# EvaCun 2025 Shared Task: Lemmatization and Token Prediction in Akkadian and Sumerian using LLMs

**Shai Gordin**
Ariel University
Land of Israel
and Archaeology
shaigo@ariel.ac.il

**Aleksi Sahala**
University of Helsinki
Helsinki, Finland
aleksi.sahala@
helsinki.fi

**Shahar Spencer**
The Hebrew University
of Jerusalem
shahar.spencer@
mail.huji.ac.il

**Stav Klein**
Ariel University
Land of Israel
and Archaeology
Stav.Klein@
msmail.ariel.ac.il

## Abstract

The EvaCun 2025 Shared Task, organized as part of ALP 2025 workshop and co-located with NAACL 2025, explores how Large Language Models (LLMs) and transformer-based models can be used to improve lemmatization and token prediction tasks for low-resource ancient cuneiform texts. This year our datasets focused on the best attested ancient Near Eastern languages written in cuneiform, namely, Akkadian and Sumerian texts. However, we utilized the availability of datasets never before used on scale in NLP tasks, primarily first millennium literature (i.e. "Canonical") provided by the Electronic Babylonian Library (eBL), and Old Babylonian letters and archival texts, provided by Archibab. We aim to encourage the development of new computational methods to better analyze and reconstruct cuneiform inscriptions, pushing NLP forward for ancient and low-resource languages. Three teams competed for the lemmatization subtask and one for the token prediction subtask. Each subtask was evaluated alongside a baseline model, provided by the organizers.

## 1 Introduction

Natural Language Processing for low-resource languages presents unique challenges, especially in an era where bigger models and more data are seen as the key to success. Ancient languages before the spread of the alphabet in the first millennium BCE, were primarily morphosyllabic, written using a combination of logograms (i.e. "word" signs) and syllabic signs (Fedorova; Daniels, 2023). Cuneiform in particular was used to encode more than a dozen languages across Western Asia, from languages of unknown or limited familial origin, like Sumerian or Hurrian, to several Semitic and Indo-European, languages, like Akkadian, Hittite, and Luwian.

Ancient Language Processing deals primarily with solving the challenges of the computational analysis of ancient morphosyllabic scripts, like the pictographic nature of signs, their iconically meaningful and complex visual arrangement, and lexical homonymy to name a few (Gordin, 2014; Gabriel et al., 2021). Some languages, particularly Semitic ones, are even more difficult due to their rich morphology, which leads to complex word forms and intricate grammatical structures (Weninger et al., 2011; Zitouni, 2014). Additionally, ancient languages often suffer from fragmented texts because the sources we rely on—inscriptions, manuscripts, and other historical records—are incomplete due to damage, erosion, and loss over time. These challenges make two key downstream NLP tasks, token prediction (used, for example, in BERT pretraining) and lemmatization, particularly difficult. To address this, we introduce a shared task with two subtasks: lemmatization, which reduces words to their base forms, and token prediction, which predicts the original token replaced with a mask.

The lemmatization and token prediction tasks for EvaCun 2025 focus on Akkadian and Sumerian cuneiform texts. Even though cuneiform was used to write on clay for more than 3,000 years, many cuneiform languages are low-resource languages. Existing corpora of texts consist of a relatively limited amount of data for each historical period of cuneiform, which is moreover divided into different geographical areas, archaeological contexts, and text genres.[1]

Existing language models have relied mostly on the tens of thousands of first millennium BCE Assyrian and Babylonian archival documents and royal inscriptions from the *Open Richly Annotated Cuneiform Corpus* (ORACC) (Gordin et al., 2020; Lazar et al., 2021; Gutherz et al., 2023), as well as the many thousands of sporadic Akkadian and Sumerian sources on the *Cuneiform Digital Library*

[1] For a good textual and linguistic overview of Akkadian and its periodization see (Vita, 2021)

*Proceedings of the Second Ancient Language Processing Workshop associated with NAACL 2025*, pages 164–172
May 4, 2025. ©2025 Association for Computational Linguistics

*Initiative* (CDLI) (Pagé-Perron et al., 2017; Chen et al., 2023). We therefore wanted to introduce new text genres and large scale corpora that have become systematically available over recent decades in the *Electronic Babylonian Library* (eBL) and *Archibab*. For more details on the content and genre of the text in the dataset provided for the shared task see Data section below.

## 2 Previous Research

The origins of Computational Assyriology can be traced back to the 1960s, and since then over 200 relevant papers have been published. Almost all aspects of Assyriological research were experimented with computationally, from artifact reconstruction to transliteration of cuneiform, text annotation, and content analysis. In this section, we briefly summarize past attempts on cuneiform tablet reconstruction and lemmatization of Akkadian. For a more detailed survey on the history of Computational Assyriology see Sahala (2021), on vision related tasks for cuneiform see Bogacz and Mara (2022), and for Assyriological digital resources see Charpin (2014), and the DANES resources on the OpenDANES platform.

### 2.1 Lemmatization

Traditionally Akkadian lemmatizers have been based on dictionary look-ups or morphological analysis. The first[2] published lemmatizer and morphological analyzer of Akkadian was implemented by Kataja and Koskenniemi (1988), but this system was more of a tech-demo to demonstrate, how discontinuative morphology could be implemented as a finite-state grammar (FSG). Further morphology-based models were published for Babylonian by Barthélemy (1998), Macks (2002), Sahala (2014) and Sahala et al. (2020b), and for Old Assyrian by Bamman (2012). To date, the most used lemmatizer for Akkadian, and cuneiform languages in general, is L2 (Tinney, 2019). L2 is a dictionary based lemmatizer, transcriber and POS-tagger that uses bigram look-up for disambiguation. It has been used to annotate ORACC texts, one of the largest open collection of annotated cuneiform texts.

Both, dictionary and morphology based lemmatizers have their shortcomings, which ultimately emerge from the Akkadian spelling variation and discontinuative morphology. Dictionary-based models suffer from spelling-variation and morphology induced out-of-vocabulary words (OOV) that they are unable to lemmatize. Morphology-based models, on the other hand, suffer from the ambiguity and irregularity of the Akkadian writing system, especially concerning the spelling of phoneme quantities. For this reason, morphology-based models rely almost exclusively on phonologically transcribed inputs, which limit their usability, since most unannotated digitized texts exist in transliteration. The only exception to this is Bamman's Old Assyrian morphological analyzer, which uses a brute-force approach to map between transliteration and transcription.

Treatment of discontinuative morphology was a long-standing challenge in Natural Language Processing, since it could not be elegantly expressed with FSGs. Over time, various extensions were introduced to FSGs, such as the compile-replace algorithm, flag diacritics, memory registers and multi-tape automata (Cohen-Sygal and Wintner, 2006), and after the memory requirements allowed it, some implementations relied on linearizing the morphology with procedural pregeneration. Yet, whereas the state-of-the-art analyzers for morphologically concatenative languages had been dominated by FSGs since the 1980s, still in the 2000s, some state-of-the-art computational models of discontinuative morphologies were implemented procedurally, such as Buckwalter (2002) for Arabic.

During the last decade, neural sequence-to-sequence models have opened new avenues for lemmatizing languages (Bergmanis and Goldwater, 2018; Kanerva et al., 2018). These models have introduced promising ways to deal with complex orthographies and morphologies, as well as synchronic and diachronic variation, like those found in Akkadian. Training neural models for lemmatizing Akkadian has been largely possible only due to Oracc's open data policy and the invaluable effort of dozens of Assyriologists, who have contributed their data to Oracc and annotated it semi-automatically using Tinney's L2.

The first neural network based attempt to linguistically annotate Akkadian was Sahala et al. (2020a). This system phonologically transcribed Akkadian using sequence-to-sequence models feeding the output into a finite-state transducer to produce lem-

[2] Giorgio Buccellati built tools for Akkadian already in the 1970s but to our knowledge these have not been published Buccellati (1977); for the goals of his project see the website of Cybernetica Mesopotamica. Tools were also created for the Neo-Assyrian Text Corpus Project by Simo Parpola and Robert M. Whiting. Their dictionary-based lemmatizer, however, remains also unpublished.

mata, POS-tags and morphological labels. This approach suffered from morphological ambiguity and the lemmatization pipeline was later simplified into BabyLemmatizer (Sahala et al., 2022), which predicted the lemmata directly from transliteration without intermediate steps. Another successful Akkadian neural network-based lemmatizer was published by Ong and Gordin (2024), who developed AkkParser, a language model implemented within the spaCy framework, with customized pipeline components for morphological analysis and syntactic dependency parsing specifically adapted to Akkadian cuneiform texts. This model was trained through an iterative bootstrapping methodology on a treebank of Neo-Assyrian letters, with human annotators providing corrections to progressively improve performance across annotation cycles. The only model so far specifically trained to annotate lemmas in Old Babylonian is Smidt et al. (2024), who conducted experiments on Part-of-Speech tagging for Old Babylonian letters using the Flair toolkit, finding that Multilingual BERT Transformer-based embeddings achieved good accuracy, despite working with a limited training corpus.

### 2.2 Token Prediction

Clay tablets, the medium on which the texts of ancient mesopotamia were written, are often found in fragmentery condition, causing a sigificant potential loss of text (Fetaya et al., 2020). Work has been conducted to collate 3D-scanned cuneiform tablet fragments by using join-surface heatmaps (Collins et al., 2014) and script feature analysis (Cammarosano, 2014; Fisseler, 2019). Systems for joining disconnected transliterated fragments have also been implemented (Tyndall, 2012; Simonjetz et al., 2024).

Token prediction differs fundamentally from these reconstruction approaches, as it aims to infer the content of missing text rather than identifying fragment matches in a database. Although relatively underexplored, some studies have employed machine learning models to reconstruct missing sign sequences in cuneiform texts. In Fetaya et al. (2020), RNN models are used to predict missing tokens. Another study, by Lazar et al. (2021), frames the problem using a masked language modeling approach similar to BERT pretraining, leveraging multilingual training with BERT-based models. This task is also popular in works on other ancient languages, see Sommerschield et al. (2023).

## 3 Data

Statistics for the shared task datasets are provided in tables 1, 2 and 3. Our data comes from two primary sources: the Electronic Babylonian Library Dataset (eBL) (Cobanoglu et al., 2024) and Archibab. The eBL data is drawn from transliterated cuneiform tablets via the eBL API provided by Enrique Jiménez on Nov. 2024; an earlier image of the data is also published on Zenodo (Cobanoglu, 2023). The data used is novel for NLP purposes as it focuses on a considerable number of literary texts. Although like ORACC texts it is also dated to the first millennium BCE, the eBL corpus make up very different kinds of literary and scientific genres subsumed here under the term canonical, using the accepted terminology of Hallo (1991). Archibab texts, on the other hand, primarily consist of Old Babylonian archival documents from the early second millennium BCE (2004–1595 BCE), of which a subset mostly made up of letters was provided for the shared task with the kind permission of Dominique Charpin (Collège de France) and Marine Béranger (FU Berlin). Where more metadata was provided in the dataset itself, as in the case of Archibab texts, or available via the eBL API, we included information about genre, find location, and language. To avoid potential bias, we replaced tablet IDs with randomized numbers. Additionally, any words that were entirely missing from the texts were removed.

| Dataset | Split | Fragments | Unique Values |
|---|---|---|---|
| Lemmatization | Total (Train + Test) | 10,214 | 46,966 |
| | Train | 8,171 | 40,640 |
| | Test | 2,043 | 15,539 |
| Token Prediction | Total (Train + Test) | 28,472 | 118,550 |
| | Train | 22,777 | 102,639 |
| | Test | 5,695 | 38,825 |

Table 1: Statistics for Lemmatization and Token Prediction datasets.

| Dataset | Category | Details |
|---|---|---|
| Lemmatization | Akkadian | 377,000 |
| | Sumerian | 51 |
| | Emesal | 2 |
| | test function words | 17,686 / 73,357 samples |
| | test OOV | 7,379 / 73,357 samples |
| Token Prediction | Akkadian | 970,237 |
| | Sumerian | 130,596 |
| | Emesal | 33,237 |
| | test function words | 3,826 / 44,517 samples |
| | test OOV | 4,161 / 44,517 samples |

Table 2: Breakdown for Lemmatization and Token Prediction datasets. Language is reported per word in the dataset as tablets may have words in multiple languages.

| Dataset | Genre | Count |
|---|---|---|
| Lemmatization | Canonical | 4659 |
| | Unclassified | 4217 |
| | Archival | 869 |
| | Administrative letter | 242 |
| | Monumental | 119 |
| | Political letter | 72 |
| | Other | 26 |
| | Private letter | 8 |
| | Diplomatic letter | 2 |
| Token Prediction | Canonical | 11332 |
| | Unclassified | 10994 |
| | Archival | 2940 |
| | Other | 2321 |
| | Monumental | 344 |
| | Administrative letter | 276 |
| | Political letter | 242 |
| | Private letter | 13 |
| | Diplomatic letter | 8 |

Table 3: Lemmatization and Token Prediction Genre Distribution.

Each dataset was split into training and testing sets, with 80 percent of the tablets allocated to the training split, which was provided to participants in the first step. The remaining 20 percent were used for evaluation, and all results are based on this held-out test set. It is worth noting that the two datasets had slight differences in transliteration conventions: this is taken into account during evaluation, as detailed below.

### 3.1 Lemmatization Data

For the lemmatization data, we applied cleaning steps to ensure consistency and usability. If a word in a given context had several possible lemmatic interpretations, we kept only the first lemma from that list. Any words that lacked a corresponding lemma were filtered out, ensuring that all remaining tokens in the dataset had a valid lemmatized form.

### 3.2 Token Prediction Data

For the word completion task, we focused on removing noise and ensuring that only complete, readable words were included. We excluded any words that contained fragmentary markers (such as "...", "[", "]", "x", "X", or "?"), as well as any numbers. Additionally, we cleaned the "value" column by removing non-alphabetical characters like < and #, which are additional editorial marks. We masked 20 percent of the data in each of the splits.

## 4 Shared Task

The task was structured to ensure consistency and transparency in assessing the performance of the lemmatization and token prediction models. Participants submitted both their generated predictions for the test set and technical reports through the SoftConf system. The train datasets provided contained pre-processed cuneiform texts, ensuring all participants worked with the same linguistic resources without modifications. While no strict measures were in place to prevent fine-tuning on the test set, the competition relied on participant integrity to avoid unfair data contamination. The evaluation compared submitted predictions against a held-out test dataset, with participants encouraged to document their methodologies in detail in the technical report, which were reviewed by the organizers. To promote replicability, participants were expected to share their scripts, system source code, and, where possible, trained models on platforms such as GitHub and Hugging Face.

## 5 Evaluation Metrics

We use accuracy as our primary evaluation measure in both, lemmatization and token prediction tasks, that is, the percentage of valid predictions over the whole evaluation category.

We structured our results according to distinct categories to assess performance across different linguistic phenomena:

Function vs. non-function words: Function words (e.g., conjunctions, prepositions) typically have high-frequency, well-attested forms, while non-function words (e.g., nouns, verbs) exhibit more variation and complexity.

In-vocabulary (in-vocab) vs. out-of-vocabulary (OOV) words: In-vocab words appear in the training data, while OOV words do not. OOV performance is particularly important for evaluating a model's generalization ability.

### 5.1 Flexible Matching

We considered predictions valid within the range of certain flexibility to prevent false negatives affecting the lemmatization evaluation results.

Firstly, as the eBL dataset contains Roman numerals indicating homonyms, while the Archibab dataset does not, we removed all numerals from both datasets and from the predictions for both tasks to ensure consistency.

For the evaluation of the lemmatization task, we aimed to allow variations that arise due to differences between the datasets, since dialect or chronolect identification was not part of the task list, and in reality separate models should be trained for different domain s for maximum efficiency. To achieve this, we implemented two steps. First, we wrote a harmonization function that standardizes most lemmatization conventions across datasets. For example, we unified macrons and circumflexes (e.g., *parāsu* vs. *parâsum*; *Anunnakū* vs. *Anunnakû*), made mimation optional (*šarru* vs. *šarrum*) and considered dictionary forms with and without initial waw equivalent (*alādu* vs. *walādu*). Second, we curated a special list of ca. 200 additional lemmatization variants to ensure that reasonable spelling differences did not unfairly impact accuracy. This list handles variation such as *nuhatimmum* vs. *nuhtimmu*. Naturally, all insonsistencies could not be handled, but the implemented rules covered most of the cases where the evaluation could have probably given false negatives. This harmonization was only ran at the evaluation phase when the predictions and the gold standards were matched with each other. Therefore, all models had to deal with the same inconsistencies in the training phase.

For the evaluation of the token prediction task, we focused on exact matches. However, we acknowledge that multiple valid completions can exist. For example, different experts might propose different reconstructions for the same missing segment based on contextual interpretation. Future work should incorporate methods to allow for semantic flexibility in evaluation.

## 6 Baseline Systems

To compare the shared task results with the existing publicly available systems, we used two baselines in lemmatizer evaluation and one baseline for token prediction evaluation

### 6.1 Maximum Likelihood Estimator

Our first baseline lemmatizer is an MLE dictionary look-up that assigns each word form with its most common lemma found in the training data. This simulates the simplest possible lemmatizer for a language and gives an estimate how well the more sophisticated models can handle ambiguity.

### 6.2 BabyLemmatizer 2.2

Our second baseline is BabyLemmatizer 2.2, a hybrid state-of-the-art annotation pipeline that combines the strengths of neural networks and shallow context-aware dictionary look-ups. Previously it has been used for lemmatizing several languages, such as Egyptian, Coptic, Demotic (Sahala and Lincke, 2024), Akkadian, Sumerian, Urartian, Greek and Latin (Sahala and Lindén, 2023). Evaluations have shown an accuracy ranging from 82% to 98% depending on the script and language. In Akkadian lemmatization the reported accuracy is ca. 95% using in-domain training data.

BabyLemmatizer treats lemmatization as a machine translation task. Its neural network architecture comprises a two layer BiLSTM encoder for reading the input sequence, and a unidirectional LSTM decoder with input feeding attention for generating the output. The neural network's output is then validated, corrected and confidence-scored with a heuristic dictionary look-up.

For all BabyLemmatizer models, we split the given training dataset into chunks of ten fragments each, which of we always take the first eight as our training data and the remaining two as development data, yielding 80/20 training/development split.

#### 6.2.1 Lemmatizer Model

Since the dataset used in the shared task does not contain part-of-speech (POS) labels and BabyLemmatizer relies on them for lemma disambiguation, we train two separate models for lemmatization and disambiguation and use them in tandem.

The initial **context-blind model** lemmatizes words without their sentence contexts and estimates their ambiguity using BabyLemmatizer's built-in confidence scoring system. The **disambiguation model** then attempts to correct the low-confidence lemmata by observing their contexts (in transliteration) using a symmetric window of three words. Both models use the default logo-phonemic tokenization that treats logograms and determinatives as indivisible symbols, and syllabograms and phonetic complements as divisible phoneme sequences. This setting collapses homonymous syllabic signs such as **ša** and **ša**$_2$ together but keeps logograms such as **DU** and **DU**$_3$ separate, since their meanings and readings are generally unrelated. The lemmatizer model works only on the word level and does not take the fragment metadata into consideration.

The advantage of the dual-model approach is marginal, providing only ca. 1% increase in lemmatization accuracy in comparison to using either of the sub-models alone.

#### 6.2.2 Token Predictor Model

For the token prediction task we train two models, the basic model and an augmented one. We train BabyLemmatizer similarly to the lemma disambiguation model, but instead of predicting the lemma we predict transliteration for each masked word based on its surrounding context with a symmetric window of three words. We segment the input using BabyLemmatizer's logo-syllabic tokenizer using translitered signs as minimal units, and generate the output sequence similarly. The token prediction model does not take into account the language or genre metadata and relies purely on sign-to-sign relations.

The augmented model is trained in the same manner, but the training data is concatenated with itself for 15 times before the train/dev split. The masked words are then randomized in a way that 15% of the total words are masked. Motivation for this additional model was to provide a more comparable baseline with the team 32's model that used the same augmentation approach.

## 7 Results

Three teams competed for the lemmatization task, and one for the word prediction task. The model numbers refer to submissions in this volume - submission 29 is "Lemmatization of Cuneiform Languages Using the ByT5 Model", submission 33 is "Beyond Base Predictors: Using LLMs to Resolve Ambiguities in Akkadian Lemmatization" and submission 53 is "A Low-Shot Prompting Approach to Lemmatization in the EvaCun 2025 Shared Task". for the token prediction task, submission 32 is "Finetuning LLMs for EvaCun 2025 token prediction shared task".

| Subset | 29 | 33 | 53 | MLE | BL |
|---|---|---|---|---|---|
| all | 0.84 | **0.94** | 0.31 | 0.83 | 0.93 |
| func | **0.98** | **0.98** | 0.83 | **0.98** | **0.98** |
| non func | 0.80 | **0.93** | 0.15 | 0.79 | 0.92 |
| in vocab | 0.89 | **0.97** | 0.34 | 0.92 | 0.96 |
| oov | 0.48 | **0.72** | 0.07 | 0.00 | 0.65 |

Table 4: Accuracy results for lemmatization. Results for teams 29, 33, 53, along with MLE baseline (pick most common lemma for each token), and BabyLemmatizer baseline.

| Subset | 32 | BL | BL+AUG |
|---|---|---|---|
| all | **0.21** | 0.14 | **0.21** |
| func | 0.36 | 0.36 | **0.46** |
| non func | **0.19** | 0.12 | **0.19** |
| in vocab | 0.22 | 0.16 | **0.23** |
| oov | **0.03** | <0.01 | <0.01 |

Table 5: Accuracy results for Token Prediction. Results for Team 32, along with Babylemmatizer baseline, and Babylemmatizer baseline with augmented data.

### 7.1 Lemmatization

The performance of the submitted lemmatization models varied significantly based on the complexity of the word forms and their frequency in the training data. Table 4 presents the overall accuracy results for lemmatization across all systems. One key observation was that function words were significantly easier to lemmatize than non-function words, as seen in the high accuracy scores across all models. This is expected, given their lower morphological variation and higher frequency in the training data. OOV words, by contrast, posed a greater challenge, highlighting the difficulty in handling previously unseen forms. In fact, OOV items represented the only notable bottleneck in lemmatization performance, as in-vocabulary words were almost perfectly lemmatized by BabyLemmatizer and team 33. This suggests that both systems exhibit strong context-awareness, allowing them to accurately determine the relevant lemma based on contextual cues.

### 7.2 Token Prediction

The results in table 5 show that 32 and BL+AUG outperform BL overall (0.21 vs. 0.14), with augmentation significantly improving function-word (0.46) and in-vocabulary (0.23) accuracy. However, OOV handling remains poor across all models, with 32 performing very slightly better (0.03).

## 8 Discussion

The results highlight both progress and remaining challenges in lemmatization and token prediction in ancient Akkadian. For lemmatization, the high accuracy of BabyLemmatizer and team 33's model shows that hybrid models combining neural networks and rule-based approaches are effective for Akkadian's complex morphology. However, the performance gap between in-vocabulary and out-of-vocabulary words suggests that generalizing to unseen forms remains a significant challenge.

Token prediction proved more difficult, reflecting the uncertainty in reconstructing missing text from fragmentary sources. Function words were easier to predict accurately than content in-vocabulary words, which exhibit greater variability. Out-of-vocabulary words were almost impossible to predict.

This shared task reinforces a pattern of successful collaborations between cuneiform specialists and computer scientists, or individuals with expertise in both domains. The complexity of ancient languages like Akkadian and Sumerian, with their rich morphological structures and varied orthographic conventions, demands both computational innovation and philological expertise. The challenges of this field may be noted by the fact that out of sixteen teams that initially expressed interest in the shared task, only four submitted final systems, with three completing the lemmatization task and only one the token prediction task.

The new corpora was made available through the years-long work of cuneiform specialists working on the eBL and Archibab digital projects. Their specialized knowledge ensured high-quality data that enhanced model performance. The original data files were furthermore preprocessed for consistency and machine readability by experts with experience in both computer models and cuneiform texts and their digital representations. Building thus on the works of others, the task force has resulted in robust models for new periods and genres of Akkadian texts that were previously underrepresented in computational studies. These new models enable more comprehensive analyses of Akkadian's diachronic development and genre-specific characteristics, ultimately enriching our understanding of this pivotal language in ancient Near Eastern history.

The task force has demonstrated that the collaboration between domain experts and computational scientists does not need to be direct–their complementary contributions across different stages of the ancient language processing pipeline create an environment conducive to breakthrough results that benefit the entire field. The promising performance on the lemmatization task, particularly by hybrid approaches combining neural networks with rule-based systems, demonstrates that these methodologies can be successfully applied to other under-resourced ancient languages. This could potentially transform our ability to analyze and understand historical texts at scale, opening new avenues for research across multiple disciplines within the humanities.

## Acknowledgments

EvaCun 2025 represents a collaborative scholarly endeavor made possible through the exemplary sponsorship of two foundational digital cuneiform initiatives: the electronic Babylonian Library (eBL), directed by Enrique Jiménez at LMU Munich, and the ARCHIBAB project, led by Dominique Charpin (Collège de France) with substantial contributions from Marine Béranger (FU Berlin). Their provision of extensive textual corpora and competitive financial incentives has fundamentally transformed what would otherwise have remained a theoretical exercise into a vibrant scholarly forum. Their investment in this emerging methodological approach demonstrates remarkable institutional foresight. We would also like to thank Luis Sáenz for creating a comprehensive list of function words for the evaluation splits, the Academy of Finland for funding the Centre of Excellence in Ancient Near Eastern Empires (PI Saana Svärd), and the Origins of Emesal project (PI Krister Lindén).

## References

David Bamman. 2012. *11-712 NLP Lab Report: Akkadian-morph-analyzer*.

François Barthélemy. 1998. A morphological Analyzer for Akkadian Verbal Forms with a Model of Phonetic Transformations. In *Computational Approaches to Semitic Languages*.

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with lematus. In *16th annual conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1391–1400. Association for Computational Linguistics (ACL).

Bartosz Bogacz and Hubert Mara. 2022. Digital Assyriology—Advances in Visual Cuneiform Analysis. *Journal on Computing and Cultural Heritage*, 15(2):1–22.

Giorgio Buccellati. 1977. The old babylonian linguistic analysis project: goals, procédures and first results. In *Computational and mathematical linguistics: proceedings of the International Conference on Computational Linguistics: vol. I.-(Biblioteca dell'Archivum romanicum; 36)*, pages 385–404. LS Olschki.

Tim Buckwalter. 2002. Buckwalter arabic morphological analyzer version 1.0. *Linguistic Data Consortium, University of Pennsylvania*, pages 86–93.

Michele Cammarosano. 2014. 3d-joins und schriftmetrologie: A quantitative approach to cuneiform palaeography. In *Current Research in Cuneiform Paleography. Proc. of a Workshop held at the 60th Rencontre Assyriologique Internationale. University of Warsaw*.

Dominique Charpin. 2014. Ressources assyriologiques sur internet. *Bibliotheca Orientalis*, 71(3):331–357.

Danlu Chen, Aditi Agarwal, Taylor Berg-Kirkpatrick, and Jacobo Myerston. 2023. CuneiML: A Cuneiform Dataset for Machine Learning. *Journal of Open Humanities Data*, 9(1):30.

Y. Cobanoglu, J. Laasonen, F. Simonjetz, I. Khait, S. Cohen, Z. Földi, A. Hätinen, A. Heinrich, T. Mitto, G. Rozzi, L. Sáenz, and E. Jiménez. 2024. Transliterated cuneiform tablets of the electronic babylonian library platform. *Journal of Open Humanities Data*, 10(1):19.

Yunus Cobanoglu. 2023. Transliterated Fragments of the Electronic Babylonian Literature Project (eBL). *Zenodo*.

Yael Cohen-Sygal and Shuly Wintner. 2006. Finite-state registered automata for non-concatenative morphology. *Computational Linguistics*, 32(1):49–82.

Tim Collins, Sandra I Woolley, Luis Hernandez Munoz, Andrew Lewis, Eugene Ch'ng, and Erlend Gehlken. 2014. Computer-assisted reconstruction of virtual fragmented cuneiform tablets. In *2014 International Conference on Virtual Systems & Multimedia (VSMM)*, pages 70–77. IEEE.

Peter T. Daniels. 2023. "Look with thine ears": Why Writing Is Syllable-based. *WORD*, 69(1):91–116. Publisher: Routledge.

Liudmila L. Fedorova. On the Typology of Writing Systems. In *Grapholinguistics in the 21st Century*, pages 805–824, Online. Fluxus Editions.

Ethan Fetaya, Yonatan Lifshitz, Elad Aaron, and Shai Gordin. 2020. Restoration of fragmentary babylonian texts using recurrent neural networks. *Proceedings of the National Academy of Sciences*, 117(37):22743–22751.

Denis Fisseler. 2019. *Contributions to computer-aided analysis of cuneiform tablet fragments*. Ph.D. thesis, Dissertation, Dortmund, Technische Universität, 2019.

Gösta Gabriel, Karenleigh A. Overmann, and Annick Payne, editors. 2021. *Signs - sounds - semantics: nature and transformation of writing systems in the Ancient Near East*. Number 13 in Wiener offene Orientalistik. Ugarit Verlag, Münster.

Sh. Gordin. 2014. *Visualizing Knowledge and Creating Meaning in Ancient Writing Systems Proceedings of the International Workshop of the Research Group "Notational Iconicity", 24-25 Sep. 2010*. Berliner Beiträge zum Vorderen Orient. PeWe-Verlag, Gladbeck. Type: Edited Book.

Shai Gordin, Gai Gutherz, Ariel Elazary, Avital Romach, Enrique Jiménez, Jonathan Berant, and Yoram Cohen. 2020. Reading Akkadian cuneiform using natural language processing. *PLOS ONE*, 15(10):e0240511.

Gai Gutherz, Shai Gordin, Luis Sáenz, Omer Levy, and Jonathan Berant. 2023. Translating Akkadian to English with neural machine translation. *PNAS Nexus*, 2(5):pgad096.

W. W. Hallo. 1991. The Concept of Canonicity in Cuneiform and Biblical Literature: A Comparative Apprasial. In K. L. Younger, W. W. Hallo, and B. F. Batto, editors, *The Biblical Canon in Comparative Perspective: Scripture in Context IV*, pages 1–19. Edwin Mellen Press, Lewiston. Type: Book Section.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 133–142.

Laura Kataja and Kimmo Koskenniemi. 1988. Finite-state Description of Semitic Morphology: A Case Study of Ancient Accadian. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*.

Koren Lazar, Benny Saret, Asaf Yehudai, Wayne Horowitz, Nathan Wasserman, and Gabriel Stanovsky. 2021. Filling the gaps in Ancient Akkadian texts: A masked language modelling approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4682–4691, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Macks. 2002. Parsing akkadian verbs with prolog. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*.

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.

Émilie Pagé-Perron, Maria Sukhareva, Ilya Khait, and Christian Chiarcos. 2017. Machine Translation and Automated Analysis of the Sumerian Language. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–16, Vancouver, Canada. Association for Computational Linguistics.

Aleksi Sahala. 2014. Babyparser: Muinaisbabylonian morfologian ohjelmallinen jäsentäminen. *Helsinki: University of Helsinki, MA Thesis*.

Aleksi Sahala. 2021. Contributions to computational assyriology. *Helsinki: University of Helsinki*.

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

Aleksi Sahala and Eliese-Sophia Lincke. 2024. Neural lemmatization and pos-tagging models for coptic, demotic and earlier egyptian. In *Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024)*, pages 87–97.

Aleksi Sahala and Krister Lindén. 2023. A Neural Pipeline for Lemmatizing and POS-tagging Cuneiform Languages. In *Proceedings of the Ancient Language Processing Workshop at the 14th International Conference on Recent Advances in Natural Language Processing RANLP 2023*, pages 203–212.

Aleksi Sahala, Miikka Silfverberg, Antti Arppe, and Krister Lindén. 2020a. Automated Phonological Transcription of Akkadian Cuneiform Texts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3528–3534.

Aleksi Sahala, Miikka Silfverberg, Antti Arppe, and Krister Lindén. 2020b. BabyFST: Towards a Finite-State Based Computational Model of Ancient Babylonian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3886–3894.

Fabian Simonjetz, Jussi Laasonen, Yunus Cobanoglu, Alexander Fraser, and Enrique Jiménez. 2024. Reconstruction of cuneiform literary texts as text matching. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13712–13721.

Gustav Ryberg Smidt, Katrien De Graef, and Els Lefever. 2024. Keep me PoS-ted: experimenting with Part-of-Speech prediction on Old Babylonian letters. *it - Information Technology*, 65(6):264–274. Publisher: De Gruyter Oldenbourg.

Thea Sommerschield, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, and Nando De Freitas. 2023. Machine Learning for Ancient Languages: A Survey. *Computational Linguistics*, 49(3):703–747.

Steve Tinney. 2019. *L2: How it Works*.

Stephen Tyndall. 2012. Toward automatically assembling hittite-language cuneiform tablet fragments into larger texts. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 243–247.

Juan-Pablo Vita, editor. 2021. *Vita, J.-P. (Ed.). (2021). History of the Akkadian language. Brill.* Number volume 152/1-2 in Handbook of Oriental studies = Handbuch der Orientalistik : Section one. The Near and Middle East. Brill, Leiden.

Stefan Weninger, Geoffrey Kahn, Michael P. Streck, and Janet C. E. Watson, editors. 2011. *The Semitic Languages: An International Handbook*. Number 36 in Handbooks of Linguistics and Communication Science. De Gruyter, Berlin.

Imed Zitouni, editor. 2014. *Natural language processing of semitic languages*. Theory and applications of natural language processing. Springer, Berlin Heidelberg.