Automatic Segmentation of Classical Tibetan Texts into Autochthonous and Allochthonous Regions

Guy Bilitski[1,2], Lev Shechter[1,2], Sonam Jamtsho[3], Nir Marciano[1,2], Nicola Bajetta[3], Rebecca Sunden[3], Omri Drori[1,2], Kai Golan Hashiloni[1,2], Orr Zwebner[1,2], Asaf Shina[1,2], Orna Almogi[3], Dorji Wangchuk[3] and Kfir Bar[1,2]

[1]Data Science Institute, Reichman University, Herzliya, Israel

[2]Efi Arazi School of Computer Science, Reichman University, Herzliya, Israel

[3]University of Hamburg, Hamburg, Germany

{guy.bilitski, lev.shechter, nir.marciano, omri.drori, kai.golanhashiloni, orr.zwebner, asaf.shina}@post.runi.ac.il

kfir.bar@runi.ac.il

{sonam.jamtsho, nicola.bajetta, rebecca.sunden, orna.almogi, dorji.wangchuk}@uni-hamburg.de

Abstract

We introduce a new computational framework for segmenting Classical Tibetan texts into *autochthonous* and *allochthonous* regions, distinguishing between indigenous Tibetan compositions and translated materials, primarily from Sanskrit sources. To support this task, we release the first annotated Tibetan corpus for autochthonous–allochthonous segmentation and evaluate several multilingual encoders, including mBERT, XLM-R and our own pre-trained model, fine-tuned for sequence labeling. Our model had the best performance, achieving strong alignment with expert annotations, showing that multilingual representations can effectively capture philological boundaries in low-resource settings. This work contributes new resources and methods for computational philology and sheds light on linguistic markers that trace the intercultural evolution and transmission of Buddhist texts in Tibet.

Keywords: Tibetan, segmentation, multilingual models, philology, digital humanities

1. Introduction

Historical Tibetan texts often combine materials originating from different compositional layers and sources. In Tibetan philology, this phenomenon is studied through stratification analysis, which seeks to identify stylistic and contextual shifts that signal changes in provenance or compositional origin. In this work, we study the automatic segmentation of Tibetan texts into their autochthonous (AUTO) and allochthonous (ALLO) components, framing the task as a computational analogue of stratification analysis. By revealing the internal structure of Tibetan works, our approach enables new quantitative studies of authorship, translation practices, and the dynamic exchange between inherited and indigenous strands of Buddhist intellectual history. At the core of such questions are issues concerning scripturalization, authentication, and canonization of the numerous new Buddhist scriptures (and non-scriptural works) that emerged within the Tibetan cultural sphere. To establish authenticity, these works were presented as Indic texts in translation. To address this phenomenon, editors of the Tibetan Buddhist Canon took great care to establish criteria for determining authenticity. One key criterion was proven Indic origin. Consequently, works lacking translation colophons naturally aroused suspicion within the tradition itself, while dubious works that possessed such colophons were scrutinized by Tibetan scholars using additional criteria (Almogi, 2020).

Most scriptures considered doubtful in Tibet can be classified into three categories: (a) Indic, (b) hybrid Indic-Tibetic, and (c) Tibetic (Almogi, 2019). Our investigation focuses on the second category—hybrid Indic-Tibetic—which comprises works containing allochthonous (translated) Indic texts in Tibetan translation interwoven with autochthonous (indigenous) Tibetan texts to form new scriptures. Since scriptures are by definition the Word of the Divine, they contain no explicit citations; the allochthonous texts are interwoven through "borrowing" (or "reuse"), with no lexical markers denoting their origin. The automatic identification of segments as allochthonous or autochthonous will, for the first time, enable researchers to study their compositional history independently of locating textual matches or parallels in allochthonous works.

Classical Tibetan poses particular challenges for computational analysis. It is a syllable-delimited language without explicit word boundaries, characterized by flexible use of grammatical particles and considerable orthographic variation. These properties complicate segmentation and parsing and limit the effectiveness of models trained on modern or high-resource languages.

Beyond classifying entire works, our approach seeks to identify switch points that mark transitions between ALLO and AUTO segments as shown in Figure 1, thereby revealing the internal structure of hybrid compositions. The challenge lies in detecting subtle linguistic and stylistic cues rather than overt lexical markers, which are in any case

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 1017–1030
11–16 May 2026.

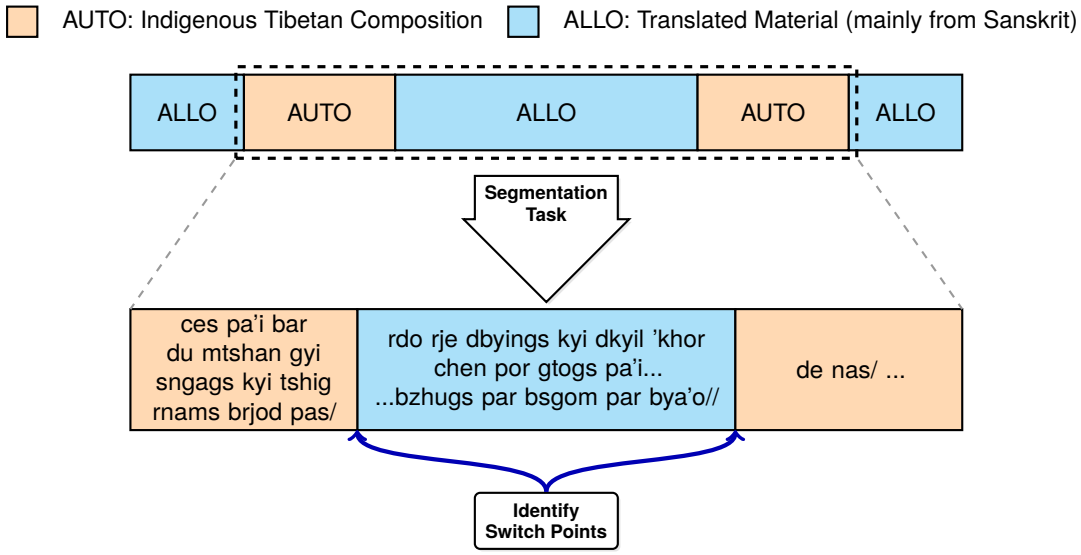

Figure 1: An illustration of the Classical Tibetan text-segmentation task. The upper diagram illustrates an AUTO text that includes citations, represented by the two outer ALLO segments, as well as near verbatim borrowing, represented by the central ALLO segment. The lower diagram magnifies a continuous three-segment transition (AUTO $\rightarrow$ ALLO $\rightarrow$ AUTO), highlighting the objective of identifying exact textual switch points.

lacking in scriptures. It makes this challenge an ideal testbed for evaluating the ability of language models to capture domain-specific signals in a low-resource classical language.

We experiment with both general-purpose large language models (LLMs) and specialized encoder-based architectures fine-tuned on our annotated Tibetan corpus. The results show that the task is far from trivial: heuristic and random baselines yield near-zero performance, confirming that segmentation depends on deep contextual understanding. While closed-weights, generative LLMs such as Gemini 2.5 Flash (Comanici et al., 2025) perform poorly ($F_{\beta=2} < 0.4$ as shown in Table 3), fine-tuned encoders trained with proximity-aware objectives achieve substantially higher accuracy, reaching $F_{\beta=2} = 0.77$ within a five-token tolerance window. These findings highlight the limits of zero-shot prompting for linguistically specialized domains and the value of supervised adaptation of pretrained encoders grounded in Classical Tibetan. Objective design and post-processing further improve structural consistency and interpretability, which are essential for scholarly applications.

In this work, we make three primary contributions. First, we introduce a new computational task: the automatic segmentation of Classical Tibetan texts into allochthonous and autochthonous segments. Second, we present the first annotated corpus designed for this purpose, providing a foundation for quantitative and comparative research on translation and authorship patterns in Tibetan literature. Finally, we establish a set of baseline models based on multilingual encoders fine-tuned for sequence labeling, demonstrating the feasibility of this task and outlining challenges for future work in this emerging area of computational philology.

2. Related work

2.1. Tibetan Corpora

The development of large annotated resources has enabled substantial progress in Tibetan NLP. The Annotated Corpus of Classical Tibetan (ACTib) contains hundreds of millions of segmented words provided along with part-of-speech (POS) tags. The corpus integrates earlier resources from the Buddhist Digital Resource Center (BDRC)[1] and employs both rule-based and data-driven segmentation approaches.[2]

At the methodological level, a key turning point was the formulation of Tibetan word segmentation as a *sequence labeling* problem, that is, predicting for each syllable in a sequence whether it marks the beginning, inside, or end of a word. Early work by Liu et al. (2011) applied Conditional Random Fields (CRFs) using a BMES (Begin, Middle, End, Single) tagging scheme over syllables, establishing strong baselines for Tibetan sequence labeling. Later studies extended this paradigm using neural CRF architectures such as BiLSTM-CRF, improving accuracy on segmentation and named entity recognition tasks (Wang and Yang, 2018). Data-driven taggers tailored for Classical Tibetan (Meelen and

[1] https://www.bdrc.io/
[2] https://zenodo.org/records/3951503

Hill, 2017) also proved effective and have since been integrated into broader processing pipelines.

2.2. Boundary Detection

The ALLO/AUTO segmentation task can be reformulated as a *boundary detection* or *switch-point localization* problem, focusing on identifying transitions between segments rather than labeling each token individually. This framing parallels research in discourse and topic segmentation, where evaluation metrics such as P_k (Beeferman et al., 1999), WindowDiff (Pevzner and Hearst, 2002), and Boundary Similarity (Fournier, 2013) explicitly account for near–miss predictions. These metrics recognize that minor deviations from the gold boundary often preserve the semantic correctness of the segmentation.

Inspired by this line of work, our evaluation adopts a tolerance-based approach, where predicted switch points are considered correct if they fall within a small token window around the gold annotation. This accounts for slight positional variation while emphasizing the model's ability to localize true transition regions.

2.3. Analogous Tasks: Code-Switching and Style-Change Detection

The ALLO/AUTO segmentation task bears conceptual resemblance to several well-studied linguistic boundary detection tasks. In particular, *code-switching* detection seeks to identify boundaries between linguistic systems in multilingual utterances. Benchmarks such as GLUECoS (Khanuja et al., 2020) and LinCE (Aguilar et al., 2020) evaluate both token-level and boundary-level accuracy, providing transferable insights into cross-lingual segmentation and transition modeling.

Similarly, the *style-change* and *authorship* boundary detection tasks studied in the PAN shared tasks (Kestemont et al., 2018) treat abrupt stylistic shifts as segmentation cues. Both domains emphasize detecting transition regions rather than enforcing exact boundary matches, highlighting the value of tolerance-based metrics that capture functional correctness even when token alignment is slightly offset.

Low-resource languages such as Tibetan benefit greatly from multilingual encoders like `mBERT` (Devlin et al., 2019) and `XLM-RoBERTa` (Conneau et al., 2020), which enable cross-lingual transfer and provide a robust foundation for sequence labeling tasks. Continued pretraining on in-domain corpora, tokenizer adaptation, and fine-tuning have been shown to substantially improve downstream performance in such settings. Recent Tibetan-specific work, `Banzhida` (Pan et al., 2025), continues pretraining `Qwen2.5-7B-base` (Qwen, 2024) on a 72 GB curated Tibetan corpus with staged language-balanced and long-context training, extends the tokenizer, and reports consistent gains over general multilingual and Tibetan-tailored models on bespoke benchmarks. Complementing this, Andryushchenko and Ivanov (2025) systematically compared several tokenizer adaptation strategies: Full Vocabulary Transfer (FVT), Focused Subword Transfer (FOCUS), and Zero-shot Tokenizer Transfer (ZeTT), and found ZeTT to be particularly effective. They also highlighted trade-offs between preserving cross-language coverage and maximizing target-language efficiency, providing a methodological framework for vocabulary expansion and tokenizer transfer directly relevant to Tibetan NLP.

2.4. Transfer Learning and Tibetan-centric Language Models

Recent progress includes the release of Tibetan-specific resources such as the TLUE benchmark and the TIB-STC corpus, alongside the development of Tibetan-centric large language models such as `Sun-Shine` and `T-LLaMA`, which integrate Tibetan linguistic and cultural data into multilingual pretraining pipelines (Huang et al., 2025; Lv et al., 2025). Building on this progress, we extend multilingual foundational models such as `mBERT` through continued pretraining on Tibetan corpora to obtain a Tibetan-specialized encoder used in our experiments.

2.5. Positioning

Most prior work in Tibetan NLP focuses on word- or syllable-level segmentation. In contrast, our work explicitly targets the detection of *switch points* between ALLO (translated) and AUTO (indigenous) textual segments. This reformulation aligns the task more closely with discourse segmentation, code-switching, and style-change detection, emphasizing flexible, tolerance-based evaluation rather than strict token-level matching. Such an approach better serves the practical needs of philologists and digital humanities scholars investigating the compositional structure of Tibetan texts.

2.6. Link to the Present Work

This body of research motivates our formulation of ALLO/AUTO segmentation as a boundary localization problem. Rather than enforcing exact token-wise alignment, we evaluate whether models can accurately localize transition regions between ALLO and AUTO segments within a predefined tolerance window, reflecting the methodological principles established in discourse segmentation and code-switching evaluation.

3. Method

The task of identifying ALLO and AUTO segments in text is formulated as a sequence labeling problem. Our focus lies on accurately detecting *switch points*, the positions in the text where ALLO or AUTO segments begin. These points are crucial since, once identified, they deterministically define the boundaries of both ALLO and AUTO segments. We refer to this as the ALLO/AUTO segmentation task. We employ both encoder-based and decoder-based language models to perform the task.

3.1. Evaluation

Rather than requiring an exact token-level label-match per token, we focus on switch points, that is, the specific tokens at which the text transitions from ALLO to AUTO or vice versa. We note that when evaluating switch-point predictions, the model may pinpoint a location near the true transition rather than matching it exactly. To account for this, we allow for a tolerance window around the annotated switch points. Specifically, a predicted switch point is considered correct if it falls within a predefined threshold (e.g., ±5 tokens) of the gold annotation. This accounts for minor boundary variations while still capturing whether the model is effectively pointing to the correct transition region.

For each switch direction, for example from AUTO to ALLO, precision and recall are evaluated independently, computing the values for each type. To provide a balanced measure across both transition directions, we report their macro-averaged precision and recall, defined as the average of the corresponding values for the two transition directions.

We adopt the $F_{\beta=2}$ score as our main evaluation metric, placing greater emphasis on recall than on precision. With $\beta = 2$, recall is weighted twice as heavily, reflecting the practical preference to highlight all potential translated regions rather than risk missing genuine switch points. This combination of proximity-based evaluation and recall-oriented scoring offers a more flexible and informative assessment of model performance for scholarly text segmentation tasks.

3.2. Data

To fine-tune and evaluate models on the segmentation task, we asked Tibetan philology scholars to manually annotate texts into ALLO and AUTO segments, forming the gold-standard dataset used throughout our experiments. Each text represents a continuous passage of classical Buddhist literature. The annotators provided positive samples of allochthonous texts interwoven in autochthonous texts, including known cases of borrowing and citations of allochthonous texts in autochthonous ones, and negative samples of allochthonous texts interwoven in allochthonous ones and autochthonous texts interwoven in autochthonous ones. The two main annotators, who are experienced Tibetologists (both are post-doctoral scholars), segmented AUTO texts by locating passages that can be identified as ALLO with a high degree of confidence on the basis of various criteria. Each annotator selected texts within their area of expertise and manually marked ALLO and AUTO segments directly in the text. Subsequently, we applied an automated procedure to convert these annotated segments into a token-level classification format suitable for model training and evaluation.

The texts were taken from BuddhaNexus[3], rKTS[4], and ACIP[5]. The specific text identifiers can be found in Appendix 8.6.

For the train, validation, and test split, each text was divided into contiguous chunks based on the ends of sentences or paragraphs, resulting in approximately 250 to 450 tokens in length to standardize the input size for model training and evaluation. Since many of these chunks contain no code-switches (i.e., they consist entirely of a single segment type), they are less informative from a machine-learning perspective. Therefore, we maintained a controlled distribution in the dataset: 33% of the chunks were selected to contain no switches, while the remaining 67% include at least one switch point. Each chunk is therefore annotated with zero or more transitions between ALLO and AUTO, serving as the fundamental unit for supervised learning and evaluation.

The annotated corpus was divided into training, validation, and test splits, maintaining an approximately balanced distribution between AUTO and ALLO segments across all subsets. Table 1 summarizes the dataset statistics.

Metric	Train	Val	Test
Total number of chunks	456	91	121
Chunks with switches	298	67	75
Total number of tokens	140K	27K	37K
Total number of switches	785	179	210
Switches to AUTO	358	81	94
Switches to ALLO	427	98	116

Table 1: Dataset statistics for the training, validation, and test splits.

[3] buddhanexus.kc-tbts.uni-hamburg.de/

[4] www.rkts.org/rktsneu/sub/index.php

[5] www.buddhism-dict.net/ebti/textinput/acip.html

Since ALLO segments represent passages cited or borrowed from other works, which often appear verbatim or nearly so across multiple texts, we explicitly searched for duplicate or near-duplicate occurrences within our corpus (train, validation, and test sets). Any identical chunks found were removed from the training data to prevent the model from simply memorizing previously seen ALLO sequences. This step ensures that model performance reflects genuine generalization rather than lexical recall. Notably, the same phenomenon complicates human annotation: scholars often identify ALLO passages precisely because they recognize them from other sources, highlighting the inherent intertextuality of Tibetan canonical literature.
The annotated dataset can be found in the paper's GitHub repository.[6]

3.3. Models

The data, fine-tuning, and evaluation pipelines for all models were kept identical and fully reproducible to ensure a fair and consistent comparison across experimental settings.

3.3.1. Encoder-based LMs

We designed a specialized ALLO/AUTO segmentation pipeline based on encoder language-model architectures, which offer efficiency and practical applicability for real-world use cases. We began with `mBERT` (Devlin et al., 2019), a multilingual encoder pre-trained on over one hundred languages with a vocabulary of approximately 120k tokens. We used `mBERT` primarily to evaluate which token-classification architecture best fits our task, comparing two alternatives: a named entity recognition (NER)-style model with four labels ($B\text{-}ALLO$, $I\text{-}ALLO$, $B\text{-}AUTO$, $I\text{-}AUTO$), where B- marks the beginning of a segment and I- denotes tokens within it; and a code-switching model with three labels ($switch\text{-}AUTO$, $switch\text{-}ALLO$, O), where the first two indicate transitions into AUTO and ALLO segments, respectively. We introduce additional techniques for optimization of the objective function, token-alignment post-processing, and a method to reward the model to predict switch points at more natural switching points, identified by domain scholars. We describe these techniques in detail in Section 3.3.2.

Base models. We experimented with several base models for this task. We considered two multilingual models: `mBERT` and `XLM-RoBERTa` (Conneau et al., 2020), as well as two Tibetan-specialized models: `CINO` (Yang et al., 2022) and `Tibetan RoBERTa`.[7]

We additionally trained a Tibetan-adapted variant of `mBERT`, which we call `ALTO`, by continually pre-training the multilingual model on a 13.8 GB corpus (Appendix 8.2) of cleaned and Tibetan canonical, para-canonical, and non-canonical scriptures and non-scriptures. Some of the data were converted between Unicode and Wylie using a tool we developed and release as part of this work (Full details are in Appendix 8.3). The tokenizer was trained from scratch with a 32K WordPiece vocabulary optimized for Tibetan, yielding a 30.8% reduction in token count compared to the original `mBERT`. Tokenizer details are provided in Appendix 8.1 and Table 4. The continued pre-training (cpt) followed standard masked language modeling (See Appendix 8.4), before fine-tuning on the segmentation task.

All models were trained under a unified setup for consistency: batch size 8, sequence length 512, learning rate $2e^{-5}$, AdamW with warmup and weight decay, FP16 precision, and gradient clipping at 1.0, using the dataset in Table 1.

3.3.2. Objective Function Optimization for Segmentation

Building on our model architecture and empirical findings, we sought to refine the training objective to better align model behavior with the specific requirements of ALLO/AUTO segmentation. While standard token-level cross-entropy loss encourages exact label matching, segmentation quality depends more on identifying accurate switch regions—the boundaries between AUTO and ALLO segments—rather than on token-level precision alone. To address this, we designed a series of proximity- and structure-aware loss functions that make the model more sensitive to plausible transition points while maintaining linguistic and structural consistency.

Multiplicative proximity-aware objective (MUL). To reflect the inherent fuzziness of segment boundaries, we introduce a proximity-aware weighting factor $\alpha_{i,j}$ that scales the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ according to the distance between predicted and true switch points. Within a tolerance window of $\tau = 5$ tokens, nearby predictions are rewarded, while distant ones are penalized:

Reward: Let S and $\hat{S}$ be the golden and prediction sets of switch points, respectively. When a gold switch at position $j \in S$ is matched by a predicted switch $\hat{k} \in \hat{S}$ such that $|j - \hat{k}| \leq \tau$, the corresponding loss term is scaled by a reward (r) factor $\gamma_r < 1$,

[6] https://github.com/Intellexus-DSI/alloauto

[7] https://huggingface.co/sangjeedondrub/tibetan-roberta-base

reducing the penalty for "close enough" predictions. We use $\gamma_r = 0.1$ for exact matches ($d = 0$), with a linear interpolation up to $\gamma_r = 1.0$ at the window boundary:

$$\gamma_r(d) = 0.1 + 0.18d \quad \text{for } 0 < d \leq \tau.$$

Penalty: Predictions farther than τ tokens from any gold switch are scaled by a penalty (p) factor $\gamma_p = 10.0$, discouraging spurious or distant switches.

This multiplicative objective effectively balances recall around true boundary regions with robustness to false positives.

Additive proximity objective (ADD). We also explored an additive formulation that adjusts the loss directly through positive or negative offsets rather than scaling factors. For each token i, the adjusted loss is defined as:

$$\mathcal{L}_{\text{add}}(i) = \mathcal{L}_{\text{CE}}(i) + \Delta_i,$$

where Δ_i depends on the proximity between the prediction and any gold switch within a $\tau = 5$ token window. Predictions close to true boundaries receive negative offsets (rewards), while distant ones receive positive offsets (penalties). This formulation promotes smooth gradient updates and stable convergence, while encouraging recall in plausible switch regions and discouraging excessive switching.

Segmentation alignment heuristic (SAH). To further incorporate linguistic knowledge, we introduce a segmentation alignment heuristic (SAH) that leverages Tibetan orthographic structure. Because Tibetan texts often mark clause or sentence boundaries using a *shad* ("/" or "//"), predicted switches are encouraged to align with such markers. Formally, a modifier $\beta_{i,j}$ adjusts the loss contribution based on proximity to segmentation marks, where i indexes sequences in the batch and j indexes token positions within each sequence.

Reward: Loss is reduced by $\delta_r = 0.3$ (r denotes reward) if a predicted switch occurs at or immediately before a *shad*.

Penalty: Loss is increased by $\delta_p = 3.0$ (p denotes penalty) if a predicted switch occurs one or two tokens after a *shad*.

The resulting composite objective,

$$\mathcal{L} = \sum_{i,j} \beta_{i,j} \mathcal{L}_{\text{CE}}(y_{i,j}, \hat{y}_{i,j}),$$

encourages linguistically plausible segmentation while preserving the model's flexibility to adapt to varied textual patterns.

Post-processing for structural consistency (PP). Finally, to ensure coherence in predicted label sequences, we implement a deterministic post-processing layer. Although training data enforces valid token transitions (e.g., $B\text{-}AUTO$ cannot be followed by $I\text{-}ALLO$, models occasionally produce inconsistent outputs during inference. The PP module corrects such inconsistencies by enforcing logical constraints on label transitions—prohibiting adjacent switches, ensuring valid directionality (AUTO→ALLO or vice versa), and maintaining a minimal gap between consecutive switches. This step improves segmentation precision without affecting the underlying learned representations.

In summary, these loss refinements, MUL, ADD, and SAH, combined with a rule-based PP layer, collectively steer the model toward boundary-aware, linguistically grounded, and structurally coherent segmentation, leading to measurable improvements over standard token-level objectives.

3.3.3. Generative LLMs

We leverage the in-context learning capabilities of frontier LLMs through their official APIs. To balance performance with cost and prompt engineering efficiency, the Gemini 2.5 Flash model was used during development to iteratively refine and optimize the task prompt (see Appendix 8.5.1).

The final prompt was then used to evaluate the other models. Within the prompt context, the model was instructed to act as an expert in Tibetan philology and explicitly guided to identify switch points between allochthonous and autochthonous text segments. All LLMs (GPT-4o,[8] Sonnet 4.5,[9] Llama 4-Scout,[10] and Gemini 2.5 (Comanici et al., 2025)) were evaluated in a zero-shot setting and temperature of 0.3 for stability.

3.4. Baseline Models

As baseline methods, we evaluated several approaches to the ALLO/AUTO segmentation task. **(1) Random:** a stochastic baseline that draws the number of switch points per segment from a Poisson distribution (λ equal to the empirical average number of switches in the test set, typically $\lambda \approx 3.5$). Switch positions are then sampled uniformly from the token range (excluding the first and last 10 tokens), and each switch alternates the current mode between AUTO and ALLO. Segments with zero switches are assigned a single random mode for all tokens. (2) **Heuristic**: In this baseline, we fine-tuned the `RoBERTa` (Liu et al., 2019) model

[8] https://openai.com/
[9] https://www.anthropic.com/
[10] https://ai.meta.com/

on the ALLO/AUTO segmentation task using a labeled dataset from `DharmaBench` (Hashiloni et al., 2025). The dataset consists of 600 training samples and 400 test samples of text chunks and their class of either ALLO or AUTO. We trained the base model with a binary classification head. The classifier achieved an accuracy of 99%, with a macro-average $F_{\beta=2}$ score of 0.99 (precision 0.99, recall 1.00) on binary classification of ALLO or AUTO. For evaluation, we took the test data we curated, which is detailed in Table 1. We segmented them heuristically by splitting at line boundaries and classifying each line individually. Switch points were determined at the transitions between ALLO and AUTO line labels.

4. Results

Our experimental process involved evaluating both the three-label and four-label architectures across all base models (`mBERT`, `ALTO`, and others). After identifying the leading model from this comparison, we applied the different objective functions and additional techniques introduced in Section 3.3.1 to further assess their individual and combined contributions. The complete results are presented in Table 2. The post-processing methods we developed do not apply to the three-class architecture and are therefore omitted. Table 3 presents the results obtained with the generative LLMs, which performed noticeably worse on the ALLO/AUTO segmentation task. We discuss the results in the following section.

5. Discussion

5.1. Discussion of Results

Task difficulty. As shown in Table 2, both heuristic and random baselines yield nearly zero $F_{\beta=2}$, indicating that the ALLO/AUTO segmentation task cannot be solved through simple pattern recognition. This demonstrates that meaningful progress requires models capable of capturing subtle contextual and stylistic cues beyond surface features.

Generative LLMs perform substantially worse in our setup. Closed API-based generative models (Table 3) perform poorly, with $F_{\beta=2}$ scores below 0.4 even for the best-performing model. Despite their strong general language modeling capability, these models fail to reliably identify precise switch points even under the relaxing proximity-based evaluation. We attribute this to two factors: (i) generative models produce dynamic output, which is hard to parse, rather than direct token-level predictions; and (ii) LLMs lack explicit domain grounding in Classical Tibetan syntax and translationese indicators.

Fine-tuned encoders achieve substantial gains. Encoder-based token-classification models achieved significantly better results. It improves precision and recall, with ALTO reaching an $F_{\beta=2}$ of 0.797 under a 5-token tolerance evaluation (Table 2). This confirms that supervised adaptation, rather than zero-shot prompting, is essential for capturing the nuanced linguistic cues of autochthonous texts.

Post-processing improves structural consistency. As seen in Table 2, the post-processing (PP) layer notably increases precision (e.g., from 0.477 to 0.611 in ALTO+MUL) with only a small effect on recall. This demonstrates that rule based enforcement of valid label transitions effectively reduces over segmentation and improves interpretability, a valuable property for scholarly applications where false negatives are more costly than precision.

Best-performing configurations. Multiple ALTO configurations achieve similar $F_{\beta=2}$ values around 0.78, suggesting most of the improvement was gained from the continual pretraining and not from the fine tuning using the special objective functions. Arguably, further calibrations may increase the results, especially the trade-off between precision and recall.

5.2. Qualitative Analysis

While prompting the LLMs to predict switch points, we requested that each output include a dedicated "reasoning" field. This served two purposes: first, to encourage explicit model chain-of-thought (Wei et al., 2022), and second, to expose the linguistic and stylistic cues the models rely on when identifying ALLO/AUTO segments.

We subsequently compiled all model outputs alongside their corresponding gold labels and re-evaluated them using `Gemini 2.5 Flash`. We instructed the model to analyze its own reasoning traces from Tibetan segmentation outputs, focusing not on segmentation accuracy but on the logic and coherence of its explanations. Specifically, it was asked to identify recurring reasoning patterns, and provide a global assessment of reasoning quality. The complete prompt is included in Appendix 8.5.2. This analysis revealed that the model's reasoning patterns were systematic and evidence-based, frequently referencing citation markers, Sanskrit loanwords, and stylistic transitions as indicators of ALLO/AUTO boundaries. However, this surface-oriented approach often led to overgeneralization, because relying on Sanskrit loanwords often led to poetic or ritual Tibetan passages being incorrectly classified as translated. Moreover, referencing ci-

Model	$F_{\beta=2}$		Precision		Recall	
	3c	4c	3c	4c	3c	4c
Heuristic	0.005	0.005	0.013	0.013	0.005	0.005
Random	0.032	0.032	0.019	0.019	0.038	0.038
Tibetan RoBERTa	0.212	0.185	0.148	0.141	0.238	0.200
CINO	0.671	0.251	0.461	0.495	0.757	0.224
mBERT	0.673	0.667	0.489	0.405	0.743	0.795
XLM-RoBERTa	0.689	0.611	0.489	0.333	0.767	0.771
ALTO	0.791	0.776	0.681	0.631	0.824	0.824
ALTO + MUL	0.796	0.750	0.662	0.477	0.838	0.876
ALTO + MUL + PP	–	0.770	–	0.611	–	0.824
ALTO + MUL + SAH	0.783	0.776	0.653	0.501	0.824	0.900
ALTO + MUL + SAH + PP	–	**0.793**	–	0.622	–	0.852
ALTO + ADD	0.779	0.778	0.512	0.530	0.895	0.881
ALTO + ADD + PP	–	0.781	–	0.623	–	0.833
ALTO + ADD + SAH	**0.797**	0.765	0.527	0.495	0.914	0.886
ALTO + ADD + SAH + PP	–	0.766	–	0.579	–	0.833

Table 2: Results on the ALLO/AUTO segmentation task with a 5-token tolerance. Metrics are reported for both 3-class (3c) and 4-class (4c) settings as explained in Section 3.3.1.

Model	$F_{\beta=2}$	Precision	Recall
GPT-4o	0.011	0.025	0.010
Sonnet 4.5	0.030	0.038	0.029
Llama 4-Scout	0.032	0.029	0.033
Gemini 2.5 Pro	0.281	0.250	0.290
Gemini 2.5 Flash	0.389	0.381	0.390

Table 3: Performance comparison of generative LLMs on the ALLO/AUTO segmentation task.

tation markers stressed transition points resulting from citations over those resulting from borrowings, for which linguistic analysis is required. These findings suggest that improved segmentation performance will require models to incorporate broader contextual and semantic cues, rather than relying solely on surface-level markers.

5.3. Case Study

In addition to the model analyses reported above, we applied our best-performing model in a real-world case study to evaluate its potential usefulness for scholars working on this task. For this case study, we selected a paragraph from the text 020UMA,[11] which was not included in the training or evaluation sets, in order to assess the model's segmentation performance through manual inspection by a domain expert. The following Tibetan passage was examined (provided here in the Wylie transliteration system):

```
 don spyi dang med pa gsal ba
gnyis shes pa las tha dad pa'i
yul zhig yin na | yul rnam pa ni
rags | yid nye bar gtad pa ni yod
| 'khrul pa'i rgyu mtshan ni med
pas | yul snang rung gcig na gnas
pa'i gang zag gzhan gyis kyang
mthong bar 'gyur te | yul snang
rung du khyad par med pa'i phyir
bum pa thams cad kyis mthong ba
bzhin no | de skad du'ang rnam
'grel las | rnam pa de ni tha
dad na | gzhan gyis kyang ni
rtogs par 'gyur | zhes gsungs
pa ltar ro | don nyid yin kyang
lus kyi nang | mi mthong bzhin
du 'khrul snang gnyis | bdag la
rtag tu 'brel ba'i phyir | gzhan
gyis rtogs pa med ce na | don
spyi dang med pa gsal ba gnyis
po yul du grub kyang rang gi lus
kyi nang na gnas pa don yin yang
mi mthong ba ltar gzhan gyis mi
mthong ste | bdag gi blo dang
'brel ba'i phyir ro zhe na | nang
gi lus ni yul rung min | des na
rang gis kyang mi mthong | nang
gi lus mi mthong ba dper mi rung
ste | de ni yul rung ba ma yin
pa'i phyir rang nyid kyis kyang
mi mthong la | rung ba na gnas
na mthong bar 'gyur ro | rang
dang rtag tu 'brel pas na | gnyis
po'ang yum rung min zhe na | gal
te don spyi dang skra shad gnyis
```

[11] http://www.sakyalibrary.com/Assets/files/020UMA.pdf

```
kyang gzhan gyi snang du rung
ba'i yul ma yin gyi rang gi blo
dang dus rgyun du 'brel pas so
snyam na | rang blo kho na dang
'brel phyir | brjod kyang gzhan
gyis rtog mi 'gyur | rang rang gi
blo kho na dang 'brel na rang rig
pa bzhin du gzhan la bstan kyang
go bar mi 'gyur ro | de'ang rnam
'grel las | bdag la rtag tu 'brel
yin na | brjod kyang rtogs par
mi 'gyur ro | zhes gsungs pa ltar
ro |
```

Our best model correctly segmented the following segments as ALLO:

1. `rnam pa de ni tha dad na | gzhan gyis kyang ni rtogs par 'gyur |`

2. `bdag la rtag tu 'brel yin na | brjod kyang rtogs par mi 'gyur ro |`

Based on a qualitative analysis done by a professional scholar, the model demonstrated a strong capacity to distinguish between autochthonous Tibetan and allochthonous Indic discourse within this unseen passage. The segmentation algorithm successfully identified the transitions between the two textual strata, confirming the feasibility of automated stratification in Tibetan Buddhist corpora.

Additional examples of the model's segmentation output are provided in Appendix 8.7.

6. Conclusion

We introduced the first computational framework for segmenting classical Tibetan texts into *autochthonous* and *allochthonous* regions, supported by a newly annotated corpus and a suite of baseline models. Our experiments demonstrate that surface heuristics and zero-shot prompting approaches are insufficient for this highly specialized task, whereas supervised fine-tuning of multilingual encoders produces robust boundary detection. The proposed `ALTO` model, a continually pretrained Tibetan-adapted encoder derived from `mBERT`, achieves high boundary localization accuracy within a narrow tolerance window, confirming the feasibility of automated stratification as a practical tool for philological and historical research.

Beyond the methodological contributions, this work also aims to strengthen the broader Tibetan NLP ecosystem. As part of this research, the ALLO/AUTO annotated corpus, along with trained segmentation models, scripts, and evaluation protocols, has been made available to the public research community through Intellexus' GitHub repository[12] and Intellexus' HuggingFace.[13]

Furthermore, this token-classification method, together with the 3-class and 4-class classification schemes, post-processing techniques, and architectural decisions, can be applied to other manuscripts and different segmentation tasks.

Looking ahead, we plan to expand the corpus with additional canonical and non-canonical works to capture greater stylistic and temporal variation and to explore multilingual and multimodal extensions that integrate visual cues from manuscript layout and paratextual markers.

Limitations

While the proposed framework and experiments establish a promising foundation for Tibetan ALLO/AUTO segmentation, three main limitations should be acknowledged.

Dataset scale. The annotated corpus used in this study remains relatively small compared to benchmarks commonly used in sequence-labeling research. This limits the diversity of linguistic contexts the models are exposed to. Future work will benefit from enlarging the corpus with additional canonical and non-canonical texts to capture a broader range of stylistic and historical variation, and particularly with more samples containing borrowings, to enable better identification of stylistic features, while reducing the dependency on surface-oriented markers.

Single-annotator design. Each text was annotated by a single expert rather than by multiple independent annotators. Because reliable ALLO/AUTO segmentation requires deep philological familiarity with each work, the selected annotators were scholars who had studied the same texts for months or even years, thus ensuring sufficient accuracy for this exploratory stage.

Cross-model prompt stability. Finally, there is a limitation in our generative LLM evaluation setup regarding cross-model prompt stability. Because the prompt was iteratively refined and optimized exclusively using Gemini 2.5 Flash, it may favor this specific model. Prompt engineering is rarely perfectly transferable; this phenomenon means that reusing a prompt engineered for one model on another frequently yields degraded performance. Different model families often exhibit sensitivities to specific formatting constraints, role-playing directives, and in-context definitions.

[12] https://github.com/Intellexus-DSI/alloauto

[13] https://huggingface.co/Intellexus

Acknowledgments

This study is supported in part by the European Research Council (Intellexus, Project No. 101118558). 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 authorities can be held responsible for them.

7. Bibliographical References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

O. Almogi. 2020. *Authenticity and Authentication: Glimpses Behind the Scenes of the Formation of the Tibetan Buddhist Canon*. Indian and Tibetan studies. Department of Indian and Tibetan Studies, Universität Hamburg.

Orna Almogi. 2019. The human behind the divine: An investigation into the evolution of scriptures with special reference to the ancient tantras of tibetan buddhism. *Unearthing Himalayan Treasures: Festschrift for Franz-Karl Ehrhard*, pages 1–26.

Georgy Andryushchenko and Vladimir V. Ivanov. 2025. Evaluating tokenizer adaptation methods for large language models on low-resource programming languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 823–833, Vienna, Austria. Association for Computational Linguistics.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Chris Fournier. 2013. Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1702–1712, Sofia, Bulgaria. Association for Computational Linguistics.

Kai Golan Hashiloni, Shay Cohen, Asaf Shina, Jingyi Yang, Orr Meir Zwebner, Nicola Bajetta, Guy Bilitski, Rebecca Sundén, Guy Maduel, Ryan Conlon, Ari Barzilai, Daniel Mass, Shanshan Jia, Aviv Naaman, Sonam Choden, Sonam Jamtsho, Yadi Qu, Harunaga Isaacson, Dorji Wangchuk, Shai Fine, Orna Almogi, and Kfir Bar. 2025. DharmaBench: Evaluating language models on buddhist texts in Sanskrit and Tibetan. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2088–2110, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Cheng Huang, Fan Gao, Nyima Tashi, Yutong Liu, Xiangxiang Wang, Thupten Tsering, Ban Ma-bao, Renzeg Duojie, Gadeng Luosang, Rinchen Dongrub, et al. 2025. Sun-Shine: A large language model for tibetan culture. *arXiv e-prints*, page https://arxiv.org/html/2503.18288v2.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613.

Mike Kestemont, Michael Tschuggnall, Efstathios Stamatatos, Walter Daelemans, Günther Specht, Benno Stein, and Martin Potthast. 2018. Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection. In *Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al.*, pages 1–25.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Huidan Liu, Minghua Nuo, Longlong Ma, Jian Wu, and Yeping He. 2011. Tibetan word segmentation as syllable tagging using conditional random field. In *Proceedings of the 25th Pacific Asia conference on language, information and computation*, pages 168–177. Waseda University.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hui Lv, Chi Pu, La Duo, Yan Li, Qingguo Zhou, and Jun Shen. 2025. T-LLaMA: a tibetan large language model based on LLaMA2. *Complex & Intelligent Systems*, 11(1):72.

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

Leiyu Pan, Bojian Xiong, Lei Yang, Renren Jin, Shaowei Zhang, Yue Chen, Ling Shi, Jiang Zhou, Junru Wu, Zhen Wang, Jianxiang Peng, Juesi Xiao, Tianyu Dong, Zhuowen Han, Zhuo Chen, Yuqi Ren, and Deyi Xiong. 2025. Advancing large language models for tibetan with curated data and continual pre-training.

Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Qwen. 2024. Qwen2.5: A party of foundation models.

Lili Wang and Hongwu Yang. 2018. Tibetan word segmentation method based on bilstm_ crf model. In *2018 International Conference on Asian Language Processing (IALP)*, pages 297–302. IEEE.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Ziqing Yang, Zihang Xu, Yiming Cui, Baoxin Wang, Min Lin, Dayong Wu, and Zhigang Chen. 2022. CINO: A chinese minority pre-trained language model. *arXiv preprint arXiv:2202.13558*.

8. Appendix

This appendix provides additional details and examples that complement the main paper.

8.1. Tokenizer Training

The Tibetan tokenizer was trained from scratch using the `tokenizers` library's WordPiece model. The training corpus consisted of all cleaned `.jsonl` files produced after preprocessing, streamed line by line to avoid memory overhead. Tokenization relied on whitespace splitting. The model was trained with a target vocabulary size of 32,000 and the standard special tokens `[UNK]`, `[CLS]`, `[SEP]`, `[PAD]`, and `[MASK]`. Upon completion, the tokenizer was serialized to `tokenizer.json` and verified using example Wylie strings to confirm correct segmentation behavior. The cpt-mBERT uses 30.79% fewer tokens.

Tokenizer	Vocab Size	Fertility
mBERT (original)	119,547	2.07
Tibetan mBERT (ours)	32,000	1.43

Table 4: Comparison between the original and Tibetan-adapted tokenizers.

8.2. Data Collection and Processing for Pretraining

We compiled a corpus of e-texts of Classical Tibetan canonical, para-canonical, and non-canonical scriptures and non-scriptures. The data sources include publicly available repositories such as Spither Tibetan corpus (https://huggingface.co/datasets/spsither/tibetan_monolingual_A_

filtered_deduped), as well as in-house collections curated by our team of scholars (some of whom are co-authors). We also incorporated corpora that may contain limited amounts of modern Tibetan, such as the MC2 corpus (https://huggingface.co/datasets/pkupie/mc2_corpus). To ensure data quality, we applied document-level deduplication using the MinHash algorithm (Broder, 1997) combined with a Locality-Sensitive Hashing (LSH) index (Indyk and Motwani, 1998) for efficient similarity lookups. Documents with an estimated Jaccard similarity of ≥ 0.8 were considered duplicates and removed. Subsequently, we performed rule-based cleaning and normalization to remove comments (e.g., lines starting with %%), HTML artifacts, and long numerical sequences. We also applied paragraph-level filtering, discarding paragraphs with fewer than four tokens, excessively repetitive word patterns, or consecutive duplicates, issues that often result from OCR errors in digitized manuscripts. All preprocessing scripts are publicly available in the project's repository. After preprocessing, the final corpus comprises 13.8 GB of clean running text.

8.3. Transliteration Conversion

Our tool operates through a three-stage cascade: (1) Unicode Heuristic: identifies the dominant script family using Unicode code-point categories. (2) Statistical Profile Classifier: applies extended langdetect n-gram models trained on Sanskrit and Tibetan transliteration profiles. (3) Deterministic Fallback: uses curated regular-expression templates for Sanskrit schemes and structural stack patterns for Tibetan (validated through pyewts round-trip checks). It supports Sanskrit (Devanāgarī, IAST, SLP-1, Harvard–Kyoto, Velthuis) and Tibetan (Unicode, Wylie/EWTS, ACIP), with additional coverage for Modern Chinese (CJK) and English. The converter module unifies script and scheme representations using open-source Sanskrit and Tibetan libraries combined with custom normalization routines.

8.4. Hyperparameter Configuration

The continued pretraining was conducted using `bert-base-multilingual-cased` as the base model with a custom Tibetan tokenizer (vocabulary size: 32,000). Training used a batch size of 8 per device, sequence length of 512, learning rate of 2e−5, AdamW optimizer with weight decay of 0.01, warmup ratio of 0.06, FP16 mixed precision, gradient clipping at a max norm of 1.0, and a total of 3 epochs. The masked language modeling (MLM) probability was set to 0.15.
Fine-tuning followed a similar configuration, using a learning rate of 2e−5 with early stopping, AdamW optimizer with a weight decay of 0.01 and a warmup ratio of 0.1, mixed-precision (FP16), and gradient clipping at 1.0. The model was trained for 10 epochs with a batch size of 8.

8.5. Prompts

8.5.1. Segmentation Prompt For Generative LLMs

The following system and human prompts were used to predict ALLO/AUTO segmentation predictions.

```
You are a Tibetan Buddhist
philology expert and
computational linguist.
Your task is to read Tibetan text
and segment it into contiguous
spans of two types:
- AUTO (autochthonous Tibetan):
passages originally composed in
Tibetan.
- ALLO (allochthonous Tibetan):
passages translated into Tibetan
(typically from Sanskrit or
related Indic sources).

We are building a profile for
Tibetan text segmentation.
After segmenting, you must output
detailed reasoning inside the
JSON object under the key
"reasoning".
Do NOT include any explanations
or commentary outside the JSON.

Return ONLY a JSON object of the
form:
{
  "reasoning": "<short
explanation of why and where the
text switches>",
  "first_segment": "auto" |
"allo",
  "prediction": [i1, i2, ...]
}

Definitions:
- "reasoning" briefly explains in
2-3 sentences why you placed
those boundaries (e.g.,
indicators of translationese,
mantra markers, syntax shifts,
stylistic transitions, etc.).
- "first_segment" is the label of
the first span in the text:
"auto" or "allo".
- "prediction" is a list of
0-based WORD indices that mark
the start of a new segment after
a label change.
- If there is no switch, output:
{"reasoning": "No clear switch
detected.", "first_segment":
```

```
"auto"|"allo", "prediction":
[]}.

Word indexing rules:
- Normalize consecutive
whitespace to single spaces for
counting.
- The first word has index 0.
- Do not include index 0 in
prediction.
- Indices must be integers,
unique, strictly ascending, each
within [1, {total_tokens}].

Strict formatting:
- Output only valid JSON with all
three keys: "reasoning",
"first_segment", "prediction".
- Do not include any text or
commentary outside the JSON.
```

8.5.2. Reasoning Prompt for Explainability Effort

```
You are a Tibetan Buddhist
philology researcher and
computational linguist.
Your task is to analyze a batch
of reasoning traces from a
Tibetan segmentation model.

Each example includes:
- The original Tibetan text.
- The model's reasoning for its
segmentation decisions.
- The model's predicted switch
indices.
- The true (gold) switch indices.

You are not evaluating linguistic
content or segmentation quality
itself.
Instead, focus on the meta-logic
behind the reasoning - how the
model explains its decisions.

Your goals:
1. Observe recurring reasoning
patterns or motifs that appear
across examples.
2. Identify how these patterns
relate to prediction correctness
(consistent logic
vs. mismatches).
3. Assess the quality and
coherence of reasoning - whether
it is evidence-based, repetitive,
speculative, or inconsistent.

Return only valid JSON with the
following structure:
{
  "key_observations": "A short
paragraph describing recurring
reasoning patterns, logic
structures, and consistency
across examples. Highlight when
reasoning repeats, shifts, or
contradicts itself (use up to 10
examples).",
  "summary": "A concise global
evaluation of reasoning quality
across all samples - how
systematic, evidence-based, or
error-prone it is overall."
}

Formatting rules:
- Use double quotes, no trailing
commas.
- Every key must appear.
- Focus on reasoning patterns,
not segmentation details.

Human message:
Analyze the following {n_samples}
examples:

{entries}

Each example contains: Text,
Model Reasoning, Model
Prediction, and Gold Indices.
Study how the reasoning logic
behaves across examples, find
repeated patterns,
and summarize which reasoning
styles lead to correct
vs. incorrect segmentation.
Return only the JSON profile
object.
```

8.6. List of Annotated Source Files

bshes pa'i springs yig gi 'grel pa don gsal, Byang chub kyi sems bsgom pa'i rgyud, D805, Gpb052.015, rNam gsum bshad pa, S00271I, S05531E, S05533E-1, S05533E-2, S05534E, S05535E, S05536E, S05537E, S05538E, S05540E, S05541E, S05543E, S05544E, S05545E, S05546E, S05547E, S05548E, S05558E, S05559E, S05560E.

The texts were taken from BuddhaNexus,[14] rKTS,[15] and ACIP.[16]

8.7. Additional Segmentation Examples

8.7.1. Example: Passage from *ACIP Sungbum*

The following passage was examined using our ALTO + MUL + SAH model. The model's token-level predictions are visualized below, with autochthonous (auto) segments highlighted in yellow

[14] https://buddhanexus.kc-tbts.uni-hamburg.de/

[15] http://www.rkts.org/rktsneu/sub/index.php

[16] http://www.buddhism-dict.net/ebti/textinput/acip.html

and allochthonous (allo) segments highlighted in green.

> las dge ba chung du las kyang ’bras bu bde ba shin tu che ba ’byung la| las mi dge ba chung ngu las kyang ’bras bu sdug bsngal shin tu chen po ’byung bas nang gi rgyu ’bras kyi ’phel ’dra ba ni phyi rol gyi rgyu ’bras la med do / de yang tshoms las / sdig pa chung du byas pas kyang / ’jig rten pha rol ’jigs chen dang / phung krol chen po byed ’gyur te / khong par song ba’i dug bzhin no / bsod nams chung du byas pas kyang / ’jig rten pha rol bde chen ’dran / don chen dag kyang byed ’gyur te / ’bru rnams phun tshogs smin pa bzhin / zhes so / gsum pa ni / bde sdug myong ba’i rgyur gyur pa’i las ma bsags na las de’i ’bras bu bde sdug gtan mi myong ba ste / ston pas tshogs grangs med pa bsags pa’i ’bras bu la spyod pa rnams kyis kyang de’i rgyu thams cad bsag mi dgos kyang cha gcig bas dgos so / bzhi pa ni / las dge mi dge byas pa rnams kyis yid du ’ong mi ’ong gi ’bras bu ’byin pa ste / khyad par ’phags bstod las / bram ze rnams ni dge sdig dag / byin pa len bzhin ’pho bar smra / khyod kyis byas chud mi za dang / ma byas pa dang ’phrad med gsungs / zhes dang / ting nge ’dzin rgyal po las kyang / de yang byas nas rig par mi ’gyur min / gzhan gyis byas pa tsho bar ’gyur ba’ang med / ces dang / lung las kyang / las rnams bskal pa brgyar yang ni / chud mi za ba’ang tshogs dang dus /