.1515/9783110519082-006

Nathan W. Hill. 2012. A note on the history and future of the "Wylie" system. *Revue d'Etudes Tibétaines* 23 (2012), 103–105.

Nathan W. Hill and Edward Garrett. 2017. A part-of-speech (POS) lexicon of Classical Tibetan for NLP. http://doi.org/10.5281/zenodo.574876

Nathan W. Hill and Di Jiang. 2016. Introduction: Tibetan natural language processing. *Himal. Linguist.* 15, 1 (2016), 1–11. DOI: https://doi.org/10.5070/H915131516

Di Jiang. 2003. *A New Perspective for Modern Tibetan Machine Processing and its Development: An Insight Into the Method of Computerized Automatic Understanding of Natural Languages in Terms of Chunk Identification.* 现代藏语的机器处理及发展之路，徐波，孙茂松，靳光瑾主编,科学出版社 Kexue Chubanshe, 438–448.

C. Kang, D. Jiang, and C. Long. 2013. Tibetan word segmentation based on word-position tagging. In *Proceedings of the International Conference on Asian Language Processing*. 239–242. DOI: https://doi.org/10.1109/IALP.2013.74

Huidan Liu, Congjun Long, Minghua Nuo, and Jian Wu. 2015. Tibetan word segmentation as sub-syllable tagging with syllable's part-of-speech property. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally*

[21] https://zenodo.org/record/3951503.

Annotated Big Data, Maosong Sun, Zhiyuan Liu, Min Zhang, and Yang Liu (Eds.). Springer International Publishing, Cham, 189–201.

Ning Ma, Yachao Li, and Xiangzhen He. 2016. Fusion of word clustering features for Tibetan part of speech tagging based on maximum entropy model. *Int. J. Simul. Syst. Sci. Technol.* 17, 8 (2016), 19.1–19.5. DOI: https://doi.org/10.5013/IJSSST.a.17.08.19

Marieke Meelen and Nathan Hill. 2017. Segmenting and POS tagging Classical Tibetan using a memory-based tagger. *Himal. Linguist.* 16, 2 (2017), 64–89.

Marieke Meelen, Nathan W. Hill, and Christopher Handy. 2017a. The Annotated Corpus of Classical Tibetan (ACTib), Part I—Segmented version, based on the BDRC digitised text collection, tagged with the Memory-based Tagger from TiMBL. DOI: https://doi.org/10.5281/zenodo.823707

Marieke Meelen, Nathan W. Hill, and Christopher Handy. 2017b. The Annotated Corpus of Classical Tibetan (ACTib), Part II—POS-tagged version, based on the BDRC digitised text collection, tagged with the Memory-based Tagger from TiMBL. DOI: https://doi.org/10.5281/zenodo.823707

M. Meelen and É. Roux. 2020. Meta-dating the PArsed Corpus of Tibetan (PACTib). In *Proceedings of the 19th Workshop on Treebanks and Linguistic Theories*. 31–42.

Yuan Sun, Xiaodong Yan, Xiaobing Zhao, and Guosheng Yang. 2009. Design of a Tibetan automatic word segmentation scheme. In *Proceedings of the International Conference on Information Engineering and Computer Science*. DOI: https://doi.org/10.1109/iciecs.2009.5366542

Tashi Tsering 扎西次仁. 1999. 一个人机互助的藏文分词和词登录系统的设计 *Design of a Word Segmentation System for Word Segmentation and Word Registration.* 民族出版社 Nationalities Publishing House, 322–327.

Tshe Ring Rgyal 才让加 and Mchog Thar Rgyal 吉太加. 2005. 基于藏语语料库的词类分类方法研究 Studies on a Taxonomic Approach to Part of Speech Identification in the Tibetan Corpus. 西北民族大学学报（自然科学版） *J. Northwest Univ. National. (Natural Sci.)* 26, 57 (2005), 39–42.

Lili Wang, Ziyan Chen, and Hongwu Yang. 2019. TPOS tagging method based on BiLSTM_CRF model. In *Proceedings of the 5th International Conference of Pioneering Computer Scientists, Engineers and Educators (ICPCSEE'19)*, Xiaohui Cheng, Weipeng Jing, Xianhua Song, and Zeguang Lu (Eds.). Springer, Singapore, 490–503.

Z. Q. Wu, H. Z. Yu, and S. H. Wan. 2014. Research on automatic tagging of parts of speech for Tibetan texts based on the condition of random fields. *Appl. Mech. Mater.* 519–520 (2014), 784–787. DOI: https://doi.org/10.4028/www.scientific.net/amm.519-520.784

Received December 2019; revised May 2020; accepted July 2020