Lemmatization of Cuneiform Languages Using the ByT5 Model

Pengxiu Lu[1,2], Yonglong Huang[1,2], Jing Xu[1,2], Minxuan Feng[1,2], Chao Xu[1,2]
[1]School of Chinese Language and Literature, Nanjing Normal University, China
[2]Center of Language Big Data and Computational Humanities, Nanjing Normal University, China
fomalhaut2001@gmail.com

Abstract

Lemmatization of cuneiform languages presents a unique challenge due to their complex writing system, which combines syllabic and logographic elements. In this study, we investigate the effectiveness of the ByT5 model in addressing this challenge by developing and evaluating a ByT5-based lemmatization system. Experimental results demonstrate that ByT5 outperforms mT5 in this task, achieving an accuracy of 80.55% on raw lemmas and 82.59% on generalized lemmas, where sense numbers are removed. These findings highlight the potential of ByT5 for lemmatizing cuneiform languages and provide useful insights for future work on ancient text lemmatization.

1 Introduction

Cuneiform writing systems, used by ancient Mesopotamian civilizations like the Sumerians and Akkadians, provide valuable insights into early human civilization. However, despite their historical significance, computational methods for processing cuneiform texts remain relatively underdeveloped. One of the key challenges in natural language processing (NLP) for these ancient languages is lemmatization — the task of reducing words to their base or dictionary forms—a process that is particularly complex due to the high degree of inflection, polysemy of signs, and extensive morphological variation characteristic of these languages.

Among the languages written in cuneiform, Akkadian and Sumerian are two of the most extensively documented, yet they pose distinct computational challenges. Akkadian, a Semitic language, exhibits root-based morphology with non-linear inflectional patterns, while Sumerian, a language isolate, follows an agglutinative structure with extensive prefixation and suffixation. Both languages also feature logographic and syllabic writing elements, further complicating automated linguistic analysis.

Among existing approaches, BabyLemmatizer (Sahala and Lindén, 2023) employs a neural encoder-decoder model to perform joint POS tagging and lemmatization, achieving 94–96% accuracy. Similarly, AkkParser (Ong and Gordin, 2024) combines rule-based morphological analysis, dictionary matching, and dependency parsing, providing robust performance on Neo-Assyrian texts. Despite their success, the variability in orthographic forms and the vast morphological richness of cuneiform languages still present challenges.

Recent advancements in transformer-based models, such as T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020), have significantly improved performance across a wide range of NLP tasks, including sequence-to-sequence applications like translation and lemmatization (Riemenschneider and Krahn, 2024). Building upon this foundation, ByT5 (Xue et al., 2022) was introduced as a variant of T5, designed to process text at the byte level. Unlike traditional token-based models, ByT5 operates directly on raw byte sequences, eliminating the need for predefined vocabularies and tokenization schemes. This token-free approach has proven advantageous in multilingual tasks such as grapheme-to-phoneme conversion (Zhu et al., 2022), where ByT5 has outperformed token-based models. Its architecture has also proven effective in lemmatization tasks—particularly for morphologically rich languages such as Latin (Wróbel and Nowak, 2022)—highlighting ByT5's ability to handle complex morphological variation with minimal preprocessing. Moreover, its strong zero-shot learning capabilities (Stankevičius et al., 2022) enable it to generalize to previously unseen languages, making it especially valuable for under-resourced historical languages such as Akkadian and Sumerian, where annotated corpora remain limited.

The aim of this study is to evaluate the effectiveness of ByT5 in lemmatizing Akkadian and

Proceedings of the Second Ancient Language Processing Workshop associated with NAACL 2025, pages 173–181

Sumerian texts, with a focus on assessing its ability to overcome the challenges posed by the morphological complexity and spelling irregularity of these ancient languages.

2 Methodology

2.1 Dataset

The original dataset consists of several fields, including: `fragment id`, `fragment line num`, `index in line`, `word language`, `domain`, `place discovery`, `place composition`, `value`, `clean value`, and `lemma`. The primary input to the model during training is the `clean value`, and the target is the `lemma`.

A particular challenge in this task arises from words that have multiple meanings or senses, a phenomenon particularly prominent in cuneiform lexicon. For example, the lemma *abāru* exhibits various senses, each with a specific definition in the *Concise Dictionary of Akkadian* (Black et al., 2000). The different senses of *abāru* are often marked with Roman numerals to denote the specific sense, as outlined below:

1. ***abāru I***: This sense refers to "(the metal) lead." It appears in texts such as A.GAR$_5$ and in 1st millennium royal inscriptions, specifically noted as A.BÁR. In Middle Assyrian, the phrase is also written as *annuku abāru*.

2. ***abāru II***: This sense has two distinct meanings:
   (a) Babylonian literary meaning: "A kind of clamp".
   (b) Standard Babylonian (Jungbabylonisch) transferred meaning: "embrace" or "physical strength", often used in reference to gods or kings.

3. ***abāru III***: This sense refers to "to embrace" in Old and Standard Babylonian. It is often used in magical contexts to mean "embrace intensely" or "bind" (e.g., limbs or persons). In legal contexts, it is used to mean "accuse someone" or "denounce".

Given this ambiguity, two distinct forms of the dataset are created to account for the different levels of semantic granularity.

- **Raw Lemma Dataset** retains sense numbers (e.g., *abāru I*) to capture semantic distinctions, as shown in Table 1.

Surface Form	Lemma
A.BAR$_2$	*abāru I*
a-ba-ri	*abāru II*
ub-bir	*abāru III*

Table 1: Examples from the Raw Lemma Dataset

- **Generalized Lemma Dataset** removes sense numbers for morphological normalization, as shown in Table 2.

Surface Form	Lemma
A.BAR$_2$	*abāru*
a-ba-ri	
ub-bir	

Table 2: Examples from the Generalized Lemma Dataset

2.2 Model Architecture

The primary model used for lemmatization of cuneiform languages is the ByT5 model, a variant of the T5 architecture that operates directly on the raw character sequences of texts at the byte level. ByT5 is built on a transformer-based architecture, where input sequences pass through multiple layers of attention mechanisms and feed-forward networks. It employs a standard encoder-decoder framework: the encoder processes the input text, while the decoder generates the corresponding output based on the encoded information.

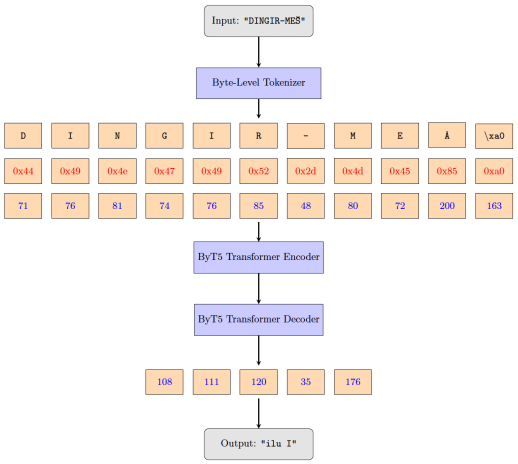

Figure 1: ByT5 Lemmatization Architecture with Byte-Level Tokenization

In this study, ByT5 is trained to map sequences of byte tokens to a sequence of output tokens,

where each output token corresponds to the canonical lemma (or generalized lemma) of the input word, as illustrated in Figure 1.

As an additional model for comparison, the mT5 model was also used. mT5 is a multilingual variant of T5, capable of processing text in multiple languages. mT5 also follows a transformer-based architecture, using word-level tokenization and is suited for handling multiple languages with varying scripts. For the purpose of this study, mT5 serves as a baseline model to evaluate how well ByT5 performs relative to a more traditional, multilingual approach.

2.3 Training Setup

Both models are trained using a standard sequence-to-sequence learning approach. For ByT5, the input sequence length is limited to 128 tokens, while for mT5, it is restricted to 32 tokens. The input text is prefixed with a task-specific indicator like "Convert:", following the approach inspired by the T5 model. The model's output is the predicted lemma, which can either be a raw lemma with sense numbering (for the Raw Lemma Dataset) or a generalized lemma (for the Generalized Lemma Dataset). Both datasets are split, with 95% used for training and the remaining 5% for validation.

We utilize pre-trained weights from the Hugging Face Transformers library and fine-tune the model on both datasets. The training process uses the Adam optimizer with a learning rate of 2e-5 and a batch size of 16. Models are fine-tuned for 10 epochs or until convergence. Training is conducted on an Apple M3 Pro (18GB) chip, leveraging the MPS backend for accelerated computation.

3 Experimental Results

3.1 Performance Metrics

To evaluate the effectiveness of the models, we used the following metrics:

Accuracy (Exact Match): This metric measures the percentage of instances where the predicted lemma exactly matches the target lemma.

$$Accuracy = \frac{\text{Number of Correct Lemma Predictions}}{\text{Total Number of Words}} \times 100\% \quad (1)$$

Accuracy serves as the primary metric for assessing the accuracy of the lemmatization process.

3.2 Results

The following tables present the performance of ByT5-small and mT5-small on the two datasets: one with raw lemmas (where sense numbers are retained) and another with generalized lemmas (where sense numbers are removed).

Model	Accuracy (%)
ByT5-small	80.55
mT5-small	77.38

Table 3: Performance on Raw Lemmas (Sense Number Retained)

Model	Accuracy (%)
ByT5-small	82.59
mT5-small	79.28

Table 4: Performance on Generalized Lemmas (Sense Number Removed)

4 Error Analysis

4.1 Challenges in Lemma Prediction

In this section, we conduct an error analysis based on the predictions made by the ByT5-small model on the raw lemma dataset, which consists of 39,621 unique word forms (transliterations) that have not been normalized and 8,021 unique lemmas, reflecting the diversity and complexity of the cuneiform lexicon. The model was trained on a dataset of 290,294 instances, learning to map surface forms to their corresponding lemmas. To assess its performance, we evaluated it on a validation set of 15,279 instances, where it produced 2,972 erroneous predictions, resulting in an overall accuracy of 80.55%. The detailed statistics of the dataset and its partitions are presented in Table 5 [1].

Dataset	Instances	Unique Word Forms	Unique Lemmas
Raw Lemma Dataset	305,573	39,621	8,021
Training Set (95%)	290,294	38,464	7,898
Validation Set (5%)	15,279	5,257	2,459

Table 5: Detailed Statistics of Raw Lemma Dataset

To better understand the sources of errors, we analyzed the incorrect predictions and categorized

[1] For conciseness, we will refer to the training set as TS and the validation set as VS in following tables.

them into three main groups: (1) surface forms that were most frequently mispredicted, (2) lemmas that were most frequently predicted incorrectly, and (3) erroneous lemma predictions that the model frequently produced. These insights highlight specific challenges in lemma disambiguation and the complex mappings required for accurate lemmatization.

Based on the validation set, the following table summarizes the five most frequently mispredicted surface forms, the five lemmas that were most commonly misclassified, and the five incorrect lemma predictions that the model frequently produced:

Category	Word/Lemma	Freq
Most frequently mispredicted surface forms	IGI	50
	NU	41
	ša	33
	KI	32
	BI	31
Most frequently misclassified lemmas	*amāru I*	40
	ul I	36
	ša	32
	ana	29
	šamšu I	27
Most frequently produced incorrect lemma predictions	*pānu I*	77
	lā I	52
	itti I	35
	ša I	33
	šū I	31

Table 6: Common Errors in Lemmatization

Building on the analysis above, we can identify three major challenges in the lemmatization process. First, polysemy poses a significant issue: without explicit syntactic or semantic context, the model struggles to accurately disambiguate multiple possible meanings of a given form. Second, inconsistencies in scribal conventions contribute to further complexity, leading to variability in representation. Third, the model exhibits a frequency bias, tending to over-predict high-frequency lemmas even in contexts where they are incorrect. These three challenges will be examined in detail in the following discussion.

4.1.1 Polysemy in Surface Forms

A major source of error in the ByT5-small model's predictions stems from the inherent polysemy in surface forms. Polysemy arises when a single surface form corresponds to multiple meanings or senses, each associated with a distinct lemma. Our analysis identified 2,194 surface forms exhibiting polysemy, accounting for a significant proportion of the dataset.

We observe that many of the most frequently mispredicted surface forms—IGI, NU, KI, BI—are Sumerograms, logographic signs borrowed from Sumerian into Akkadian. Unlike phonetic spellings, Sumerograms encode meaning rather than sound, making them particularly challenging for lemmatization. The interpretation of a single Sumerogram often depends on its contextual usage, as it can correspond to multiple lemmas. For instance, IGI can signify "eye" (*īnu I*) or "to see" (*amāru I*), among other meanings.

The semantic range of the Sumerogram IGI, as documented in the *Concise Dictionary of Akkadian*, along with their frequency distribution in the training dataset, is presented in the table below.

Sign	Lemma	Meaning	TS Freq
IGI	*pānu I*	face	772
	amāru I	to see	475
	mahru II	front	270
	naṭālu I	to look	79
	mahra I	in front; before; earlier	51
	īnu I	eye	41
	mahāru I	to face; oppose; receive	10
	pānātu I	front	7
	lapān I	in front of	5
	mahrum	/	4
	āmeru I	that sees, reads	2
	mehretu I	opposite side; front	2
	panû I	to face; be ahead	1
	nawāru I	to be(come) bright, shine	1

Table 7: Frequency and Semantic Range of Sumerogram IGI

Similar to IGI, the cuneiform logogram IM can correspond to four distinct lemmas: *ṭuppu I*, *šāru I*, *ṭīdu I*, and *ešēru I*. To illustrate this challenge, Table 8 presents the distribution of IM's lemmas in the training and validation sets and the model's

predictions. Despite the diverse occurrences of IM in the dataset, the model consistently predicted *ṭīdu I* across all instances, failing to account for the other possible lemmas.

Surface Form	**TS Count**	**TS Lemma Distribution**	
IM	226	ṭīdu I (57), ṭuppu I (66), šāru I (102), ešēru I (1)	
	VS Count	**VS Lemma Distribution**	
	16	ṭīdu I (3), ṭuppu I (9), šāru I (4)	**Prediction**: ṭīdu I (16/16)

Table 8: Distribution of IM's Lemmas in Training and Validation Sets vs. Model Prediction

The majority of most frequently misclassified lemmas and most commonly produced incorrect lemma predictions are closely associated with Sumerograms with multiple semantic variants. As shown in Table 6, *amāru I* and *pānu I* correspond to the Sumerogram IGI, *lā I* corresponds to NU, and *šū I* corresponds to BI. Moreover, these Sumerograms occur with high frequency in the training set, making them some of the most common lexical items (e.g., IGI: 1,720 occurrences; NU: 1,993 occurrences; BI: 1,352 occurrences). This high frequency, combined with their multiple semantic interpretations, constitutes a major source of prediction errors in the model.

Notably, the inclusion of texts from different historical periods and source traditions (as discussed in later sections) may further contribute to inconsistencies in lemmatization, as variations can arise due to differences in transcription conventions for cuneiform signs or historical shifts in the writing system. For example, the original form *ṭup-pi* can be lemmatized as *ṭuppi I*, *ṭuppu I*, or *ṭuppum*, depending on scribal practices. However, the lemma *ṭuppum* appears only twice in the training dataset and is more likely a morphological variant of *ṭuppu I* rather than a distinct lemma.

Overall, these challenges highlight the inherent complexities of cuneiform languages, where a single word form can have multiple interpretations depending on context or transcription conventions. Among all mispredictions, 1,253 errors were attributed to such one-to-many mappings. The model struggles to effectively disambiguate these cases, primarily due to its limited ability to capture the contextual cues that differentiate semantic variants. This issue is fundamentally rooted in the constraints of a simple sequence-to-sequence architecture, in which the model takes a surface form as input and generates a single corresponding lemma as output. Hence, lacking the capacity to incorporate broader contextual information necessary for disambiguation makes the existing model architecture inadequate for handling one-to-many mappings, which eventually leads to frequent misclassifications.

4.1.2 Orthographic Variation in Lemmas

As previously noted, *ṭuppu I* as a lemma may be reconstructed from multiple surface forms, such as *ṭup-pi* or IM, illustrating the intricate mapping between surface forms and lemmas. A single surface form may correspond to multiple lemmas, while a single lemma may also be associated with multiple surface forms (although the latter does not introduce ambiguity in one-to-one lemmatization processes).

Therefore, in addition to polysemy, orthographic variation presents another challenge, wherein a single lemma can be represented by multiple surface forms. Our analysis of the raw lemma dataset revealed that 4,865 lemmas—comprising 60.65% of the total—are associated with multiple surface forms, indicating a significant presence of spelling variants. Noticeably, among the 2,972 lemmas incorrectly predicted by the model (i.e., the lemmas that the model erroneously generated rather than the correct lemmas that were misidentified), 2,788 errors were traced to these orthographic variations. This finding suggests that a substantial proportion of mispredictions can be attributed to the model's inclination to favor frequently occurring variants, likely due to the disproportionate representation of such cases in the training data. A clear example is *pānu I*, which exhibits significant spelling variation and is frequently mispredicted by the model.

Another illustrative case is the lemma *šapârum*, which corresponds to 92 distinct word forms, many of which exhibit considerable morphological complexity and subtle variations (e.g., *ši-ta-ap-pa-ra-am*, *iš-pu-ra-am*, *šu-up-ra-nim*, *áš-tap-ra*, *li-iš-ur-ma*, etc.). While the model successfully predicts the lemma in the majority of cases, it occasionally produces an entirely nonexistent lemma. For

instance, for "ša-ap-pa-ra-ak-kum", the model incorrectly generates "*šapparakkum*"—a form unattested in the dataset. This pattern of errors further underscores the model's difficulty in distinguishing between legitimate orthographic variants and erroneous extrapolations, ultimately complicating the lemmatization process.

4.1.3 Frequency Effects in Lemma Prediction

An important consideration in the model's performance is the effect of lemma frequency on prediction accuracy. In the dataset, some lemmas appear far more frequently than others, creating a potential imbalance in the model's learning process. To systematically analyze this, we classified low-frequency lemmas as those appearing at most once (Q1 = 1.0), mid-frequency lemmas as those appearing between Q1 and Q3 (2 to 12 times), and high-frequency lemmas as those appearing 13 times or more (Q3 = 13.0).

Our analysis revealed that 2,349 errors (79.0% of the total errors) were made by the model on high-frequency lemmas, 529 errors (17.8%) on mid-frequency lemmas, and 94 errors (3.2%) on low-frequency lemmas. The relatively high number of errors on high-frequency lemmas suggests that, despite their prominence in the training data, these words still present challenges for the model. This can be attributed to the polysemy and orthographic variation issues discussed above, where the model's familiarity with a lemma's high-frequency forms does not guarantee its ability to handle less common senses or spelling variants. On the other hand, low-frequency lemmas, while less problematic in terms of sheer error counts, may be underrepresented in the training data, leading to occasional mispredictions when these lemmas do appear in the validation set. For instance, in the training corpus (comprising 290,294 instances), there were only nine occurrences of the surface form "*im*", mapping to seven distinct lemmas: *ṣâbum* (2 instances), *ne'rârum* (2 instances), *epêšum* (1 instance), *eqlum* (1 instance), *šapârum* (1 instance), *âlum* (1 instance), and *makârum* (1 instance). Given the extremely limited number of training examples, the model struggled to learn the correct mappings, ultimately producing an erroneous output (e.g., *ṭuppu I*). The lack of sufficient representation of variant forms in the training data makes it even more difficult for the model to generalize accurately.

These findings highlight the impact of data imbalance, where the model's performance is skewed toward frequently occurring lemmas while remaining less reliable on rarer ones.

4.2 Comparative Evaluation on Archibab and eBL Corpora

As part of our error analysis, we conducted an additional evaluation by dividing the validation set into two subsets based on their sources: Archibab[2] and the Electronic Babylonian Library (eBL)[3]. This allowed us to assess the model's performance separately on texts from distinct historical periods and linguistic traditions, providing further insights into its strengths and limitations. The division was necessary due to significant differences between these two corpora. Archibab consists of Old Babylonian texts from the early second millennium BCE, primarily legal, administrative, and epistolary documents. These texts adhere to lemmatization conventions shaped by their historical and linguistic context. In contrast, eBL comprises first-millennium BCE literary and scholarly texts, which reflect later linguistic developments and more standardized scribal practices. With approximately 1000 years separating these corpora, their divergent lemmatization practices posed unique challenges for the model.

To conduct this evaluation, we refined the dataset by further splitting the training and validation sets accordingly, after which we obtained the following distribution of instances, as shown in Table 9. Notably, the Archibab dataset does not include sense numbers in its lemmatization annotations, which may influence the ability of certain models to handle this subset effectively.

Source	Training Set	Validation Set
eBL	292,423	14,619
Archibab	13,150	660

Table 9: Data Distribution across Different Sources

By evaluating performance on each subset independently, we aimed to determine whether the model could generalize across different stages of cuneiform languages or whether it showed biases toward a particular linguistic tradition. Specifically, we evaluated models trained on two different datasets: the *Raw Lemma Dataset* and the *Generalized Lemma Dataset*. The results are summarized in Table 10 and Table 11.

[2] https://www.archibab.fr/home

[3] https://www.ebl.lmu.de

Model	Dataset	Accuracy (%)
ByT5 (Raw Lemma)	eBL	82.39
ByT5 (Raw Lemma)	Archibab	39.85
mT5 (Raw Lemma)	eBL	79.30
mT5 (Raw Lemma)	Archibab	34.85

Table 10: Performance of models trained on the Raw Lemma Dataset.

Model	Dataset	Accuracy (%)
ByT5 (Generalized Lemma)	eBL	83.80
ByT5 (Generalized Lemma)	Archibab	55.76
mT5 (Generalized Lemma)	eBL	80.60
mT5 (Generalized Lemma)	Archibab	50.00

Table 11: Performance of models trained on the Generalized Lemma Dataset.

Across all models, lemmatization accuracy on the eBL dataset was significantly higher than on the Archibab dataset. This discrepancy can largely be attributed to the imbalance in training data, where eBL data greatly outnumbered Archibab data (292,423 vs. 13,150 instances, a ratio of approximately 22.2:1). This imbalance likely led the model to develop a stronger bias toward the linguistic patterns found in eBL, resulting in higher accuracy for that subset.

Furthermore, models trained on the Raw Lemma Dataset exhibited particularly low performance on Archibab data. This is likely because these models were trained to predict sense numbers, whereas the Archibab dataset lacks sense-number annotations. As a result, the models trained on Raw Lemma data tended to incorrectly assign sense numbers when lemmatizing Archibab instances, leading to a notable decrease in accuracy. In contrast, models trained on the Generalized Lemma Dataset showed higher accuracy on Archibab, as they were explicitly trained to generalize across datasets without relying on sense-number distinctions. This suggests that generalizing lemma annotations can help improve model performance when dealing with corpora that follow different lemmatization conventions.

5 Conclusion

The results from our experiments demonstrate that the **ByT5-small** model outperforms **mT5-small** in accuracy across both generalized and raw lemmatization tasks. Results also indicate that predicting raw lemmas (including sense numbers) is more challenging than predicting generalized lemmas, which is reflected in the lower accuracy scores for the raw lemma dataset, suggesting that incorporating sense numbers adds a layer of complexity to the task.

The effectiveness of ByT5's byte-level tokenization is particularly evident in Akkadian and Sumerian lemmatization, as it eliminates the need for complex, language-specific tokenization strategies that traditionally require specialized cuneiform expertise. In previous approaches to processing these ancient languages, pre-tokenization often relied on in-depth linguistic knowledge, such as the logo-syllabic tokenization employed by BabyLemmatizer[4]—a process tailored to the structure of cuneiform writing systems. In contrast, ByT5 leverages a byte-level vocabulary of only 256 basic tokens, enabling it to represent all cuneiform symbols and their transliterations without additional tokenization preprocessing.

This is particularly beneficial for Akkadian and Sumerian transliterations, which often include diacritics (e.g., š, ṭ), subscript numerals (e.g., $_2$ and $_3$, to distinguish between homophones or different readings of the same cuneiform sign), determinatives (e.g., $\{d\}$) and special notations for broken or uncertain readings (e.g., *?*). ByT5's ability to handle these symbols directly allows for a simpler yet effective architecture that achieves competitive performance without relying on intricate domain-specific tokenization rules. This suggests that byte-level models can possibly serve as a more accessible and adaptable approach to lemmatization in low-resource, complex linguistic settings, reducing

[4] https://github.com/asahala/BabyLemmatizer

dependence on specialized cuneiform processing techniques.

We acknowledge that a key limitation of our experiment is the lack of contextual integration. Without leveraging broader contextual information, further performance improvements are impossible, particularly in distinguishing sense variations. Future work could explore incorporating sentence- or discourse-level context, as ByT5 with contextual awareness might yield interesting results and further enhance lemmatization accuracy. Additionally, expanding the training data and refining the lemmatization pipeline may further improve performance, particularly for datasets with sparse annotations like Archibab.

Acknowledgments

This research was supported by the National Language Commission Project of China (YB145-41), the National Social Science Funds of China (21&ZD331, 22&ZD262), and the project *Research on Graded Reading of Children's Chinese Classics*, funded by the Shenzhen IREAD Foundation. We are grateful to the reviewers for comments which helped us to improve the paper.

References

Jeremy Black, Andrew George, and Nicholas Postgate. 2000. *A Concise Dictionary of Akkadian*, volume 5. Otto Harrassowitz Verlag.

Dominique Charpin. 2014. The assyriologist and the computer. the "archibab" project. *Hebrew Bible and Ancient Israel*, 3(1):137–153.

Evelien de Graaf, Silvia Stopponi, Jasper Bos, Saskia Peels-Matthey, and Malvina Nissim. 2022. Agile: The first lemmatizer for ancient greek inscriptions. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC)*, pages 5334–5344. European Language Resources Association (ELRA).

John L. Hayes. 2019. *A Manual of Sumerian Grammar and Texts*, 3rd revised ed. edition. Undena Publications.

Abraham Hendrik Jagersma. 2010. *A Descriptive Grammar of Sumerian*. Ph.d. dissertation, Leiden University.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. 2024. Myte: Morphology-driven byte encoding for better and fairer multilingual language modeling. *arXiv preprint*.

Matthew Ong and Shai Gordin. 2024. Linguistic annotation of cuneiform texts using treebanks and deep learning. *Digital Scholarship in the Humanities*, 39(1):296–307.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Frederick Riemenschneider and Kevin Krahn. 2024. Heidelberg-boston @ sigtyp 2024 shared task: Enhancing low-resource language analysis with character-aware hierarchical transformers. In *Proceedings of the 6th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 131–141. Association for Computational Linguistics.

Aleksi Sahala, Tero Alstola, Jonathan Valk, and Krister Lindén. 2022. Babylemmatizer: A lemmatizer and pos-tagger for akkadian. In *Proceedings of the CLARIN Annual Conference*, pages 14–18. CLARIN ERIC.

Aleksi Sahala and Krister Lindén. 2023. Babylemmatizer 2.0–a neural pipeline for pos-tagging and lemmatizing cuneiform languages. In *Proceedings of the Workshop on Ancient Language Processing*, pages 203–212. INCOMA.

Aleksi Sahala, Miikka Silfverberg, Antti Arppe, and Krister Lindén. 2020. Babyfst: Towards a finite-state based computational model of ancient babylonian. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3886–3894. European Language Resources Association (ELRA).

Lukas Stankevičius, Mantas Lukoševičius, Jurgita Kapočiūtė-Dzikienė, Monica Briedienė, and Tomas Krilavičius. 2022. Correcting diacritics and typos with a byt5 transformer model. *Applied Sciences*, 12(5):2636.

Chahan Vidal-Gorène and Bastien Kindt. 2020. Lemmatization and pos-tagging process by using joint learning approach: Experimental results on classical armenian, old georgian, and syriac. In *Proceedings of LT4HALA 2020–1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 22–27.

Wolfram von Soden. 1969. *Grundriss der Akkadischen Grammatik*, 3rd ed. edition. Pontificium Institutum Biblicum.

Krzysztof Wróbel and Krzysztof Nowak. 2022. Transformer-based part-of-speech tagging and lemmatization for latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 193–197. European Language Resources Association.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Byt5 model for massively multilingual grapheme-to-phoneme conversion. *arXiv preprint*.