The Deep Past Akkadian Project

Aditya Behera [1] **Abhi Karthana** [1]

1. Problem

The ancient language of Akkadian is one of the earliest recorded languages used to document mercantile transactions in ancient Mesopotamia. Despite its revolutionary role in transcribing business, finance, and literature, the language itself is incredibly difficult to translate. Only under a dozen linguists today are able to translate these ancient texts into modern English accurately. As a result, only half of the surviving tablets have been fully translated. If translations could be done more efficiently, we could uncover not only the remarkable events of ancient Mesopotamia but also financial insights from one of the first trade networks in history. Nearly 23,000 tablets survive documenting the Old Assyrian trading networks that connected Mesopotamia to Anatolia.

[1]Georgia Institute of Technology, Atlanta, USA. Correspondence to: Aditya Behera <abehera31@gatech.edu>, Abhi Karthana <akarthana3@gatech.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

2. Data and Pre-Processing

For our data, we used a small dataset of roughly 1,500 valid translations currently available on Kaggle. The data is provided by the Old Assyrian Research Environment (OARE), a lab at UChicago that catalogs Akkadian texts. Each entry in the training set includes an `oare_id` referencing the original OARE database record, an Akkadian `transliteration` extracted from the tablet, and an English `translation`. The test set follows a similar schema but provides no ground-truth translations, as it was used during the Kaggle competition to score submissions. Both datasets also include metadata such as document identifiers (`text_id`) and line boundaries (`line_start`, `line_end`) that mark sentence spans within the original tablets.

It is worth mentioning that test.csv does not provide the ground truth translation since it was used to score the teams during the Kaggle competition. We separated the provided training data into a 90/10 split for training and validation.

We created a script to clean out annotations from tablets and standardize the original texts to account for multiple symbols representing the same meaning. This included removing scribal annotations that were included by the human scholar who transcribed the tablet and any indications on how confident the scholar was about certain symbols. We also have to do the same for English texts as there are annotations and ratings in currently translated works as well.

Because the Akkadian language does not use the

same punctuation as us (ie. periods), we had to split their paragraphs into sentences using human annotations from OARE Assyriologists. We used a supplemental dataset that marks where all the sentence breaks in the train.csv file should go, and then used a script to complete the punctuation-wise mapping. Akkadian also does not follow the same grammatical structure as English does. Since Akkadian is an SOV (Subject-Object-Verb) language and English is SVO, we tried to reorder the sentences to match the SVO order that ByT5 was expecting, but it was far too difficult to properly reorder the sentences without scrambling the overall meaning.

Additionally, each sample could use accented characters, CDLI, ORACC, or Cuneiform Unicode, as every scholar was using an unstandardized system, so we had to convert all of these to one format to standardize all the transliteration signs. Unifying scholarly variations (e.g. sz $\rightarrow$ s, ś $\rightarrow$ s, ḫ $\rightarrow$ h) and subscript vowels (e.g. a2 $\rightarrow$ a, b3 $\rightarrow$ b) reduced the vocabulary size and increased the model's ability to recognize consistent semantic roots.

In addition to the correctly translated data, the datasets have supplemental links that have English definitions of many Akkadian words. We chose to scrape the API of the Electronic Babylonian Library (EBL) to collect these definitions and append them to every Akkadian word in order to develop an in-line dictionary for the majority of words in each transliteration.

3. Methods and Techniques

Our proposed technique is to utilize the ByT5 model, an out-of-the-box model widely used to process text in any language. The ByT5 model is a widely popular byte-to-byte model which is particularly useful to use with noisy text-based data. While our standard definition of noisy text data might be occasional typos, unclear handwriting, and other factors involved with the actual text, this particular problem's noise comes from the source material. In the specific context of Akkadian, the human annotators often used uncommon Unicode characters in their transliterations for each tablet. As such, a standard tokenizer might break or produce 'UNKNOWN' tokens when hitting these irregularities, but ByT5 operates on raw UTF-8 bytes, allowing it to process uncommon characters without the vocabulary bottlenecks of a traditional tokenizer.

For this project specifically, we chose to use the ByT5-Base variant to avoid the memory requirements of the XL and Large models. We also used LoRA with a rank of 16 instead of fine-tuning the entire model. The target modules were only the q and v projections because of our computational constraints. This allowed the 580M-parameter ByT5-base model to be fine-tuned locally in a reasonable amount of time. In comparison to XL, which requires 2 T4 GPUs, our base model can be trained on a single M2 16GB MacBook in 1.5 hours. We attempted to use 8-bit and 4-bit quantization via the bitsandbytes library, but bitsandbytes is not supported on MacOS. We took inspiration from the top teams in the Kaggle competition who also used the ByT5 models, though they used the XL variants. The top teams also often trained an ensemble of multiple ByT5-XL models and used a small gating network to choose which model to use. This worked really well for them since it allowed them to have multiple experts for different subsets of the dataset, however we were too compute constrained to replicate this.

4. Analysis and Comparison

The standard baseline for Akkadian translation often relies on traditional NMT architectures like mBART-50 or Transformer-base. These models typically achieve a BLEU score of 37.0 on standard datasets but fail significantly on "noisy" data (broken tablets). The Kaggle competition used hu-

man annotated transliterations and the top teams achieved a BLEU score of >40 on the hidden evaluation set.

In our local BLEU evaluation, the fine-tuned ByT5-base model achieved a 1-gram precision of 11.08, but this plummeted to 1.47 for 2-gram precision. This sharp decline, paired with a validation cross-entropy loss of 14.93, indicates that while the model is learning character-level and word-level distributions, it struggles to maintain multi-word syntactic coherence. The results of our training reflect this as seen in the samples below.

```
Sample 1:
  Input: um-ma k -ru-um k -ni-ia-ma a-na
      aa-q -il <gap> da-tim a -ip-ri-ni
      k -ar k -ar-ma wa-bar-ra-tim {(a
       small Ass. trading colony) OA;} q -
       b i -ma mup-pu-...
  Prediction: a aa-ra aa-ra aa-ra aa-ra aa-
      ra aa-ra aa-ra aa-ra aa-ra aa-ra aa-
      ra aa-ra aa-ra aa-ra aa-ra aa-ra aa-
      ra aa-ra aa-ra aa-ra aa-ra aa-ra aa-
      ra aa-ra aa-r...

Sample 2:
  Input: i-na mup-p -im aa a-lim{ki} ia-
      t u -m -im a-nim {this, those,
      this matter, one way or another,
      various, some ..., others ..., thus,
      in the same way} ...
  Prediction: -mu-ni i-na mup-p -im aa a-
      lim {to suffer} ia-t u -mu-ni i-
      na mup-p -im aa a-lim {to suffer} ia
      -t u -mu-ni i-na mup-p -im aa a-
      lim {to suffer} ia-t ...

Sample 3:
  Input: ki-ma mup-p -ni ta- a -me-a-ni a-
      ma-kam {there, from there} lu a-na
      a -m -im a-na .GAL-lim {palace
      service; palace, review p., arsenal,
      rear p., p. of...
  Prediction: an ai-m -im a-na .GAL-lim
       {palace service; palace, review p.,
      arsenal, rear p., p. of rest, the
      government, palace official, queen,
      palace official ,...

Sample 4:
  Input: me-+e-er mup-p -ni a-na k -ar
      k -ar-ma wa-bar-ra-tim {(a small
      Ass. trading colony) OA;} a -b i -
      l K .AN {status of a maid, female
      slave; meteoric i...
  Prediction: -ra-tim {to raise a slave; to
       raise a slave; to raise a slave; to
      raise a slave; to raise a slave; to
      raise a slave; to raise a slave; to
      raise a slav...
```

Our model's output highlights two primary failure modes common in low-data and low-parameter NLP:

1. Degenerate Repetition: Samples 1 and 4 exhibit "stuttering" loops (e.g., "aa-ra aa-ra"). This suggests the model has not yet internalized the EOS (end-of-sentence) token or is over-fitting on high-frequency byte sequences.

2. Identity Mapping: Sample 2,3, and 4 shows the model parroting phrases from the transliteration (including the in-line definitions we gave it) rather than translating it. This "copying" behavior occurs when the attention mechanism recognizes the input structure but lacks sufficient weights to perform the semantic transformation to English.

Despite these failures, the preprocessing flow is promising. With more computation power and possibly using the XL model or an ensemble approach, we could improve from our baseline by following some of the future directions below.

5. Future Directions

For more data, there are 8,000 supplemental translations from the AI Cuneiform Corpus (AICC) that could have also been incorporated if the smaller dataset proved to not contain enough information. AICC also has an additional 30K translation on its website that we could have scraped (https://aicuneiform.com). It should be noted that OARE claims many of these translations are of poor quality, but it should also be noted that OARE and AICC are in direct competition with each other for who can catalog and translate the most Akkadian texts.

Another key component in more accurate models is having more data. However, Akkadian text is finite and no longer produced, so the next best alternative is to create synthetic data. Many of the teams in the Kaggle competition prompted cloud LLMs (ChatGPT, Claude, etc.) to generate more synthetic data in the same style as the given samples. This allowed them to nearly 10x the amount of samples they had, which significantly improved the quality of their translations (one of the teams did an ablation study and concluded that creating more synthetic data helped their models more than increasing model size). Because we are financially constrained, we did not have enough OpenAI or Anthropic tokens to pursue this strategy.

Even though one of the teams concluded that more data was better than a larger model, that does not mean a larger model would not help at all. Using the Large or XL variants of the ByT5 model would likely improve our translations significantly. Additionally, many of the teams used an ensemble approach where they trained multiple ByT5 models on specific subsets of the dataset. Given no compute limits, we would have trained multiple larger models in an ensemble to match their performance.

This is not something that any team did in the Kaggle competition, but many researchers are moving away from transliteration entirely as it is too reliant on a very small number of Assyriologists to provide human translations. These researchers are training models to translate directly from Unicode Cuneiform glyphs or raw tablet images using Vision-Transformer (ViT) models.

6. Contributions

We, Aditya Behera and Abhi Karthana, worked together on The Deep Past Akkadian project as our NLP final project this semester. We went through the entire process collaboratively, from finding the topic, writing the proposal, collecting/processing data, training, and the write up. The GitHub repository is hosted by Aditya who pushed the majority of the code while both of us worked on finding preprocessing and training techniques during the literature review. Abhi focused on the write-up and keeping documentation up to date with current progress on the model. Overall, it was a successful learning experience for the both us as we applied the NLP techniques we learned in class as well as more advanced techniques we found during our literature review.

7. External Resources

Our Github:
https://github.com/adityabehera2004/deep-past-akkadian

Kaggle competition:
https://www.kaggle.com/competitions/deep-past-initiative-machine-translation

Old Assyrian Research Environment translations:
https://oare.byu.edu

Electronic Babylonian Library dictionary:
https://www.ebl.lmu.de

Google ByT5:
https://huggingface.co/google/byt5-base