

# GILgaMeSH: Glyph-Interpreting Language Models for Sumerian History

Stanford CS224N Custom Project

**Cole Simmons**

Department of Computer Science  
Stanford University  
coles@stanford.edu

## Abstract

To date, archaeologists have unearthed more than 100,000 Sumerian texts, a corpus that reflects both the apparent genesis of writing itself, as well as a rich written tradition spanning three millennia. Reading these texts, however, requires years of specialized training in Assyriology. Each glyph in the language’s cuneiform script can have dozens of different readings depending on the neighboring context, the genre, and the period. The laborious task of resolving each glyph into a particular reading is known as *transliteration*, wherein Assyriologists use a conventional system to render readings phonetically in the Latin alphabet. Transliteration is the most difficult and time-consuming step in translating Sumerian.

In this paper, we introduce *GILgaMeSH*, the first neural machine transliteration pipeline for Sumerian. We demonstrate that—despite Sumerian’s status as a low-resource language and language isolate—large pretrained multilingual language models can be adapted to perform the sequence-to-sequence task of transliterating a sequence of unicode cuneiform glyphs with remarkable accuracy. With such an abundance of extant texts and so few specialists capable of reading them, our work represents a significant step towards making a complete translation of the most ancient corpus available in all modern languages.


## 1 Key information to include

- Mentor: Soumya Chatterjee

## 2 Introduction

Sumerian is the oldest attested written language. First written in southern Mesopotamia (modern-day Iraq south of Baghdad) as early as 2900 BCE, Sumerian continued to be used, depending on the time and place, for administration, liturgy, or scholarship until around 100 CE. It then went extinct. However, because texts were written on clay—unlike later biodegradable substrates such as papyri—they survived in great quantity (Finkel and Taylor, 2015). Rediscovered in modernity, the struggle to completely decipher the language continues on to this day, hindered by Sumerian’s status as a language isolate, the fragmented condition of most texts, and by the indistinguishable nature of genuine yet underrepresented language features, scribal mistakes, and dialectical or orthographic variation.

Mesopotamian scribes formed glyphs (i.e. signs) by composing stylus impressions on wet clay tablets. In context, a glyph can function as a logogram (word sign), phonogram (phonetic sign), or determinative (unvoiced semantic classifier). Most glyphs have many readings

across these categories of no necessary semantic relation: e.g.  can represent the verb “to speak”, the ergative marker on a noun, pronominal agreement on a verb, or just the syllable *e*. We cross-analyzed our dataset with a Sumerian dictionary and found that the average number of different readings for a glyph, weighted by glyphs’ frequency, is 29.22.

Therefore, to read a tablet you must first painstakingly sort through the myriad of possibilities for every single glyph given those that surround it. Assyriologists do this through the process of transliteration. Using a conventional system, they render selected readings in the Latin alphabet in a manner that reflects best-guess phonetics. Resulting homophones are distinguished through subscripts: e.g. *e* and *e*<sub>2</sub> are homophonic but otherwise unrelated readings semantically. While each glyph can have many possible readings, each reading is backed by only one glyph. Further minimizing ambiguity in the transliteration, nominal or verbal roots and their affixes are joined with hyphens. In a lecture by Professor Gina Konstantopoulos, she stated that “once you have a transliteration, you’re about 80% of the way to a translation.”

Transformer-based deep learning architectures have had demonstrated great success in language modeling when trained on either a masked language modeling (MLM) task—predicting a masked word given the surrounding words—or a causal language modeling (CLM) task—predicting the next word given the preceding words. However, this approach typically requires a significant number of examples in a language. But by training models on a multilingual corpus, the model can learn cross-lingual representations that can be transferred to a new language.

Our project demonstrates remarkable success in adapting large multilingual models to model and transliterate Sumerian, despite the highly fragmented nature of the extant texts and the language being both low-resource and an isolate. By first fine-tuning an XLM-R model on a masking task, and then connecting it to another XLM-R model in an encoder-decoder architecture, we were able to achieve an average BLEU score of 90.07 on unseen test data. In comparison, generating transliterations by sampling from the glyphs’ possible readings yields an average BLEU score of 7.2.

### 3 Related Work

#### 3.1 Language Modeling with Transformers

In the traditional approach to language modeling, practitioners applied methods such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) to generate static embeddings for each word. They would then pass these embeddings as the inputs to an RNN-based architecture, such as a GRU or LSTM, to learn how these words in sequence related to the downstream task, e.g. document classification.

The advent of transformers (Vaswani et al., 2017) marked a turning point for natural language processing, with models exhibiting unprecedented abilities in understanding and generating text. BERT (Devlin et al., 2019) demonstrated the efficacy of combining transformers and a bidirectional training strategy. RoBERTa (Liu et al., 2019) improved performance by removing the next sentence prediction objective and training on a larger corpus.

#### 3.2 Multilingual Models and Sequence-to-Sequence

While BERT and RoBERTa were theoretically language-agnostic, they were trained and evaluated as monolingual models. XLM (Lample and Conneau, 2019) showed that multilingual models can effectively learn cross-lingual representations. XLM-R (Conneau et al., 2020) pushed the envelope further, training on a corpus of 100 languages.

A shortcoming of the aforementioned models is that they are “encoder-only,” meaning that they can produce meaningful representations of a sequence but are not designed to produce an output sequence in the same or a different language. The earlier work, however, of Sutskever et al. (2014) showed that encoder models can be paired with a decoder to perform these sequence-to-sequence tasks.

### 3.3 Low-resource NMT and Sumerian

Languages such as English have immense corpora due to their ubiquity on the internet. But other languages—both ancient and modern—have much scarcer resources with which to work. A promise of multilingual models is that the cross-lingual understanding can be leveraged to fine-tune models to work with languages that do not have sufficient resources on which to train one from scratch.

Li et al. (2023) shows that successive rounds of zero-shot translation and back-translation can be used to generate synthetic parallel examples between low-resource languages and English. Finally, Bansal et al. (2021) is the only other research effort on the Sumerian language. They show that simpler models outperform deeper models on analytical tasks like POS tagging and NER.

## 4 Approach

Although Sumerian is a language isolate, it shares grammatical features with other modern languages: like Basque, it has ergative–absolute alignment; like Turkish and Japanese, it is agglutinative; and like Korean, it is SOV (Michalowski, 2004). Therefore, the key to our approach is to leverage XLM-R—which was pretrained on all of these languages—as both the encoder and decoder in an encoder-decoder model. We do so in three steps.

### 4.1 Encoder Fine-Tuning

We began by fine-tuning a pretrained XLM-R encoder on the unicode cuneiform glyphs using a masked language modeling (MLM) task. XLM-R’s tokenizer is based upon SentencePiece (Kudo and Richardson, 2018). We found that because the unicode cuneiform glyphs were not in the vocabulary, using the tokenizer led to a sequence of cuneiform glyphs being treated as a single token, rather than each glyph being treated as its own token. To account for this, we iterated over the data and added each unique glyph to the vocabulary, a total of 505 new items.

An initial aim with this step was to test whether XLM-R could be adapted to learn representations for the new vocabulary and the structure of the language independent of the downstream sequence-to-sequence task.

### 4.2 Encoder-Decoder Fine-Tuning w/ Frozen Encoder Weights

Once the first step yielded a model capable of learning effective internal representations for the unicode cuneiform glyphs, we then integrated the model as the encoder in an encoder–decoder model. For the decoder, we again used an XLM-R model, but this time one trained on a causal language modeling (CLM) task capable of autoregressive generation. We performed an initial round of fine-tuning with the encoder weights frozen so that the decoder could update its weights to better align with the relationship between the encodings produced by the model in the first step and the desired output.

### 4.3 Encoder-Decoder Fine-Tuning w/ Unfrozen Encoder Weights

Because the encoder was trained to learn the best encodings for the MLM task, the decoder has limited ability to translate these into the desired output. The previous step shows large initial gains that quickly plateau. Once this occurred, we unfroze the encoder weights so that the model could learn the most effective way to encode and then decode the input unicode sequence to produce the transliteration sequence.

## 5 Experiments

### 5.1 Data

Our dataset was derived from the Electronic Pennsylvania Sumerian Dictionary (ePSD2)<sup>1</sup>, which provides metadata and transliterations for 84,098 texts via JSON files. These transliterations were produced over decades of changing conventions and evolving knowledge of Sumerian vocabulary and grammar. After significant preprocessing, we were able to standardize the formatting and minimize editorializing about glyphs that were broken away, left out, or erroneously included.

We added four special tokens to maintain structural information about the tablet:

- <SURFACE> – The start of a surface. While most texts only have writing on a single surface, some have writing on multiple surfaces.
- <MISSING\_LINES> – An unknown number of missing lines. This usually means that the upper and/or lower portion of a text is broken away.
- <MISSING> – An unknown number of missing glyphs.
- <NEWLINE> – Line break. These are important to retain because it is extremely rare that morphological glyphs run over to a subsequent line.

With the transliterations in shape, we worked backwards to construct the unicode glyph sequences. Table 1 provides an example of the input and target sequences.

We have made this dataset available on Hugging Face<sup>2</sup> to be utilized in subsequent work.

We then split the data into train, validation, and test sets using a 90%/5%/5% split. As both an artifact of what was produced and what sites have been archaeologically excavated, there is a considerable imbalance in the historical periods represented in the corpus. To ensure that we were training, testing, and validating evenly on how the language was used throughout time, we stratified the splits by period. Table 2 shows the number of examples in each. Because the genres of texts produced at excavated sites correlates strongly with period, stratifying by period also provides a similar distribution of genre (Table 3).

Finally, to ensure that the splits remained consistent over the models and configurations, we included this split as part of the Hugging Face dataset, uploaded before any model training occurred.



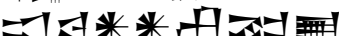

Glyphs (Inputs)	<SURFACE>  <NEWLINE>  <NEWLINE>  <NEWLINE>  ...
Transliteration (Targets)	<SURFACE>d-en-lil <sub>2</sub> <NEWLINE>lugal kur-kur-ra <NEWLINE>ab-ba dingir-dingir-re <sub>2</sub> -ne-ke <sub>4</sub> <NEWLINE>inim gi-na-ni-ta ...

Table 1: Example (ca. 2600–2300 BCE)

<sup>1</sup><https://oracc.museum.upenn.edu/epsd2/json>

<sup>2</sup><https://huggingface.co/datasets/colesimmons/sux>

<sup>3</sup><https://oracc.museum.upenn.edu/epsd2/about/articles/index.html>

Period	Dates <sup>3</sup>	Train	Val	Test
Ur III	ca. 2100–2000 BCE	72,236	4,014	4,014
Old Akkadian	ca. 2300–2200 BCE	5,010	278	278
Early Dynastic IIIb	ca. 2600–2300 BCE	3,515	195	195
Old Babylonian	ca. 2000–1600 BCE	1,536	86	85
Lagash II	ca. 2200–2100 BCE	815	45	45
Early Dynastic IIIa	ca. 2800–2600 BCE	759	42	42
Unknown		159	9	9
Neo-Babylonian	ca. 625–539 BCE	26	2	2
Neo-Assyrian	ca. 900–612 BCE	25	1	1
Ebla	ca. 2500–2250 BCE	10	1	1
Middle Babylonian	ca. 1350–1000 BCE	7	0	1
<b>Total</b>		84,098	4,673	4,673

Table 2: Texts by Period

Genre	Train	Val	Test
Administrative	78,449	4,373	4,355
Royal Inscription	2,762	138	149
Literary	1,131	63	67
Letter	733	35	43
Legal	555	31	35
Unknown	338	23	18
Lexical	69	4	1
Liturgy	37	5	2
Royal/Monumental	9	0	1
Mathematical	7	0	1
Scientific	3	0	1
Ritual	2	0	0
Hymn-Prayer	1	0	0
Lexical; School	1	0	0
Astronomical	1	0	0
<b>Total</b>	84,098	4,673	4,673

Table 3: Texts by Genre

## 5.2 Evaluation method

We evaluated the encoder during the fine-tuning process by measuring the accuracy of same token masking task that the model was using to train, but on the validation set. Specifically, every 200 steps we would randomly mask 10% of the tokens in each example in the validation set, have the model predict what tokens were being masked, and log the average accuracy of the predictions across all examples. Once the training process had plateaued, we ran the same task on the test set to make sure performance was comparable.

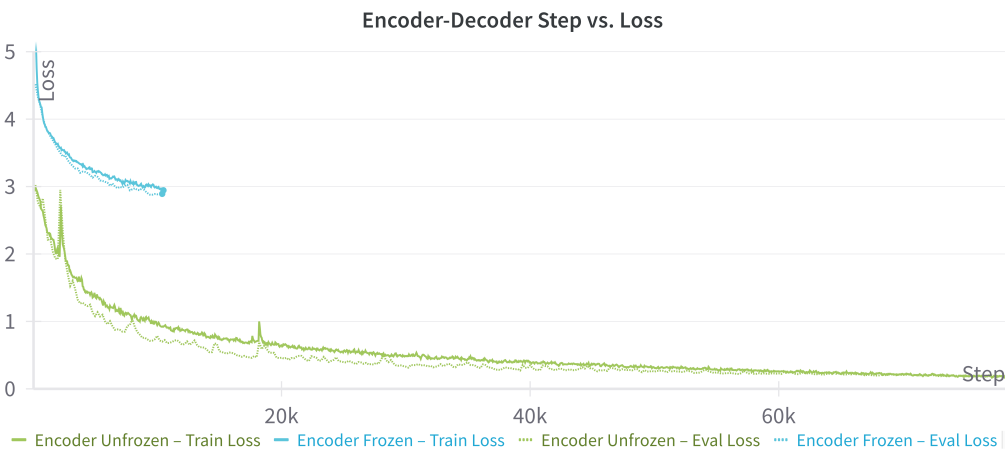
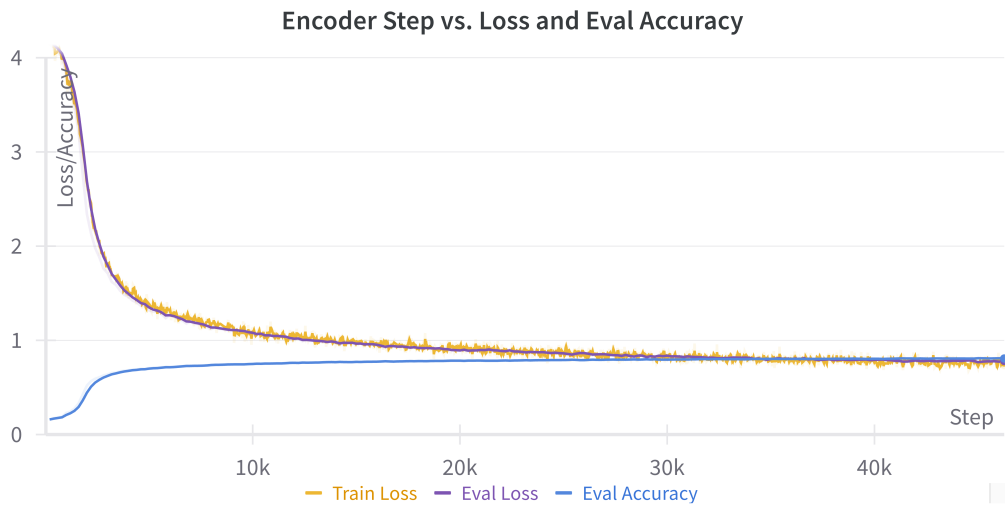
For the encoder-decoder model, we evaluated the final result by first selecting a maximum sample of 20 examples from each genre and generating examples using values for num\_beams: 1, 3, 5. We then compared the resultant generated transliterations against the true transliterations by calculating a BLEU score to see what effect the beam search was having on overall performance. When it was determined that beam search was not helping the results in any statistically significant manner, we again generated transliterations for glyph sequences in the test set, but this time with a maximum of 150 examples from each genre. Finally, we compared the BLEU scores against those of the baseline and performed qualitative evaluation.

### 5.3 Experimental details

The encoder was initialized with XLM-R weights and fine-tuned with a learning rate of  $5e-05$  and the AdamW optimizer. Train and eval batch sizes were 8. The MLM probability—that is, the probability of any token being masked in a given step—was set to 0.10. It ran for 20 epochs, or 46,220 gradient steps, on a single A100 SXM 80GB.

The encoder-decoder with frozen encoder weights was trained with a learning rate of  $5e-05$  given 200 warmup steps and the AdamW optimizer. Train and evaluation batch sizes were set to 16, and train and eval loss was logged every 200 steps. It was trained for 15 epochs, or 10,400 gradient steps, on a single A100 SXM 80GB. Then, the encoder-decoder with the encoder weights unfrozen was trained with the same hyperparameters for 78,850 steps.

### 5.4 Results



The encoder plateaued around 80% accuracy on the validation set, spending the last 16,000 steps unable to break out of the 80-81% accuracy range. Train and eval loss stayed together, showing no signs of overfitting. The model was tested on samples from the test set on a similar token masking probability of 0.1, and the results demonstrated that the eval loss was generalized.

These were promising initial results regarding the ability of cross-lingual models to learn Sumerian. We connected this model to an XLM-R decoder, enabled cross-attention, froze

the encoder weights, and found that the decoder was able to make rapid initial gains in aligning itself with the encoder to perform the objective. Once it began to plateau, we unfroze the encoder weights. Loss decreased slowly but steadily.

Frankly, we were astounded by the performance of the model to generate nearly identical translations as experts when given glyph sequences from the test set, first performing a qualitative evaluation on a handful of examples. Next, we compared the BLEU (removing special tokens and using character tokenization) scores on a sample of the test set to determine the effects of greedy search or beam search with a variable number of beams affected the results. Since it did not appear to have any notable effect, we proceeded with calculating the BLEU scores, once more with special tokens removed and with character tokenization, on a larger sample, taking at most 150 examples from each genre. The results are presented in Table 4 and Table 5.

Genre	# of Examples	Average BLEU Score	Baseline BLEU Score
Royal/Monumental	1	100.00	12.58
Administrative	150	97.23	7.65
Unknown	18	93.78	9.11
Legal	35	93.64	7.04
Letter	43	91.55	6.71
Royal Inscription	149	90.59	7.43
Literary	67	71.71	5.57
Liturgy	2	65.39	5.39
Mathematical	1	63.72	10.21
Scientific	1	58.88	3.91
Lexical	1	11.47	4.96
<b>Overall Average</b>	–	<b>90.07</b>	<b>7.20</b>

Table 4: Results by genre. BLEU scores of generated transliterations versus transliterations generated by sampling from the set of each glyph’s possible readings.

Genre	# of Examples	Average BLEU Score	Baseline BLEU Score
Ur III	298	95.66	7.35
Neo-Assyrian	1	95.42	9.07
Old Akkadian	20	91.00	7.73
Early Dynastic IIIa	5	87.63	8.15
Lagash II	18	84.73	7.98
Early Dynastic IIIb	38	83.34	8.17
Unknown	6	81.16	9.24
Neo-Babylonian	2	74.92	4.21
Old Babylonian	80	74.57	5.96
<b>Overall Average</b>	–	<b>90.07</b>	<b>7.24</b>

Table 5: Results by period using the same methodology as Table 4. Baseline texts were regenerated.

- True transliteration: 1(u) 1(aš) ku3 gin2 la2 igi- gal2 lugal-iti-da ku3-dim2-ra lugal-iti-da aga3-us2-e gu2 i3-ni-gar 5(aš) še lid2-ga 2(diš)-kam-ma-še3 gu2 i3-ni-gar lugal-en-nu ur-d-pa-e3 ur-li lu2-ki-inim bi3
- Generated: 1(u) 1(aš) ku3 gin2 la2 igi gal2 lugal-iti-da ku3-dim2-ra lugal-iti-da aga3-us2-e gu2 i3-NI-gar 5(aš) še lid2-ga 2(diš)-kam-ma-še3 gu2 i3-NI2-gar lugal-en-nu ur-d-pa-e3 ur-li3 lu2-ki-inim PI

## 6 Analysis

These results are both very exciting and intuitive given the nature of the underlying data. Because the training data is dominated by administrative examples, it is natural that that would be the best performing category (excluding the single Royal/Monumental example). Legal, letter, and royal inscription texts are formal and highly standardized texts, so it is understandable why it performs well on those.

On the other hand, the literary, liturgical, mathematical, and scientific examples have a much different style, vocabulary, and form than the rest of the corpus. These are the genres that provide the greatest challenge to even renowned Assyriologists, and many are still not completely understood. The latter three genres also have either one or two examples, so it is difficult to say how generalizable the results are to the genre as a whole.

Finally, the outlier lexical text is also expected. In retrospect, these probably should have been excluded from the corpus altogether. Lexical texts are just lists of words, so there is not much of a consistent pattern for the model to learn.

Performing manual error analysis, there is a particularly common form of error. A widespread convention in transliteration is to represent the glyph with an uppercase name of a reading when the reading is uncertain. As to be expected, the model is unable to discern these from valid readings. For example, for one text, the true transliteration has "dumu inim šara2" but the model predicted "dumu KA šara2." Here, "KA" is, in fact, the correct way to represent the glyph for "inim." This can be mitigated in future work by replacing these occurrences in the training data with a sample from the sign's most common readings.

## 7 Conclusion and limitations

This success is despite some notable limitations of the project. First, a model such as this can only be as good as the underlying data. One Professor of Assyriology told us that the administrative transliterations were done by a single team rapidly and decades ago. As a result, there are a number of errors. However, so long as these errors aren't systematic, our model can have isolate them and suggest potential fixes.

The quality of the data is further hindered by the broken condition of most of the tablets, limiting the ability of the model to make informed predictions based on what context remains. Additionally, administrative documents have an out-sized representation in the training data, and subsequent work should mitigate this by more evenly representing the genres by excluding administrative documents or duplicating examples in other genres.

Our results show the remarkable efficacy of multilingual models in the sequence-to-sequence task of transliterating the world's oldest writing. More than just applications in Assyriology, our research demonstrates the applicability of modern natural language processing techniques in the humanities, even on ancient scripts. It is also not just a theoretical win. *GILGāMeSH* has a number of tangible applications in advancing the field, from transliterating new tablets to highlighting areas of review for specialists to using the encoder to predict the tokens in a broken-away segment.



## References

- Rachit Bansal, Himanshu Choudhary, Ravneet Punia, Niko Schenk, Jacob L Dahl, and Émilie Pagé-Perron. 2021. How low is too low? a computational perspective on extremely low-resource languages.
- 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.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.
- Irving L. Finkel and Jonathan Taylor. 2015. *Cuneiform*. British Museum.
- Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining.
- Bryan Li, Mohammad Sadegh Rasooli, Ajay Patel, and Chris Callison-Burch. 2023. Multilingual bidirectional unsupervised translation through multilingual finetuning and back-translation.
- 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.
- Piotr Michalowski. 2004. *Sumerian*. Cambridge University Press, Cambridge ; New York.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.