


Assembling a Digital Toolkit: An Introduction to Text-mining for Modern Tibetan

Robert Barnett (SOAS University of London)
and
James Engels (University of Edinburgh)

he rapid development of new computational tools and methodologies for use in the humanities and social sciences raises both hopes and challenges for those who try to work with them. In this introduction to this special issue “From Print to Pixels: Building Digital Tools for Modern Tibetan Textual Analysis” of the *RET*, we describe some of the emerging tools designed to assist researchers working in the digital humanities on texts in the Tibetan language and discuss some of the challenges that are part of that effort. Our discussion complements the paper by Meelen, Nehrdich and Keutzer (2024), which described field-wide developments in a wide range of computational tasks involving Tibetan, including Handwritten Text Recognition (HTR), post-processing correction, speech-to-text technology, and digitally-assisted language learning for Tibetan. By contrast, this special issue presents, through a set of seven studies, a step-by-step summary of the main tasks involved in developing a single digital humanities project in the Tibetan studies field.

Our project, “Divergent Discourses,”¹ aims to identify the various discourses and narratives produced in newspapers within Tibet (after

¹ The Divergent Discourses project received funding from the Deutsche Forschungsgemeinschaft (DFG) under project number 508232945 (<https://gepris.dfg.de/gepris/projekt/508232945?language=en>), and from the Arts and Humanities Research Council (AHRC) under project reference AH/X001504/1

1950, in China) and Tibetan print media produced outside Tibet (mainly in India or Nepal) in the late 1950s and early 1960s. The project aims to track the historical evolution of discourses on both sides of the Himalayas during such pivotal episodes as the “democratic reforms”, the various uprisings by Tibetans in the later 1950s, the flight to India, and the ensuing “elimination of the rebellion” within Tibet. To do this, the project team has worked for the last two years on developing or refining a number of computational tools and processes that are designed to facilitate text-mining, and which we have adapted for use with modern Tibetan texts. Developing those tools has involved a number of tasks, which we briefly describe in this introduction. These involved primarily (1) finding, collecting and scanning the source documents; (2) checking and improving the images; (3) training a machine to transcribe the image content into textual form; (4) post-processing, including normalisation, paragraph extraction and adding metadata; (5) training a Tibetan-language model to enable the text-mining platform to work with Tibetan texts; (6) developing or adapting tools to recognise words in Tibetan and their parts-of-speech (POS) in order to produce training data for the language model; and (7) developing a set of text-mining tools that rely on a technology that is distinct from that used by the text-mining platform, in case the latter failed to work well with Tibetan.

Each of the papers in this special issue of *RET* describes some of the choices, considerations and (eventually) successes that we faced in carrying out these tasks. In addition, this introduction includes brief sketches of the historical background of these processes and the rapid changes taking place in the field of computationally-assisted text mining, together with the implications of these changes for digital Tibetan studies. The papers also include a description of the Divergent Discourses corpus of early Tibetan newspapers and their history, a study of Tibetan transcribing practices for foreign names, an analysis of religious policy in Tibet in the last decade, and a survey of a

(<https://gtr.ukri.org/projects?ref=AH%2FX001504%2F1>). For more information on Divergent Discourses, see <https://research.uni-leipzig.de/diverge/> (accessed on January 10, 2025) and the other contributions to this special issue.

particular discourse in an overseas Tibetan newspaper in the 1950s and early 1960s. We hope these papers will be helpful as a kind of travel guide to other researchers setting out on similar journeys in the future.

1 *Building the Corpus*

The project's source materials are 16 Tibetan-language newspapers published in the 1950s and the early 1960s. These add up to 16,718 newspaper pages. Although still a comparatively small collection, it would be very hard for a single researcher or a small team to study this volume of texts unless assisted by digital tools that allow automated or semi-automated textual analysis. It is not enough, however, just to acquire such tools from the internet and then apply them to the texts in our collection. Before any such application or analysis can be performed, a number of preparatory tasks are necessary, including, in particular, training these tools and their underlying platforms or systems so that they are able to work with texts that are in the Tibetan language, and specifically, in our case, in modern Tibetan. Only then can we reliably use computational tools to identify keywords, topics, sentiments, names and other details in the texts, which in turn will enable us to trace in detail the narratives and discourses circulating at the time when these articles were published.

Developing and adapting these tools involves numerous decisions and value judgements, most of them requiring considerable technical expertise.² In this introduction, we summarise some of the factors involved in those decisions and the basic concepts underlying them. As many researchers in the field of Tibetan studies know well, these concepts are largely drawn from the field known as Natural Language Processing (NLP). They include, firstly, the notion of a corpus, which means a large number of machine-readable texts compiled into a single body, enabling what is popularly called "big data" analysis and text-

² The project's technical decisions were guided and implemented primarily by the team's technologists and research assistants, James Engels and Christina Sabbagh in London and Yuki Kyogoku in Leipzig.

mining. The composition of the Divergent Discourses Corpus is described in the paper in this issue by **Franz Xaver Erhard**, “The Divergent Discourses Corpus: A Digital Collection of Early Tibetan Newspapers of the 1950s and 1960s” (Erhard 2025a). Erhard provides extensive historical background about the emergence of Tibetan-language newspapers since the early 1900s and describes the rapid increase in the publication of such papers on both sides of the Himalayas during the 1950s. Noting their exceptional importance in the case of Tibetan studies, not least because access to archives in Tibetan areas of China is rarely permitted for foreign (or even local) researchers, he also shows how our collection, like any corpus, contains structural biases as a result of incomplete collections, lost items, illegible copies, and so forth. Nevertheless, the corpus represents the first known attempt to make a comprehensive set of Tibetan-language newspapers from that era publicly available.

Creating a corpus is more or less an essential requirement for applying the various tools or techniques of text-mining that are so far available. Obviously, the main task in corpus creation is acquiring copies of each page of each newspaper, which might initially take the form of photographs, microfilm, or microfiche, and then, if necessary, converting any chemically-produced (sometimes incorrectly termed analogue) images into a digital format by scanning them to produce a digital image file. But a number of operations – usually referred to as “pre-processing” – have to be carried out on each image before it can be added to a corpus. Among these is improving the quality of any damaged or sub-standard images by using image-enhancement tools. As Erhard noted in his paper, sub-standard images can bias the representativeness of a corpus, leading researchers to assume that their findings reflect the totality of the corpus contents when, in fact, they may only include those with sufficient image quality to be accurately analysed and transcribed by a machine.

It was to reduce this risk that **Christina Sabbagh** developed a procedure for assessing and improving the quality of images in the corpus. As she explains in her paper, “Enhanced HTR Accuracy for Tibetan Historical Texts – Optimising Image Pre-processing for Improved Transcription Quality” (Sabbagh 2025), this task was

particularly necessary in the case of a microfilm edition published by the China National Microforms Import and Export Corporation, Beijing, in the 2000s. That microfilm edition contains the only known copies of several thousand newspaper pages in Tibetan, but many of its images are missing, incomplete, underexposed, stained, or obscured by dark patches. She describes her methodology for automating the identification of those damaged images, and, in some cases, for digitally improving them.³ She notes that no single procedure can fully restore all poor-quality images due to their varying levels of degradation, but enough to recover and extract the majority of the text contained. Sabbagh suggests additional post-processing strategies could be explored in the future to try to compensate for these difficulties, including the Turing Institute's MapReader tool and the Impreso Project's keyword suggestion tool (see also the description of the Norbu Ketaka project in Luo and van der Kuijp 2024, which uses NLP and computer vision models to “clean up” or improve machine-transcribed Tibetan texts). Sabbagh emphasises the importance of recognising that automatic text recognition is prone to errors, and that downstream applications using machine-readable text must be designed with this assumption in mind to yield truly representative conclusions.

The next major step towards a machine-readable corpus is automatic transcription or “text recognition” – using a machine to extract visual information, such as letters and words, from an image file and to convert that information into textual form. The textual form, in this case, means a file that can be read by a machine such as a word processor, so that it can be processed and manipulated by computational tools. This process is conventionally known as “optical character recognition” or OCR. In his paper, “Text and Layout Recognition for Tibetan Newspapers with Transkribus” (Erhard 2025b), **Franz Xaver Erhard** points out that strictly speaking automatic transcription cannot be termed “OCR” when it involves Tibetan texts, because in the case of Tibetan the transcribing algorithms are trained

³. The project's image enhancement code is at https://github.com/DivergentDiscourses/dd_custom_preprocess (see also Sabbagh *et al.* 2024a, b).

to recognise the shapes created by strings of letters (usually a line) on the page, not the individual characters. This shape-recognising technology, known as Handwritten Text Recognition or HTR, was first discussed in more detail in the Tibetan context in Griffiths 2024.⁴ In his paper, Erhard describes the complex and time-consuming task of training an HTR machine to transcribe Tibetan newspaper texts. One first has to select an automatic transcription programme, in this case, the online service Transkribus,⁵ and then produce what is called “Ground Truth”, meaning hundreds of pages of accurate and verified manual transcriptions that can be fed to the machine alongside the digital images of those pages. These train the HTR algorithm to recognise the specific handwriting or typeface of one or other publication or source. For best results in model training, the Ground Truth used for training must adhere to a standard set of transcribing principles to ensure that all pages are transcribed identically. While this transcription imperative is easily accepted in theory, in practice many challenges arise.

One of these challenges involves Tibetan punctuation practices. The Tibetan writing system uses a number of unique symbols to mark texts, such as the “fish-eye”, the “bullseye”, the “black up-pointing triangle”, or the *che mgo* preceding the name of high incarnate lamas. In addition, Tibetan newspapers sometimes incorporate symbols borrowed from other writing systems such as Chinese or English. The transcription model, therefore, has to be instructed as to which Unicode character should be used to transcribe each of these symbols or marks. Details of the transcribing conventions developed by the project are given by Erhard in his “Manual for transcribing historical Tibetan newspapers (in Transkribus)”, included as an appendix to his paper on “Text and Layout Recognition”.

In the case of modern printed books, HTR is a relatively straightforward task because the physical layout of a page in such a

⁴ Erhard uses the term Automatic Text Recognition (ATR) as an umbrella term including both OCR and HTR.

⁵ For more on Transkribus (www.transkribus.org), see Kahle *et al.* 2017.

book is usually standardised.⁶ But, as Erhard shows in his article, this is not the case with early Tibetan newspapers, which have highly complex layouts with varying numbers of columns as well as photographs, advertisements, mastheads, headings, sub-headings, captions, page numbers, and other forms of design or layout features on a page (these are known as “text regions”). A transcription model has to be trained to recognise these regions and to follow the different reading conventions needed to correctly transcribe texts within each of them. To teach these conventions to the machine, Erhard had to develop what is known as a “Field Model” in Transkribus, an algorithm trained to recognise the different text regions, as well as to train models that would recognise what are called “baselines”, the line which, conceptually, runs through the centre of each line of text, and “line polygons”, which are boxes that one teaches the machine to recognise as tightly enclosing each line of text. In his paper, Erhard describes the innovative three-step workflow he developed to enable automatic text recognition to compensate for the complex layouts of Tibetan newspapers. It took some two years of work, but with this method, he was able to successfully train a Transkribus model to read early Tibetan newspapers for the Divergent Discourses project. Whereas Transkribus advises that transcription models with a Character Error Rate of 10% or less should be considered successful, our model, TibNews4All 0.2, has a Character Error Rate of 2.52%.⁷

2 After the HTR stage: Post-processing

Once the text recognition process has produced digital transcriptions of the collected images with an acceptable level of accuracy, a number

⁶ The Divergent Discourses project published a model on Transkribus for modern printed Tibetan books from the PRC in March 2024, Tibetan Modern U-chen Print 0.1 (TMUP 0.1). It is the first Transkribus HTR model for printed Tibetan language publications in Uchen (འུ་ཅེན་ *dbu can*) script, available at <https://readcoop.eu/model/tibetan-modern-u-chen-print/>; see also Erhard *et al.* 2023.

⁷ The TibNewsOne4All 0.2 (ID 169581) is available in Transkribus at <https://www.transkribus.org/model/tibnewsone4all> (accessed on January 14, 2025).

of post-processing steps have to be taken to maximise the value of the corpus for researchers. The first of these is normalisation. This step, which is carried out after the transcription process on a copy of the raw transcriptions, involves writing code that standardises variations in orthographic or other practices in the document. In our case, we wrote code that included ensuring that all signs and characters in the text are encoded according to the Unicode UTF-8 standard, spelling out abbreviations, replacing any non-Unicode characters, standardising numbers, and removing non-significant spacing or *tshegs* (the small inline dot or inverted triangle (·) used to separate syllables).⁸ Without this process, searches and similar tools will fail to capture all, or at least most, instances of any given term or feature – searching for the number “8”, for example, would find only texts which used the Arabic numeral 8, and not those that use the Tibetan numeral ༘. A well-designed corpus will usually retain an option for researchers to view the original or “raw” form of each transcribed text in case they wish to see the spellings, numbers or punctuation used in the transcribed text prior to normalisation.⁹

In our case, the normalisation process brought our attention to a specific issue in Tibetan orthography: the wide range of variation in the spelling of non-Tibetan words and names. This led to the paper by **Franz Xaver Erhard** and **Xiaoying**, “Foreign Names and Places in Tibetan Newspapers” (Erhard & Xiaoying 2025), which presents an innovative study of Tibetan naming practices in the 1950s and 1960s. It shows the remarkable variation in spellings of foreign terms in Tibetan texts at that time, a reflection, the authors note, of the rushed nature of China’s translation project in Tibet, where seemingly officials had no time to set up bodies in their new territory to standardise such translation practices. Erhard and Xiaoying show the wide variations in spellings that this produced and explain some of the reasons for those discrepancies – they found, for example, twelve forms in Tibetan of the name of the former Chinese premier, Zhou Enlai, many of them related

⁸ For our code for post-transcription normalisation, see Kyogoku *et al.* 2024.

⁹ The project’s normalisation process is discussed in Kyogoku *et al.* 2025, section 3.3. For the code, see <https://github.com/Divergent-Discourses/TibNorm>.

to differences in pronunciation practices in different Tibetan or Chinese dialects. Again, these variants would ideally be normalised if a researcher is going to be able to find all instances of a given name in the corpus.¹⁰

In some cases, the texts to be included in a corpus do not need automatic transcription because they are “born digital” – that is, they already exist in a machine-readable form, such as those that are found on the internet in HTML format. This is rare in the case of historical documents, but even with texts that can be directly downloaded or

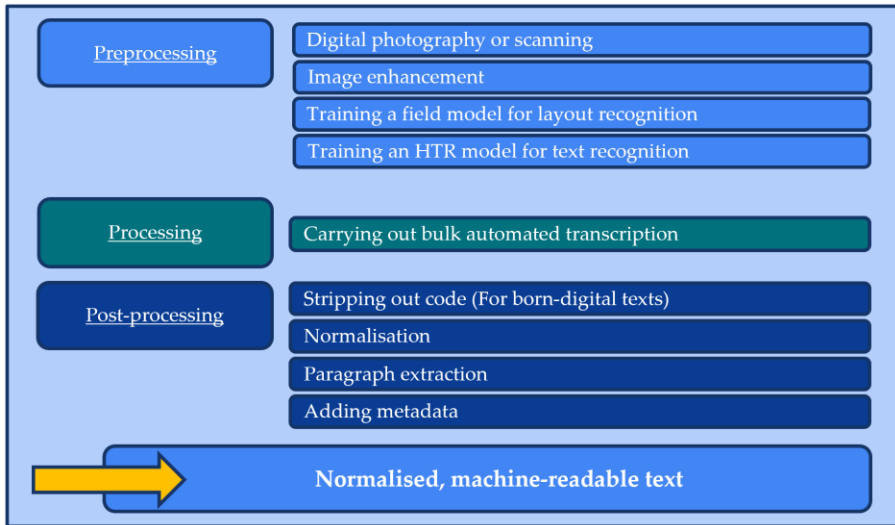


Figure 1 The basic steps required to produce normalised, machine-readable text.

“scraped” directly from websites in such a format, one then needs to invoke a code or procedure that will strip out non-essential features like formatting marks, non-standard characters, or HTML coding. Such code is, however, already available in many forms online.¹¹

An essential step in the pre-processing pipeline, before a corpus can be used effectively, is the addition of metadata to each item in the collection, including the title of the source publication, the date and

¹⁰ The name-lists compiled by Erhard and Xiaoying are available at <https://zenodo.org/records/14526125>. See Erhard, Xiaoying *et al.* 2024.

¹¹ See for example Leonard Richardson, “beautifulsoup4 4.12.3”, <https://pypi.org/project/beautifulsoup4/> (accessed January 21, 2025).

place of publication, and the number of the page on which it was originally published. Basic metadata is attached by Transkribus to each transcribed text in an XML file that it outputs together with the transcription. This includes the file name of the source and the tags for identified text regions. For our project, we initially wrote code that aimed to add metadata extracted from the online catalogues of the libraries which had held the original copies of the newspapers, but for technical reasons this failed and we instead added such metadata by hand. In addition, because Tibetan texts do not consistently mark sentences, we decided to treat the paragraph as our smallest unit of analysis and so wrote code that subdivided each transcribed text into paragraphs or segments of text. This code attached metadata to each of these paragraphs, linking each to its source document.¹²

3 *Preparing Tools for Text-analysis*

After the pre-processing stages, the texts can be grouped into a collection or corpus, which can be simply a folder on a hard drive containing the raw texts. The process of developing or applying tools for analysis of that corpus can then begin. There are four main analytical tools or methods which are commonly used in corpus-based text-mining:

- **keyword searches**, where the computer searches for occurrences of a particular word or phrase (a “string” in computer-speak) and computes their frequency;
- **sentiment analysis**, where the programme identifies texts that share a particular sentiment, such as happiness or disapproval;
- **topic modelling**, where the computer recognises which passages focus on a particular topic and groups paragraphs or sentences into shared topics or themes; and
- **named entity recognition (NER)**, which identifies the elements in a text that are names of entities such as a person, institution or place.

¹² See https://github.com/Divergent-Discourses/transkribus_utils and Engels *et al.* 2024.

Over the last four decades or so, computational linguists have developed four main approaches or techniques for carrying out these tasks. These approaches have evolved in terms of sophistication and complexity. The first approach, which we can term traditional, is rule-based: the computer is taught a set of rules that roughly replicates the grammatical features of a given language. Initially, the rules developed for the purposes of this approach were of a static or first-order type of complexity, meaning that they accounted only for basic elements of a language, such as “a space marks a word” or “a final -s marks a plural”. In time, rules of a second-order or dynamic type were developed to address more complex conditionalities, such as recognising “Great” and “Britain” as one word. These rules provide the computer with a basic model for the language of the texts in its corpus.

From the early 1990s, developers began to introduce a second approach, based not on rules but on machine learning. This approach, also known as supervised learning, which is now also considered traditional, involved feeding large quantities of hand-annotated (“marked up”) texts to the computer. The computer would then transform the text into numerical representations based on hand-selected features (like word counts or grammatical tags) and learn to recognise patterns in the language through probability assessment used by those texts.

The third approach, which emerged in the early 2000s, also involves machine learning, but is based on the use of vectors. Vectorisation (often called “embedding”) is the term used by computational linguists for the process where texts or words are transformed into numbers or numerical strings and then analysed mathematically. In most cases, this approach again requires large amounts of pre-annotated training data for each given language.¹³ As with the previous two approaches, this method requires the computer to be

¹³ Approaches of this type that are known as “unsupervised learning” allow the computer to identify patterns without pre-labelled examples, so they do not require pre-annotated training data.

provided with a model of each language used by the texts it works with.

Fourthly, since 2017, a new machine-learning approach has been developed which uses what is called “transformer” architecture or technology.¹⁴ The transformer breakthrough resulted from the publication of an article that introduced a feature known as the “attention mechanism” (Vaswani *et al.* 2017). This innovation has revolutionised the field of vectorisation, and thus of NLP. The transformer process works firstly by reducing all the words in a text or corpus to unique numerical strings, one for each word (or token) and sometimes one for each paragraph or document, and then uses the complex algorithms which make up this mechanism to guess with a high degree of accuracy, if the words have been supplied to a computer in sufficiently large quantities, which word or token is most likely to follow or precede any given one.

This has produced the technologies now termed “generative AI”, which have led, among a number of outcomes, to “large language models” (LLMs). These systems are again based on vectorisation, but have greatly enhanced sensitivity to local context within a text or sentence. Transformers, much as with earlier forms of machine learning, require very large quantities of training data – but they do not require that data to have been pre-annotated, and they do not need to know what language is used by the texts they are analysing or to be given a language model in advance.

These four approaches are often combined – rule-based methods can be integrated with machine learning, for example – to enhance accuracy, efficiency, and other criteria. The rule-based approach can be demonstrated by a basic corpus- or text-analysis tool such as Antconc.¹⁵ Antconc can be run on a local drive, and it can use simple

¹⁴ Broadly speaking, machine learning refers to an algorithm or sequence of algorithms that can be trained and can then make predictions or judgments about unseen data. This includes everything from a basic conditional probability calculation to black-box transformers like GPT-4.

¹⁵ Antconc is freely available at <https://www.laurenceanthony.net/software/antconc/>. Using it does not require any technical knowledge. To use it for Tibetan, go to “Global Settings/Token Definition”, select “Use Following Definition”, paste

with non-programmers in mind, with a relatively straightforward user interface, it wraps together a number of useful but otherwise largely inaccessible NLP functions. Specifically, the iLCM's most useful features for us are interfaces for full-text keyword searches and metadata storage for individual documents in corpora. It also analyses word frequency and can carry out co-occurrence detection (especially useful for detecting slogans or jargon), topic modelling, and semantic volatility analysis (changes in word meanings over time). The iLCM includes tools for annotating corpora or coding sections of text and so is especially helpful for those using Qualitative Content Analysis on texts. It can also display results in visual or graphic forms, such as network diagrams or charts showing the distribution of a term or topic over time (sometimes called "dynamic topic modelling"), which facilitates diachronic interpretation of a text or corpus.

4 *Developing a Tibetan Language Model*

Like all such programmes, the iLCM is not a stand-alone application: it runs on a particular platform or software package which renders incoming data intelligible and that underlying platform has to be trained to work with Tibetan. In the case of the iLCM, this underlying platform is a popular NLP software library called spaCy.¹⁸ However, spaCy does not include Tibetan among its default-supported languages. So, before being able to use the iLCM to analyse Tibetan texts, we first had to develop a Tibetan language model for use with spaCy. To develop such a model requires preparing large quantities of pre-annotated training data. These texts have to have been cleaned up (extraneous code and so forth has to be removed) and every word in the texts, and every sentence or phrase (or "utterance") in that text, has to have been tokenised – that is, they should be marked as distinct from a previous or subsequent word or utterance. This is the process known as "segmentation" or, more commonly, as "tokenisation". In

¹⁸ See <https://spacy.io/>. SpaCy is a general-purpose software package for end-to-end NLP applications, including word segmentation, topic modelling and NER.

some cases, and in particular for training a language model for a platform such as spaCy, training data also has to be marked up with tags that identify the part of speech of each word (“POS tagging”).¹⁹

Considerable quantities of such training data already exist for Tibetan, prepared and annotated by previous projects.²⁰ Most of that data, however, consists of texts in classical Tibetan, and a model trained with these will produce numerous errors or omissions when used to read modern Tibetan texts. For our language model, we therefore needed to produce large quantities of annotated data in modern Tibetan. Without these, spaCy would struggle to recognise what is a word or meaningful particle in Tibetan and our analysis tool, the iLCM, would be more or less unable to identify frequencies, topics, entities and other features within the corpus.

In the interim, while developing our annotated training data for modern Tibetan, we carried out a test to see if a makeshift Tibetan-language model using spaCy would enable the iLCM to work with Tibetan. We called this test model “Tibetan for spaCy 1.1” (see Kyoguku *et al.* 2025 in this issue, section 5). We used the training data produced by Dakpa *et al.* (2021a, b) – it consists of only 13 megabytes

¹⁹ Some NLP procedures work better if they have also been trained to recognise the “lemma” or root form of each word and the syntactic structure of the language, though for our project we have so far not found it necessary to apply lemmatisation or to add syntactical information to our texts.

²⁰ Classical Tibetan corpora include, for example, the Asian Classics corpus (available at <http://resources.christian-steinert.de/download/acip-release6-wylie.zip>); the BDRC Corpus, available at <https://zenodo.org/records/821218#.Xu5IOOdYxld> (Wallman *et al.* 2017), annotated as the “ACTib” corpus (<https://zenodo.org/records/821218#.Xu5IOOdYxld>; see Meelen and Roux 2020); the “Tibetan in Digital Communication” corpus (<https://zenodo.org/records/574878>; see Hill & Garrett 2017); the “Lexicography in Motion” corpus (<https://zenodo.org/records/4727108>; see Faggionato *et al.* 2021); and the OpenPecha corpus (<https://github.com/OpenPecha/openpecha-catalog>). Corpora of modern Tibetan texts include the “Nanhai corpus” produced by Esukhia (<https://github.com/Esukhia/Corpora/tree/master/Nanhai>), probably 2017; the Fudan NLP Tibetan Classification corpus, produced by the Natural Language Processing Laboratory of Fudan University, probably 2017 (<https://github.com/FudanNLP/Tibetan-Classification>); and the “Modern Tibetan Corpus” produced as part of “Lexicography in Motion” (see Dakpa *et al.* 2021a, 2021b).

of annotated modern Tibetan texts, a very small amount by NLP standards – which was already available in CoNLL-U, a format which spaCy's default models can natively interpret and learn from. We then added a space after each *tsheg* in the data, fed the training data into spaCy, and instructed it to treat the input data as if it were English. SpaCy thus used its English language model to read the data, interpreted the space after each Tibetan token as indicating the end of an English word, and so added these (actually Tibetan) “words” (actually syllables) to its inbuilt English lexicon. This method produced numerous errors of tokenisation and so forth, but it worked: it enabled the iLCM to read and process the Tibetan documents in our test corpus, as well as to show topics, relational distributions of words and phrases, and even political slogans (Engels *et al.* 2023).

This interim model thus worked as a temporary measure to test the feasibility of using the iLCM, but, since it used a “phoney” tokenisation method that produced frequent errors, it could not be used for any serious textual analysis. To achieve a durable language model for spaCy and hence for the iLCM – or for any pre-transformer approach to NLP and textual analysis involving modern Tibetan texts – we needed to produce sufficient quantities of annotated training data in modern Tibetan to feed to the underlying platform used by our corpus-analysis programme. Only then would that programme be able to search a corpus for particular Tibetan terms, concepts, topics, names or semantic features of interest to the user.

5 *Tokenising: Rule-based vs. Transformer-based approaches*

To produce training data consisting of accurately annotated modern Tibetan texts, we needed to find tools that could efficiently carry out tokenisation and POS tagging on such texts. The result of our search for such tools, which enabled us in time to develop a fully-functioning Tibetan language model for spaCy, is described in the paper in this issue by **Yuki Kyogoku, Franz Xaver Erhard, James Engels** and others, “Leveraging Large Language Models in Low-resourced

Language NLP: A spaCy Implementation for Modern Tibetan” (Kyogoku *et al.* 2025).

Tokenisation can be carried out by a basic NLP tool like Antconc just by differentiating lexical units by spaces, punctuation, or other signs; it can then count the tokens, organise them, compute their frequency, and so forth. But tools or procedures that carry out more advanced forms of NLP analysis, like the iLCM and spaCy, need to be able to recognise features of the language they are processing with more precision. In particular, they need to be able to recognise individual words, including polysyllabic ones, as an initial step before any further analysis can be done. In the early stages of NLP, such tools were trained to recognise separate words and utterances by using a rule-based approach to tokenisation. These early tools were designed for use with European languages, mainly English, and so their rules were over-tuned to the specific grammatical requirements of European languages, typically using white spaces to split raw text, along with a basic punctuation set plus apostrophes.

In Tibetan, however, where each syllable is separated by a *tsheg*, only monosyllabic words are marked as distinct. Polysyllabic words are not marked and thus would be invisible to our corpus-analysis tool if our Tibetan texts were tokenised simply by assuming every syllable before a *tsheg* is a word. This would lead, for example, to a collocation such as བོད་རང་སྐྱོང་ལྗོངས་ (Tibet Autonomous Region) being defined as three words, rather than as a single one, albeit with three *tshegs*, four syllables and three recognisable words in the string. Similar difficulties arise with the complex forms of noun and verb morphologies in Tibetan. Rule-based methods of tokenisation for Tibetan are based on Tibetan syntax or on checking words in dictionaries (a process known as “dictionary look up”), but this tends to bias longer words even if the component units of that word should at times be parsed as shorter words positioned consecutively. Tokenisation for Tibetan hence requires a more sophisticated approach, one which can follow second-order rules that address the context of words or that can do probability calculations using a machine-learning approach.

At least one such tokeniser exists: Botok. It is a product of collaboration between the Buddhist Digital Resource Center (BDRC)

and other organisations such as OpenPecha and Esukhia, a nonprofit that specialises in developing digital resources related to Tibetan languages and their textual traditions. Unlike previous tokenisers produced through this collaboration that were purely rule-based,²¹ Botok's tokenisation procedure involves first splitting on the *tsheg*, so that every syllable is isolated in sequence. Combinations of syllables are then evaluated using an internal dictionary search, and decisions about what classes of words to search are made based on statistical evaluations of preceding contexts. In addition, Botok is context-dynamic: it has the capability to edit the rules internally when its input data exhibits a repeating but previously unlearned pattern. Importantly, Botok's internal dictionary can be modified or updated. Furthermore, it has the advantage that it is consistent: once its settings have been adjusted to the user's needs, it will tokenise any set of morphemes in the same way.

A number of other approaches have been developed for tokenising Tibetan. These include a machine-learning approach known as a "Memory-based tagger" used by Meelen & Hill (2017) to produce an archive of high-quality annotated training data, and the combined rule-based, memory-based, and deep-learning method used by Meelen, Roux and Hill (2021) to annotate their "ACTib" corpora (see also Faggionato, Hill and Meelen 2022). However, these projects used classical Tibetan texts for their training data and so their tokenisation methods are not attuned to modern vocabulary, particularly compounds such as བོད་རང་སྐྱོང་ལྗོངས། (Tibetan Autonomous Region), ལྷན་ཁྲིམས་ལྷན་ཁུངས། (committee) or གུང་ཁག་ཏང་ (CCP). We therefore turned to the newly emerging transformer-based technologies to explore their capabilities in tokenisation for Tibetan. Using probability-based calculations, these technologies can infer which syllables should be treated as a word or token. To do this, they require very large quantities of training data in the given language, but that training data does not need to have been pre-annotated: the most recent transformer-based models can infer all

²¹ Botok is built from a tokeniser called PyBö, developed by a collaboration between Esukhia and the BDRC and completed in 2021, although Botok has a larger training dataset than PyBö. See <https://github.com/OpenPecha/Botok>.

necessary information about the language in their training data based solely on pattern recognition in that data. As a result, if they have been given data of sufficient quality, these models are now developing the ability to carry out essential tasks, including tokenisation, on texts where they have not been given information in advance about the language of those texts.²²

A prominent language model of this kind, developed by Google in 2018, is called BERT (“Bidirectional Encoder Representations from Transformers”). BERT uses transformer architecture to predict what text might come before and after other text. This form of machine learning, however, has to be trained on very large amounts of raw text, which presents a problem for a relatively low-resource language like Tibetan. Nevertheless, a number of applications of BERT have been developed for use with Tibetan by scholars in Tibet or China, and some of these have been made publicly available, including a BERT model for Tibetan called TiBERT (Sun *et al.* 2022) and one called Tibetan BERT (Zhang *et al.* 2022).²³ However, BERT was designed to help computers analyse the meaning of a word based on the words surrounding it, an approach which is applicable for tasks involving classification, such as

²² However, because transformer models often rely on shared linguistic structures from related languages and then transfer their learning to the unseen language, they perform less well on unseen languages that do not share characteristics with those on which they have previously been trained.

²³ The two models mainly differ in size and computational complexity: like BERT Base, TiBERT contains 12 transformer blocks, 768 hidden dimensions, 12 self-attention heads, and 110 million parameters (available at <http://tibert.cmli-nlp.com/>). Tibetan BERT is a scaled-down version of BERT with a decreased computational load, focusing on minimal decrease in performance for use in situations with heavier resource constraints. Tibetan BERT has 4 transformer blocks, 256 hidden dimensions, 4 attention heads, and does not report its parameter figure. TiBERT is available through the creators’ own distribution (https://huggingface.co/UTibetNLP/tibetan_bert), but does not include the training data or a detailed description of it. A Tibetan computer developer in Amdo, Sangjee Dondrub, has built a tokeniser for Tibetan using an improved version of BERT known as RoBERTa, available as of May 2022 at <https://huggingface.co/sangjeedondrub/tibetan-roberta-causal-base> (accessed January 29, 2025).

document classification,²⁴ sentiment analysis, question answering, NER, and text summarisation. It is not designed for non-classificatory tasks like text-generation or for those requiring logical reasoning or domain-specific knowledge. Its tokenisation method is based not on linguistic rules but on frequency, and it splits “words” or syllables into sub-tokens or “sub-words”, often consisting of a single character or character stack, which are not always recognisable to a human reader and so cannot be checked. A similar tokeniser is used by Tibetan LLaMA (Lv *et al.* 2024), a transformer-based LLM based not on BERT but on LLaMA,²⁵ the open-source LLM created by Meta (the owner of Facebook) as a competitor to OpenAI’s GPT-4 or Microsoft’s Bing Chat.²⁶ These tokenisation methods would not be suitable for a tool or procedure that needs to recognise actual words and to learn their linguistic functions.

Since late 2023, however, mainstream LLMs trained primarily on English texts have begun to demonstrate Tibetan-language capabilities. Initially, their ability was somewhat limited, probably because of the limited amount of raw training data in Tibetan fed to them by that stage. For example, in April 2024 we asked GPT-4 to

²⁴ When tested on a validation sample of news articles in Tibetan, Sun *et al.* (2022) reported that their TiBERT model correctly classified 86% of the unseen texts. Tibetan BERT also achieved an 86% accuracy rate on a similar test, classifying unseen texts in a partition of its news article training corpus, suggesting that Tibetan BERT has no inherent disadvantage over TiBERT despite its much lower computational demand.

²⁵ The newest version of LLaMA is at <https://www.llama.com> (accessed January 29, 2025).

²⁶ Tibetan Lama (T-LLaMA) was trained on 11 gigabytes of data in Tibetan and has so far been shown to be effective at text classification, basic news text generation and text summarisation. The developers note that it needs further development if it is to be ready for “tasks such as dialogue, reasoning, and translation” (Lv *et al.* 2024: 72). The T-LLaMA model is available at <https://huggingface.co/Pagewood/T-LLaMA>. It used a tokeniser called SentencePiece, which, like BERT, basically treats each character or character stack as a token and calculates probabilities for which token might follow it – again, not a suitable method for us to use to tokenise our training data for the spaCy language model.

tokenise a sample Tibetan paragraph taken from an online news article:²⁷

གྲུང་གོ་ས་ཡོམ་ལས་ཁུངས་ཀྱི་བཟུང་བའི་སྤྱི་ཚད་འཇལ་གཏན་འཁེལ་བྱས་པར་གཞིགས་ན། ཟླ་དང་ལོའི་ཚེས་23ཉིན་གྱི་ཚུ་ཚོད་2དང་སྐར་མཉེན་ཤིན་ཅང་ཨ་ཁི་སུ་ལུ་ས་ཁུལ་ལྷན་ཁྲུང་གི་རྒྱུ་རྐྱེན་དུ་རིམ་པ་7.1ཅན་གྱི་ས་ཡོམ་བྱུང་བ་དང་། xxཡོམ་ཁུངས་ཀྱི་གཏིང་ཚད་སྤྱི་ལའི་22ཡིན་ལ། ཡོམ་ལྷན་གྱི་བྱང་གི་འབྲེན་ཐིག་ཏུ་41.26དང་ཤར་གྱི་གཞུང་ཐིག་ཏུ་78.63ཡིན་པ་དེད།

GPT-4, although it was already more advanced in Tibetan than its predecessor GPT-3.5, responded by dividing the sentence into five utterances or phrases each ending with a *shad*, seeing each utterance as a token. It thus could not, at this stage, distinguish between a word and an utterance. By September 2024, it could identify individual words, but in most cases missed polysyllabic words (for example, it tokenised ས་ཡོམ་ལས་ཁུངས་ as four separate words). By that time, however, Google’s Gemini Ultra (Gemini Pro 1.5) model was already able to perform effectively when asked to tokenise the first two phrases of the above sentence, producing good results:

- གྲུང་གོ་ / ས་ཡོམ་ལས་ཁུངས་ / བཟུང་བའི་ / སྤྱི་ཚད་འཇལ་ / གཏན་འཁེལ་ / བྱས་པར་ / གཞིགས་ན།
 གྲུང་གོ་ (China) / ས་ཡོམ་ (earthquake) / ལས་ཁུངས་ (department/agency) / བཟུང་བའི་ (strictly) / སྤྱི་ཚད་འཇལ་ (in reality/actually) / ཚད་འཇལ་ (measurement) / གཏན་འཁེལ་ (fix/establish) / བྱས་པར་ (having done) / གཞིགས་ན། (when looking at/considering)
- ཟླ་དང་ལོ་ / འི་ / ཚེས་23 / ཉིན་གྱི་ / ཚུ་ཚོད་2 / དང་ / སྐར་མཉེན་ / ཤིན་ཅང་།
 ཟླ་དང་ལོ་ (first month) / འི་ (of: genitive particle) / ཚེས་23 (date: 23rd) / ཉིན་གྱི་ (of the day) / ཚུ་ཚོད་2 (hour: 2) / དང་ (and) / སྐར་མཉེན་ (minute: 9) / ཤིན་ཅང་ (on/upon)

On a separate test in September 2024, Gemini had improved enough to give results similar to those of Claude and Gemini Ultra. In that test, we compared their results to those given by the linguist and Tibetologist Camille Simon for the 40-syllable sentence བརྗོད་དོན་གཙོ་བོའི་སློབ་གསོ་ལག་གཉིས་པ་སློབ་འཇུག་ཞིབ་ཚགས་བྱས་ནས་ལག་བསྟར་བྱ་རྒྱུ་ནི་མིག་སྲེའི་བོད་སྲོལ་ཡོད་པའི་གྱི་ས་གནས་ལག་དང་ཚན་པ་ལག་གི་གཤམ་ཆེའི་ལས་དོན་ཞིག་ཡིན། (see Kyogoku *et al.* 2025 in this issue, Appendix F). ChatGPT4 and

²⁷ http://tb.tibet.cn/tb/index/news/202401/t20240123_7562609.html.

ChatGPT4o both tokenised this almost entirely on the *tsheg*, ignoring all two-syllable words except for two. Their error rate was 61.5% and 60% respectively, so we saw little improvement there. But Claude and Gemini had no problem in recognising most polysyllabic words and differed from the expert version only in not dividing particles, such as the genitive suffix, from a root word, which reflects a difference in tokenising method rather than an error. These two LLMs are thus already capable of tokenising a non-tokenised Tibetan passage based on their self-trained, probability-based understanding of the structure of the Tibetan language.

Nevertheless, although very promising overall, LLMs tend to be inconsistent in their decisions, so the same prompt may return slightly different results at different times. Gemini, when asked again to tokenise the sentence above about earthquakes three months later, divided it into 19 tokens rather than the 17 it had produced in its first attempt.²⁸ As they are fed more training data or their parameters are refined, the abilities of LLMs with a particular task or language – especially a low-resourced one – can even diminish (Pramanik 2025). In addition, with LLMs, there is no code or programming that an end-user can access or adjust apart from modification of their parameters.

Consequently, as Kyogoku, Erhard and their colleagues describe in their paper (section 3.1), the project chose to use Botok to tokenise the training data for the new language model. Botok has sufficient accuracy, it is consistent, it can easily be added to a pipeline or workflow so as to integrate with other tools such as spaCy, and Botok's open-source code allows the end-user to adjust their local versions should the need arise. Although Botok was designed to be used on classical Tibetan texts, the project team was thus able to adapt it for use with modern Tibetan texts by adding extensive wordlists of modern Tibetan terms to its internal dictionary (see Kyogoku *et al.* 2025, section 4.2). As a result, our adapted form of Botok, which we have termed

²⁸ In V1, Gemini rendered གཏན་འཁེལ་ and བྱམ་པར་ as two tokens, but treated them as one (གཏན་འཁེལ་བྱམ་པར་) in V2. It rendered དང་ལོ་ and འི་ as ལྷ་དང་ལོ་ and འི་ in V1 and as ལྷ་ and དང་ལོ་འི་ in V2. Numerals were not treated as separate tokens in V2, but grouped with their reference (i.e., ལྷ་ཚོ་2 / དང་ / ལྷ་མ་9) in V2.

“modern Botok”,²⁹ performs well with modern Tibetan, scoring 13/14 on a simple test (see Kyogoku *et al.* 2025, Appendix D).

6 Tagging in Tibetan

To develop a Tibetan language model for spaCy, we also needed to produce training data that is POS tagged, i.e., where each word in a text is identified according to its syntactic function as a noun, adjective, verb, or so forth. Annotation of this kind is also important for our project because it helps in the development of NER (named-entity recognition), a capability which we will need in order for our search tools to recognise automatically which words are names of places, people, offices and other entities.³⁰ In Section 3.2 of their paper, Kyogoku, Erhard and their colleagues describe their assessment of existing tools to see which might be suitable for producing POS tagged training data in modern Tibetan. They found that the rule-based tools were not able to make decisions based on context, and so could not resolve cases where a word might have more than one syntactic function. Neither could they adjust to unfamiliar genres or registers of writing. Shallow machine-learning tools such as ACTib produced much better results but, as noted above, had been trained on classical Tibetan and would again need to be fed large amounts of pre-annotated training data in modern Tibetan in order to work with modern texts.

As with tokenisation, however, the team found a remarkable improvement in transformer-based approaches to POS tagging. By

²⁹ See <https://github.com/Divergent-Discourses/modern-botok> and Erhard, Kyogoku *et al.* 2024. To compile this extended wordlist we combined relevant dictionaries from Christian Steinert’s collection (<https://github.com/christiansteinert/tibetan-dictionary/tree/master/input/dictionaries/public>) with Botok’s built-in Grand Monlam Dictionary and our own list of personal and place names derived from the newspaper corpus.

³⁰ Unlike those European languages which have capital letters and definite articles to mark some types of named entities, Tibetan offers few transparent tricks to identify them. The simplest heuristic, given the tendencies of Tibetan grammar, would be to search for proper nouns and their immediate neighbours.

September 2024, GPT-4 could POS-tag a Tibetan sentence with around 80-90% accuracy, and a comparison of POS tagging in Tibetan by GPT-4, Claude 3.5 Sonnet, and Gemini shows a similar error rate of 10-20% (see Kyogoku *et al.* 2025, Appendix F), which is low given the likely rate of progress.

The tagging by LLMs again showed inconsistencies, but, unlike with tokenisation, the resulting problems were not critical and in most cases could be corrected: by using rule-based coding at the post-processing stage, we were able to correct many of the recurrent errors in Gemini's tags. By August 2024, the Diverge team had therefore abandoned our initial plan to use a shallow machine-learning tool to produce POS tagged training data for our Tibetan language model and instead used Gemini Pro 1.5 to POS tag our training data. By late August, we had generated sufficient quantities of tokenised and POS tagged training data in modern Tibetan to develop from scratch the first Tibetan-language model for spaCy (Kyogoku *et al.* 2025, Sections 4 and 5).³¹

7 *Transformer-based Tools: Topic Modelling and Semantic Searching*

Our success in developing a Tibetan-language model for spaCy meant that, two years after we began the project, we were finally able to test our text-mining platform, the iLCM, for its capacity to read and analyse Tibetan texts. We have not yet completed the testing process, but initial results indicate that iLCM can process Tibetan without significant errors, although some issues remain concerning its handling of unfamiliar Unicode characters. We anticipate that other such problems will arise, as is frequently the case when using software developed for use with high-resource languages to work with low-resource languages. Overall, however, it is now likely that in time we will be able to use the full capabilities of the iLCM as well as related tools and platforms. This will then provide us with a sophisticated set

³¹ For the Modern Tibetan spaCy code and data set, see Kyogoku *et al.* 2024b, c.

of tools to identify “divergent discourses” – changing topics or narrative foci – in our corpus.

However, we had to allow from the outset for the possibility that we would not be able to assemble sufficient annotated training data from modern Tibetan texts, or that, even if could, the iLCM and similar tools might not work with our Tibetan model. We therefore pursued a parallel strategy in our project which avoided dependency on tools requiring pre-annotated training data. This strategy, initiated and led by Ronald Schwartz, involved the development of tools that employ vector embeddings derived from transformer-based LLMs to analyse Tibetan texts. As we have seen with LLMs, these tools do not need tokenising or POS-tagging to have been carried out on their training data.

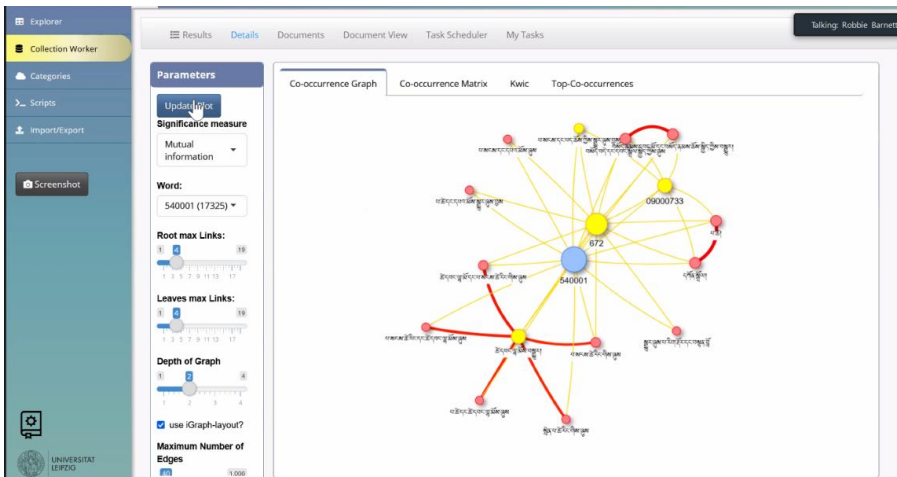


Figure 3 An iLCM visualisation showing relations between certain terms and names in the corpus

As **Ronald Schwartz** and **Robert Barnett** explain in their paper for this issue, “Religious Policy in the TAR, 2014–24: Topic Modelling a Tibetan-Language Corpus with BERTopic” (Schwartz & Barnett 2025), embeddings are numerical strings into which a word, phrase, sentence, or paragraph has been encoded. Those encodings are calculated with reference to context, so that the number used for a particular word or text is adjusted to account for the numbers of words or passages that adjoin it or are near it. This ability to incorporate

contextual information goes further than just word co-occurrences. Thus the encoding for the word “bear” in the sense of a large furry animal would differ from the same word used as a verb in the sense of carrying something. In addition, the word “bear” followed by the word “arms” would be encoded differently from the word “bear” followed by “fruit”. As a result, encodings of this kind do not just represent the lexical form of a word or text; they in effect convey the semantic content or meanings of those texts.

In the transformer-based approach, these encoded numbers are treated mathematically as if they are situated within a conceptual space that consists of multiple dimensions. As a result, numbers that are close together algebraically in the vector space represent words or texts that are close together in meaning. This has a consequence of great importance for NLP: it means that these transformer-based forms of vectorisation can carry out tasks that identify meanings of different kinds within a text, even if not apparent to a reader or a conventional search tool. As noted above, these transformer-based tools can perform these tasks without needing task-specific training data.

The project therefore set out to develop two such tools for use with modern Tibetan. One is topic modelling – the automatic, corpus-scale identification of textual foci. Topic modelling identifies topics or themes in a text or corpus based on the above-mentioned algebraic forms of detection, whether or not the explicit name of a given topic is mentioned. Topic modelling in this case is not guided – the tool determines for itself what are the different topics in a text. The second tool is semantic search, which is a guided form of the same technology: a user enters a query and the semantic search tool finds any texts in the corpus which resemble that query – even if those texts do not include any of the words found in the query. Thus, if one asked a semantic search tool to search for similar passages to the phrase “people who study Tibetan texts”, it could return not only all the passages in that corpus about Tibetanists, but also those about Buddhas, animals or deities that read novels in Sanskrit, Chinese or perhaps Uyghur. Both these tools are of particular value to textual analysts because by their nature they are likely to find passages or texts that are not anticipated by the searcher yet are relevant to the query.

At least in principle, they thus can contribute to some extent to decreasing the risk of confirmation bias by an analyst.

The paper by Schwartz and Barnett explains the principles of this approach to topic modelling and to semantic searching and demonstrates an application of the topic modelling tool in the Tibetan context. Using a set of some 4,000 “born-digital” articles collected by Schwartz from online Tibetan-language newspapers over the last ten years, the paper shows how Schwartz first cleaned up the texts, subdivided them into paragraphs or “chunks”, and compiled them into a corpus. He then created a “subcorpus” consisting only of articles including the Tibetan terms for “religion” or “Tibetan Buddhism”. The topic modelling tool was then run on the subcorpus to identify variations in discussions of religion in the subcorpus. It divided the paragraphs into groups, each with a different narrative emphasis. Schwartz then used an LLM called Claude Sonnet 3.5, which is already able to read and process Tibetan texts, to analyse the top ten paragraphs from each group (meaning those identified by the tool as most “representative” of the narrative focus of that group) and to add a label and a set of keywords that typified the topic of that group. With this information, Schwartz and Barnett were then able to identify a number of topics (about half of the total number of topics) indicating drives by the Chinese authorities to regulate the behaviour of monks and nuns in Tibetan monasteries. They also identified a third of the topics as representing ideological drives to reshape thinking about acceptable forms of religious belief among the Tibetan public, and a set of topics or themes that represented ongoing background arguments or opinions circulated by the Chinese government concerning religion in Tibet. Since the topic model could display its findings in terms of occurrences of each topic over time (a function known as “dynamic topic modelling”), the analysts could also map the duration of each drive and identify their peaks of activity, in terms of occurrences of the topics in the official Tibetan media.

In their paper for this issue, “Developing a Semantic Search Engine for Modern Tibetan” (Engels & Barnett 2025), **James Engels** and **Robert Barnett** describe Engels’ construction of a semantic search engine for Tibetan texts that can be used by members of the public

without any knowledge of coding or other computational techniques. They first explain the differences between semantic searches and conventional keyword searching. In the latter, a user finds all words in a corpus that are identical to the query term or (in the case of fuzzy searches or lemmatized systems) resemble it lexically; in the former, search results produce texts similar in meaning, but not necessarily in form, to the query. The paper goes on to explain the concepts behind the historical development of semantic searching, describing the use of transformer-based approaches to vectorisation and summarising in lay terms the broad theories underlying such a system and the history of these major advances in NLP. In particular, they credit Schwartz with realizing that one company specialising in AI, Cohere, has already developed the technology for creating vector embeddings for less-resourced languages, including Tibetan.³² In effect, Cohere's embedding model can "understand" Tibetan at an advanced level of complexity. Cohere allows any user to submit a text or an entire corpus which it then converts into embeddings and returns to the user, for a small fee. It is this that makes semantic searching possible in Tibetan. In addition, Cohere's model is multilingual: since it deals with meanings numerically, translation is an emergent feature of its capabilities. With Cohere's embeddings, one can thus submit a query in Chinese or Vertical Mongolian and get results in Tibetan or whatever is the language of one's corpus, if Cohere has been trained on that language or one that is linguistically similar.

The paper also provides more detail about the distinction between topic modelling and semantic searching. Engels characterises the former as "outside-in" analysis of a corpus, such as looking at what topics tend to occur over time within a corpus or identifying what is represented in a particular article. He describes semantic searching as an "inside-out" method, where a researcher knows what kinds of things they want or expect to see but do not know where to find them or what words might be used to signify them.

³² Cohere's Multilingual Model 2.0 is at <https://cohere.com/> (accessed January 15, 2025). Both Schwartz and Engels advise against using Cohere's Multilingual Model 3.0 for Tibetan texts.

In the second half of his paper, Engels describes the techniques he used to create a semantic search tool for Tibetan that would not require programming knowledge by its users but would be fully accessible to the public. By adapting code initially developed by Schwartz for semantic searching first in Chinese and then in Tibetan, Engels produced a publicly accessible version of the tool for use with Tibetan texts that is hosted on a university server and has a simple user interface.³³ It lists the date, source, title of article for each search result and allows the user to rank results either by relevance (semantic match) or by date, or by one of the other metadata fields. It includes a toggle allowing non-readers of Tibetan to see a translation of the query or the results in English. For this he linked the system to Bing Translate, which currently appears to be the most reliable provider of online translations from Tibetan to English.

Currently, the projects' semantic search engine is linked to a test corpus of recent Tibetan newspaper articles harvested from online sites, but in future it will be linked to the Divergent Discourses corpus of historical newspapers. It is, we believe, the first online system for searching historical Tibetan newspapers using state-of-the-art NLP tools. It can easily be linked to any corpus of Tibetan (or other) texts by anyone with basic programming knowledge; search systems of this kind, Engels notes, are not difficult to build and do not require expertise in software design.

8 *Identifying a discourse: 'Friendship' in the Tibet Mirror*

The final paper in this issue is **Natalia Mikhailova's** study, "*The Tibet Mirror, 'Friends' of Tibet, and the Internationalisation of the Tibet Question*" (Mikhailova 2025). Her article offers an example of the project's aim – the identification and discussion of discourses in Tibetan-language newspapers from the 1950s to the early 1960s. Her contribution is distinct from the other papers in this issue, firstly in

³³ The public link to the project's semantic search tool is <https://tibetcorpus.uni-leipzig.de/search/> (accessed January 30, 2025).

that it is an example of applied research rather than a discussion of methodology, and secondly, in that it is an example of research in the pre-digital humanities era: she carried out the study without the advantage of text-mining tools, before our corpus has become available, and before our tools have been completed. Using nothing more than photographs or scans of newspaper pages obtained from libraries, and studying the often imperfect images without the aid of any search or other tools, Mikhailova shows the enormous labour that goes into research of this kind, identifying themes in the articles, supplying lengthy translations, and connecting them to their historical context and, where available, to previous studies.

Her focus is on the most prominent of all Tibetan-language newspapers outside Tibet, the *Tibet Mirror* (the *Yul phyogs so so'i gsar 'gyur me long*). The *Mirror* was published from the north-eastern Indian city of Kalinpong from 1925 to 1963, and Mikhailova's study looks at its output in the years following the annexation and incorporation of Tibet by the People's Republic of China (PRC) in the early 1950s. She includes the years after the mass exodus of Tibetans and their leader, the 14th Dalai Lama, from Tibet to India following the failed uprising of March 1959. Her main finding is that, among other discourses, the *Tibet Mirror* aimed to construct for its readers a narrative of international support for Tibet in its struggle against the PRC. The paper, she shows, focused at times on presenting the governments of Great Britain, India, and the United States as "friends" or supporters of the Tibetans, particularly those who had fled into Exile. The paper also represented the Chinese nationalist government in Taiwan as supportive of the exiles, and even as having said that, if it were to regain power in China, it would support Tibet's independence. She also shows how the paper proposed arguments, based on Nepal's application to the United Nations for membership of that body in 1949, that could be used to argue for Tibet's independence. As she explains, these arguments came to dominate exile discussions of Tibetan independence for many decades afterwards.

This study of a single narrative thread in one Tibetan newspaper offers an early example of the kind of findings that are of interest to the project, and which other team members might explore in future

studies. Those studies, however, will have the advantage of access to the project’s computational tools and to its corpus, which will shortly be ready for use. Over the next year, as the project moves from the development of text-mining tools to the application of those tools to the study of discourses in Tibetan newspapers, we will see how the use of digital methodologies confirms, extends or varies Mikhailova’s findings.

9 Conclusion

At the current rate of progress, a transformer-based Tibetan LLM will soon outmode even the best tools based on earlier approaches. At the time of writing, Monlam AI’s promised first-generation Tibetan LLM (including various functions like a Dalai Lama chatbot), which is being developed jointly with Esukhia, is still in early stages of development, but it is likely to expand its abilities rapidly as its range of training data is expanded.³⁴ At Berkeley, Sebastian Nehrdich and colleagues have produced the initial elements of a Tibetan transformer model.³⁵ If Tibetan LLM technology improves enough to gain “intuitive” understandings of modern Tibetan, it is possible that dedicated POS taggers will no longer be needed. Meanwhile, a number of English-language-based LLMs from major corporate entities have advanced astonishingly quickly in their capacity to understand and analyse Tibetan text, albeit with different strengths. At the end of 2023, no public-facing LLM could produce intelligible interpretations of Tibetan data. By the autumn of the following year, some major LLMs – such as OpenAI’s GPT-4, Google’s Gemini, and Anthropic’s Claude 3.5 Sonnet – had developed translation capacity and other

³⁴ See <https://monlam.ai/about> (accessed December 18, 2024), and <https://github.com/MonlamAI> (accessed December 18, 2024).

³⁵ See <https://huggingface.co/buddhist-nlp/byt5-mitra-bo>. Other initiatives include “TibetaMind” (“an advanced language model based on the Llama 3-8B-Instruct architecture, further fine-tuned using extensive Tibetan language corpora”, available at <https://huggingface.co/DaydreamerF/TibetaMind>) and the T-LLaMA model (Lv *et al.* 2024). Both sites accessed January 30, 2025.

competencies in modern Tibetan. For example, we tested them with the following sentence:

གྲུང་གོང་དོ་ཤོ་ལ་བྱེད་མཁན་རྒྱལ་ཕྱོགས་ཕྱོགས་གཏོགས་ཀྱིས་དྲ་ལའི་རུ་ཚོགས་ལ་བརྟེན་ནས་གྲུང་གོ་ལ་ཕྱལ་འཛོལ་ཞིག་ཏུ་གཏོང་བའི་ལྷོག་གཡོ་གཤོས་རྒྱ་སྐབར་ཕྱི་གསལ་དུ་འགྱུར་བ་བྱུང་མེད་ལ། དྲ་ལའི་རུ་ཚོགས་ཀྱིས་“བོད་རང་བཙན”ཡོང་འབྱེད་པའི་ལ་ཕྱལ་འཛོལ་ལུགས་ཀྱི་དོ་བོ་ཡང་སྐབར་ཕྱི་གསལ་དུ་འགྱུར་བ་བྱུང་མེད།

GPT-4's translation of this sentence, as of September 2024, was “Western-backed anti-China forces have not yet resorted to external interference or attempted to secretly incite a sudden breakup of China through reliance on groups like the Dalai clique.”³⁶ This reversed the actual meaning of the sentence and missed a number of details.³⁷ Claude, however, translated it correctly as “The Western forces opposing China have not changed their covert plot to use the Dalai Lama's group to split and disintegrate China, whether in the past, present, or future. Similarly, the separatist nature of the Dalai Lama's group's hope for “Tibetan independence” has also not changed over time.”³⁸ Gemini's translation was similar.³⁹ The translations by both

³⁶ <https://chatgpt.com/c/66dae400-2168-8002-975c-e082e71bd3c7>. No longer accessible.

³⁷ GPT-3.5 was almost entirely incorrect except for a few words, underlined here: “Those who engage in the struggle for the establishment of Tibet's independence, from the perspective of the Tibetan people, are not caught in the virtue of a particular kind of nobility. With the justification of the Tibetan struggle becoming apparent, there is no change in the recognition of the essence and nature of the pursuit of 'Bod rang btsan' (Tibetan independence) even three times in the past” (<https://chat.chatbotapp.ai/chats/-O6sEh0wWsm0uJUb0fXP?model=gpt-3.5>; no longer accessible).

³⁸ See <https://claude.ai/chat/6faa50e1-f10b-44f7-98c0-ce854b6037a1> (accessed January 30, 2025).

³⁹ “The Western powers, who oppose China, have relied on the Dalai Lama group to split China. However, their covert attempts to achieve this have not changed from the beginning to the end. Similarly, the Dalai Lama group's separatist ideology, which seeks “Tibetan independence,” has not changed from the beginning to the end” (<https://gemini.google.com/app/eaaf83e2455372f9>; accessed January 30, 2025).

Claude and Gemini were far more precise and detailed than the online translations of the same sentence by Bing or Google.⁴⁰

Claude Sonnet has demonstrated even more advanced Tibetan-language capability. It can produce high-quality translations of modern texts, perform generative tasks such as summarising, finding keywords, and generating labels, and it can do these directly in Tibetan without working through an English translation. GPT-4o, unlike previous generations of GPT, also is capable of summarising and extracting themes and topics from modern Tibetan texts in a remarkably human-like way. Future generations of LLMs are thus likely to have the capacity to understand and analyse Tibetan texts to a degree not markedly different from any other language for which the machine has been supplied with sufficient training data.

For those working with Tibetan texts, this means that in the near future, transformers will likely be the tool of choice for translation, summarisation, classification, topic modelling and semantic searches. They will also be very strong options for tokenising and POS tagging, if those tasks are still needed. However, they remain “black-box” technologies: the end-user cannot adjust their code or even be certain as to how they reach their conclusions, and their results and capabilities are likely to vary over time. Given these uncertainties, the Tibetan studies community is probably going to continue for some time to need to maintain and develop tools based on more traditional approaches, whether based on rules or machine learning, in order to have robust and replicable techniques available for the analysis of Tibetan texts.

Overall, however, we have found that for our project some objectives are most easily achieved by using more traditional

⁴⁰ Google Translate, which had only recently made Tibetan available, struggled to translate this sentence and rendered it as “Western opponents of China rely on the Dalai Lama's party to see China as a divided country There is no change in the smuggling [*sic*]. The Dalai Lama's group's ‘Tibetan independence’ is also a form of separatism. It hasn't changed.” Bing Translate was more accurate than Google Translate but also missed some details, rendering this as “The conspiracy of the Western forces opposing China to use the Dalai clique to split China has never changed. The separatist nature of ‘Tibetan independence’ has never changed.”

approaches, not least because, firstly, these approaches often come with pre-developed user interfaces and, secondly, their underlying code can be re-engineered if needed. Their results are, in addition, more consistent. Other tasks, such as semantic searching, are better served by transformer-based techniques. As a result, we have found that, so far, a combination of traditional and more recent methods is preferable.

This survey of the computational tools and strategies developed or assessed for the Divergent Discourses project will, we hope, be useful for Tibetanists looking to work with digital methodologies such as text-mining. However, our survey is by no means comprehensive, focusing mainly on tools developed by BDRC, Esukhia, and ACTib, and we apologise for important contributions that we have not included here. We note, however, that most of the steps that we have described here involve specialist technical knowledge and experience, and to put them into practice takes extensive time, funding, and expert support for problem-solving. Implementation of these tools is not smooth and involves multiple, time-consuming stages of error correction, troubleshooting and consultation within the specialist community. Nevertheless, we hope this special issue of RET, by describing some of the technical considerations we have experienced in developing text-mining tools for use with modern Tibetan texts, will be helpful for others seeking to apply such methods in their work.

Bibliography

Engels, James, and Robert Barnett

“Developing a Semantic Search Engine for Modern Tibetan,” *Revue d'Etudes Tibétaines* 74, 2025, pp. 262–283.

Engels, James, Robert Barnett, Franz Xaver Erhard, and Nathan. W. Hill
 “Transkribus_utils: Paragraph Extractor (v1_Paragraph_Extractor),” *Zenodo*, 2024. [doi:10.5281/zenodo.10810509](https://doi.org/10.5281/zenodo.10810509)

- Engels, James, Franz Xaver Erhard, Robert Barnett, and Nathan W Hill
 “Tibetan for Spacy 1.1 [Data set],” *Zenodo*, 2023. [doi:10.5281/zenodo.10148636](https://doi.org/10.5281/zenodo.10148636)
- Erhard, Franz Xaver.
 “The Divergent Discourses Corpus: A Digital Collection of Early Tibetan Newspapers of the 1950s and 1960s,” *Revue d’Etudes Tibétaines* (74), 2025a, pp. 45–81.
- “Text and Layout Recognition for Tibetan Newspapers with Transkribus,” *Revue d’Etudes Tibétaines* (74), 2025b, pp. 129–172.
- Erhard, Franz Xaver and Xiaoying 笑影
 “Foreign names and places in Tibetan newspapers of the 1950s and 1960s,” *Revue d’Etudes Tibétaines* (74), 2025, pp. 173–187.
- Erhard, Franz Xaver, Yuki Kyogoku, Robert Barnett, and Nathan W. Hill
 “Modern-Botok. Custom dictionary for modern Tibetan (v0.1) [Data set],” *Zenodo*, 2024. [doi:10.5281/zenodo.14034747](https://doi.org/10.5281/zenodo.14034747).
- Erhard, Franz Xaver, Xiaoying 笑影, Barnett, Robert, and Nathan W. Hill
 “Tibetan Modern U-chen Print (TMUP) 0.1: Training Data for a Transkribus HTR Model for Modern Tibetan Printed Texts,” *Fachinformationsdienst (FID) Asien*, 2023. [doi:10.48796/20240313-000](https://doi.org/10.48796/20240313-000).
- “Toponyms and Anthroponyms from Tibetan-language Newspapers of the 1950s and 1960s: Three Name Lists,” *Zenodo*, 2024. [doi:10.5281/zenodo.14526125](https://doi.org/10.5281/zenodo.14526125).
- Dakpa, Jamyang, Tashi Dhondup, Yeshe Jigme Gangne, Edward Garrett, Marieke Meelen, and Sonam Wangyal
 “Modern Tibetan Corpus,” *Github repository*, 2021a. Available online at <https://github.com/tibetan-nlp/modern-tibetan-corpus/tree/v1.0> (accessed January 30, 2025).

"Modern Tibetan Corpus Annotated for Verb-argument Dependency Relations," *Zenodo*, 2021b, [doi:10.5281/zenodo.4727129](https://doi.org/10.5281/zenodo.4727129).

Faggionato, Christian, Edward Garrett, Nathan W. Hill, Samyo Rode, Nikolai Solmsdorf, and Sonam Wangyal

"Classical Tibetan Corpus Annotated for Verb-argument Dependency Relations," *Zenodo*, 2021. [doi:10.5281/zenodo.4727108](https://doi.org/10.5281/zenodo.4727108).

Faggionato, Christian, Nathan W. Hill, and Marieke Meelen

"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*, pp. 1–6, Marseille. European Language Resources Association. 2022. Online available at <https://aclanthology.org/2022.euralli-1.1/> (accessed January 31, 2025).

Hill, Nathan W., and Edward Garrett

"A part-of-speech (POS) tagged corpus of Classical Tibetan [Data set]," *Zenodo*, 2017. [doi:10.5281/zenodo.574878](https://doi.org/10.5281/zenodo.574878).

Kahle, Philip, Sebastian Colutto, Günter Hackl and Günter Mühlberger.

"Transkribus - A Service Platform for Transcription, Recognition and Retrieval of Historical Documents." In *14th IAPR International Conference on Document Analysis and Recognition: ICDAR 2017: proceedings*. Los Alamitos: Conference Publishing Services, IEEE Computer Society, 2017, pp. 19–24. [doi:10.1109/ICDAR.2017.307](https://doi.org/10.1109/ICDAR.2017.307)

Kyogoku, Yuki, Franz Xaver Erhard, Robert Barnett, and Nathan W. Hill

"TibNorm - Normaliser for Tibetan (Version v1)," *Zenodo*, 2024a, [doi:10.5281/zenodo.10815272](https://doi.org/10.5281/zenodo.10815272).

"Diverge-Gemini POS-tagged Corpus of Modern Tibetan (1.0) [Data set]," *Zenodo*, 2024b, [doi:10.5281/zenodo.14447192](https://doi.org/10.5281/zenodo.14447192).

"Basic Modern Tibetan SpaCy Model," *Zenodo*, 2024c, [doi:10.5281/zenodo.14494472](https://doi.org/10.5281/zenodo.14494472).

Kyogoku, Yuki, Franz Xaver Erhard, James Engels, and Robert Barnett
"Leveraging Large Language Models in Low-resourced Language NLP: A spaCy Implementation for Modern Tibetan," *Revue d'Etudes Tibétaines* (74), 2025, pp. 188–221.

Luo, Queenie and 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*, (72), 2024, pp. 26-42. Available online at https://d1i1jdw69xsqx0.cloudfront.net/digitalhimalaya/collections/journals/ret/pdf/ret_72_02.pdf (accessed January 20, 2025).

Lv, Hui, Pu Chi, La Duo, Yan Li, Zhou Qingguo and Shen Jun
"T-LLaMA: a Tibetan large language model based on LLaMA2," *Complex & Intelligent Systems* 11(1), 2024. [doi:10.1007/s40747-024-01641-7](https://doi.org/10.1007/s40747-024-01641-7).

Marieke Meelen, Nathan W. Hill, and Christopher Handy
"The Annotated Corpus of Classical Tibetan (ACTib), Part I—Segmented version, based on the BDRC digitised text collection, tagged with the Memory-based Tagger from TiMBL," *Zenodo*, 6 July 2017a. [doi:10.5281/zenodo.823707](https://doi.org/10.5281/zenodo.823707).

"The Annotated Corpus of Classical Tibetan (ACTib), Part II—POS-tagged version, based on the BDRC digitised text collection, tagged with the Memory-based Tagger from TiMBL," *Zenodo*, 2017b. [doi:10.5281/zenodo.822537](https://doi.org/10.5281/zenodo.822537).

Meelen, Marieke, Sebastian Nehrlich and Kurt Keutzer
"Breakthroughs in Tibetan NLP & Digital Humanities," *Revue d'Etudes Tibétaines*, (72), 2024, pp. 5-25. Available online at https://d1i1jdw69xsqx0.cloudfront.net/digitalhimalaya/collections/journals/ret/pdf/ret_72_01.pdf (accessed January 26, 2025).

Meelen, Marieke, and Élie Roux

"The Annotated Corpus of Classical Tibetan (actib) - Version 2.0 (segmented & Pos-tagged)," *Zenodo*, 2020. [doi:10.5281/zenodo.3951503](https://doi.org/10.5281/zenodo.3951503).

Meelen, Marieke, Élie Roux, and Nathan Hill

"Optimisation of the Largest Annotated Tibetan Corpus Combining Rule-Based, Memory-Based, and Deep-Learning Methods." In *ACM Transactions on Asian and Low-Resource Language Information Processing* 20 (1), 2021, pp. 1–11. [doi:10.1145/3409488](https://doi.org/10.1145/3409488).

Natalia Mikhailova

"*The Tibet Mirror*, 'Friends of Tibet,' and Internationalisation of the Tibet Question," *Revue d'Etudes Tibétaines*, (74), 2025, pp. 284–328.

Pramanik, Siddhartha.

"Continual Learning and Catastrophic Forgetting: The Challenges and Strategies in AI," *Medium*, January 17, 2025. Available online at <https://medium.com/@siddharthapramanik771/continual-learning-and-catastrophic-forgetting-the-challenges-and-strategies-in-ai-636e79a6a449> (accessed January 30, 2025).

Sabbagh, Christina.

"Enhanced HTR Accuracy for Tibetan Historical Texts - Optimising Image Pre-processing for Improved Transcription Quality," *Revue d'Etudes Tibétaines*, (74), 2025, pp. 82–128.

Sabbagh, Christina, Franz Xaver Erhard, Robert Barnett, and Nathan W. Hill

"Divergent Discourses Custom Image Preprocessing (Sauvola Binarisation)," *Zenodo*, 2024a. [doi:10.5281/zenodo.14525692](https://doi.org/10.5281/zenodo.14525692).

"Divergent Discourses Custom Image Preprocessing (Forked Binarisation)," *Zenodo*, 2024b. [doi:10.5281/zenodo.14523007](https://doi.org/10.5281/zenodo.14523007).

Schwartz, Ronald, and Robert Barnett

“Religious Policy in the TAR, 2014–24: Topic Modelling a Tibetan Language Corpus with BERTopic,” *Revue d’Etudes Tibétaines*, (74), 2025, pp. 222–261.

Sun Yuan, Liu Sisi, Deng Junjie, Sun Yuan and Zhao Xiaobing

“TiBERT: Tibetan Pre-trained Language Model*.” *In* IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2022, pp. 2956–2961. [doi:10.48550/arXiv.2205.07303](https://doi.org/10.48550/arXiv.2205.07303).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin

“Attention is All you Need,” *Advances in Neural Information Processing Systems*. 30. Curran Associates, 2017. [doi:10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762)

Wallman, Jeff, Zach Rowinski, Ngawang Trinley, Chris Tomlinson, and Kurt Keutzer

“Collection of Tibetan Etexts Compiled by the Buddhist Digital Resource Center,” *Zenodo*, 2017. [doi:10.5281/zenodo.821218](https://doi.org/10.5281/zenodo.821218).

Zhang Jiangyan, Deji Kazhuo, Luosang Gadeng, Nyima Trashi, and Nuo Qun

“Research and Application of Tibetan Pre-training Language Model Based on BERT.” *In* Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics (ICCIR '22). Association for Computing Machinery: New York, 2022, pp. 519–524. [doi:10.1145/3548608.3559255](https://doi.org/10.1145/3548608.3559255).

