Revue d'Etudes Tibétaines

Proceedings of the IATS 2022 Panel on Tibetan digital humanities and natural language processing

Edited by Marieke Meelen, Nathan Hill
& Christian Faggionato

numéro soixante-douze — Juillet 2024

Revue d'Etudes Tibétaines

numéro soixante-douze — Juillet 2024

ISSN 1768-2959

Périodicité
La périodicité de la *Revue d'Etudes Tibétaines* est généralement bi-annuelle, les mois de parution étant, sauf indication contraire, Octobre et Avril. Les contributions doivent parvenir au moins six (6) mois à l'avance. Les dates de proposition d'articles au comité de lecture sont Novembre pour une parution en Avril, et Mai pour une parution en Octobre.

Participation
La participation est ouverte aux membres statutaires des équipes CNRS, à leurs membres associés, aux doctorants et aux chercheurs non-affiliés. Les articles et autres contributions sont proposés aux membres du comité de lecture et sont soumis à l'approbation des membres du comité de rédaction. Les articles et autres contributions doivent être inédits ou leur réédition doit être justifiée et soumise à l'approbation des membres du comité de lecture. Les documents doivent parvenir sous la forme de fichiers Word, envoyés à l'adresse du directeur (jeanluc.achard@sfr.fr).

Comptes-rendus
Contacter le directeur de publication, à l'adresse électronique suivante :
jeanluc.achard@sfr.fr

Langues
Les langues acceptées dans la revue sont le français, l'anglais, l'allemand, l'italien, l'espagnol, le tibétain et le chinois.

La *Revue d'Etudes Tibétaines* est publiée par l'UMR 8155 du CNRS (CRCAO), Paris, dirigée par Sylvie Hureau.

Hébergement: http://www.digitalhimalaya.com/collections/journals/ret/

❖

Revue d'Etudes Tibétaines

numéro soixante-douze — Juillet 2024

Proceedings of the IATS 2022 Panel on Tibetan digital humanities and natural language processing

Edited by Marieke Meelen, Nathan Hill & Christian Faggionato

❖

Breakthroughs in Tibetan NLP & Digital Humanities*

Marieke Meelen
University of Cambridge

Sebastian Nehrdich
Heinrich Heine University Düsseldorf;
University of California, Berkeley

Kurt Keutzer
University of California, Berkeley

The field of Digital Humanities has been transformed in recent years, not just because of advances in computing software and hardware, but in particular because of breakthroughs in Natural Language Processing (NLP). In this introduction to this Special Issue related to the Tibetan NLP and Technology panel at the conference of the International Association of Tibetan Studies (IATS) in Prague in 2022, we give a brief overview of the so-called 'state-of-the-art' of NLP and Digital Humanities tools for Tibetan Studies in particular. We aim to provide accessible introductions to the contributions in this Special Issue by other panel members as well as other recent developments in the field of 'Tibetan Tech' that could benefit any scholars in the field.

1. Introduction

In the Humanities, Social Sciences, Cultural Heritage and literary communities, there is increasing interest in, and demand for, Digital Humanities and Natural Language Processing (NLP) methods to enhance our data and facilitate new lines of research. The IATS 2022 panel 'Tibetan digital humanities and natural language processing' aimed to bring together researchers from all

* Marieke Meelen, Sebastian Nehrdich & Kurt Keutzer, "Breakthroughs in Tibetan NLP & Digital Humanities", *Revue d'Etudes Tibétaines*, no. 72, Juillet 2024, pp. 5-25.

The authors would like to acknowledge support from the AHRC (Grant ref. AH/V011235/1), the ELDP (Grant ref. G114548) as well as the BDRC, Esukhia and MonlamAI. This research is furthermore partially funded by the European Union (ERC, PaganTibet, 101097364). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

these areas to discuss new technologies, NLP methods and digital humanities tools related to any aspect of Tibetan Studies (language, linguistics, culture, history, literary studies, etc.).

Recent years have seen great progress in these areas with a wide range of research projects focusing on various aspects of Tibetan NLP and Digital Humanities. The panel encouraged discussions on both the technical as well as the applied side, so that both developers and users of these new tools can collaborate and enhance their research. In 2022 already, there were major Tibetan text corpora that could be digitally mined. Nonetheless, digital techniques have not reached everyone in the field of Tibetan Studies yet. In addition, there remained a gap even between those technologically engaged Tibetologists working on resource creation (e.g. library scientists) and those working on task-based tools (e.g. computational linguists) as well as as the wider field at large. This panel provided an excellent forum for researchers working across the domains of Tibetan digital humanities, natural language processing, and library science to share results and build collaborations.

Developments in the field of NLP and Digital Humanities are quick even for languages like Tibetan that are traditionally classified as 'low-resource' and 'under-researched'. The panel and this Special Issue demonstrate, however, that rapid progress in the field has opened up a wide range of opportunities from digitising textual and audiovisual resources to get data (Section 2), to enriching data through linguistic annotation (Section 3), retrieving information from digitised and annotated resources (Section 4) as well as Machine Translation (Section 5) and language learning or textual editing (Section 6).

2. Getting Data: ASR & OCR/HTR

Advanced textual analysis tools require a corpus of digitised texts in a Unicode format, henceforth called "e-texts". Data for Tibetan languages comes from various resources. For historical data, we mainly rely on manuscripts or xylographs, whereas new data for modern Tibetan languages can be collected in both written and audiovisual format.[1]

2.1 Optical and Handwritten Text Recognition (OCR/HTR)

In recent years computational methods have been employed in order to digitise written or printed data in a more efficient way through the use of Optical and Handwritten Text Recognition that take images with a textual component

1 In more recent years, especially Modern Standard/Lhasa Tibetan data has become available as original e-texts as well (i.e. so-called 'born-digital'), which makes it easier to study this variety. In this article, however, we focus on materials that are not yet digitised.

as an input and outputs a Unicode transcription of the text that is automatically searchable. Optical Character Recognition (OCR) was originally used for typed books (depending on white space between characters), but has now become a more general term for any form of automatic recognition of any type of texts. Handwritten Text Recognition (HTR) focuses specifically on manuscripts that are handwritten and therefore present a more challenging task. When it comes to Tibetan, progress was initially made for the (printed) *uchen* script, starting in the late 1980s in a collaboration between Bell Laboratories and the University of Virginia (Baird & Lofting 1990), and a range of Chinese OCR implementations in the following decades (e.g. Wang & Ding (2003: 5296), Drup, Zhao, Ren, Sanglangjie, Liu & Bawangdui (2010)) as well as efforts to use multi-language OCR software like Tesseract and Abbyy, and, finally, custom-made software like Namsel, which was used at the Tibetan Buddhist Resource Center (TBRC) to support the production, review, and distribution of searchable Tibetan texts at a large scale (Rowinski & Keutzer 2016). Most recently, the Buddhist Digital Resource Centre (BDRC) has collaborated with the Google Cloud Vision team, and a number of scholars, to speed up the creation of e-texts from their digital image library, focusing on OCR as well as automated post-correction.

Queenie Luo's contribution to this Special Issue gives an excellent example of a post-correction pipeline. In her article, she reports on the outcomes of a joint BDRC-Harvard-Berkeley project creating a Buddhist manuscript database using Natural Language Processing (NLP) algorithms. BDRC has digitised over 8,000 volumes of Tibetan Buddhist texts in recent years, most of which were processed by Google Cloud Vision's OCR engine. Since these OCR engines do not create a perfect e-text corpus, Luo shows how using Tibetan language models (e.g BERT, Bidirectional Encoder Representations from Transformers, GRU (Gated recurrent unit) and LSTM (Long-Short Term Memory) can not only facilitate auto-correction, but also develop further tools to retrieve named entities, identify topics as well as different genres. Her workflow combines automated computational as well as human efforts to optimise the results by first automatically mapping the OCR-ed e-texts with a Tibetan dictionary, and then using a spelling-check model to auto-correct the misspelled words based on their context. Following that, human experts validated and edited all machine-corrected texts, which were then made publicly available through the BDRC online BUDA platform.

While Google OCR[2] performs reasonably well for printed books in the standard Tibetan *uchen* script, block prints usually prove more challenging. This most commonly-found form of Tibetan literature exhibits more irregu-

2 https://cloud.google.com/use-cases/ocr

larities and digital images are not as clear as modern printed publications. Early efforts on OCR of blockprint identified the key challenges with this medium, but were not very successful at addressing them (Hedayati, Chong & Keutzer 2011). The BDRC has therefore developed a model for both layout analysis as well as OCR for block prints.[3]

Meanwhile, digitising handwritten manuscripts is an even more challenging task, due to various reasons ranging from a wide variety of cursive scripts, as well as typical features of writing styles such as overlap in characters (i.e. lack of white space separating them), lack of straight lines and the use of marginalia. In her contribution to this Special Issue, **Rachael Griffiths** discusses these challenges in the context of her work for the *The Dawn of Tibetan Buddhist Scholasticism (11th-13th)* (TibSchol) project at the Austrian Academy of Sciences, which is utilising the Transkribus platform to explore possible solutions to automate the transcription of Tibetan cursive scripts, in particular Tibetan *ume*. Transkribus is a user-friendly platform for transcribing, annotating, and searching historical manuscripts, which can be run on a local machine or in an online web interface (Muehlberger, Seaward, Terras, Oliveira, Bosch, Bryan, Colutto, Déjean, Diem, Fiel et al. 2019). She presents a detailed methodology and workflow for recognising the text fields (layout analysis) as well as different models to transcribe the text after providing the system with manual training data ("Ground Truth").

Fig. 1 – Screenshot of the OCR/HTR input view in Pecha Tools

3 https://huggingface.co/BDRC

Fig. 2 – Screenshot of the OCR/HTR correction tool with list of conventions

The results form an important step forward for anyone working with historical Tibetan documents that are handwritten, since they can form the starting point for any future HTR model that aims to transcribe any type of Tibetan cursive writing, as shown by the application of the *TibSchol* models to new materials from a very different period and region investigated in the ERC-funded PaganTibet project led by Charles Ramble. For this project, a new HTR input and correction workflow and tool was developed in collaboration with Esukhia and the BDRC to facilitate transcription. Figure 1 shows an example of an easy way to input transcriptions, while Figure 2 shows an example of the HTR tool that enables manual correction of automatically-generated transcriptions. Both tools have a built-in list of conventions for transcribing particularly difficult graphemes, ligatures and abbreviations (shown in its expanded version in Figure 2).

Finally, work on the generation of synthetic data in Tibetan cursive is undertaken by BDRC in collaboration with Esukhia & MonlamAI means future training of HTR models for these script varieties should be even more straightforward. Overall, these new methods to automatically create and correct searchable e-texts form a great leap forward in efforts to make written Tibetan materials accessible to both academics and the general public anywhere in the world through collaborations with the BDRC and its BUDA platform where new manuscript images and e-texts are (and will be) stored even after

these projects have ended.

2.2 Automatic Speech Recognition (ASR)

In addition to OCR and HTR, much progress has been made with Automatic Speech Recognition (ASR) of audio(visual) data. Developing good ASR models facilitates the creation of any Speech-to-Text (STT) or Text-to-Speech (TTS) tools. In the last decade deep-learning methods have overtaken traditional hybrid models that consisted of a lexicon with a custom phoneme set for each language, handcrafted by phoneticians. The more recent end-to-end deep-learning models, on the other hand, directly map the acoustic input onto a sequence of transcribed words without the need for force-aligned data or a language-specific lexicon.

For Modern Standard (Lhasa) Tibetan, Esukhia and MonlamAI have led efforts to collect and transcribe data that can be used to train ASR system. Their 'Tibetan Voice' data set currently contains >950 hours of Tibetan films, religious teachings, natural speech, audio books as well as children's speech. Their latest model (OpenPecha run 10) achieves results of 20.42 Character Error Rate (CER) on the benchmark.[4] Based on these, MonlamAI have also built Speech-to-Text and Text-to-Speech tools, which can transcribe recordings or produce spoken Tibetan from text online.[5]

For non-English ASR in general, but in particular for any non-standard Tibetan variety for which no large transcribed and time-aligned audio datasets exist, creating high-quality end-to-end STT systems has been challenging until recent developments in multilingual deep learning. Baevski, Zhou, Mohamed & Auli (2020) show that their Wav2Vec2 model, which learns representations from speech audio alone can outperform earlier methods when it is fine-tuned on transcribed speech in any target language. Similarly, OpenAI's Whisper trained on 680k hours of multilingual web data (about a third of which is non-English) has enabled transcription in multiple other languages. Since the proportion of English data is much larger in Whisper than in Wav2Vec2, the latter proves more successful for finetuning languages like Tibetan. Because most of the training data in these models consists of European languages, transcription in a regular, romanised script with a straightforward 1-to-1 mapping of sounds and graphemes is easier for these models. Standard (Lhasa) Tibetan audio is generally transcribed in Tibetan Unicode script, which even in its romanised (Wylie) conversion is far removed from its pronunciation. Fine-tuning of these ASR systems for languages like Tibetan is

4 This dataset and their models can be found on https://huggingface.co/openpecha.

5 Speech-to-Text: https://monlam.ai/model/stt and Text-to-Speech: https://monlam.ai/model/tts.

therefore most successful when used in combination with language-specific pre- and post-processing tools that can convert scripts and use language-specific dictionaries and spell-checkers.

The real strength of these multilingual models, however, lies in their ability to enable fine-tuning of extremely low-resource and endangered languages, like most non-standard modern Tibetan varieties. For example, O'Neill, Meelen, Coto-Solano, Phuntsog & Ramble (2023), based on earlier work on endangered languages by Coto-Solano (2021) and Coto-Solano, Nicholas, Datta, Quint, Wills, Powell & Feldman (2022), show that fine-tuning a Wav2Vec2 model for Dzardzongke (South Mustang Tibetan) can be particularly useful in language preservation, as it forms an efficient way to address the well-known transcription bottleneck in endangered language documentation (Shi, Amith, Castillo García, Sierra, Duh & Watanabe 2021). Meelen, O'Neill & Coto-Solano (2024) demonstrate that results can be further improved through transfer learning (i.e. using converted Standard Lhasa Tibetan data to enhance the dataset) as well as signal and output modifications. For example, pitch and amplitude modifications yield Character Error Rates (CER) of <10 with transcribed input of less than two hours of Dzardzongke data. Adding a post-correction dictionary (even just one built-up automatically from just three hours of input data) further improves results with a CER of 8 and a Word Error Rate (WER) of 32. These results are extremely promising for other non-standard varieties of Tibetan, especially those that are in danger of disappearing in the near future.

3. *Linguistic Annotation*

Transcription of materials is generally not sufficient for research into language variation and change, or any other form of linguistics. Especially for historical stages of the language, where native speakers are not available, it is essential to have access to well-annotated corpora to get the most out of scarce data. Linguistic annotation can also facilitate research beyond linguistics such as history and religious studies (cf. Krishna, Vidhyut, Chawla, Sambhavi & Goyal (2020) for an investigation of large, annotated religious corpora) or literature (cf. Reiter, Gius, Strötgen & Willand (2017) on performance gains in finding narratives structures when the corpus is accurately annotated). In general, good word and sentence segmentation is often essential to feed into off-the-shelf digital humanities tools for topic modelling, document classification, information retrieval, etc. (see also Section 4 below).

3.1 Normalisation, Tokenisation & Segmentation

Linguistic annotation can be done on various levels, from surface-level normalisation, tokenisation and (sentence) segmentation to mid-level morphosyntactic annotation as well as the addition of semantic and pragmatic features in deeply-annotated corpora. Especially in the absence of a reliable automatic lemmatisation tool, which groups together different inflected or conjugated forms of the same word, it can be useful to preprocess historical texts by normalising the orthography and/or standardising certain aberrant features. For Old and Classical Tibetan, for instance, this can be done using Faggionato & Garrett's 2019 Constraint Grammar Formalism. When spelling variation is the focus of research, these steps can (and should, probably) be skipped unless links to the original versions are kept, e.g. through multi-layered XML or JSON formats that preserve diplomatic transcriptions alongside normalised forms. Preprocessing and normalisation specifically, however, is very beneficial for any subsequent annotation tasks, such as segmentation, Part-of-Speech (POS) tagging or parsing.

Tokenisation (splitting into meaningful words or tokens) and sentence segmentation are non-trivial tasks in languages like Tibetan in which the script does not indicate meaningful word or sentence boundaries. Early tokenisation attempts focusing on meaningful linguistic units in particular include the syllable-tagging and recombination method developed by Meelen & Hill (2017), building on earlier work by Garrett, Hill, Kilgarriff, Vadlapudi & Zadoks (2015). Since the Tibetan script marks syllable boundaries, either by a *tsheg* ་ or by a *shad* །, these can be used to automatically split syllables. To facilitate multisyllabic meaningful units or 'words' alongside monosyllables, Meelen & Hill (2017) recast tokenisation as a syllable-tagging task with labels for beginning, middle and end syllables with a postprocessing procedure that combines these into meaningful linguistic units. More recent Tibetan tokenisers such as OpenPecha's Botok[6] use dictionaries to split a text into meaningful words or tokens. Depending on the specific Tibetan variety, the wordlist can be adjusted to get better results for different dialects. In addition to word segmentation, Botok also has the option to split sentences and/or paragraphs. Similarly, the ACTib segmenter and POS tagger[7] can do both word and sentence segmentation for Old and Classical Tibetan, using a combination of the Botok tokeniser and a syllable-tagging method with a number of post-processing rules that focus on getting meaningful linguistic segments on the word and sentence level (cf. Meelen, Roux & Hill (2021) and Faggionato, Hill & Meelen (2022)). Consistent and well-thought-through normalisation &

6 https://github.com/OpenPecha/Botok
7 https://github.com/lothelanor/actib

segmentation are often essential to improve word embeddings and large language models (like those forming the basis of tools like ChatGPT, cf. section 4 below). Sentence segmentation in particular is also crucial for the development of well-working machine translation tools.

3.2 Adding morphosyntactic and other information

For more sophisticated linguistic analyses, normalisation and segmentation of the data are not nearly enough. Adding detailed and reliable morphosyntactic information is not only useful for both synchronic, comparative and diachronic linguistic research, it can also enhance other NLP tools such as word embeddings (cf. Garcia-Silva, Denaux & Gomez-Perez (2021)).

In a series of papers, Edward Garrett, Nathan Hill and Abel Zadoks and colleagues presented one of the first attempts of adding morphosyntactic tags to each token (i.e. Part-of-Speech 'POS' tagging) of a small Classical Tibetan corpus . They used a rule-based tagger to disambiguate Tibetan verb stems Garrett, Hill & Zadoks (2013) and POS tag four Classical Tibetan texts (Garrett, Hill & Zadoks (2014) and Garrett et al. (2015)). They present a detailed set of morphosyntactic tags, going much beyond the usual set of 10-15 Universal Dependency tags[8] to facilitate more detailed linguistic research. Meelen & Hill (2017) built on this first manually-corrected, POS-tagged corpus to train a memory-based tagger, achieving 95% Global Accuracy in a ten-fold cross-validation with a tagset consisting of 79 morphosyntactic tags, which Faggionato & Meelen (2019) extend to Old Tibetan as well.[9] Meelen et al. (2021) optimise the annotation pipeline and test neural-based methods for annotation as well, while Meelen & Roux (2020a) focus on adding crucial metadata as well as constituency parses (i.e. syntactic information) to a very large diachronic corpus (Meelen & Roux 2020b). Although this corpus is not manually corrected yet, with over 166 million tokens from over 5000 texts across 11 centuries, it has opened up a wide range of research opportunities for anyone with an interest in the history of the Tibetan language. The latest version of the above-mentioned ACTib POS tagger furthermore includes additional morphological information for over 100 specific Tibetan auxiliary and light verbs to facilitate more complex diachronic linguistic research in particular.

In his contribution to this Special Issue **Christian Faggionato** demonstrates how to add syntactic information in the form of dependency parses to historical Tibetan texts. He shows how to implement a rule-based dependency parser written in the Constraint Grammar (CG-3) formalism to create the first Classical Tibetan treebank that can be part of the Universal Depen-

8 https://universaldependencies.org/u/pos/
9 https://zenodo.org/records/4727552

dencies (UD) collection.[10] Having syntactic relations encoded (either in dependency or in hierarchical phrase-structure format) facilitates more complex linguistic annotation in other domains too. Faggionato et al. (2022), for example, show how semantic and information-structural annotation can be added to Tibetan corpora by making use of an even further extended POS tag set and with syntactic annotation, Darling, Meelen & Willis (2022) show that this can be used for coreference resolution tracking noun phrases in Early Irish, which can be transferred to languages like historical Tibetan where omission of arguments is prevalent too. With categories like the animacy of noun phrases as well as the distribution of foci and topics in the sentence, more fine-grained and complex questions on the emergence of egophoric and/or switch-reference marking can be answered (cf. Meelen & Hill (2023)).

Linguistic annotation is not just beneficial to linguistics, but also forms a stronger basis for other research in a wide range of fields by enhancing opportunities for Information Retrieval, discussed in the next section.

4. Information Retrieval

Information Retrieval (IR) is the NLP task of gathering specific information from a corpus, based on any research question. It can range from simple queries like "Which people or place names can be found in which text?" to more complex tasks like Text Classification and identifying parallel content, but not necessarily identical passages in different corpora.

4.1 Named-Entity Recognition

The first question can be addressed through automatised Named-Entity Recognition (NER), i.e. adding appropriate labels to proper names, organisations, dates, institutions and other 'named entities' like 'Tashi', 'Lhasa', 'Tibetan New Year' or 'United Nations', etc. Because the Tibetan Unicode script does not use capital letters, detailed morphosyntactic annotation, e.g. distinguishing proper nouns like the personal name 'Nyima' from the regular noun 'nyima' meaning 'sun', is essential to facilitate automatic NER (cf. Suzuki, Komiya, Sasaki & Shinnou (2018)). This can be used in combination with Semantic Role Labelling (SRL), which identifies the relations with other named entities and/or verbs in the sentence, e.g. 'Tashi' is a personal name + the undergoer ('patient') of the action in the sentence 'Tashi was pushed aside by his brother.' (Zhang, Xia, Zhou, Jiang, Fu & Zhang 2022). In addition, 'his brother' can be linked to Tashi as a close family relationship. This type of information is not

10 https://universaldependencies.org/

just beneficial for linguists, but crucial for anyone doing historical research if NER and SRL annotation is provided for diachronic corpora in particular. In addition, scholars of religious studies, sociology and anthropology can benefit from this type of richly-annotated data as it enables them to extract named entities together with geographic and temporal indicators from a body of texts as well as what roles the protagonists potentially played and how they were related to each other.

For Tibetan, Liu & Wang (2018) presented one of the first NER methods, but acknowledge that the lack of good training data hindered progress. Barnett, Faggionato, Meelen, Yunshaab, Samdrup, Hill & Diemberger (2021a) and Barnett, Hill, Diemberger & Samdrup (2021b) aimed to remedy that presenting the a detailed annotation scheme of 17 Named-Entities ranging from the more conventional DATE, PLACE and PERSON to more specific TITLE, RELIGIOUS ORGANISATION and IDEOLOGY as well as a unique new set of training data consisting of almost 10k annotated terms.[11]

4.2 Text similarity & classification

A real breakthrough in the field of NLP came with development of computational methods that could 'understand', not just in frequencies, forms and structures, but also *meaning*. To get closer to letting computers gain insight into distributional semantics, i.e. deriving the meaning of words from their context, Mikolov, Yih & Zweig (2013) developed so-called 'word embeddings' using Word2Vec, a neural network-based algorithm that learns numerical representations of words. For historical languages and data sets with more orthographic variation, Meta built a character-based extension of this,[12] which Meelen (2022) used to train the first semantic model for Classical Tibetan.[13] Word embeddings are representing words for text analysis in the form of real-valued, static vectors of numbers, that can be extended to dynamic models and full-blown Language Models if enough data (i.e. billions of words with little or no orthographical variation) are available. With larger amounts of data from (Early) Modern Tibetan, for example, Engels, Erhard, Barnett & Hill (2023) developed a Language Model for SpaCy.[14]

These numerical representations of large amounts of texts are useful for more complex NLP tasks since the come closer to a semantic model of the language. Even when only smaller amounts of data are available and large language models are not an option, well-curated static word embeddings can

11 https://zenodo.org/records/4536516
12 https://fasttext.cc/
13 https://zenodo.org/records/6782247
14 https://zenodo.org/records/10148636

Fig. 3 – Buddhanexus text-view display with a Tibetan text on the left hand side, called inquiry text, a match in Sanskrit and a match in Tibetan in the middle column and the Tibetan text that has a match with the inquiry text on the right hand side, called hit text.

facilitate tasks like text classification, as Meelen (2022) shows in a pilot study that aims to classify texts based on different types of religious features, ranging from Buddhist to Bön and Pagan, on the other end of the scale. In addition to finding differences, both word embeddings and large language models help Information Retrieval tasks like finding textual similarity. Felbur, Meelen & Vierthaler (2022) show that if Classical Tibetan embeddings are combined with Classical Chinese models, it is possible to retrieve similar passages from Tibetan canonical texts based on Chinese input.

On a larger and user-friendly scale, the Khyentse Center for Tibetan Textual Scholarship, University of Hamburg launched the BuddhaNexus platform shown in Fig. 3.[15] This is a text-matching database that provides advanced functionality for the research of intertextual matches in Pāli, Sanskrit, Tibetan, and Chinese. BuddhaNexus provides not only a huge dataset of more than 120 million textual matches, but also advanced filter and different display options to make working with such a large dataset as comfortable as possible.

On the technical level, BuddhaNexus uses word embeddings in combina-

15 https://buddhanexus.kc-tbts.uni-hamburg.de/

tion with *k* nearest neighbor search (kNN) in order to calculate intertextual links within a large textual database efficiently (cf. Nehrdich (2020)). BuddhaNexus also features bilingual matches between Sanskrit texts and their Tibetan translations. These matches have been created using a combination of contextual word embeddings of the BERT type and vector-based sentence alignment (cf. Nehrdich (2022)). Since its launch, a number of studies have appeared that used the unique capabilities of BuddhaNexus in order to conduct intertextuality research on a scale that was not possible before (e.g. Dorjee (2021), Almogi (2022a), Almogi (2022b), Cheung (2023), and Nehrdich (2023)). In addition to researchers at western institutions, Buddhanexus is also used extensively by Tibetan khenpos.

5. *Machine Translation*

When it comes to automatic translation, in April 2023, the Dharmamitra project started at the Berkeley AI Research Lab (BAIR), UC Berkeley, under the guidance of Kurt Keutzer together with Sebastian Nehrdich. The goal of Dharmamitra is the creation of a range of Large Language Model enabled tools to help academics and translators interact with material in Tibetan and other Classical Asian languages. These include machine translation, semantic search, and Question/Answer systems. The first outcome of the Dharmamitra project, the Monlam-Mitra machine translation model, was made public through the MonlamAI website in Autumn 2023.[16]. This model is based on Meta's *No Language Left Behind* (NLLB Team 2022) Large Language Model, which was then finetuned on more than two million Tibetan-English sentence pairs collected by the Monlam AI team. Figure 4 shows the Monlam AI website providing Tibetan to English machine translation. In February 2024, the Dharmamitra team created a new version based on MADLAD-400 (Kudugunta, Caswell, Zhang, García, Choquette-Choo, Lee, Xin, Kusupati, Stella, Bapna & Firat 2023), which, in addition to English, also supports translation from Sanskrit, Chinese and Pāli into and from Tibetan. Figure 5 shows how this model provides translations from Classical Buddhist Chinese into English. The Tibetan-English model is currently accessible in interactive applications.[17]

A natural question is: What does this tool offer its users? Before its public release, feedback on Monlam-MITRA was solicited from twenty experienced translators and scholars. There is a natural positive confirmation bias in those who responded, as those who saw no particular value were unlikely to respond. First it should be made clear that no one indicated that the resulting machine translations were immediately usable. Everyone felt errors had to be

16 https://monlam.ai/

17 See monlam.ai and dharmamitra.org for the multilingual model.

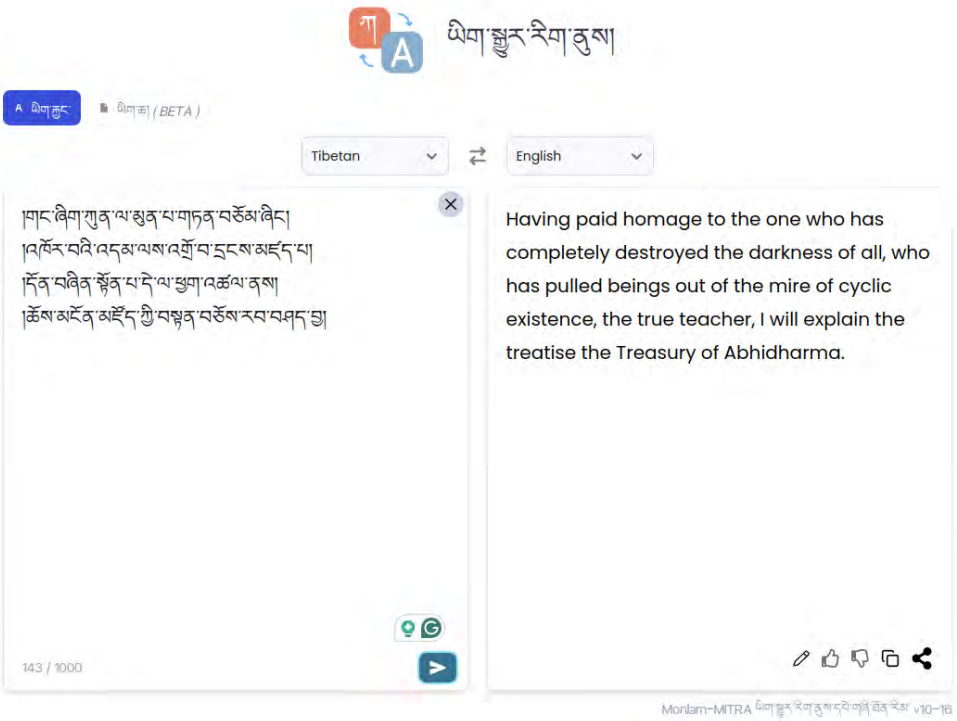

Fig. 4 – monlam.ai machine translation from Tibetan into English.

Fig. 5 – dharmamitra.org machine translation from Chinese into Tibetan.

corrected and translations reworked. Nevertheless, many translators felt that the initial translation gave a great point of departure. Surprisingly, even experienced translators found consistent productivity gains using the Monlam-MITRA translation tool. They found it very easy to quickly identify mistakes in the machine translation and move on. In addition to being faster in translation, scholars found that the interactive translation capability, with instantaneous results, made it easy to navigate unfamiliar literature. One translator, working closely with ethnic Tibetan translators, observed that Tibetans with modest English skills greatly benefited from the tool as well, making it easier for them to obtain a draft translation in the target language. Another encouraging sign of use is that there are 20,000 individual translations done per day now.

Regarding the shortcomings of the system, the most recurrent errors were incomplete translations (i.e. portions of the original entirely missing in the translation), oscillatory hallucinations in which portions of translations are repeated in a fugue-like manner (e.g. "May all sentient beings find comfort and joy. May all sentient beings find comfort and joy"). Translations of material with missing training data, such as particular types of Tibetan medicine, were non-sensical. All of these issues are fairly straightforwardly addressable, however, and improvements have been seen even over the last couple months.

6. *Learning & Editing*

In addition to the above-mentioned NLP applications, Tibetan studies is also moving forward in areas of language learning as well as text collation and editing. In his contribution to this Special Issue, **Dirk Schmidt** introduces *Dakje*,[18] a Tibetan education tool. Since pronunciation of most Modern Tibetan varieties is far removed from the orthography due to diglossia in the spoken and written language (Ferguson 1959), he explains that learning how to read and write Tibetan is a challenging task, with high rates of illiteracy from 21% to 34% (Reddy & Bhole 2023, Textor 2022). Building on earlier work (Schmidt 2020), he shows how *Dakje* aims to address these issues by unpacking the learning-to-read process in Tibetan step-by-step. Using two Esukhia speech corpora (the Nanhai Corpus (Esukhia 2020) and The Children's Speech Corpus (Esukhia 2022a)), he proposes a new data collection technique for writing beginning reading material, the creation of story-specific 'mini speech corpora', and how to use word segmentation in both *Dakje* (Esukhia 2022b) and *Botok* (OpenPecha 2023) for editing and reading level identifica-

18 https://dakje.io/

tion. This provides readers who wish to write, edit, or analyse early reading materials with the practical information, tools, and resources they need.

When it comes to reading and editing historical texts in digital environments, recent years have seen progress here too in a variety of ways. First, an ever-increasing number of digital images and eTexts are available through the Buddhist Digital Resource Center's (BDRC) BUDA platform.[19] These texts are not only provided with detailed metadata, but also linked to other platforms through IIIF and Linked Open Data (LOD) protocols. In addition, BUDA facilitates searching not just for textual strings, but also for people, places, topics, collections, works, etc. making it the largest digital library of Buddhist and other Tibetan material in the world.

When it comes to creating editions of texts, the BDRC and Esukhia have collaborated to develop *Pydurma*, a tool that can create a clean e-text version of any Tibetan work from multiple sources.[20] It can very efficiently collate different texts (in multiple formats) and has a configurable system of weights to choose a preferred reading as the one presented in the diplomatic edition (most common reading, best OCR confidence index, conformance to spelling standards, etc.). The result is a new clean version called a 'vulgate edition'. This is useful for publishers who need clean copies of texts, but also developers who need clean data to train new computational models.

7. Conclusion and further developments

Computer-aided methods in the Humanities and Social Sciences based on Natural Language Processing (NLP) techniques have led to considerable breakthroughs in recent years. The field of Tibetan Studies has already seen a number of changes as more state-of-the-art tools have become available not just to those with a technical background, but also as user-friendly online applications that facilitate research. This brief introduction and the Special Issue in general are not meant to be exhaustive, but merely aims to provide a taste of the wide range of opportunities these tools have to offer.

Bibliography

Almogi, Orna. 2022a. Editors as canon-makers: The formation of the tibetan buddhist canon in the light of its editors' predilections and agendas. In Orna Almogi (ed.), *Evolution of scriptures, formation of canons: The buddhist case*, vol. 13 Indian and Tibetan Studies Series, 351–458. Hamburg: Department of Indian and Tibetan Studies, Universität Hamburg.

19 https://library.bdrc.io/
20 https://github.com/buda-base/pydurma

Almogi, Orna (ed.). 2022b. *Evolution of scriptures, formation of canons: The buddhist case*, vol. 13 Indian and Tibetan Studies Series. Hamburg: Department of Indian and Tibetan Studies, Universität Hamburg.

Baevski, Alexei, Henry Zhou, Abdelrahman Mohamed & Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations.

Baird, Fossey Henry S., Henry S. & P. Lofting. 1990. The typestyle jockey: Putting the horse out front in Devanagari and Tibetan. In *Nordic Institute of Asian Studies report*, 5–30.

Barnett, Robert, Christian Faggionato, Marieke Meelen, Sargai Yunshaab, Tsering Samdrup, Nathan Hill & Hildegard Diemberger. 2021a. *Named Entity Recognition (NER) for Tibetan and Mongolian Newspapers*. Poster presented at the Cambridge Language Sciences Symposium. doi:10.33774/coe-2021-xhw9l-v2.

Barnett, Robert, Nathan Hill, Hildegard Diemberger & T Samdrup. 2021b. Named-Entity Recognition for Modern Tibetan Newspapers: Tagset, Guidelines and Training Data [Data set]. doi:https://doi.org/10.5281/zenodo.4536516.

Cheung, Daisy. 2023. "Madhyamakanising" Tantric Yogācāra: The Reuse of Ratnākaraśānti's Explanation of maṇḍala Visualisation in the Works of Śūnyasamādhivajra, Abhayākaragupta and Tsong Kha Pa. *Journal of Indian Philosophy* 51(5). 611–643.

Coto-Solano, Rolando. 2021. In *Proceedings of the first workshop on natural language processing for indigenous languages of the americas. association for computational linguistics.*, 173–184. Explicit tone transcription improves ASR performance in extremely low-resource languages: A Case Study in Bribri.

Coto-Solano, Rolando, Sally Akevai Nicholas, Samiha Datta, Victoria Quint, Piripi Wills, Emma Ngakuravaru Powell & Isaac Feldman. 2022. Development of Automatic Speech Recognition for the Documentation of Cook Islands Māori. In *Proceedings of the thirteenth language resources and evaluation conference*, 3872–3882. https://aclanthology.org/2022.lrec-1.412.

Darling, Mark, Marieke Meelen & David Willis. 2022. Towards coreference resolution for Early Irish. In *Proceedings of the CLTW 4 @ LREC2022*, 85–93. European Language Resources Association (ELRA).

Dorjee, Khenpo Tashi. 2021. *Chos bzang rigs pa'i rnam dpyod | Rong zom ma hā paṇḍita'i theg chen tshul 'jug rjod byed zhib dpyad zhu dag lung khungs ngos 'dzin dang | brjod bya gnas lugs rig par rtsad zhib tshom bu |*, vol. 3 sNga 'gyur rnying ma'i zhib 'jug. Bylakuppe, Mysore: Ngagyur Nyingma Institute, Ngagyur Nyingma Research Centre.

Drup, Ngo, Dongcai Zhao, Puts Ren, Daluo Sanglangjie, Fang Liu & Bian Bawangdui. 2010. Study on printed Tibetan character recognition. In *2010 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1, 280–285. IEEE.

Engels, James, Xaver Erhard, Robert Barnett & Nathan Hill. 2023. Tibetan for Spacy 1.1 [Data set]. https://doi.org/10.5281/zenodo.10148636.

Esukhia. 2020. The nanhai corpus. https://github.com/Esukhia/Corpora/tree/master/Nanhai.

Esukhia. 2022a. Children's Stories Speech Corpus. https://github.com/Esukhia/Corpora/tree/master/Childrens_Stories.

Esukhia. 2022b. Dakje. https://github.com/Esukhia/dakje-desktop.

Faggionato, Christian & Edward Garrett. 2019. Constraint Grammars for Tibetan Language Processing. In *Nealt proceedings series 33:3*, 12–16.

Faggionato, Christian, Nathan Hill & Marieke Meelen. 2022. NLP pipeline for annotating (endangered) Tibetan and newar varieties. In *Proceedings of the workshop on resources and technologies for indigenous, endangered and lesser-resourced languages in eurasia within the 13th language resources and evaluation conference*, 1–6. Marseille, France: European Language Resources Association. https://aclanthology.org/2022.eurali-1.1.

Faggionato, Christian & Marieke Meelen. 2019. Developing the Old Tibetan treebank. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 304–312. Varna, Bulgaria: INCOMA Ltd. doi:10.26615/978-954-452-056-4_035.

Felbur, Rafal, Marieke Meelen & Paul Vierthaler. 2022. Crosslinguistic Semantic Textual Similarity of Buddhist Chinese and Classical Tibetan. *Journal of Open Humanities Data* doi:10.5334/johd.86.

Ferguson, Charles A. 1959. Diglossia. *WORD* 15(2). 325–340. doi:10.1080/00437956.1959.11659702.

Garcia-Silva, Andres, Ronald Denaux & Jose Manuel Gomez-Perez. 2021. On the impact of knowledge-based linguistic annotations in the quality of scientific embeddings. *Future Generation Computer Systems* 120. 26–35.

Garrett, Edward, Nathan W Hill, Adam Kilgarriff, Ravikiran Vadlapudi & Abel Zadoks. 2015. The contribution of corpus linguistics to lexicography and the future of Tibetan dictionaries. *Revue d'Études Tibétaines* 32. 51–86.

Garrett, Edward, Nathan W Hill & Abel Zadoks. 2013. Disambiguating Tibetan verb stems with matrix verbs in the indirect infinitive construction. *Bulletin of Tibetology* 49(2). 35–44.

Garrett, Edward, Nathan W Hill & Abel Zadoks. 2014. A rule-based part-of-speech tagger for Classical Tibetan. *Himalayan Linguistics* 13(2).

Hedayati, Fares, Jike Chong & Kurt Keutzer. 2011. Recognition of tibetan

wood block prints with generalized hidden markov and kernelized modified quadratic distance function. In *Proceedings of the 2011 joint workshop on multilingual ocr and analytics for noisy unstructured text data*, 1–14.

Krishna, Amrith, Shiv Vidhyut, Dilpreet Chawla, Sruti Sambhavi & Pawan Goyal. 2020. SHR++: An interface for morpho-syntactic annotation of Sanskrit corpora. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 7069–7076.

Kudugunta, Sneha, Isaac Caswell, Biao Zhang, Xavier García, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna & Orhan Firat. 2023. MADLAD-400: A Multilingual And Document-Level Large Audited Dataset. *ArXiv* abs/2309.04662. https://api.semanticscholar.org/CorpusID:261682406.

Liu, Fei-Fei & Zhi-Juan Wang. 2018. Active Learning for Tibetan Named Entity Recognition based on CRF. In Jinhua Du & Mihael Arcan (eds.), *Lrec 2018 workshop mlp–moment*, 18–45.

Meelen, Marieke. 2022. Tibetan word embeddings: from distributional semantics to facilitating Tibetan NLP. *International Association of Tibetan Studies - Tech Panel presentation* .

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

Meelen, Marieke & Nathan Hill. 2023. From co-reference to evidentiality: how syntax, semantics and information structure interact to create a new grammatical feature. *Himalayan Languages Symposium* doi:10.13140/RG.2.2.22106.72646.

Meelen, Marieke, Alexander O'Neill & Rolando Coto-Solano. 2024. ASR for endangered languages in Nepal. In *Proceedings of the Comput-EL workshop at the EACL*, 83–93.

Meelen, Marieke & Élie Roux. 2020a. Meta-dating the PArsed Corpus of Tibetan (PACTib). In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, 31–42.

Meelen, Marieke & Elie Roux. 2020b. The Annotated Corpus of Classical Tibetan (ACTib) - Version 2.0 (Segmented & POS-tagged) [Data set]. doi:https://doi.org/10.5281/zenodo.3951503.

Meelen, Marieke, Élie Roux & Nathan Hill. 2021. Optimisation of the Largest Annotated Tibetan Corpus Combining Rule-based, Memory-based, and Deep-learning Methods. *ACM Transactions on Asian and Low-Resource Language Information Processing* 20(1). 1–11. doi:10.1145/3409488.

Mikolov, Tomáš, Wen-tau Yih & Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational lin-*

guistics: Human language technologies, 746–751.

Muehlberger, Guenter, Louise Seaward, Melissa Terras, Sofia Ares Oliveira, Vicente Bosch, Maximilian Bryan, Sebastian Colutto, Hervé Déjean, Markus Diem, Stefan Fiel et al. 2019. Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. *Journal of documentation* 75(5). 954–976.

Nehrdich, Sebastian. 2020. A Method for the Calculation of Parallel Passages for Buddhist Chinese Sources Based on Million-scale Nearest Neighbor Search. *Journal of the Japanese Association for Digital Humanities* 5(2). 132–153.

Nehrdich, Sebastian. 2022. SansTib, a Sanskrit - Tibetan Parallel Corpus and Bilingual Sentence Embedding Model. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the thirteenth language resources and evaluation conference*, 6728–6734. Marseille, France: European Language Resources Association.

Nehrdich, Sebastian. 2023. Observations on the intertextuality of selected abhidharma texts preserved in chinese translation. *Religions* 14(7).

NLLB Team. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. *ArXiv* abs/2207.04672. https://api.semanticscholar.org/CorpusID:250425961.

O'Neill, Alexander, Marieke Meelen, Rolando Coto-Solano, Sonam Phuntsog & Charles Ramble. 2023. Language Preservation through ASR. doi:10.33774/coe-2023-rm6wq-v2.

OpenPecha. 2023. Botok. https://github.com/OpenPecha/Botok.

Reddy, Rahul K. & Omkar Bhole. 2023. Analysing China's Census Report .

Reiter, Nils, Evelyn Gius, Jannik Strötgen & Marcus Willand. 2017. A Shared Task for a Shared Goal: Systematic Annotation of Literary. In *Digital humanities*, .

Rowinski, Zach & Kurt Keutzer. 2016. Namsel: An optical character recognition system for Tibetan text. *Himalayan Linguistics* 15(1). 12–30.

Schmidt, Dirk. 2020. Grading Tibetan Children's Literature: A Test Case Using the NLP Readability Tool "Dakje". *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19(6). doi:10.1145/3392046. https://doi.org/10.1145/3392046.

Shi, J., J. D. Amith, R. Castillo García, E. G. Sierra, K. Duh & S. Watanabe. 2021. Leveraging end-to-end ASR for endangered language documentation: An empirical study on Yolóxochitl Mixtec. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics* 1134–

1145.

Suzuki, Masaya, Kanako Komiya, Minoru Sasaki & Hiroyuki Shinnou. 2018. Fine-tuning for named entity recognition using part-of-speech tagging. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, 632–640.

Textor, C. 2022. Illiteracy rate in China in 2021, by region. https://www.statista.com/statistics/278568/illiteracy-rate-in-china-by-region/.

Wang, Hua & Xiaoqing Ding. 2003. New statistical method for multifont printed Tibetan/English OCR. In *Document recognition and retrieval xi*, vol. 5296, 155–165. SPIE.

Zhang, Yu, Qingrong Xia, Shilin Zhou, Yong Jiang, Guohong Fu & Min Zhang. 2022. Semantic role labeling as dependency parsing: Exploring latent tree structures inside arguments. In *Proceedings of the 29th International Conference on Computational Linguistics*, 4212–4227.

Norbu Ketaka: Auto-Correcting BDRC's E-Text Corpora Using Natural Language Processing and Computer Vision Methods*

Queenie Luo[1]

Leonard W.J. van der Kuijp[2]
Harvard University

This paper discusses the Norbu Ketaka project, which employs the latest cutting-edge technologies in deep learning to process one million pages of Tibetan texts. This includes labeling training data, adjustments to neural-network architectures, and creating new Natural Language Processing (NLP) and Computer Vision models. Additionally, a staff administration system was designed and implemented using the Google Docs and Google Drive APIs, which facilitated the distribution of 12,000 documents to 40 part-time annotators based on their preferences and time availability, as well as tracking and recording the edited documents. We have donated one million pages of "cleaned" texts, along with the models and algorithms utilized in this project, to BDRC as the ***Norbu Ketaka*** collection which is publicly accessible on BDRC's website.[1]

1. *Introduction*

Locating exact words in ancient manuscripts is a crucial but time-consuming task for scholars in the humanities to study cultural, historical, and linguistic changes. In recent years, Optical Character Recognition (OCR) technology has greatly facilitated the progress made in the digitization of ancient

* Queenie Luo & Leonard W.J. van der Kuijp, "Norbu Ketaka: Auto-Correcting BDRC's E-Text Corpora Using Natural Language Processing and Computer Vision Methods", *Revue d'Etudes Tibétaines*, no. 72, Juillet 2024, pp. 26-42.

Listen to this article here: https://github.com/queenieluo/NorbuKetaka. We thank the Harvard China Fund for its generous sponsorship of this project. We are grateful to the Buddhist Digital Resource Center [BDRC] for their ready cooperation in providing text images and OCR-ed files, and the Department of East Asian Languages and Civilizations of Harvard University for extending additional administrative and financial support. Last but not least, we also thank Professor Kurt Keutzer of UC Berkeley for his generous support and comments on the technical aspect of this project.

1 Norbu Ketaka Collection: https://library.bdrc.io/show/bdr:PR1ER1?uilang=en

and more recent texts. The Buddhist Digital Resource Center (BDRC) [2] has archived over 15 million pages of culturally significant Buddhist works. Sponsored by Google's Tibetan OCR, BDRC digitized over 8,000 volumes of Tibetan Buddhist texts to create a searchable database. However, the digitized e-text corpus contains many errors that scholars cannot reliably use for their research. Some hand-written Tibetan graphs, such as "ཀ" and "ཏ", "ཀ" and "ཅ", are often indistinguishable even to the naked eye, and an OCR program cannot be expected to capture these nuances. Scholars in this field have to rely on their linguistic and philological knowledge to discern the correct errors based on word context and familiarity with the language. In addition, in the case of some of the texts' antiquity, most wooden block prints and manuscripts have faded ink and minor stains on pages that further challenge Google OCR's performance.

The original raw OCR output from Google's Tibetan OCR contains lots of "noise" or spelling errors due to misidentified characters in the image. For example, Google's Tibetan OCR system tended to read stains and black border lines as actual texts and produce several lines of unreadable textual noise. Current spelling-checkers cannot correct Tibetan texts, and are not designed to auto-correct OCR-ed output. The Norbu Ketaka project used the latest cutting-edge technologies in deep learning to clean one million pages of Tibetan texts, including labeling training data, tweaking neural-network architectures, and training models from scratch. We also implemented a staff administration system using Google Docs and Google Drive APIs to automatically dispatch 12,000 documents to 40 part-time employees based on their preferences and time availability, as well as track and record the edited document.

Fig. 1 shows the pipeline of this project. The raw images underwent an initial pre-processing stage using computer vision models, involving three distinct stages of rotation, border removal, and contrast adjustment, to minimize OCR errors. The images were then sent to Google OCR. Using the output from Google OCR, the system extracted each syllable's confidence score, which indicated the degree of accuracy in syllable identification. Subsequently, the system used these confidence scores to label both the images and the texts. These labeled images and texts were then inserted into a Google Docs document. A Tibetan BERT model was connected to the generated Google Docs documents to auto-correct low-level errors. The pre-cleaned documents were then dispatched to the annotators at Sichuan University via Google Drive. Once the annotators completed their corrections and uploaded the documents back to Google Drive, a staff administration program was activated to track

2 BDRC's official website: https://www.bdrc.io/

and record the returned documents against the dispatched documents to identify any missing documents. We then proofread the returned documents. Once all the documents were validated and approved, the texts were inserted into our database.

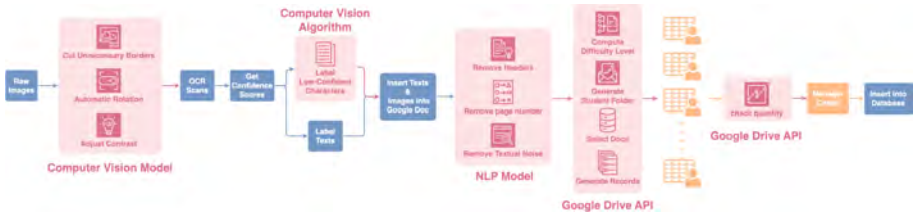

Fig. 1 – Workflow for processing images and texts, and distributing 12,000 documents to annotators.

2. *Literature Review*

The fields of computer vision and Natural Language Processing (NLP) have seen remarkable advances in recent years. In computer vision, several models have achieved notable prominence due to their remarkable performance in image recognition tasks. Convolutional neural-networks (CNNs) (LeCun, Bottou, Bengio & Haffner 1998) form the bedrock of many computer vision solutions. They are particularly effective due to their ability to process raw pixel data and learn complex patterns from images (Krizhevsky, Sutskever & Hinton 2017). Microsoft introduced ResNet (Residual neural-network), an advanced architecture built upon CNN (He, Zhang, Ren & Sun 2016). ResNet stands out for its remarkable capability to train exceptionally deep networks by incorporating "skip connections" or "shortcuts" that allow information to bypass certain layers. This approach effectively addresses the vanishing gradient problem, which can hinder training in deep networks. In addition to ResNet, other notable architectures include Mask R-CNN (He, Gkioxari, Dollár & Girshick 2017) and Imagenet (Krizhevsky et al. 2017). More recently, Facebook AI Research has developed Detectron2 (Wu, Kirillov, Massa, Lo & Girshick 2019), an advanced library that offers cutting-edge algorithms for image detection and segmentation. It supports various computer vision research projects and production applications in Facebook. In our project, we utilized the benefits of this publicly available Detectron2 model for image preprocessing. However, since Detectron2 is pre-trained on large and general datasets, it may not be perfectly adapted to domain-specific tasks. We fine-tuned this model by training it further on a smaller and task-specific dataset.

Parallelly, various Natural Language Processing (NLP) models have demonstrated robust performance in various language tasks, including spelling-error correction. The Transformer model, first introduced by Vaswani et al. in "Attention is All You Need" (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser & Polosukhin 2017), has become a cornerstone in NLP. Its architecture includes attention mechanisms that allow the model to capture contextual relationships between words and thus solve the problem of long-term dependencies for long sentences. This has led to more sophisticated Transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin, Chang, Lee & Toutanova 2018). Developed by Google, BERT is pre-trained on a large corpus of text and has been proven effective in a variety of NLP tasks. It uses a masked language model to understand word context in both directions (left and right of a word). Since the primary objective of our project is to identify and eliminate low-level errors in OCR-ed texts, such as noise. The architecture of the BERT model has been proven to be particularly effective in this regard. To leverage this efficiency, we pre-trained a BERT Tibetan model and augmented it with an additional layer to perform a binary classification task. This setup allowed us to automatically identify incorrect or noisy text sequences from Google OCR's output.

Efforts have been made to utilize computational methods for processing Tibetan texts. Meelen, Roux & Hill (2021) showcase a pipeline capable of converting Tibetan documents in plain text or XML format into a fully segmented and part-of-speech (POS) tagged corpus. Implemented on the extensive collection from the Buddhist Digital Resource Center, this study shows the rule-based, memory-based, and neural-network methods led to an end-to-end accuracy rate of 91.99% for the automatic pipeline, making the resultant corpus a valuable resource for linguists and Tibetan studies scholars. Notably, other works have also explored tasks such as Classical Tibetan word segmentation and POS tagging (e.g., Jiangdi (2003); Kang, Jiang & Long (2013); Hill & Meelen (2017), Li, Li, Wang, Lv, Li & Duo (2022), Xiangxiu, Qun, Renqing, Nyima & Zhao (2022)). An & Long (2021) address the challenges of Tibetan dependency analysis by introducing a new dataset and proposing a neural-based framework. The proposed model achieves promising performance in Tibetan dependency analysis tasks by automatically extracting feature vectors for words and predicting their head words and dependency arc types. Our project does not focus on NLP tasks of word segmentation, POS tagging, and extracting feature vectors, but aims to clean and provide high-quality Tibetan dataset that supports research, analysis, and further advancements in NLP and the broader field of Tibetan studies.

3. *Technological Advancements in pre-processing data and generating documents*

This section covers the technological advancements employed in the Norbu Ketaka project. These advancements encompass pre-processing scanned images through computer vision neural-network models, utilizing the confidence scores from Google OCR output to label OCR-ed texts and images, employing the Google Docs API to generate and manage 12,000 Google Doc documents, and training and implementing a Tibetan BERT model to automatically correct low-level errors.

3.1 Pre-processing scanned images using computer vision neural-network models

The image pre-processing step involved three distinct stages: rotation, border removal, and contrast adjustment. The raw images used in the project were scanned copies of manuscripts or books, resulting in some images not being perfectly oriented with the vertical axis of the page. To tackle this issue, we developed a CNN rotation model specifically designed to rotate tilted images. The model architecture consisted of two convolutional layers, two max-pooling layers, and two fully connected layers. For training data generation, we selected 200 perfectly aligned images and employed PyTorch's torchvision library[3] to randomly rotate these images and recorded the rotation angles. The rotated images served as the inputs for the model, and the *negative* values of the rotation angles served as both the training labels and the model's outputs. To evaluate the accuracy of the rotation model, we utilized the Mean Squared Error (MSE) metric, measuring the discrepancy between predicted rotation angles and ground truth angles. The model achieved a low MSE, ranging between 0-0.5 degrees, after training for 100 epochs.

Next, we fine-tuned Facebook Research's Detectron2 (Wu et al. 2019) for the purpose of removing unnecessary borders in the images. Cutting the borders was crucial due to the significant number of errors and noise they introduced to the OCR output, particularly in the case of tainted images. For fine-tuning Detectron2, we utilized only 150 training labels. An example of the training data can be seen in Fig. 2, where the target bounding box was highlighted in red against a transparent background. These 150 training labels were hand-labeled. This model obtained 99.1% accuracy after 800 epochs of training. The errors were generally attributed to images that had fainted inks.

3 The torchvision.transforms.functional module in PyTorch's torchvision library provides a set of functional image transformation operations for data augmentation and manipulation, which can be accessed here: https://pytorch.org/vision/stable/transforms.html

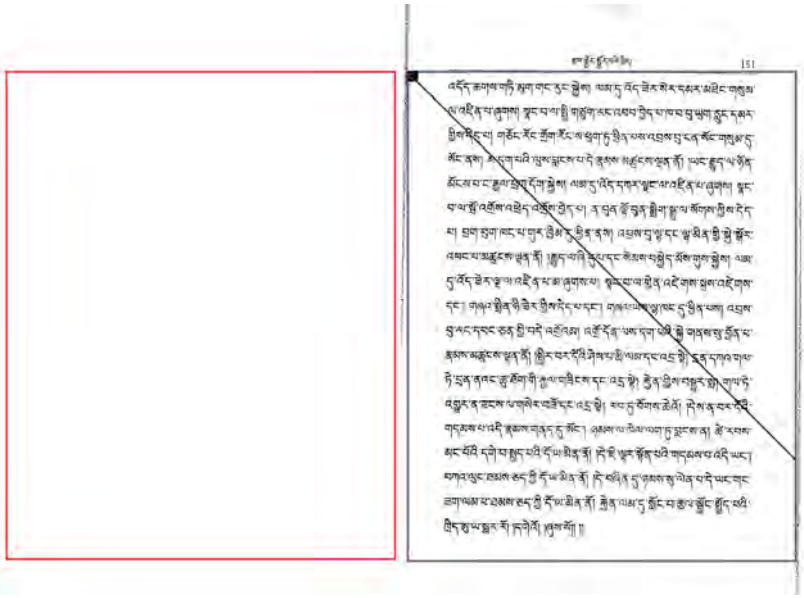

Fig. 2 – Training label for fine-tuning Detectron2: The red box on the left represents the actual training label, while the image overlaid with a bounding box on the right is for illustrative purposes.

Lastly, before sending them to Google's OCR system, we applied further contrast adjustments to the cropped images using the torchvision library.

Overall, the amount of OCR errors dramatically decreased using the pre-processed images by a factor of 2 - 10 (depending on the quality of the original image), compared to the amount of OCR errors using the original scanned images. Fig. 3 and Fig. 4 show comparisons between the original and pre-processed images.

3.2 Using the confidence score from Google OCR output to label OCR-ed texts and images

Next, we forwarded the pre-processed images to the Google OCR system. The OCR system's output included a confidence score for each identified character. Based on human validation, characters with confidence scores exceeding 80% exhibited nearly perfect accuracy, while characters with lower confidence scores, around 30% for example, often required correction. Thus, we utilized these scores as valuable indicators to label low-confident characters

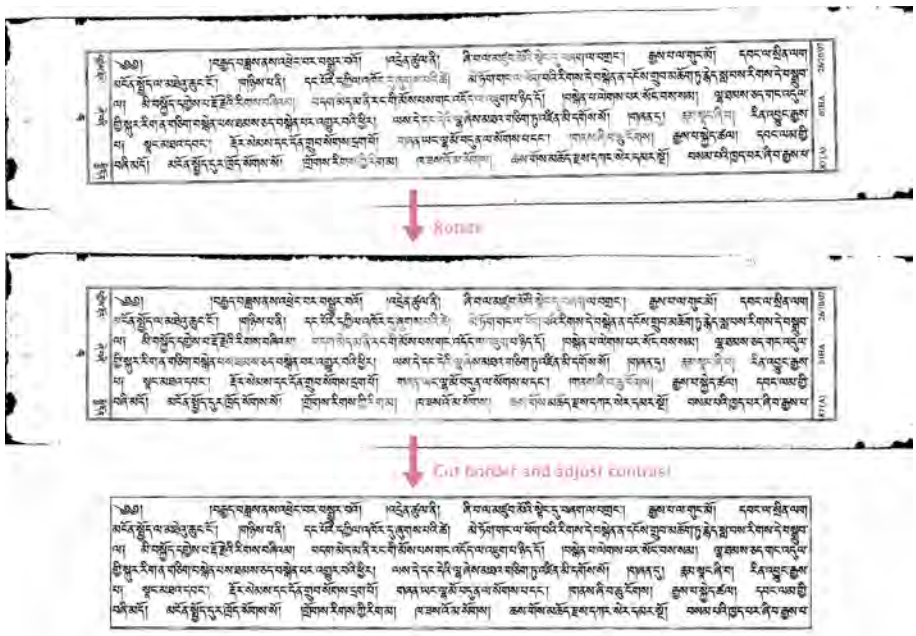

Fig. 3 – Comparison of original and pre-processed Images. Pre-processed images lead to a 2-10 fold decrease in OCR-ed errors.

and informed human annotators to focus on these characters (see Fig. 6).

However, it is important to note that the 80% confidence score threshold is not always an optimal choice for all texts. The quality of the texts could vary greatly, and some texts might have characteristics that made them more difficult for OCR to recognize accurately. For example, texts with brownish paper, red ink, or faded characters produced lower confidence scores even though the OCR output was correct (see Fig. 5). Additionally, Google OCR might have faced difficulties in identifying texts that contained a mixture of multiple languages. This was particularly true for bilingual dictionaries where Google OCR was unable to preserve the original two-column layout and accurately recognized Chinese characters. To solve these problems, we manually filtered out some texts and adjusted the confidence score thresholds for atypical texts.

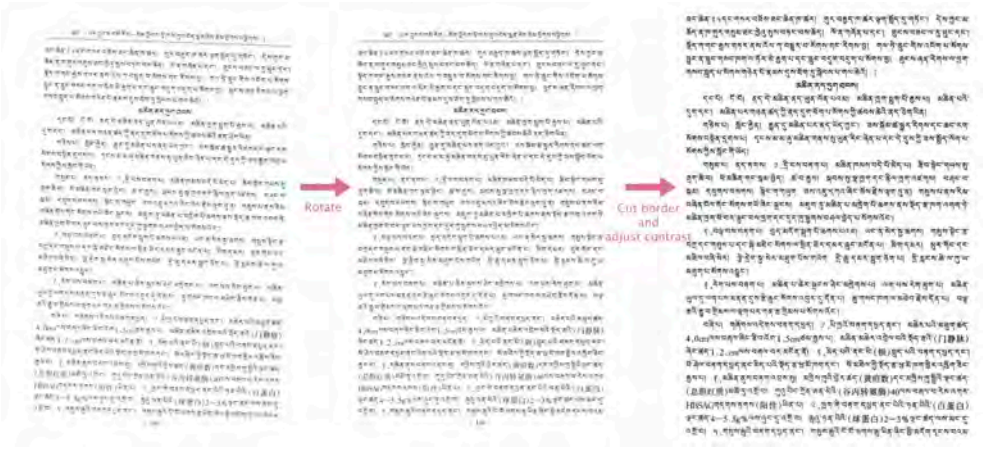

Fig. 4 – Comparison of original and pre-processed images. Pre-processed images lead to a 2-10 fold decrease in OCR-ed errors.

3.3 Using Google Docs API to generate and manage 12,000 Google Doc documents

To process one million pages of Tibetan texts, we had to generate around 12,000 Word documents, with each document containing a maximum of 100 pages of text. We set the maximum page number as 100 to avoid the document size growing too big. Managing such a large number of documents manually is not feasible. Thus, we utilized the Google Docs API[4] which allows for programmatic access and manipulation of Google Docs. By leveraging this API, we significantly improved the efficiency and accuracy of creating and managing large-scale data.

The Google Docs API allowed us to automate the creation and editing of Google Docs documents. We were able to seamlessly insert images and their respective OCR-processed texts into a single document, situating the image above its corresponding text (see Fig. 6). This resolved the issue of managing disjointed sets of images and texts. We also formatted low-confident characters in red to facilitate annotators to solely on the highlighted text within the documents, and edited the document's background color to light green to comfort their eyes.

4 Google Docs API is an interface provided by Google that allows developers to programmatically access and manipulate Google Docs documents, enabling automated document creation, editing, and collaboration: https://developers.google.com/docs/api.

Fig. 5 – Sample images that tend to produce low confidence scores.

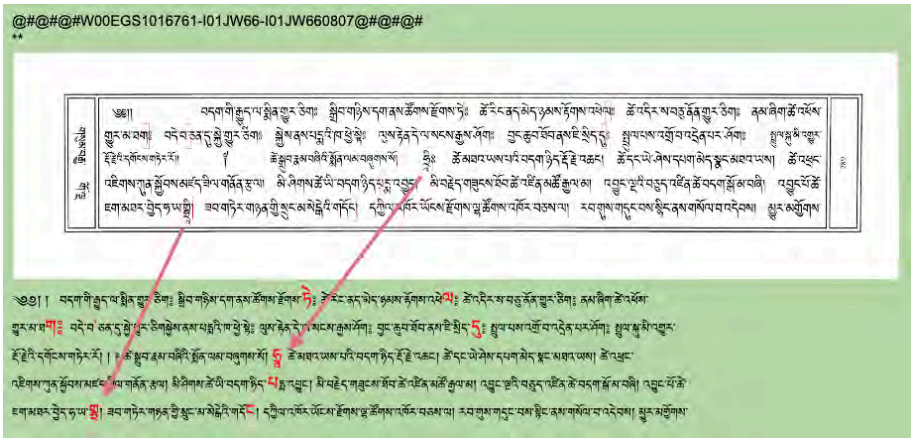

Fig. 6 – Screenshot of a Google document we generated using Google Docs API, in which the low-confidence characters are labeled in both the images and texts.

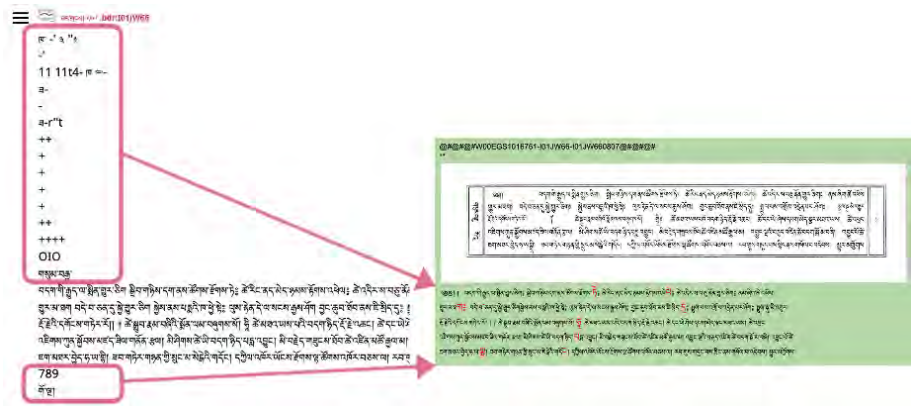

Fig. 7 – A comparison between BDRC's current website and our pre-processed texts. The image on the left is a screenshot from BDRC's website (acquired on 06/22/2022), and the image on the right is a screenshot of a Google document, in which all texts are pre-processed using our AI models.

Furthermore, we used Google Docs API to integrate the 12,000 generated documents with Google Drive and Google Cloud Storage[5] through Google Colab[6]

5 Google Cloud Storage is a scalable and durable object storage service provided by Google Cloud Platform, offering secure and flexible storage for various types of data with high availability and global accessibility: https://cloud.google.com/storage.

6 Google Colab is a cloud-based Jupyter notebook environment provided by Google that allows

and Google Drive API[7] to ensure every single document is ordered, counted, and proofread.

3.4 Training and using a Tibetan BERT model to auto-correct low-level errors

As mentioned above, the pre-processed images were able to significantly reduce noise and errors in the OCR output. However, we observed remaining errors that could be further edited using language models, such as page numbers, headers, and low-level spelling errors like repeated instances of "˝" and phrases that are mixed with non-Tibetan scripts. Therefore, we decided to train a Tibetan language model to auto-correct these low-level errors. It is important to note that in order to avoid the accidental correction of spelling variants, our language model does not intend to correct high-level mistakes in the main texts.

As introduced in the literature review section, the BERT architecture (Devlin et al. 2018) has been proven to be particularly robust in spelling error correction. Therefore, we chose the BERT architecture to implement a Tibetan spelling error correction model. We first pre-trained the BERT model using an architecture with 6 hidden layers, 12 attention heads, a maximum sequence length of 256 in which the tokenizer uses the byte pair encoding tokenizer algorithm to obtain subword units. The final tokenizer has a vocabulary size of 8,000. The training data consisted of 103 volumes of the Derge Kangyur and 213 volumes of the Derge Tengyur that are publicly accessible from the GitHub repository (Esukhia 2023). We then attached a fully connected layer to the pre-trained model to perform a binary classification task, wherein label 0 indicated that the given sentence was a standard Tibetan sentence, and label 1 indicated that the sentence was noise, a page number, or a header. Table. 1 presents examples of the two classes we wish to classify.

Fig. 8 shows the ROC curves of our Tibetan BERT model, which indicate the true positive rates versus the false positive rates for thresholds ranging from 0 to 1. BERT_fcc1, BERT_fcc2, and BERT_fcc3 represent three different fully connected layer configurations, with 64, 128, and 256 neurons, respectively. As indicated by the curves, the fully connected layer with 64 neurons presents the most favorable ROC curve, and we therefore adopted this model for our project. In our scenario, a false positive suggests the model erroneously classifying a correct sentence as a noisy one, leading to an un-

users to write, run, and collaborate on Python code seamlessly, with access to free GPU and TPU resources: https://colab.research.google.com.

7 Google Drive API is a programming interface provided by Google that enables developers to integrate their applications with Google Drive, allowing for file management, sharing, and synchronization: https://developers.google.com/drive/api.

Text	Label
བཅུ་གཅིག	1
Ir꤬ཆ꤬rཀྱིཟིཐ	1
ལ114rat	1
ཇཡཀདཆཀྱིFཀwwr	1
བའི་གནས་སུ་ཀུར་ཡ་ཞིག་ཏུ་སྣང་ངོ་།།	0
ཞལ་ནས་ཀྱང་རྨི་ལྟས་འདི་དག་ལ་དཔག་ན་ཙང་ལོང་ཐེག་དང་""གཉིས་ཀྱི་ཐོག་ནད་ཚབ་ཆེན་པོ་རླུང་རིང་བྱང་བ་རིམ་གྱི་ ུ ེ ོ	0
དང་སྲས་ཆེ་བ་གཉིས་གཤེགས།བཙུན་མོ་དང་སྲས་གཞན་རྣམས་མ་མཐུན་པས་མངའ་འབངས་བགོ་ལ་འཁའ་བྱས་ཏེ་སོ་སོར་གྱེས་ ུ ེ ོ	0
ངེད་ཀྱང་ལུགས་གྱི་ལོ་གཉིས་ཀར་ན་ཚ་ཐུ་མོ་བྱང་སོང་བས་ཡོན་མཆོད་གཉིས་ཀར་གེགས་མཆག་ཆེ་བའི་ལྟས་སུ་མངོན།།	0

Table 1 – Label 0 indicates the given sentence is a normal Tibetan sentence, and label 1 indicates the given sentence is either noise, page number or header.

intended deletion of an accurate sentence in the document; this forces annotators to manually retype the sentence. On the other hand, a false negative occurs when the model misclassifies a noisy sentence as a correct one, leading to the retention of a noisy sentence in the document; this requires annotators to manually remove it. In essence, erroneously deleting a correct sentence (a false positive) imposes a greater cost than failing to eliminate a noisy sentence (a false negative). Therefore, we set a high threshold of 0.8 for the model to maintain a low false positive rate. As can be observed from the ROC curve, with a threshold of 0.8, the false positive rate is close to 0.

After training and testing the model, we integrated it with our generated Google Docs documents via APIs to automatically identify and remove erroneous sentences. Prior to the automatic cleaning implemented by BERT, our annotators spent, on average, 20 minutes on the removal of these sentences in each document. With the integration of BERT, our annotators could now focus solely on the main texts.

The integration of computer vision models for image pre-processing and a Tibetan BERT model to address low-level errors led to an average reduction of errors in OCR output by 80% - 90% across various collections of Tibetan texts, significantly reducing labor and costs. In the next section, we discuss the challenges and solutions related to staff administration that we encountered during this project.

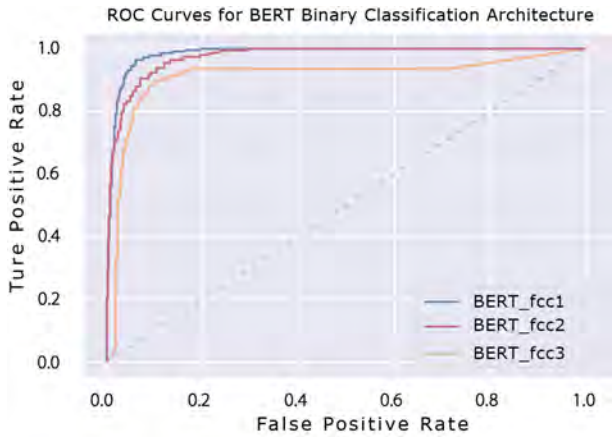

Fig. 8 – ROC curves for BERT binary classification architecture.

4. *Staff Administration*

Due to the vast number of documents and a large number of part-time employees, this project was not only a technical AI project. It was also a social-personnel management project. That is, we not only needed to ensure the high quality and accuracy of the models but also had to bear responsibility for recruiting and supervising the annotators at Sichuan University.

Our team was composed of 40 part-time employees. The generated documents presented varying amounts of errors - some demanded significantly more time to process than others, and each employee had different preferences and time availability. As a result, randomly distributing documents was impractical. To establish a fair and manageable workflow, we classified our documents into different difficulty levels and associated each level with a unit price. The difficulty levels were determined based on the count of low-confidence characters in the document. Table. 2 displays the estimated completion time corresponding to different document levels. Generally, level-0 documents took about 10 minutes to finish, while level-7 documents took 45 to 60 minutes. Employees were then assigned documents according to their preferred difficulty level.

Next, distributing the appropriate number of documents that meet employees' preferences is a complex mathematical problem. To address this issue, we implemented a system that utilized the Google Drive API to automatically dispatch, track, and record documents. Similar to the Google Docs API,

Level	Estimated Time	Number of Low-Confident Characters
0	10 - 13 mins	0 - 500
1	13 - 15 mins	500 - 1000
2	13 - 15 mins	1000 - 1500
3	15 - 17 mins	1500 - 2000
4	17 - 20 mins	2000 - 2500
5	20 - 30 mins	2500 - 3000
6	30 - 45 mins	3000 - 3500
7	45 - 60 mins	3500 - 4000

Table 2 – Estimated time required for different document levels. Level-0 documents typically take approximately 10 minutes to complete, whereas level-7 documents may require 45-60 minutes to complete.

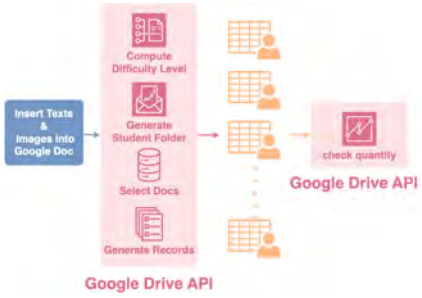

Fig. 9 – Staff administration pipeline to dispatch, track and check 12,000 documents.

the Google Drive API is capable of handling large quantities of files, making it ideal for organizing extensive document collections. Based on each employee's requirements, our system was able to automatically create a folder using the employee's name, calculate and select documents that align with the employee's preference, copy the relevant documents to the employee's folder, and generate an Excel sheet that records all the document names, their difficulty levels, unit prices, and the total payment for the entire folder (see Fig. 9).

Once the employees had finished editing and returned their documents, the system checked the dispatched documents against the returned documents to identify any missing documents. As depicted in Fig. 10, blue represents completed folders, while red represents incomplete ones. As indi-

Fig. 10 – Status of employees' progress: The graph showcases the status of document completion, where blue present the completed documents and red denotes incomplete ones.

cated by the graph, one employee inadvertently omitted a document from their folder. In short, our staff administration system guarantees a seamless process for accurately distributing and collecting 12,000 documents, ensuring equitable compensation, personalized document selection based on employee preferences, and no missing documents.

5. Conclusion

In conclusion, we implemented two systems for this project: a technical system that integrated NLP and computer vision models to pre-clean Tibetan texts, and a staff administration system designed to dispatch, track, and record documents. We have donated one million pages of "cleaned" texts, along with the models and algorithms utilized in this project, to BDRC as the ***Norbu Ketaka*** collection. Ultimately, our project paves the way for linguistics and humanities research, offering a promising methodology for constructing extensive textual corpora in other under-researched languages. The innovative approach of our project can serve as a model for similar endeavors in processing old manuscripts on a large scale. The success of our project undoubtedly benefits scholars in the field of Tibetan studies, as well as the broader communities involved in NLP, and low-resource language research.

Bibliography

An, Bo & Congjun Long. 2021. Neural dependency parser for tibetan sentences. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 20(2). doi:10.1145/3429456. https://doi.org/10.1145/3429456.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)*, Association for Computational Linguistics.

Esukhia. 2023. Derge tengyur. https://github.com/Esukhia/derge-tengyur.

He, Kaiming, Georgia Gkioxari, Piotr Dollár & Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the ieee international conference on computer vision*, 2961–2969.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren & Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 770–778.

Hill, Nathan W & Marieke Meelen. 2017. Segmenting and pos tagging classical tibetan using a memory-based tagger. *Himalayan Linguistics* 16(2). 64–86.

Jiangdi, Kang Caijun. 2003. The methods of lemmatization of bound case forms in modern tibetan [c]. In *Ieee international conference on natural language processing and knowledge engineering*, .

Kang, Caijun, Di Jiang & Congjun Long. 2013. Tibetan word segmentation based on word-position tagging. In *2013 international conference on asian language processing*, 239–242. IEEE.

Krizhevsky, Alex, Ilya Sutskever & Geoffrey E Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60(6). 84–90.

LeCun, Yann, Léon Bottou, Yoshua Bengio & Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11). 2278–2324.

Li, Yan, Xiaomin Li, Yiru Wang, Hui Lv, Fenfang Li & La Duo. 2022. Character-based joint word segmentation and part-of-speech tagging for tibetan based on deep learning. *Transactions on Asian and Low-Resource Language Information Processing* 21(5). 1–15.

Meelen, Marieke, Élie Roux & 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). doi:10.1145/3409488. https://doi.org/10.1145/3409488.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,

Aidan N Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems 30*, 6000–6010.

Wu, Yuxin, Alexander Kirillov, Francisco Massa, Wan-Yen Lo & Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

Xiangxiu, Cairang, Nuo Qun, Nuobu Renqing, Trashi Nyima & Qijun Zhao. 2022. Research on tibetan part-of-speech tagging based on transformer. In *2022 3rd international conference on pattern recognition and machine learning (prml)*, 315–320. IEEE.

Handwritten Text Recognition (HTR) for Tibetan Manuscripts in Cursive Script*

Rachael M. Griffiths
(Austrian Academy of Sciences)

The use of advanced computational methods for the analysis of digitised texts is becoming increasingly popular in humanities and social science research. One such technology is Handwritten Text Recognition (HTR), which generates transcripts from digitised texts with machine learning approaches, to enable full-text search and analysis. Up to now, HTR models for Tibetan manuscripts in cursive script have not been available. This paper introduces work carried out as part of the *The Dawn of Tibetan Buddhist Scholasticism (11th-13th)* (TibSchol) project at the Austrian Academy of Sciences, which is utilising the Transkribus platform to explore possible solutions to automate the transcription of Tibetan cursive scripts. It presents our methodology and preliminary results along with a discussion of the limitations and potential of our current models.

1. *Introduction*

Handwritten Text Recognition (HTR) is an active research field that has developed significantly over the last decade, making great strides in its ability to automatically transcribe texts, especially those in Roman script (Nockels 2022). Focus now is being applied to extending this to other scripts–including Devanagari (Merkel-Hilf 2022), Hebrew (Digitizing Jewish Studies (DiJeSt) 2020), and Pracalit script (O'Neill & Hill 2022)–and offers great potential to those studying texts in a wider range of languages. In the context of Tibetan, several initiatives have been undertaken to develop Optical Character Recognition (OCR) systems. Notable OCR implementations for

* Rachael Griffiths, "Handwritten Text Recognition (HTR) for Tibetan Manuscripts in Cursive Script", *Revue d'Etudes Tibétaines*, no. 72, Juillet 2024, pp. 43-51.

The writing of this paper was facilitated through the project *The Dawn of Tibetan Buddhist Scholasticism (11th-13th)* (TibSchol). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101001002 TibSchol). The results presented are solely within the author's responsibility and do not necessarily reflect the opinion of the European Research Council or the European Commission who must not be held responsible for either contents or their further use.

Tibetan include Namsel OCR[1] and Google Drive/Google Docs. Additionally, projects and organisations such as the Buddhist Digital Research Centre (BDRC, https://www.bdrc.io) and Esukhia (https://github.com/Esukhia/) are actively engaged in research and development efforts in OCR and HTR. However, despite these endeavors, publicly available HTR models for Tibetan are currently unavailable.

Expanding the abilities of HTR models to Tibetan manuscripts is one strand that is being explored as part of the ERC-funded project *The Dawn of Tibetan Buddhist Scholasticism (11th–13th)* (TibSchol) at the Austrian Academy of Sciences. The project is carrying out an extensive study of the formative phase of Tibetan Buddhist scholasticism, utilising a large number of recently surfaced works. Notably it explores texts that were published as manuscript facsimile in the *bKa' gdams gsung 'bum* (*Collected Works of the Kadampas*) (dPal brtsegs bod yig dpe rnying zhib 'jug khang 2006-2015). Additional relevant manuscript sources are accessible online via BDRC. As TibSchol is a text-based project, HTR offers the possibility of facilitating full-text mining and analysis for a broad-scale approach to the corpus by relying on machine-readable transcriptions of these sources.

2. *Method*

For this task, we have been using Transkribus, a popular platform for transcribing, annotating, and searching historical manuscripts, which can be run on a local machine or in an online web interface (Kahle, Colutto, Hackl & Mühlberger 2017). It allows users to train text recognition models based on images of handwritten text that are lined up with corresponding diplomatic transcriptions which are called 'ground truth'. It also provides pre-trained HTR models in a range of scripts, although no public model is currently available in any Tibetan script. We tested a private model trained by Esukhia for handwritten *uchen* (Tib. *dbu can*, 'headed script'), however it was unable to transcribe the *ume* (Tib. *dbu med*, 'headless script') works in our corpus. As such, we have had to train a new HTR model from scratch.

As a guideline for creating an HTR model, Transkribus recommends preparing 5,000-15,000 words (25–75 pages) of transcribed material. In general, the more training data used, the higher the accuracy of the HTR model will be. The metric used to assess the accuracy of an HTR model in Transkribus is Character Error Rate (CER), that is the percentage of character-level errors in the recognised text compared to the ground truth text. A CER under 10% is considered efficient for automatic transcription, however, to maximise the

1 https://github.com/zmr/namsel

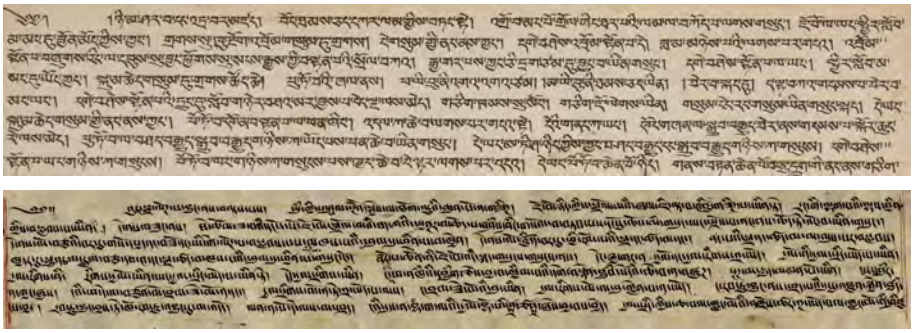

Fig. 1 – Examples of handwritten uchen (top, BDRC W3CN2815) and ume (bottom, BDRC W1NLM1277).

usability of transcribed texts for the text mining and analysis that will form the bedrock of our project, we are aiming for a CER of 5% or lower.

Our workflow began with making fundamental decisions about which scripts and texts the HTR will be required to read and accordingly, which images and transcripts will form a suitable ground truth. The manuscripts in our corpus are written in a variety of scripts (Fig. 2), which makes the possibility of training a general model more challenging. A further challenge is the quality of the images, which varies considerably throughout the corpus, and many folios contain interlinear and/or marginal insertions that are difficult to read. As such, we decided to begin with training a script-specific model, choosing *drutsa* (Tib.*'bru tsha*) due to the quality of images available in this script. We selected five manuscripts (totalling approximately 300 folios and 2500 lines) in *drutsa* script for training that were clearly legible and for which we already had a transcription in the so-called "Wylie" system, which uses Roman characters, without diacritics, to render univocal combinations of letters in Tibetan syllables (Wylie 1959).

The next step was running the Layout Analysis (LA) tool on manuscript images imported into Transkribus. This tool, which is integrated into the platform, automatically analyses the structure and layout of a document to identify its different components such as text regions, images, and other elements. Before it can transcribe, the HTR model must be able to clearly define what it is seeking to transcribe. The default LA tool did not generate accurate results on the Tibetan manuscripts we had selected for training; results often included multiple text regions across one folio, the omission of lines or lines only being partially recognised, and stains or markings on the manuscript

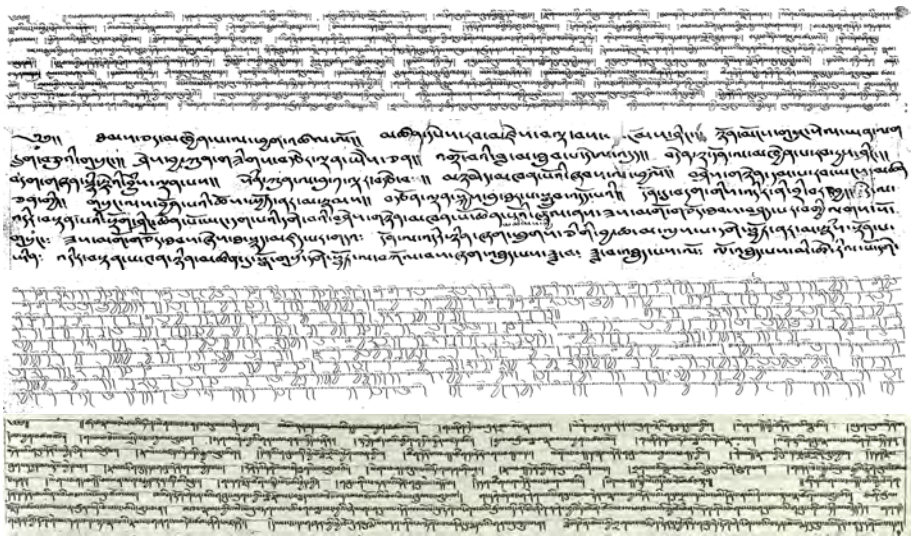

Fig. 2 – Examples of scripts found in the TibSchol corpus (top to bottom, BDRC W26453, BDRC W1CZ1224, BDRC W12170, BDRC W1KG12371).

marked as a text. The number of errors were significant enough that manual correction was not a feasible option (see first image in Fig. 4).

Initially, improvement in the LA was achieved through pre-processing the images before uploading them to Transkribus. We used OpenCV (https://www.opencv.org) to affect image sharpness, resolution, and noise using the filter2D and fastNlMeansDenoising functions (Fig. 3). These image enhancements improved the results of the default LA tool, although it continued to segment stains and markings as baselines, which required manual deletion. We also trialed a Python pre-processing script developed by Esukhia, which automatically generates baselines that stretch across an entire text region (https://github.com/Esukhia/custom-script-for-transkribus). The baselines created are straight lines, which unfortunately are not compatible with our images, some of which are warped and contain marginalia and interlinear additions. In June 2022, Transkribus launched a new feature, where users can train a baseline model specific to their document typology (see Transkribus (2022)). We tested this tool to see if it could train a customised baseline model based on examples from our corpus.

The baseline model was trained on 160 folios that had been manually segmented, keeping aside 10% as a Validation Set (a Validation Set is used to evaluate the model's performance on unseen data during the training process). A

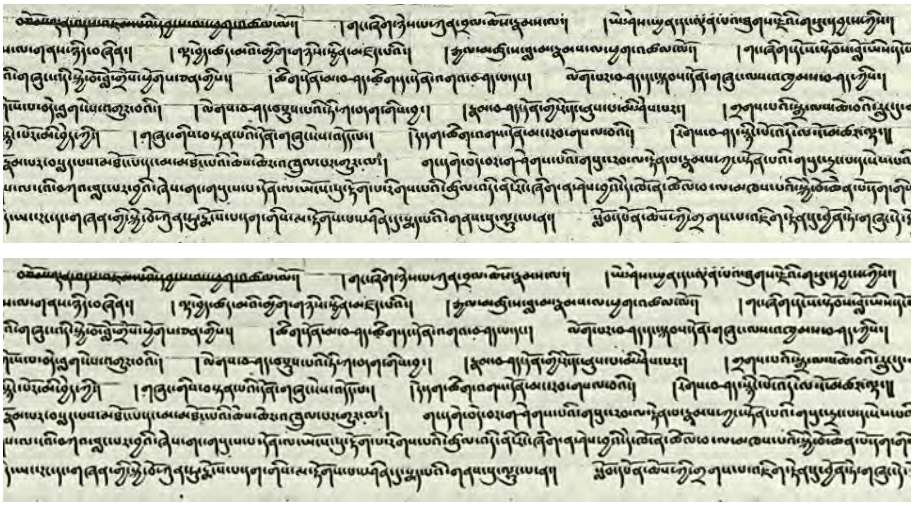

Fig. 3 – Section of folio from Tshad ma rnam par nges pa'i Ti ka legs bshad bsdus pa (BDRC W1KG12371) before (top) and after (below) pre-processing using OpenCV.

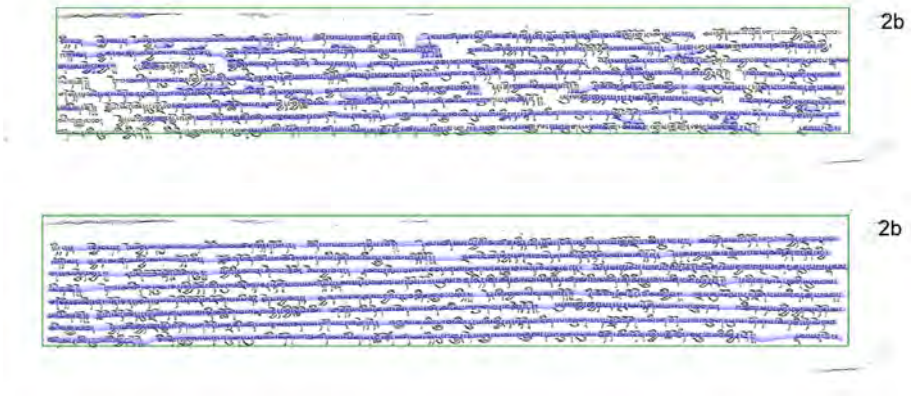

Fig. 4 – Results of the default LA tool (top) and custom baseline model (bottom) on the same folio from Tshad ma'i bstan bcos sde bdun rgyan gyi me tog (BDRC W00KG03838).

Loss on the Training Set (LTS) and Loss on the Validation Set (LVS) below 10% allows for an effective automatic segmentation. The TibSchol baseline model has an LTS of 4.3% and LVS of 3.8%. As can be seen in Fig. 4, the model still requires some manual correction, although this is minimal. As we were satisfied with the results, the baseline model was then run on the remaining folios uploaded to Transkribus.

Once the layout was finalised, the transcriptions were added to the document editor and HTR training could begin. When we first started working with Trankribus, it hosted two HTR engines: HTR+, developed by the CITlab of the University of Rostock, and PyLaia, a PyTorch-based model developed by the Technical University Valencia. During initial tests, PyLaia appeared to struggle with reading 'curved' lines (see the results of Model 5 in Fig. 6)– some of our images are warped–while the HTR+ engine produced better results. Thus we decided to continue using HTR+ for training.

To more accurately gauge a model's performance on the Validation Set, it is possible to compare the ground truth of a page with the transcription produced by the model using 'Compare Text Version' (Fig. 5). This not only flags sticking points for the model but also errors in our ground truth, which were then manually corrected before training a new model with more data.

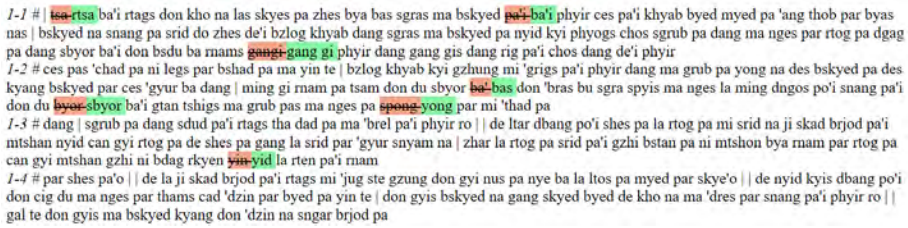

Fig. 5 – Errors in the Validation Set are marked in red. In green, the word is shown as it is written in the ground truth transcription.

3. *Results*

The project initially iterated through six models. Fig. 6 shows the size of the training data as number of lines, beside the CERs of each model. Although the results were promising, the CER of the Validation Set remained above 5%, and so work continued on improving the model's accuracy. This included reinspecting the segmentation of images and proofreading transcriptions (see Griffiths 2022a and Griffiths 2022b).

In October 2022, our HTR model was trained on 269 folios (2310 lines), with validation performed on 27 folios. Using 250 epochs, the trained model

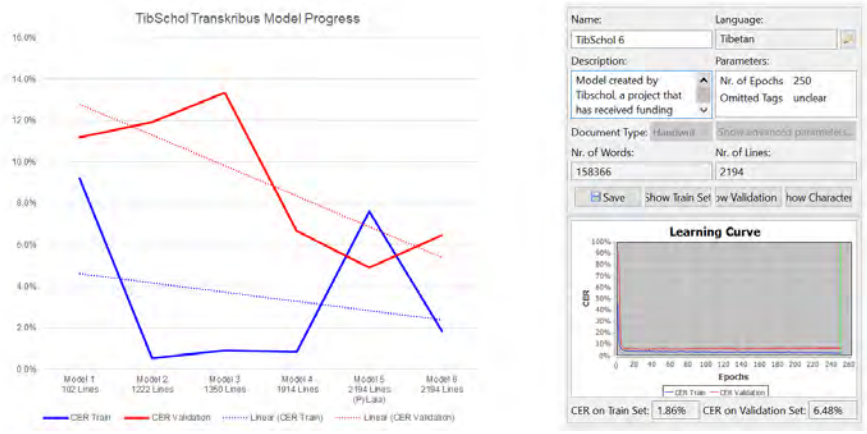

Fig. 6 – Results of HTR Models 1-6 trained in Transkribus.

had a CER of 1.15% for the Training Set and a 2.33% for the Validation Set. Satisfied with the results, we then tested the model on other manuscripts written in *drutsa* that were not part of our dataset and were pleased to see that it produced accurate results.

Unfortunately, in November 2022, HTR+ recognition within Transkribus was deactivated. All models trained using HTR+ can no longer be used, although they can be retrained with the PyLaia engine. Training with PyLaia (using the same data set and number of epochs) initially produced poorer results; a CER of 10.20% for the Training Set and 8.80% for the Validation Set. A recurring issue in the generated transcript was a string of random letters that appeared mostly at the end of each line, although sometimes at the beginning (Fig. 7), limiting the use of the model.

1-2 pa'i shes pa de ni zhes pa dbye ba la khyab pas sngar brjod pa don la phyin ci ma log par 'jug pa nyid yin pa 'am | bstan par bya ba'i gtso' bo yin pas khyad par tshad ma'i shes pa'o | | gang dbye ba'i dngos po ni mngon sum dang ni rjes su dpag ces bya ba'o zhes mo da a pa a ha pa pa a a ya ya pa pa pa a

1-3 don la brtags pa'i sgra yin la | de yang gnyis ste | yi ge'i 'bru las thob pa bshad pa'i rgyu mtshan can dang | don la grags pas go ba 'jug pa'i rgyu mtshan can gyi sgra'o | | de la bshad pa dang 'jug pa'i sgra de dag 'brel pa nyid du 'pug pa pa ya ya pa pa ha pa pa pa pa pa

Fig. 7 – Example of errors in transcript produced by PyLaia model before dewarping.

The random string of letters was linked to PyLaia's difficulty in reading curved lines. Fortunately, Transkribus has developed a new tool that tackles this issue: line dewarping (accessed via 'PyLaia advanced parameters'). We found that selecting 'dewarp' significantly improved training results. Our most recent model was trained with a CER of 2.2% on the Training Set and 1.40% on the Validation Set.

4. Next Steps

Having successfully trained a model that gives us a CER <3%, it is clear that our research so far indicates that Transkribus works extremely well at recognising historical documents and can be applied to the yet unexplored Tibetan texts in *drutsa* in our corpus.

There are however, two areas of focus to improve our current and future models, as identified through our current validation. The first of these is the occurrence of rarer textual elements, such as numerals, Tibetan rendering of Sanskrit words, and certain symbols and punctuation. Due to their infrequency, the model struggles to recognise these and, instead, produces something that appears similar e.g., a *shad* ། instead of the number one ༡. We have been able to significantly improve the model's recognition of numbers through identifying texts with a higher frequency of numerals and adding these to the training model. The second emerging issue is that the model also transcribes similar looking letters such as *pa* and *ba*, *zha* and *na* with lower confidence, especially with lower quality images and/or damaged or soiled manuscripts. To some degree, these issues will lessen as the model is trained on a greater volume of texts, however, further investigation is required to allow tailored solutions.

The next stage of the project will be to train HTR models for the other scripts found within our corpus. Currently, when tested, the *drutsa* model has a higher CER (>10%) when applied to other scripts. However, used as a base model, it will require significantly fewer pages to train models for other scripts. We estimate around 30 to 50 folios of new ground truth, as opposed to the 300 folios required for the base model. Based on our results with *drutsa*, we are satisfied that our approach of developing multiple HTR models is appropriate for obtaining a reasonably accurate transcription of a large quantity of data in multiple scripts. We would then be able to explore the possibility of training scripts together to create a generic model for Tibetan manuscripts in cursive script.

The development of HTR models that can transcribe multiple Tibetan scripts would open up the possibility of unlocking a vast trove of fully searchable texts that could be available to a much broader audience. To this end, one

of our project's aims is to publish our models on Transkribus and make our ground truth datasets publicly available.

Bibliography

Digitizing Jewish Studies (DiJeSt). 2020. Digitize your texts with Transkribus. (last accessed: 30.01.2023). http://dijest.net/digitize-your-texts-with-transkribus/.

dPal brtsegs bod yig dpe rnying zhib 'jug khang (ed.). 2006-2015. *bKa' gdams gsung 'bum phyogs bsgrigs thengs dang po/gnyis pa/gsum pa/bzhi pa*, vol. 1–120. Si khron mi rigs dpe skrun khang.

Griffiths, R. 2022a. Transkribus in Practice: Abbreviations. *The Digital Orientalist* https://digitalorientalist.com/2022/11/01/transkribus-in-practice-abbreviations/. (last accessed: 30.01.2023).

Griffiths, R. 2022b. Transkribus in Practice: Improving CER. *The Digital Orientalist* https://digitalorientalist.com/2022/10/25/transkribus-in-practice-improving-cer/. (last accessed: 30.01.2023).

Kahle, Philip, Sebastian Colutto, Günter Hackl & Günter Mühlberger. 2017. Transkribus - a service platform for transcription, recognition and retrieval of historical documents. In *2017 14th iapr international conference on document analysis and recognition (icdar)*, vol. 04, 19–24. doi:10.1109/ICDAR.2017.307.

Merkel-Hilf, N. 2022. Ground Truth data for printed Devanagari [Dataset]. doi:10.11588/data/EGOKEI.

Nockels, J et al. 2022. Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of Transkribus in published research. *Archival Science* 22. 367–392. doi:10.1007/s10502-022-09397-0.

O'Neill, A.J. & N. Hill. 2022. Text Recognition for Nepalese Manuscripts in Pracalit Script. *Journal of Open Humanities Data* 8(26). doi:10.5334/johd.90.

Transkribus. 2022. How to Train Baseline Models in Transkribus. (last accessed: 30.01.2023). https://readcoop.eu/transkribus/howto/how-to-train-baseline-models-in-transkribus/.

Wylie, T. 1959. A Standard System of Tibetan Transcription. *Harvard Journal of Asiatic Studies* 22. 261–267. doi:10.2307/2718544.

A Universal Dependency Treebank for Classical Tibetan*

Christian Faggionato
(University of Cambridge)

The Universal Dependencies (UD) Project's goal is to create a set of multilingual standardised dependency treebanks that are built according to a universal annotation scheme. The present paper describes the work behind the creation of the first Classical Tibetan UD treebank that involves a a semi-automated NLP pipeline, with the implementation of a rule-based dependency parser written in the Constraint Grammar (CG-3) formalism.

1. *Introduction to Dependency Grammar*

Dependency grammar is a type of grammatical framework that focuses on the relationships between words in a sentence. It provides an alternative to phrase structure grammar and generative grammar, which both define a sentence as a set of constituents or phrases. Instead, dependency grammar defines the relationships between words in terms of dependency, where one word (the head) is the main word, and the other word(s) (the dependents) provide additional information about the head. Dependency grammar can be represented using a tree structure, where the root of the tree represents the main verb in the sentence, and the other words in the sentence are attached as dependents to the root or to other words in the sentence (Matthews et al. 1981).

One of the main advantages of dependency grammar is its simplicity and flexibility. In a dependency grammar, each word in a sentence is treated as a separate unit and is assigned a grammatical function, such as subject, object, or modifier. These functions are indicated by the dependencies between the words, rather than by the placement of the words within a phrase or clause.

* Christian Faggionato, "A Universal Dependency Treebank for Classical Tibetan", *Revue d'Etudes Tibétaines*, no. 72, Juillet 2024, pp. 52-69.

This work was funded by the Arts and Humanities Research Council (AHRC), UKRI, as part of the project "The Emergence of Egophoricity: a diachronic investigation into the marking of the conscious self." Project Reference: AH/V011235/1. Principal Investigator: Nathan Hill, SOAS University of London.

This means that the structure of a sentence can be represented as a set of directed dependencies between the words, rather than having a hierarchical structure (Jurafsky & Martin 2014). Figure 1 shows an example of a Classical Tibetan sentence with dependency parsing:

(1) ཨ་ཁུ་ དང་ ཨ་ནེ་ ལ་ ཤ་ཁོག་ འཛུངས།

a-khu	*dang*	*a-ne*	*la*	*sha-khog*
uncle.NOUN	and.ADP	aunt.NOUN	to.ADP	carcass.ADP
'dzungs	/			
gave.VERB				

'We presented an [entire animal] carcass to my aunt and uncle (de Jong (1959), p. 33 ln. 14)'

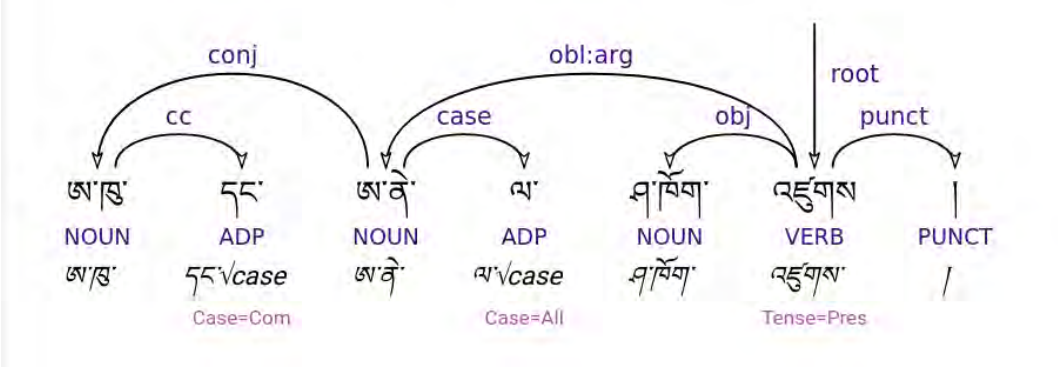

Fig. 1

In this example relations among the words are illustrated above the sentence by labelled arcs from heads to dependents; labels are drawn from a fixed set of grammatical relations and the head of the entire sentence structure is marked by a root node. Each token has three layers of information: word, Part-of-Speech tag (POS) and lemma.

2. *The Universal Dependencies Project*

The Universal Dependencies (UD) treebank project is a collaborative effort that started in 2014 and it is aimed at creating a consistent representation of syntactic dependencies across multiple languages. This is achieved by defin-

ing a set of universal dependencies (Nivre, de Marneffe, Ginter, Hajič, Manning, Pyysalo, Schuster, Tyers & Zeman 2020).[1]

The UD treebank project has been successful in creating annotated corpora for many languages including low-resourced ones. Each treebank is annotated using a common annotation scheme, which includes information about the words in the sentence, their part-of-speech tags, and the relationships between them in order to capture the underlying grammatical structure. Additionally, all the annotated corpora are available for free and can be easily accessed through the UD treebank website. This common annotation scheme generally allows for cross-lingual comparison and analysis, making it easier to develop NLP models that can be used across multiple languages and for a variety of NLP tasks, such as machine translation, text classification, and named entity recognition. Furthermore dependency relations can also provide information on the semantic relationship between predicates and their arguments which is useful for other NLP applications such as question answering and information extraction.[2]

The first dependency parsed corpus of Classical Tibetan –using the guidelines of the UD treebank project– was compiled during two AHRC funded projects that took place at the School of Asian and African Studies (SOAS) in London. During the first project, "Tibetan in Digital Communication" (2012-2015), a group of four Classical Tibetan texts were manually POS tagged: the *mdzang blun* མཛངས་བླུན་ཞེས་བྱ་བའི་མདོ་, "The Sutra of the Wise and the Foolish", the *Mi la'i rnam thar* མི་ལའི་རྣམ་ཐར་, "The Biography of Milarepa", the *Mar pa lo tsā' i rnam thar* མ་རཔ་ལོ་ཙཱ་ཡི་རྣམ་ཐར་, "The Biography of Marpa", and the *Bu ston chos ' byung* བུ་སྟོན་ཆོས་འབྱུང་, "The History of Buddhism" by *Bu ston Rin chen grub*. During the follow up project "Lexicography in Motion" (2017-2021), the corpus was expanded with two Old Tibetan texts, the *Old Tibetan Annals* and the *Old Tibetan Chronicle* and an additional Classical Tibetan text, the *Twa ra nwa tha' i rgya gar chos ' byung* ཏྭ་ར་ནྭ་ཐའི་རྒྱ་གར་ཆོས་འབྱུང་, "the History of Buddhism" by *Tāranātha Kun dga' Snying po*. All the new texts have been automatically POS tagged using the method developed by Meelen, Roux & Hill (2021) and then manually corrected.[3] During the same project we compiled a diachronic lexicon of Tibetan verbs and the dependency relations linking verbs to their arguments were manually annotated using the following scheme:

1 https://universaldependencies.org/

2 Nivre, de Marneffe, Ginter, Goldberg, Hajič, Manning, McDonald, Petrov, Pyysalo, Silveira, Tsarfaty & Zeman (2016).

3 A full manual correction of the POS tagging was only done for the two Old Tibetan texts: at the present stage, this small Old Tibetan corpus represents the Gold Standard, which can be used for training.

- *arg1* (changed to *nsubj* in accordance with UD annotation scheme): the first argument or " subject" of a verb, which may be agentive or unmarked, but not oblique.
- *arg2* (changed to *obj* in accordance to the UD annotation scheme): the second argument or " object" of a verb, which cannot be oblique.
- *arg2-lvc* (changed to extitobj-lvc in accordance to the UD annotation scheme): the second argument of a verb, which together with it forms a complex predicate.
- *argcl*: the clausal argument of a verb.
- *obl-adv*: an oblique marked nominal, which behaves like an adverb.
- *obl-arg*: an oblique marked nominal, which is considered an argument of the verb.
- *obl*: an oblique nominal that modifies a verb.

The process of manually annotating verbal arguments gave rise to the idea of developing an NLP tool that can automatically map missing dependencies for other sentence constituents.

3. *A Dependency Constraint Grammar for Classical Tibetan*

The constraint grammar formalism (CG) is a rule-based formalism for writing disambiguation and syntactic annotation grammars, originally introduced by Karlsson (Karlsson, Voutilainen, Heikkila & Anttila 1995) and successively implemented with a set of rules that creates dependency annotation (CG-3). Its VISL constraint grammar compiler (version 3) (VISL-CG3) implemented in the IDE for CG-3, is used for the compilation of constraint grammar rules. The constraint grammar analyzes the texts with a bottom-up scanning. Every disambiguation is solved step by step with the help of morpho-syntactic context. Constraint-grammar rules usually contain context conditions, domains, operators and targets. The context can be absolute, referring to a fixed token position within the text, or relative, referring to a token to the left or right with a certain distance to a specific constraint. We can modify the context using barriers made of tokens or SET of tokens that stop the scanning of the sentence. Furthermore, we can link context to other context with the LINK rule. In this way the constraint grammar works globally and creates complex syntactic relations.

Dependency treebanks can be created using human annotators, or using automatic parsers to provide an initial parse and then having annotators to

correct the parses. To facilitate the creation of the first dependency parsed corpus of Classical Tibetan I decided to write a a rule-based parser, in the CG-3 constraint grammar formalism, able to generate full dependencies from the verb arguments dependencies we already had at our disposal. I created 72 SETPARENT rules and 41 mapping rules that generate case-marking relations, dependencies for noun phrases—linking modifiers such as adjectives, determiners and demonstratives to head nouns—and dependencies linking converbs, punctuation and adverbs verbs.

In the CG-3 formalism the dependency analysis is done using specific rules, i.e. SETPARENT (mapping a token to its parent) and SETCHILD (mapping a token to its child). There are also rules used to correct and fix errors, i.e. ADDCOHORT (adding a token and all its readings), MOVECOHORT (moving a token and all its readings) and DELETE (deleting a token with all its readings). The grammar at the moment uses a set of POS tags and and the dependency tags for verbal arguments to generate a full set of dependency relations tags for all the sentence constituents. In order to establish dependency relations, the dependency tags are expressed in the following way, 5->2: the first digit indicates the absolute position of the token in the sentence and it points to the absolute position of the token representing the mother. Usually a verb points to 0, which indicates its head status (Bick & Didriksen 2015).

I tested the CG-3 dependency grammar on both Old Tibetan and Classical Tibetan texts. I opted to work also on the Old Tibetan texts due to practical considerations, as these texts represented our gold standard in terms of POS tagging and annotation of verb arguments. It is worth noting that Old Tibetan and Classical Tibetan are very similar in terms of grammar and vocabulary, but they differ sustantially in terms of spelling and orthography.Dotson & Helman-Ważny (2016) I developed a python script in order to normalize Old Tibetan to Classical Tibetan, to make the two languages to look similar.[4] An improved version of the normalization grammar is now implemented in the pre-processing python script of the NLP pipeline developed by Faggionato, Hill & Meelen (2022). The normalization process allowed the analysis of Old Tibetan with the NLP tools for segmentation and POS tagging available for Classical Tibetan (Faggionato & Meelen 2019).

The Tibetan texts have been first exported in the CoNNL-U format, where each line has 10 fields, separated by tabs, containing information for every word/token such as word index or ID, word form, lemma, universal POS tag, a list of morphological features, the head of the current word (which is either a ID value or 0) the universal dependency relation to the head and other annotations. Figure 2 shows an example of a sentence in the CoNNL-U

4 https://github.com/lothelanor/actib/blob/main/preprocessing.py

format. After each sentence there is always an empty line which represents sentence boundaries.

```
<s ref="T112">
1   རྡོ་རྗེ་  རྡོ་རྗེ་  NOUN    _   Number=Sing 0   root    _   _
2   ལ་  ཀྱི་√case  ADP _   Case=Gen    0   root    _   _
3   སྙིང་  སྙིང་  NOUN    _   Number=Sing 0   root    _   _
4   ཡ་  ཀྱི་√case  ADP _   Case=Gen    0   root    _   _
5   དལ་བུབ་  དལ་བུབ་  NOUN    _   Number=Sing 7   arg2    _   _
6   ཁྲོད་  ཁྲོད་  NOUN    _   Number=Sing 7   obl-adv _   _
7   བསད་ གསོད་VERB    _   Tense=Past  0   root    _   _
8   ནས  ནས་√cv   SCONJ   _   Case=Ela    0   root    _   _
9   |   |   PUNCT   _   _   0   root    _   _
10  |   |   PUNCT   _   _   0   root    _   _
</s>
```

Fig. 2 – CoNLL-U Format

The CG-3 input files have been created modifying the existing CoNNL-U files and retaining information such as ID, POS, lemma, dependencies for verb arguments, dependency tags and other syntactic features. Example (2) shows an extract from a VISL-CG3 input file with five tokens:

(2) "<རྣམ་མཁའ་>"
"རྣམ་མཁའ་" NOUN Number=Sing @obl-arg #1->3
"<ལས་>"
"ལས་√case" ADP Case=Abl @root #2->0
"<བཙས་>"
"འབད་√1" VERB Tense=Past @root #3->0
"<ཏེ>"
"ཏེ་√cv" SCONJ Case=Sem @root #4->0
"<|>"
"|" PUNCT @root #5->0

Example (3) shows the output file containing the dependencies and dependency labels generated by the CG-3 grammar. The generated dependencies link all the tokens within a sentence to the root element which is usually a verb or a head noun:

(3) "<ནམ་མཁའ་>"
"ནམ་མཁའ་" NOUN Number=Sing @obl-arg #1->3
"<ལས་>"
"ལས་√case" ADP Case=Abl @case #2->1
"<བབས་>"
"འབབ་√1" VERB Tense=Past @root #3->0
"<ཏེ>"
"ཏེ་√cv" SCONJ Case=Sem @mark #4->3
"<|>"
"|" PUNCT @punct #5->3

The CG-3 output file is then converted again into the CoNLL-U format and uploaded into Arboratorgrew.[5] Arboratorgrew is a popular software tool for linguistic analysis that provides a graphical representation of the tree structure of sentences, and makes it very easy and quick to manually correct dependency relations (Guibon, Courtin, Gerdes & Guillaume 2020).

The CG-3 dependency grammar is made of three main sections, and each of them serves a distinct purpose in generating the dependency treebank.

In the first section I defined the sentence boundaries creating a sentence break after specific converbs and cases: final སོ་, semifinal ཏེ་, imperative ཤིག་, imperfective ཞིང་, na-re ན་རེ་, question འམ་ and associative དང་.[6] Furthermore I introduced sentence boundaries when verbs are not followed by any converb but are followed by punctuation markers such as *shads*. The punctuation condition is necessary because we do not want to introduce any sentence segmentation when we have a chain of two or more verbs. In the same section I introduced all the POS tags and the universal dependency tags used by the grammar.

In the second section I defined three sets of helpers. Each set creates boundaries for the noun phrases, define their constituents and their nominal heads. Example (4) shows the three sets in the CG-3 formalism:

(4) SET np.elem = (NOUN) OR (ADJ) OR (NUM) OR (PROPN) OR (DET) OR (PRON) OR VN;
SET LEFT_NP_BOUNDARY = (VERB) OR (ADP) OR (SCONJ) OR (PART) OR (ADV) OR (AUX) OR (PUNCT);
SET RIGHT_NP_BOUNDARY = (ADP) OR (SCONJ) OR (Polarity=Neg) OR (PUNCT);
SET Head_NOUN = (NOUN) OR (NUM) OR (PROPN) OR (PRON) OR (DET) OR VN;

5 https://arboratorgrew.elizia.net/#/
6 Associative cases are sentence boundaries only if they follow verbal nouns.

In the third and main section, I created the dependency relations for all the tokens, using the CG-3 SETPARENT and MAP rules. A first set of rules deals with punctuations, linking them to the root verb, usually on the left of the PUNCT token. In this section I further improved the CG-3 dependencies linking subordinate clauses to the main verbs with the tag *advcl* and using the new sentence segmentation rules and annotation scheme we recently developed (Faggionato, Meelen & Hill 2023). For a detailed description of the newly developed NLP pipeline for processing Old and Classical Tibetan texts see Faggionato et al. (2022). Figure 3 shows an example of a subordinate clause linked to the main verb with the correct dependency relation *advcl*.

(5) བཞེངས་ནས་ཕྱི་ར་རྒྱངས་བཀྱེད་དེ་

bzhengs *nas* *phy* *r* *rgyangs*
get_up.VERB SCONJ outside.DET ADP far.NOUN
bkyed *de*
move_away.VERB SCONJ
'[the lady] got up in order to move far away' (de Jong (1959), p. 78 ln. 25-26)'

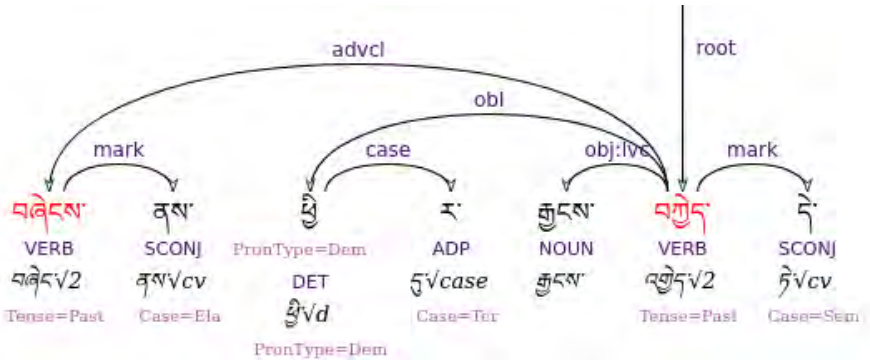

Fig. 3 – Subordinate Clause

In the main section I also created the CG-3 rules that deal with the internal dependencies of the noun phrase (NP) elements. Analyzing the distributions and common patterns of the Tibetan NP constituents, I created a set of dependency rules that solve these potential issues. These rules prevent the parser to create wrong cross-dependencies between sequences of nouns (Garrett & Hill 2015).

Patterns of head nouns followed by one or more determiner, numeral, pronoun or adjective and followed by another head noun, are not a real challenge for the parser. Also cases of head nouns, already tagged as verb arguments, followed by nouns that function as appositions are easily solved by the CG-3 dependency rules. Figure 4 shows a set of dependency relations correctly generated for the internal elements of a NP.

(6) མ་སྨད་ གསུམ་ ཀ་ ས་ དྲྭ་ སོ་

ma-smad *gsum* *ka* *s* *ngus* *so*
mother.NOUN three.NUM all.DET ADP cry.VERB PART
'the three of us, mother and children, cried' (de Jong (1959), p. 37 ln. 6)

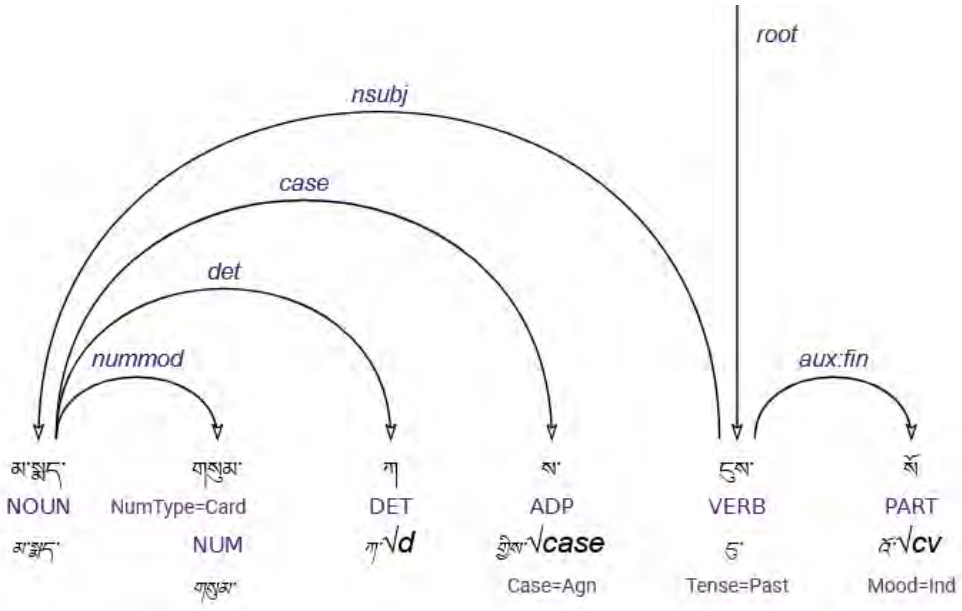

Fig. 4 – NP Elements

The CG-3 rules that creates these dependencies follow the same structure for determiners, numerals, adjectives, proper nouns and nouns that function as appositions. Here is the set of rules that creates the dependencies for nouns in apposition:

(7) SETPARENT (NOUN) (NONE p ALLPOS) TO (-1* Head_NOUN + TAGS OR (ADJ) + TAGS BARRIER LEFT_NP_BOUNDARY);

SETPARENT (NOUN) (NONE p ALLPOS) TO (-1* Head_NOUN BARRIER LEFT_
NP_BOUNDARY)(-1 LEFT_NP_BOUNDARY);

SETPARENT (NOUN) (NONE p ALLPOS) TO (-1* Head_NOUN BARRIER LEFT_
NP_BOUNDARY)(-2< (PUNCT))(NONE p ALLPOS - TAGS);

MAP (@appos) TARGET (NOUN) - TAGS (p Head_NOUN OR (ADJ) + TAGS);

The first SETPARENT rule of the set deals with cases such as NOUN + Head_ NOUN + ADJ in order to create a parental relation between the adjective and the second noun which has been manually tagged as a verb argument. Figure 5 shows the dependencies created by the first SETPARENT rule:

(8) གཡོན་ དུ་ ཐལ་བ་ སྤྱར་ གང་ ཁྱེར །

gyon	*du*	*thal-ba*	*spar*	*gang*
left.NOUN	in.ADP	ashes.NOUN	hand.NOUN	full.ADJ
khyer				
carry.VERB	PUNCT			

'carrying a handful of ashes in her left hand' (de Jong (1959), p. 36 ln. 18-19)

The second and third SETPARENT rule deals with other cases were the head noun is not an argument: the condition (-1 LEFT_NP_BOUN -DARY) forces to create a parental relation between the NP constituents and the leftmost element of the NP. The third SETPARENT rule works exactly as the second SETPARENT rule but deals with NPs at the beginning of a new sentence. Figure 6 shows the dependencies generated by these two SETPARENT rules:

(9) ནས་ ཕུལ་ གང་ ངམ

nas	*phul*	*gang*	*ngam*
barley.NOUN	handful.NOUN	full.ADJ	PART

'A handful of barley?' (de Jong (1959), p. 34 ln. 5)

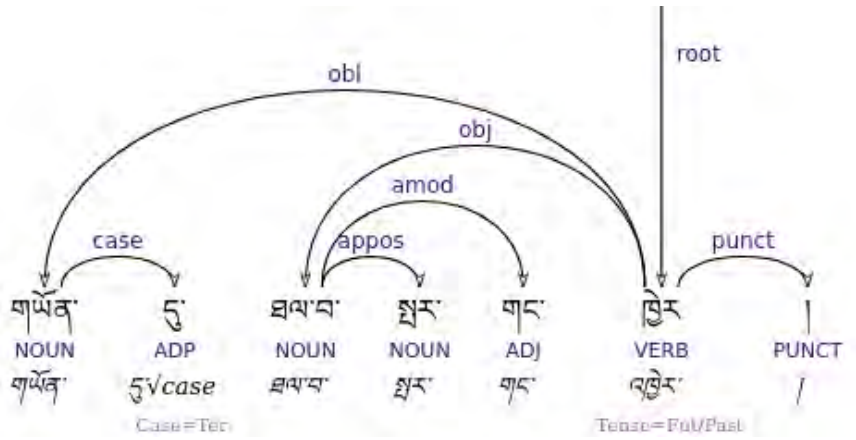

Fig. 5 – Nouns in Apposition - Rule 1

In all the three rules the MAP rule assigns the *adj* tag to the dependencies created with the three SETPARENT rules.

All the cases where nouns are in apposition to each other and are not already manually tagged as verb arguments might require manual correction in the post-processing phase.

In this case a careful analysis of the distribution or verb arguments in absolutive case might help. In fact we would expect zero-marked arguments being positioned as close as possible to the root verb, while other nominals that functions as oblique further away in the sentence. Also, when we have head nouns in absolutive case following each other, as shown in Figure 7, there will be often an intervening clitic, agentive or other case markers. All these considerations, highly reduce the chance of errors for the CG-3 parser.

(10) ཡུམ་ ཡང་ སྤྱན་ཆབ་ འདོན་ ཞེང་

yum	*yang*	*spyan-chab*	*'don*	*zhing*
wife.NOUN	PART	tear.NOUN	expel.VERB	SCONJ

'the wife burst into tears' (de Jong (1959), p. 67 ln. 23)

After the first set of rules for the NP elements, I created a second set of dependency rules for ADP (cases), PART (negations and focus clitics), SCONJ (converbs), and nouns linked to head nouns through the genitive case or the as-

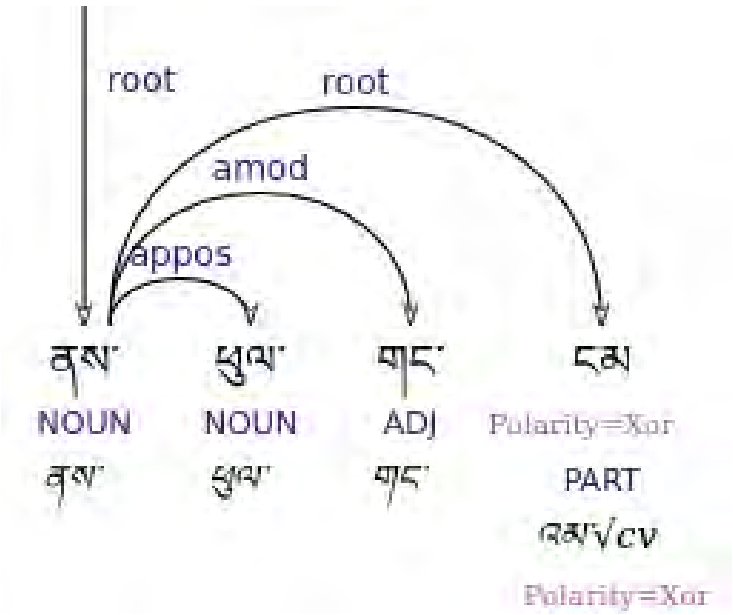

Fig. 6 – Nouns in Apposition - Rule 2 & 3

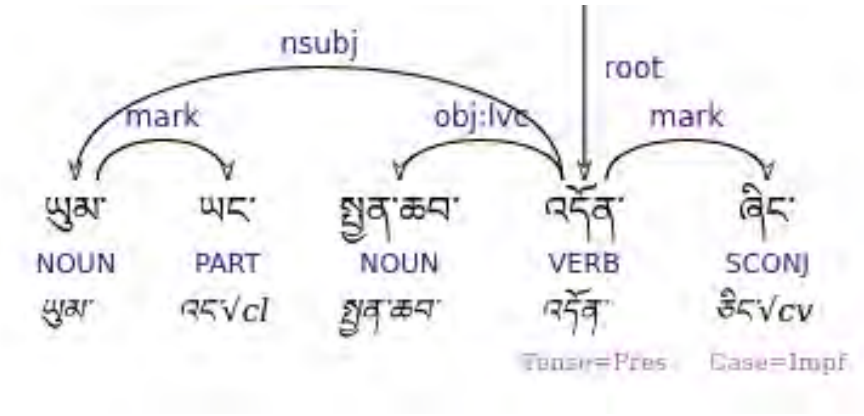

Fig. 7 – Head Nouns in Absolutive Case

sociative case: these nouns are tagged in the dependency treekbank as *nmod*, noun modifiers (see Fig. 8).

Since the parser follows the dependency rules in a sequential order, I was also able to link case markers and focus clitics straight to the head nouns, as

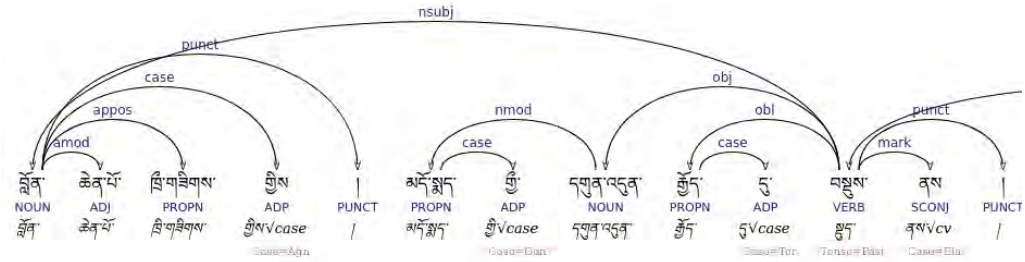

Fig. 8 – Cases and Converbs

all the other elements within the NP already have a dependency relation with their heads. Figure 8 also shows an agentive *gyis* (གྱིས་) correctly linked to its head noun *blon* (བློན་), 'minister'. This has been possible adding a (NONE p ALLPOS - TAGS) condition to the SETPARENT rule: the condition allows the parser to skip all the NP elements that already have a parental dependency relation with any head noun.

(11) བློན་ ཆེན་པོ་ ཁྲི་གཟིགས་ གྱིས ། མདོ་སྨད་ གྱི་ དགུན་འདུན་ རྒྱོད་ དུ་ བསྡུས་ ནས །

blon *chen-po* *khri-gzigs* *gyis* /
minister.NOUN great.ADJ Khri-gzigs.PROPN ADP PUNCT
mndo-smad *gyi* *dgun-'dun* *rgyod*
Mdo-smad.PROPN of.ADP winterNOUN Rgyod.PROPN
du *bsdus* *nas* /
at.ADP convened.VERB after.SCONJ PUNCT
'After Chief minister [Dba' s] Khri-gzigs convened the Mdo-smad winter council at Rgyod' (The Old Tibetan Annals, (Dotson & Hazod 2009))

In the final section of the CG-3 grammar I created a rule for relative clauses, tagged as *acl:rel*, and a section with specific rules targeting particular cases of tokens not mapped by previous dependency rules. Here is the SETPARENT rule that generates the dependencies for relative clauses:

(12) SETPARENT VN - TAGS (NONE p ALLPOS) TO (1* Head_NOUN BARRIER LEFT_NP_BOUNDARY - (Case=Gen) OR TAGS) (-1 (Case=Gen));
MAP (@acl:rel) TARGET VN - TAGS (p Head_NOUN);

The rule links a verbal noun followed by a genitive case to the head noun to its right. Figure 9 shows the dependencies created by this SETPARENT rule.

(13) ཨ་ཁུ་ དང་ ཨ་ནེ་ ས་ གཙོ་ བྱས་པ་ འི་ གཉེན་ཉེ་འཁོར་ །

a-khu *dang* *a-ne* *s* *gtso*
uncle.NOUN and.ADP aunt.NOUN ADP main.NOUN
byas-pa *'i* *gnye-nye-'khor* */*
made.VERB that.ADP close_relatives.NOUN PUNCT
'[our] close relatives, that were headed by my [paternal] aunt and uncle' (de Jong (1959), p. 31 ln. 12)

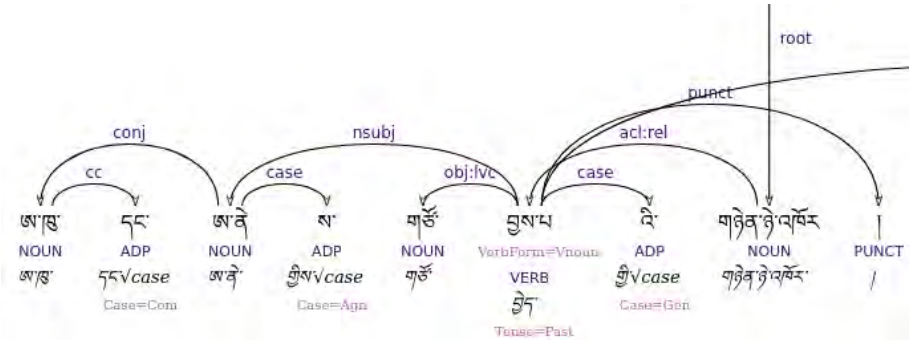

Fig. 9 – Relative Clause

4. *Evaluations and Future Improvements*

To evaluate the results of the dependency parser, I manually corrected the whole text of the Old Tibetan Annals and some sections of the *Mi la'i rnam thar*, "The Biography of Milarepa", for a total of 430 sentences and 12k tokens. The CG-3 dependency grammar achieves an accuracy level of 80% in creating dependencies and assigning dependency labels to unlabelled tokens. Additionally, the parser performs exceptionally in recognising subordinate clauses with an accuracy level of 90%. As it is expected, No automatic NLP task is

100% correct and a certain degree of manual correction is needed. The overall process of generating a full dependency treebank for Classical Tibetan it is hugely facilitated by the CG-3 dependency parser.

After the analysis of the obtained results, I identified some recurring error patterns. As already pointed out in the previous section, the challenge posed by chain of nouns leaves some of the tokens without a dependency tag. Again, this issue could be partially solved adding a layer of animacy information that will help in disambiguating head nouns. It is worth noting that according to the new and improved word segmentation guidelines that we recently developed [7]words have been split as much as possible to create more insight into the internal structure of the sentences, and that creates an additional problem with personal names and titles in terms of generating automated dependencies. Some manual correction will be needed in all these cases.

A second case where the CG-3 dependency parser needs improvement is represented by direct speech sentences that are not ending with a question converb or case. A possible approach is to use the POS tags for quotative clitics together with a lookup rule into a list of verbs of speech and create, in the first section of the CG-3 grammar, sentence boundaries after specific constructions such as *smras pa* སྨྲས་པ་, 'it is said', or *la gsol pa* ལ་གསོལ་པ་, 'said, replied'.

To create the first version of the UD Treebank for Classical Tibetan, I am curating a corpus of almost 300 sentences, carefully selected from three different texts. This approach will ensure that the corpus offers a well-rounded representation of the language from a diachronic point of view. The three texts in chronological order are: The Old Tibetan Annals (9th c.), The Old Tibetan Chronicle (10th c.) and the *Mi la'i rnam thar*, "The Biography of Milarepa" (15th.). After its validation, the treebank will be submitted and deposited in the UD project website for public use under the Creative Commons Attribution-ShareAlike (CC BY-SA) license.[8] All the material including the CG-3 grammars and the annotated CONLL-U files will be available as soon as they are ready at my Github UD_Tibetan repository.

Once the first version of the treebank is completed, it will be possible to train and test neural dependency parsers, like StanfordNLP, a transition-based neural parser trainable with CONLL-X files and word embeddings, and the UDPipe parser, a transition-based parser using a neural-network classifier which provides good results with small UD Treebanks. The UDPipe pipeline

7 The Segmentation and POS manual for Classical Tibetan are available on Zenodo at https://zenodo.org/records/7880130

8 https://universaldependencies.org/

is easily trainable on new languages with training data in CoNLL-U format and does not require additional resources such as morphosyntactic dictionaries or any feature engineering (Straka, Hajič & Straková 2016). At the same time I aim to expand the UD Treebank by incorporating additional texts and sentences, enhancing the accuracy and precision of the dependency parsers and improving their performance on a wider range of sentence structures and linguistic phenomena.

5. Conclusions

This paper outlines the procedures involved in developing a fully-annotated dependency treebank for Classical Tibetan. The process has been partially automated through the implementation of a CG-3 rule-based dependency parser. The data output provides a foundational framework, essential for the training of any neural model aimed at improving automated dependency parsing for the language.

The development of the Classical Tibetan UD corpus represents a significant contribution to both the linguistic and computational communities. For linguists, the corpus allows for a more comprehensive understanding of the grammatical structures of Classical Tibetan. It provides a wealth of data that can be used to analyze and describe the syntax and morphology of the language, including its word order and case marking. This data can be used to test linguistic theories and hypotheses, and to gain deeper insights into the nature of the language and its historical development. From an NLP point of view, the corpus has the potential to significantly improve the accuracy of NLP tools and applications for Classical Tibetan. With a complete dependency treebank, computational models can be trained to accurately parse and analyze Classical Tibetan texts, enabling the development of technologies such as machine translation and language generation. Furthermore, the corpus can be used to develop language models that can assist in automatic speech recognition, sentiment analysis, and other NLP applications.

Overall, the development of a full dependency corpus for Classical and Old Tibetan provides a valuable resource for scholars, researchers, and language enthusiasts interested in understanding and analyzing the language, as well as for developers seeking to build robust NLP tools and applications for Classical Tibetan.

Bibliography

Bick, Eckhard & Tino Didriksen. 2015. CG-3 — beyond classical constraint grammar. In *Proceedings of the 20th nordic conference of computational lin-*

guistics (NODALIDA 2015), 31–39. Vilnius, Lithuania: Linköping University Electronic Press, Sweden. https://aclanthology.org/W15-1807.

Dotson, B. & G. Hazod. 2009. *The Old Tibetan Annals: An Annotated Translation of Tibet's First History* Denkschriften (Österreichische Akademie der Wissenschaften. Philosophisch-Historische Klasse). Verlag der osterreichischen Akademie der Wissenschaften.

Dotson, B. & A. Helman-Ważny. 2016. *Codicology, paleography, and orthography of early tibetan documents: Methods and a case study* Wiener Studien zur Tibetologie und Buddhismuskunde. Arbeitskreis für Tibetische und Buddhistische Studien Universität Wien.

Faggionato, Christian, Nathan Hill & Marieke Meelen. 2022. NLP pipeline for annotating (endangered) Tibetan and newar varieties. In *Proceedings of the workshop on resources and technologies for indigenous, endangered and lesser-resourced languages in eurasia within the 13th language resources and evaluation conference*, 1–6. Marseille, France: European Language Resources Association.

Faggionato, Christian & Marieke Meelen. 2019. Developing the Old Tibetan treebank. In *Proceedings of the international conference on recent advances in natural language processing (ranlp 2019)*, 304–312. Varna, Bulgaria: INCOMA Ltd. doi:10.26615/978-954-452-056-4_035.

Faggionato, Christian, Marieke Meelen & Nathan Hill. 2023. Classical Tibetan Annotation Manual Part II - Segmentation & POS tagging. doi:10.5281/zenodo.7880130.

Garrett, Edward John & Nathan W. Hill. 2015. Constituent order in the tibetan noun phrase, .

Guibon, Gaël, Marine Courtin, Kim Gerdes & Bruno Guillaume. 2020. When collaborative treebank curation meets graph grammars. In *Proceedings of the twelfth language resources and evaluation conference*, 5291–5300. Marseille, France: European Language Resources Association. https://aclanthology.org/2020.lrec-1.651.

de Jong, J. W. 1959. *Mi la ras pa' i rnam thar*. Berlin, Boston: De Gruyter Mouton. doi:10.1515/9783112313008.

Jurafsky, Daniel & James H Martin. 2014. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Publishing.

Karlsson, Fred, Atro Voutilainen, Juha Heikkila & Arto Anttila (eds.). 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Berlin: Mouton de Gruyter.

Matthews, Peter Hugoe et al. 1981. *Syntax*. Cambridge University Press.

Meelen, Marieke, Élie Roux & Nathan Hill. 2021. Optimisation of the

largest annotated tibetan corpus combining rule-based, memory-based, and deep-learning methods. *ACM Transactions on Asian and Low-Resource Language Information Processing* 20(1). 1–11. doi:10.1145/3409488.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty & Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)*, 1659–1666. Portorož, Slovenia: European Language Resources Association (ELRA). https://aclanthology.org/L16-1262.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers & Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the twelfth language resources and evaluation conference*, 4034–4043. Marseille, France: European Language Resources Association. https://aclanthology.org/2020.lrec-1.497.

Straka, Milan, Jan Hajič & Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)*, 4290–4297. Portorož, Slovenia: European Language Resources Association (ELRA). https://aclanthology.org/L16-1680.

NLP for Readability, Graded Literature, & Materials Development in Tibetan*

Dirk Schmidt
(University of Wisconsin-Madison)

When it comes to learning-to-read, Tibetan is notoriously difficult. This is reflected in low literacy rates, low levels of reading comprehension, and struggles by early readers of all kinds—children, native speakers, heritage speakers, and second-language learners alike. The most recent statistics for the Tibetan Autonomous Region (TAR), for example, show rates of illiteracy from 21% to 34% (Reddy & Bhole 2023, Textor 2022). Beyer (1992) describes literacy in Tibetan as an "elite achievement". And earlier investigations into reading comprehension and vocabulary levels in diaspora contexts have found comprehension difficulties for Tibetan literature (Schmidt 2022a). At the core of this issue is diglossia, the gap between how Tibetan is spoken and how it is written (Ferguson 1959). Briefly put, the closer a writing system hews to a speech community's own natural language—the variety they use for everyday communication—the easier it is for speakers to learn to read. Correspondingly, the further apart speech and writing are, the more difficult literacy is (Koda, Zehler, Perfetti & Dunlap 2008).

In the case of Tibetan, written norms date to the 7th–11th centuries, and have changed little over the last 1,000 years (Tournadre & Gsang-bdag-rdo-rje 2003). This means that readers and writers must take effort to process their natural languages while decoding or encoding Tibetan text—mentally adding or subtracting letters that are no longer pronounced, replacing speech words for written corollaries, and making mental grammatical or syntactic substitutions or other changes. Tibetan text—even text written for early readers—is thus rarely highly readable. For example, following Nation's method for analysis (Hu & Nation 2000), a sample of 26 published children's stories was found to have an average readability of only 65%.[1] This is significantly lower

* Dirk Schmidt, "NLP for Readability, Graded Literature, & Materials Development in Tibetan", *Revue d'Etudes Tibétaines*, no. 72, Juillet 2024, pp. 70-85.

With thanks to Esukhia.

1 For this, I wrote Python code to import digital text from children's stories and calculate a readability percentage. To account for automation and segmentation errors, stories above 75% (rather than 98%) were deemed reasonably 'readable'. However, 9/10 stories still fell below this benchmark (Schmidt 2022b).

than the recommended vocabulary coverage of 98% for independent reading (ibid.).

This paper aims to show why this is the case, and what can be done about it. It builds on my previous work (Schmidt 2020), but is more expansive in scope, providing important updates, a thorough theoretical backing, and more technical details about the role of *Natural Language Processing* (NLP) to the work of readability. Specifically, I will cover the learning-to-read process step-by-step, and how this impacts early readers of Tibetan. Then, I will propose a new data collection technique for writing beginning reading material, the creation of story-specific 'mini speech corpora'. Finally, I will focus on how to apply these ideas to the Tibetan context using word segmentation in both *Dakje* (Esukhia 2022b) and *Botok* (OpenPecha 2023) for editing and level identification. My aim is to provide readers who wish to write, edit, or analyze early reading materials with the practical information, tools, and resources they need to do so, in the hopes that it benefits and supports readers of the Tibetan languages.

1. *Introducing Applications for NLP*

While technical and theoretical NLP work for Tibetan began in the 1990s (Hill & Jiang 2016), some of the more practical, everyday applications for the field are just now becoming widely available, or are now in their nascent stages. To provide a brief overview of NLP tools that provide practical applications for everyday users, large tech companies and small initiatives have both played important roles. Google Cloud Vision now provides Tibetan *Optical Character Recognition* (OCR) (Google 2023), building on early progress made by Namsel (Rowinski 2016). Microsoft recently released *Machine Translation* (MT) for Tibetan (Lekhden 2021). Tools like speech recognition (Ruan, Gan, Liu & Guo 2017) and spell-checking (Roux 2017) have also seen progress. These developments have followed in the footsteps of progress made in majority languages, like English.

Many of them are also dependent on progress in fundamental NLP areas like word segmentation and POS tagging (Hill & Jiang 2016). Dakje and Botok, the Python segmenter it relies on for word spacing and recognition, follow this trend of modeling itself on progress made in the larger languages. Specifically, Dakje uses Botok's word segmentation to build on ideas in vocabulary analysis, grading (or leveling), and readability scores found in tools built for grading and editing text (Chall 1948). Writers who write in English, for example, may use an editor like Hemingway (Long 2023) to analyze, simplify, and improve the readability of their text. Similarly, Dakje provides a user-friendly interface for readability editing in Tibetan. For users with ba-

sic Python coding skills, Botok provides the user with more advanced options for segmenting text for readability analysis. After briefly outlining the background to issues in readability, this article will present how word segmentation in Dakje and Botok have been used in the context of editing and analyzing Tibetan text for early reading materials for Esukhia,[2] a non-profit organization I work with that creates resources for Tibetan language learning.

2. *The Issue: Diglossia*

To understand how word segmentation is key to improving readability, it is important to first discuss the big-picture context of learning to read in general. When we take a closer look at how readers attain literacy, the ways in which diglossia creates obstacles to literacy become clear (Hudson 1992, Harbi 2022). Correspondingly, the ways in which word segmentation in Dakje or Botok helps writers clear these obstacles should also become apparent. For the purposes of this article, I will divide the road to literacy into four steps (below). This road map is greatly simplified. Learning to read is a complex process, and many of these 'steps' occur in parallel, and inform one another. Below, each of these steps will be introduced and expanded, followed by a discussion of how word segmentation works to improve the learning-to-read process:

2.1 *Developing speech skills & reading habits*

The first step on the road to literacy is developing speech skills and good reading habits. Children's (or a second-language learner's) exposure to oral language leads to the acquisition of speech skills. Meanwhile, being read to—out loud—creates motivation for reading, leading to good reading habits (Brock & Rankin 2008). Pressley & McCormick (2007) and others call this level "emergent literacy skills", while Callander & Nahmad-Williams (2011) draw important links between factors like early communication, rhythm, companionship, and social skills in early language development. The diglossic gap between speech and writing, however, make naturally-obtained speech skills

2 https://esukhia.net/

less useful in this learning-to-read process. Known words appear less frequently, and books that contain unknown words won't be understood, which impacts motivation for further reading. Sevinç & Backus (2019) give more details on this kind of language anxiety in a specific context, which is also discussed more below.

2.2 Connecting sounds to symbols

Next, a beginning reader must internalize the alphabetic principle, or what is called 'phonemic awareness'. These are the connections speakers make between the sounds from their natural language and the symbols found on the page (Brock & Rankin 2008: p.203). By sounding things out, they decode written words into speech words in order to understand the text. With practice reading out loud, oral comprehension gradually becomes reading comprehension. That is, understanding speech is what leads to understanding text. But when spellings are 'opaque'—that is, when there is not a one-to-one correspondence moving from symbols to sounds—'decoding' text becomes increasingly difficult, blocking this process. Research shows that children who learn to read in 'transparent' orthographies, for example, learn to read faster than those who learn 'opaque' ones (Koda et al. 2008). When sounding-things-out is difficult, reading is difficult; if the word that is decoded is an unknown word, it won't be understood.[3]

Modern Tibetan languages are, generally speaking, 'opaque', rather than 'transparent'. This is especially true of the Central Tibetan dialects most frequently spoken and studied in the diaspora and in the West more broadly. While the diaspora varieties are widely conflated with Central Lhasa Tibetan (Tournadre & Gsang-bdag-rdo-rje 2003), they have several unique features (Schmidt 2022a). Here, I prefer the term Zhichag Tibetan (*gzhis-chags skad*, "settlement language"), and examples from these varieties are the ones referenced in this article. They broadly share many of the pronunciation features of the Central Tibetan dialects that lead to 'opaque' spellings, such as consonant cluster reduction. It is reasonable to expect, then, that this would have an effect on speakers and learners of these varieties learning to read or write it. To give an example of orthographic depth in Tibetan:

(1) Some Tibetan words are 'transparent' (they have a 1:1 symbol:sound relationship): In *ku-shu*, ཀུཤུ, "apple", for example, all the consonants and vowels are pronounced as written, /ku-ɕu/.

3 It's worth noting that sometimes, even a 'known' word won't be comprehended (Hu & Nation 2000).

(2) Many more words, however, are 'opaque'—they contain consonant clusters that are no longer pronounced; are pronounced differently than they are written; or are otherwise inconsistent in their grapheme-to-phoneme relationships: In *'bras*, འབྲས་ "rice", for example, the initial *'a* is silent; the cluster *-br-* has been palatalized; and the final *-s* changes the vowel, but is itself not pronounced, /ʈe/.

2.3 Reading level-appropriate texts extensively

With a foundation of speech skills and the ability to decode symbols into sounds, the third step in attaining literacy is reading level-appropriate texts extensively (Jacobs & Farrell 2012). In other words, getting considerable amounts of practice at reading. By reading simple, age- or level-appropriate texts extensively, readers build up their word-recognition skills (i.e., 'automaticity'). They improve in speed, fluency, and reading comprehension. In order for this to happen, reading material must be highly readable. Again, research suggests an ideal vocab coverage of 98% (Schmitt, Jiang & Grabe 2011, Hu & Nation 2000). Below, in Figure 1, "known" vocabulary refers to words in a speaker's active vocabulary: Words they recognize, understand, and are able to use. A lack of easy, level-appropriate reading material that contains known words, however, puts a beginning reader at a disadvantage. Again, difficult texts can easily demotivate an early reader. A beginner in this situation is at risk of developing 'language anxiety', a negative feedback loop that leads to less and less reading (Sevinç & Backus 2019). Competition from other, easier literatures, can also lead a reader to prefer using another language altogether for reading and writing (for example English or Chinese).

Fig. 1 – The target vocab coverage for a level-appropriate text is 98%

2.4 *Vocabulary growth & learning from reading*

It isn't until these foundational skills have been obtained that a reader can begin using literacy to learn from reading (Nikolajeva 2014, Wolf 2010). In step four, speech vocabulary grows as the beginning reader reads more and more. Understanding and enjoying the texts that they read leads to more reading, and more reading leads to better reading skills. In contrast, reading less and less—because it is too hard, too anxiety-inducing, or too impractical—leads to not being good at reading. Not being good at reading then leads, again, to less reading. Figure 2 provides a graphic view of this feedback cycle, where not comprehending a text causes frustration; frustration leads to decreased motivation; and less practice reading leads to less comprehension (which, in turn, leads to further frustration). The opaque spellings, high level of unknown vocabulary, and lack of level-appropriate reading materials of diglossia all contribute to this "vicious cycle" of language anxiety (Sevinç & Backus 2019).

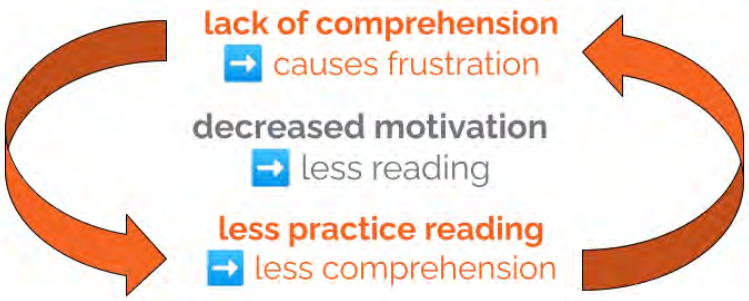

Fig. 2 – The negative feedback loop of 'language anxiety'

3. *The Solution: Readability*

Breaking this cycle requires easy-to-read texts that increase reading, motivation, and literacy. In the case of Tibetan, addressing issues like opaque spellings and shifts in grammar that have arisen from diglossia would require comprehensive language reforms, something that would take widespread social support and political will. Given the sociopolitical marginalization these languages face, this seems unlikely in the foreseeable future. However, vocabulary choice is something that any author, writer, or material developer can easily address. By choosing known words that occur naturally in speech over unknown ones, the readability of any given text can be greatly increased. And Natural Language Processing (NLP) using word-segmentation tools makes this not only possible, but easy and efficient. The following sections address

how Dakje and Botok supports writing easy-to-read text by discussing each step in the development and implementation process of these tools:

3.1 Collecting natural speech

For our purposes, then, it's important to define natural language strictly in both time and space. That's because the NLP that is useful in the context of readability is dependent on the speech community it will be used to benefit, or on the target language the learner is hoping to acquire. For Tibetan, this necessitates addressing the diglossic gap between speech and writing, as discussed above. But also requires recognizing that not all Tibetan speech communities use the same words and grammar in their natural speech—natural language is unplanned, naturally occurring, and constantly changing. Of the fifty or more Tibetan languages that exist, as defined by their mutual comprehensibility (Tournadre 2014), each have their own unique pronunciations, vocabularies, and grammars. To put it another way, a frequency list based on the words Zhichag Tibetan speakers use will not be the same as a list based on the words, say, Amdo Tibetan speakers use. Even if the target literature of Standard Literary Tibetan is the same, the early stepping stones may be different for different speakers and learners of different varieties. We need to know what words speakers know. This requires collecting natural speech data.

Methodologically, 'collecting natural speech' means recording it using a voice recorder; transcribing it as it was spoken (that is, non-prescriptively, making no edits or corrections); and organizing the data for ease of analysis. For large corpus projects, the more data, the better. For smaller projects, however, we may use data that targets a specific demographic; a particular age group; or even an individual story. Creating these kinds of 'mini corpora' for purposes of analysis and readability is one way to apply speech corpus creation to language learning and literacy (Beeching 2014, O'Keeffe, McCarthy & Carter 2007). One such example is the story "The Race", an open source children's story from Pratham's Storyweaver website (Figure 3).[4]

In our recent work creating mini speech corpora, we began by telling the story, orally, to children. Using the images (but no text), we then allowed

4 https://storyweaver.org.in/

Fig. 3 – An example story, "The Race", is used to illustrate the feedback speech corpora can provide.

the children to re-tell the story back to us, in their own words, recording the result. In this way, we are able to limit the amount of data we have to collect, while ensuring we have story-specific vocabulary to work from. Afterwards, the recordings were transcribed, resulting in a mini speech corpus. This corpus contains the speech versions from several children of the same story (Esukhia 2022a). If the speech corpus researcher is also the writer of the Tibetan version, very little post-editing is needed. The result is a graded story, told in words that the children used (and thus, words that we can be sure they know). However, to make speech corpora more widely useful, further steps are useful for improved readability. The next section will explore how to apply this data using word segmentation in Dakje and Botok.

4. The Path: NLP Tools

So far, this article has introduced the problem: Tibetan texts have low readability due to diglossia. Texts contain opaque spellings, hard words, and literary grammar that do not occur in natural speech. It has also offered a solution: Readable texts for early readers that have a high percentage of known words from natural speech. The goal of using the NLP tools Dakje and Botok is identifying words that might be hard, or giving an overall sense of the grade level of a text based on its vocabulary. We do this by processing a text input, splitting it into words using NLP tools, and comparing those words to frequency lists made from speech data. Dakje gives the words a color based on

how often they appear in natural speech. Similarly, Botok may be used within Python directly in much the same way, by segmenting text and comparing it to the word lists from natural speech. In this section, I will first discuss the details of this process; then, how it is applied in a real-world context in order to analyze, grade, or write children's stories or textbook materials.

4.1 Processing the data

The idea behind frequency lists is that the more often a word is used, the easier it is, and the more people are likely to know it. Hard words, in contrast, are rarer. They are used less often, and fewer readers are less likely to know them. Frequency has been used like this since the early days of graded reading to give researchers an idea of vocabulary difficulty (Chall 1948, DuBay 2007). For Tibetan, however, 'the word' is not an obvious unit: while the inter-syllabic Tibetan punctuation mark, or *tsheg*, indicates syllable boundaries, there is no punctuation that shows word boundaries. Ideally, we want to outsource the tasks of identifying, counting, and sorting Tibetan words to the machine. The result is word lists, ranked by frequency, that we can then split and sort into level lists.

In other words, here, a word's 'level' is defined by its frequency. While it is generally recognized that some words are easier and others harder, there is no universal, agreed-upon standard for precisely defining vocabulary levels or their lengths. For second-language learning, however, J & Alexiou (2009) provides basic guidance for length (or size) based on the Common European Framework of Reference for languages (CEFR). There, for example, 1,500 words is the suggested Beginner Level, or Level A1 (ibid). Combining this with the ideas from Chall (1948), among others, I have split and sorted words based on the principle that frequent words are easier and infrequent words are more difficult. This results in a set of lists that are used as reference points for vocabulary difficulty by level. In addition to this general data, we also have the story-specific vocabulary lists taken from the mini corpora collected for the stories.

4.2 Writing level-appropriate texts

As discussed in Section 3.1, writing or translating a beginning text directly (that is, without feedback on readability) will be successfully level appropriate if and only if the writer researches children's speech themselves. As shown in Figure 1, any percentage of unknown vocabulary beyond 2% is burdensome. For reference, this would be 9–10 unknown words every page in an article like this one. It's easy to see how unrecognizable words can lead to not

reading when they make up even more than that. Imagine not knowing 20+ (5%), 40+ (10%), 60+ (15%) or even more of the words on each of these pages! In a ten-page article, you'd encounter hundreds of words you didn't know. In contrast, while it may seem counter intuitive, easy-to-read texts lead to more and better reading. This is why level appropriate texts are so important for literary achievement.

Perhaps surprisingly, translation of a level-appropriate story written in English, for example, does not automatically yield a level-appropriate Tibetan version (Schmidt 2020). That is because vocabulary choice in Tibetan writing is heavily influenced by many factors, including both traditional literary standards, as discussed above, as well as the movement for a modern "Pure Tibetan" (Tib. *bod-skad gtsang-ma*) (Thurston 2018). The fear that modern loanwords are 'degrading' Tibetan has led to large dictionary projects that collect, define, create, and publish Chinese-English-Tibetan dictionaries for new, modern vocabulary (Blo-gros 2013). Yet, while Tibetan children do use loanwords, the rate—even in the diaspora, amongst the youngest generations of speakers—does not seem to be particularly high. For example, if we analyze transcripts of diaspora children telling stories (Esukhia 2022a), we find that modern loanwords make up less than 1% of the total words spoken. This is represented in Figure 4. While a desire to preserve and promote Tibetan language is commendable, the impact of each additional unknown word can add up, overly burdening a beginning reader. The vocabulary choices from the many versions of "The Race" found on the StoryWeaver website exemplify this issue. Each of these vehicles has a specific neologism in Pure Tibetan, and this is reflected in the translated versions; the children, however, naturally used different vocabulary when speaking during corpus collection, suggesting they may not actually know or use these terms (see Table 1).

Fig. 4 – Even in diaspora children's speech, modern loanwords make up less than 1% of the total words spoken.

English	Pure Tibetan	Speech word
bus	*spyi-spyod-rlang-'khor*, སྤྱི་སྤྱོད་རླང་འཁོར་	*bus*, བུ་སི་
auto rickshaw	*'khor-gsum-snum-'khor*, འཁོར་གསུམ་སྣུམ་འཁོར་	*auto*, ཨ་ཐོ་
car	*rlangs-'khor*, རླངས་འཁོར་	*mo-Ta*, མོ་ཊ་

Table 1 – The influence of 'Pure Tibetan' on vocabulary choice in children's stories; an example from "The Race". On the left, Pure Tibetan terms found in the published stories; on the right, natural speech loanwords found in the mini speech corpus.

4.3 Using Segmentation

As discussed above, segmentation is key to identifying non-level vocabulary, or unknown words beginning readers will find difficult. Whether done in Dakje (Esukhia 2022b) or Botok (OpenPecha 2023), the segmentation process relies on the same background processes. At its core, these tools implement a "max match" algorithm for word recognition. Tibetan input text is compared to a large dictionary—in essence, a word list—and segmented based on matches to this list. In Dakje, the general word list is a dictionary of Standard Literary Tibetan. The benefit of using this software is that Dakje will automatically segment Tibetan text, and highlight vocabulary items by level. It will also calculate the distribution percentage across those lists (Figure 5, right panel), and display the total readability of the text (Figure 5, top bar). Users can then use this feedback to edit problematic words directly in the editor.

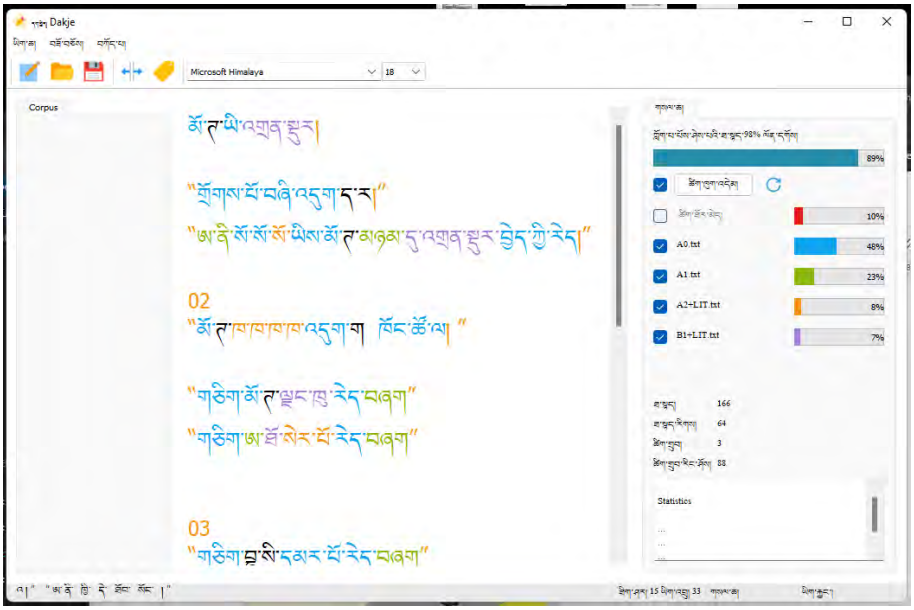

Fig. 5 – 'Dakje', an NLP-based word editing software for grading Tibetan texts, can provide actionable feedback for authors by highlighting words by level

The drawback, however, is that this the general dictionary lacks many words that are specific to speech in Modern Tibetan dialects. For those with coding ability, however, Botok allows the user to define a 'dialect pack' to improve word spacing. In the case of editing specifically for Zhichag Tibetan, for example, I prepared a word list of speech words from two Esukhia speech corpora: The Nanhai Corpus (Esukhia 2020) and The Children's Speech Corpus (Esukhia 2022a). I then loaded the speech word list into the dialect pack's "words" folder to call it when word spacing. I then segmented text by configuring Botok's Python module as below in Table 2. After word-spacing a text using this method, I then loaded the frequency lists as Python lists. This allowed me to grade texts directly by comparing them against the frequency lists. While this improved method still doesn't word-space perfectly, it is good enough for practical applications. For example, it allowed me to automatically assign levels to Esukhia's story database (Esukhia 2023). The database currently contains 42 stories, including five levels, L0–L4. These levels are roughly equivalent to the CEFR levels A0–B2, graded by word lists based on CEFR numbers (J & Alexiou 2009) and rates of unknown vocabulary (Nation & Hirsh 2020).[5] With a rise in the amount and quality of children's literature

5 These ideas have also played a role in other materials development. See, for example, stories, games, and textbooks found on Esukhia's website.

in Tibetan, the hope is that these tools will reach a wider range of authors, writers, and material developers. With more, and more readable, content, beginning readers should have more opportunities that help them along the path to literacy.

```
def word_split(text):

  """takes a string of text and word spaces it based on the Zhichag dialect pack"""

    words=[]
    if __name__ == "__main__":
        config = Config(dialect_name="zhichag", base_path= Path.home())
        wt = WordTokenizer(config=config)
        tokens = get_tokens(wt, text)
        for token in tokens:
            words.append(token['text'])

    return ' '.join(words)
```

Table 2 – A coding sample: Creating a user-defined 'dialect pack' for use in the Botok Python module.

5. Concluding Remarks

Using Dakje and Botok segmentation in the context of readability and applied linguistics is thus an important application for NLP in Tibetan. It can help authors, writers, and material developers ensure that they are providing their students, children readers, or language learners level-appropriate materials that help them develop good reading habits; connect symbols to sounds; and read extensively. Because diglossia manifests as unknown, difficult traditional or "pure" vocabulary in beginning reading materials, NLP methods and tools like Dakje and Botok have an important role to play in breaking the cycle of language anxiety that comes hand-in-hand with attempts to attain literacy in diglossic languages. My hope is that the details provided in this article will support and encourage others to explore these tools to improve the readability of their own children's stories and Tibetan language learning materials, too.

Bibliography

Beeching, Kate. 2014. Corpora in language teaching and learning. *Recherches en didactique des langues et des cultures* 11(1). doi:10.4000/rdlc.1672.

Beyer, Stephan V. 1992. *The classical tibetan language*. Albany: Suny Press.

Blo-gros, Tshul-khrims. 2013. *Rgya-bod-dbyin-gsum gsar-byung rgyun-bkol ris-'grel ming-mdzod[chinese-tibetan-english illustrated dictionary of new daily vocabulary]*. So-khron Mi-rigs Dpe-skrun-khang[Sichuan Nationalities Publishing House].

Brock, Avril & Carolynn Rankin. 2008. *Communication, language and literacy from birth to five*. Los Angeles: SAGE.

Callander, N. & L. Nahmad-Williams. 2011. *Communication, language and literacy* Supporting Development in the Early Years Foundation Stage. Bloomsbury Academic.

Chall, Dale E. 1948. A formula for predicting readability. *Educational Research Bulletin* 27. 11–20+28.

DuBay, William H. 2007. *Smart language: readers, readability, and the grading of text*. Costa Mesa, Calif.: Impact Information. OCLC: 164437606.

Esukhia. 2020. The nanhai corpus. https://github.com/Esukhia/Corpora/tree/master/Nanhai.

Esukhia. 2022a. Children's stories speech corpus. https://github.com/Esukhia/Corpora/tree/master/Childrens_Stories.

Esukhia. 2022b. Dakje. https://github.com/Esukhia/dakje-desktop.

Esukhia. 2023. Stories. https://esukhia.online/stories/.

Ferguson, Charles A. 1959. Diglossia. *WORD* 15(2). 325–340. doi:10.1080/00437956.1959.11659702.

Google. 2023. Ocr language support. https://cloud.google.com/vision/docs/languages.

Harbi, Mohammed. 2022. Arabic diglossia and its impact on the social communication and learning process of non-native Arabic learners: Students' perspective. *SSRN Electronic Journal* doi:10.2139/ssrn.4037655.

Hill, Nathan W. & Di Jiang. 2016. Tibetan natural language processing. *Himalayan Linguistics, Vol. 15(1)* .

Hu, Marcella & Paul Nation. 2000. Unknown vocabulary density and reading comprehension. *Reading in a Foreign Language* 13.

Hudson, Alan. 1992. Diglossia: A bibliographic review. *Language in Society* 21(4). 611–674. http://www.jstor.org/stable/4168395.

J, Milton & T. Alexiou. 2009. *Vocabulary size and the common european framework of reference in languages* 194–211. Macmillan.

Jacobs, George M. & Thomas S. C. Farrell. 2012. *Teachers sourcebook for extensive reading*. Charlotte, N.C: Information Age Pub.

Koda, Keiko, Annette Marie Zehler, Charles A. Perfetti & Susan Dunlap. 2008. *Learning to read: General principles and writing system variation* 13–38. Routledge.

Lekhden, Tenzin. 2021. Microsoft's translation app includes Tibetan language. https://www.phayul.com/2021/12/03/46489/.

Long, Adam Ben. 2023. Hemingway editor. https://hemingwayapp.com/.

Nation, Paul & David Hirsh. 2020. What vocabulary size is needed toread unsimplified texts for pleasure? doi:10.26686/wgtn.12560417.v1.

Nikolajeva, Maria. 2014. *Reading for learning: cognitive approaches to children's literature* (Children's literature, culture, and cognition v. 3). Amsterdam ; Philadelphia: John Benjamins Publishing Company.

O'Keeffe, Anne, Michael McCarthy & Ronald Carter. 2007. *From corpus to classroom: language use and language teaching*. Cambridge ; New York: Cambridge University Press. OCLC: ocm76935901.

OpenPecha. 2023. Botok. https://github.com/OpenPecha/Botok.

Pressley, Michael & Christine B. McCormick. 2007. *Child and adolescent development for educators*. New York: Guilford Press. OCLC: ocm67383559.

Reddy, Rahul K. & Omkar Bhole. 2023. Analysing China's Census Report .

Roux, Elie. 2017. Hunspell: Tibetan spellchecker. https://github.com/eroux/hunspell-bo.

Rowinski, Kurt, Zach;Keutzer. 2016. Namsel: An optical character recognition system for tibetan text. https://escholarship.org/uc/item/6d5781k5.

Ruan, Wenbin, Zhenye Gan, Bin Liu & Yinmei Guo. 2017. An improved tibetan lhasa speech recognition method based on deep neural network. *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)* 303–306.

Schmidt, Dirk. 2020. Grading tibetan children's literature: A test case using the nlp readability tool "dakje" . *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 19(6). doi:10.1145/3392046. https://doi.org/10.1145/3392046.

Schmidt, Dirk. 2022a. *On the yak horns of a dilemma: Diverging standards in diaspora tibetan* 127–156. Amsterdam University Press. doi:10.1017/9789048552719.006.

Schmidt, Dirk. 2022b. Tibetan readability –a python project. https://github.com/thedirk/readability/blob/main/p00.ipynb.

Schmitt, Norbert, Xiangying Jiang & William Grabe. 2011. The percentage of words known in a text and reading comprehension. *The Modern Language Journal* 95(1). 26–43. http://www.jstor.org/stable/41262309.

Sevinç, Yeşim & Ad Backus. 2019. Anxiety, language use and linguistic competence in an immigrant context: a vicious circle? *International Journal of Bilingual Education and Bilingualism* 22(6). 706–724.

doi:10.1080/13670050.2017.1306021.

Textor, C. 2022. Illiteracy rate in China in 2021, by region. https://www.statista.com/statistics/278568/illiteracy-rate-in-china-by-region/.

Thurston, Timothy. 2018. The purist campaign as metadiscursive regime in china's tibet. *Inner Asia* 20(2). 199 – 218. doi:https://doi.org/10.1163/22105018-12340107.

Tournadre, Nicolas. 2014. *The Tibetic languages and their classification* 105–130. Berlin, Boston: De Gruyter Mouton. doi:10.1515/9783110310832.105.

Tournadre, Nicolas & Gsang-bdag-rdo-rje. 2003. *Manual of standard Tibetan: language and civilization: introduction to standad Tibetan (spoken and written) followed by an appendix on classical literary Tibetan*. Ithaca, N.Y: Snow Lion Publications.

Wolf, Maryanne. 2010. *Proust and the squid: the story and science of the reading brain*. New York: Harper Perennial 1st edn.