CuReD: Deep Learning Optical Character Recognition for Cuneiform Text Editions and Legacy Materials

Shai Gordin
Ariel University
Open University of Israel
shaigo@ariel.ac.il

Morris Alper
Tel Aviv University
morrisalper@mail.tau.ac.il

Avital Romach
Yale University
avital.romach@yale.edu

Luis Sáenz
Heidelberg University
luissaenzs@gmail.com

Naama Yochai
Tel Aviv University
naamayochai@mail.tau.ac.il

Roey Lalazar
roey@gerev.ai

Abstract

Cuneiform documents, the earliest known form of writing, are prolific textual sources of the ancient past. Experts publish editions of these texts in transliteration using specialized typesetting, but most remain inaccessible for computational analysis in traditional printed books or legacy materials. Off-the-shelf OCR systems are insufficient for digitization without adaptation. We present CuReD (Cuneiform Recognition-Documents), a deep learning-based human-in-the-loop OCR pipeline for digitizing scanned transliterations of cuneiform texts. CuReD has a character error rate of 9% on clean data and 11% on representative scans. We digitized a challenging sample of transliterated cuneiform documents, as well as lexical index cards from the University of Pennsylvania Museum, demonstrating the feasibility of our platform for enabling computational analysis and bolstering machine-readable cuneiform text datasets. Our result provide the first human-in-the-loop pipeline and interface for digitizing transliterated cuneiform sources and legacy materials, enabling the enrichment of digital sources of these low-resource languages.

1 Introduction

The cuneiform writing system was used to write around a dozen different ancient languages over a period of more than three millennia. Many of these complex writing systems were logo-syllabic and of different language families, from the agglutinative Sumerian in southern Mesopotamia, to the family of Hurrian and Urartian in northern Mesopotamia and Armenia, to Indo-European Hittite and Luwian in Anatolia. While the records of many of these languages are in the hundreds or thousands, it is Semitic Akkadian with its main Babylonian and Assyrian dialects that is attested on hundreds of thousands of ancient texts (Vita, 2021). What all of them share is a similar critical apparatus: a standard Latin transcription and notation system, developed by experts in scholarly publications, from the mid-19th century to the early 20th century (see Appendix A), and is still used to this day. Legacy materials such as personal notebooks of curators or researchers, or card catalogues in universities and museums use this notation system extensively (Fig. 1).

Many publications and legacy materials have been scanned or photographed, but are largely unavailable as machine-readable text. The ability to automatically digitize them using optical character recognition (OCR) would make their contents readily available to experts and the general public. They can be further used in computational research into the languages, cultures, and history of these societies, as well as a wider use of natural language processing (NLP) techniques, such as part-of-speech tagging, named entity recognition, sentiment analysis, machine translation, and more. This in turn can further enhance cross-lingual research and the creation of linked open data entities as well as knowledge graphs (Gutherz et al., 2023; Homburg et al., 2023; Sahala and Lindén, 2023; Ong and Gordin, 2024; Smidt et al., 2024).

Existing OCR models trained on texts in other languages such as English are not suitable for this task. They do not recognize the diacritics, typographical oddities like mixed upper- and lower-case or sub- and super-script, as well as special symbols required for digitizing cuneiform transliterations. Furthermore, many off-the-shelf models are biased by their prior training on character sequences in the

Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024), pages 130–140

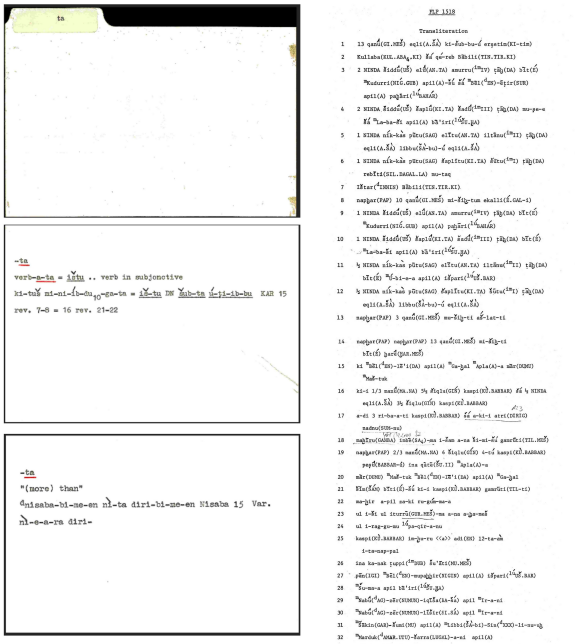


Figure 1: On the left examples of scanned lexical cards from the University of Pennsylvania Sumerian lexicography collection digitized by Anna Glenn for the University of Munich (Sjöberg, 2023). On the right an example of cuneiform transliteration from the Neo-Babylonian text corpus published in the dissertation of R. B. Dillard (Dillard, 1975).

source language. They are also typically trained on large datasets of scans with manually-labelled text, which is not available for more niche use cases such as those of cuneiform scholars.

In order to overcome these challenges, we trained a custom deep-learning based OCR model on transliterations of Akkadian, bootstrapping with artificially-generated data, and then fine-tuned with a small set of manually-labelled examples. The data for training and testing the model were taken from Open Richly Annotated Cuneiform Corpus (ORACC) and their equivalent print publications in PDF format.

However, those texts were from one period (Neo-Assyrian) and followed the same editorial conventions. To estimate the real-world usability of the model, we performed two additional digitization experiments on transliterated text produced with a typewriter during the 1970s and 80s: (1) 81 previously undigitized Neo-Babylonian administrative and archival daily documents published in the 1975 dissertation of Raymond B. Dillard (Dillard, 1975); (2) 30 index cards produced by Å. W. Sjöberg in the late 1970s and early 1980s as part of the Sumerian Lexicography collection housed in the Babylonian Section of the University of Pennsylvania Museum, now scanned in their entirety by Anna Glenn for LMU Munich, and published on the LMU library online catalogue (Sjöberg, 2023).

Both were particularly difficult to OCR, and were not a part of the model's training. In the case of the Dillard texts, we show that with fine-tuning on only 10 texts, the models' results rose from 53% to 85% accuracy. Similarly, in the case of the Sumerian lexical cards, after only 60 text lines, the model improved from 87% to 94% accuracy.

Thus, the model requires a minimal number of examples in order to be a significant assistant in the digitization process of ancient documents. The model is published on the Digital Pasts Lab GitHub repository and is freely available as an online tool in the Babylonian Engine website, which is undergoing a transformation into a standalone browser-based application. The tool and model will facilitate the digitization of hundreds of thousands of published cuneiform text lines in transliteration, which were previously unavailable for further computational or quantitative study.

2 Methods

2.1 Data preparation for training the OCR model

We used texts from the State Archives of Assyria (SAA), which are available in both print and digital forms. The transliterated Akkadian texts are hosted on the Open Richly Annotated Cuneiform Corpus (ORACC) as the State Archives of Assyria online (SAAo), which are part of the Munich Open-access Cuneiform Corpus Initiative (MOCCI) (Radner et al., 2015). Also available are scans of the books containing the texts in print; however, these cannot directly be used to train an OCR model because there is no alignment between the digitized Akkadian text and the location of its print equivalent on the scanned pages.

In order to collect usable pairs of images and corresponding digital Akkadian transliterations, we ran a heuristic algorithm which segmented and localized the transliterations within these scans, as well as extracting the digitized Akkadian transcribed text hosted on the Open Richly Annotated Cuneiform Corpus (ORACC) and aligning them. The algorithm uses computer vision (CV) methods such as thresholding and dilation to determine where there are paragraphs, and then runs a regular OCR on each paragraph to check whether this is an English

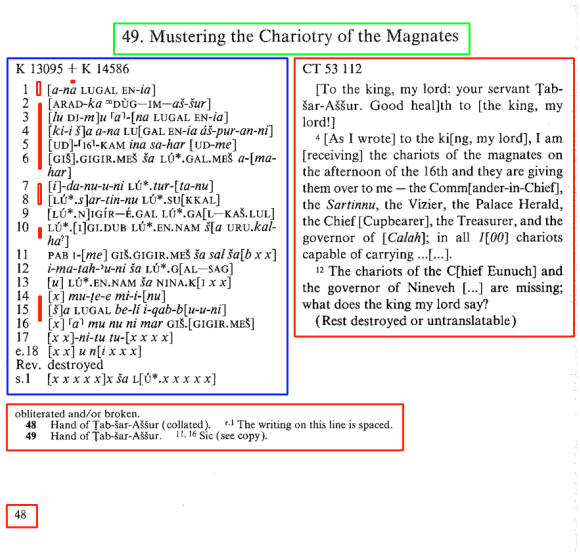


Figure 2: Segmentation Example from the State Archives of Assyria volume 1 (Parpola, 2015). Akkadian paragraph marked in blue, title marked in green.

paragraph or an Akkadian one. It also tries to locate titles (Fig. 2).

This method generated pairs of images and Akkadian text. The results were highly approximate and not clean enough to be used directly for training an OCR system, but allowed us to find examples for use in fine-tuning and evaluating our final model. In total, we manually labelled 30 of these images.

Because of this scarcity of labelled data, in order to train our OCR model from scratch we decided to bootstrap it with artificially-generated image data. We took all the digital text editions of the SAAo and generated images by rendering each line of text in the dataset as an image file. Our core OCR model was based on the open-source Kraken OCR framework developed by Benjamin Kiessling as described in Romanov et al. (2017).

[x x x x x]-u-ma ú-ta-na-ba-al [x x x x x x x x x x x x x x x x x]

Figure 3: Artificial data example generated from the SAAo corpus.

In order to increase the robustness of the model to noise and typeface variation, we added noise to the images by using Kraken's data augmentation API, with parameters alpha=0.3 (mean of folded normal distribution of foreground pixel flip probabilities) and distortion=3 (mean of folded normal distribution from which distortion values are sampled). We found that results were significantly improved by using multiple fonts to render the images. In order to match the typefaces most commonly found in the source materials, we rendered the texts in three fonts: (1) DejaVu Serif, (2) Garamond, (3) IM Fell Double Pica.

We also found that it was important to consider italic text since in Akkadian transliterations lowercase letters are normally printed in italics. Therefore, for each font we rendered all lines of the dataset in both normal and italic letters. Although this did not exactly match the scanned texts in which normal-styled uppercase letters and italic-styled lowercase letters were mixed, we found that it gave acceptable results upon bootstrapping our model.

After generating all lines of the SAAo textual dataset in all three fonts and in both normal and italic styles (416,000 images in total), we took a random subset of 194,000 of these images to use as our artificially-generated bootstrapping dataset.

In summary the data that we collected and used in our final model consisted of:

1. 30 manually-labelled pairs of scanned Akkadian transliterations and their corresponding digital texts.

2. 194,000 automatically generated images of lines of Akkadian transliterations, using various fonts and both normal and italic font styles

Since the SAAo data was used for training, we used scanned data from The Royal Inscriptions of the Neo-Assyrian Period (RINAP) as test data. We manually labeled 10 pages of these books, which gave us about 350 lines of test data.

2.2 OCR workflow and architecture

The typical OCR workflow consists of steps similar to the following:

- Preprocess images (deskewing, image binarization)
- Segmentation (localizing text on page, line segmentation)
- Core OCR (converting line to text)
- Post-processing (language model-based correction)

We found that Kraken's default preprocessing and segmentation methods were sufficient for our purposes, and focused on adapting the core OCR model to Akkadian transliterations. We assume input of the form similar to the data we collected, with paragraphs already localized.

After binarization and line segmentation, each line of input was first dewarped and resized to be of appropriate dimensions for the OCR model. After dewarping, the height of each line was resized to be 48 units, with the width scaled by the same factor and with 16 units of (white) padding added to the left and right sides of the line. Therefore, each sample input into the OCR model is a tensor of shape $(48, ?, 1)$, with ? representing the variable width of a single line of input and 1 the single (grayscale) channel of input.

The core OCR model that we trained was a hybrid CNN-RNN neural network (CRNN) selected from Kraken with the following sequential architecture:

- 2D convolutional layer (32 filters, kernel size 4×2, 4×2 stride, 1×0 padding)
- 2D convolutional layer (64 filters, kernel size 4×2, 1×1 stride, 1×0 padding)
- Max-pooling (kernel size 4×2, stride 4×2, no padding, dilation 1)
- 2D convolutional layer (128 filters, kernel size 3×3, 1×1 stride, 1×1 padding)
- Max-pooling (kernel size 1×2, stride 1×2, no padding, dilation 1)
- Reshape (converting input of shape (2, ?, 128) to output of shape (?, 256))
- BiLSTM (hidden size 256)
- BiLSTM (hidden size 512)
- BiLSTM (hidden size 256)
- Fully-connected (output size 103, linear activation)

The output of the final layer was chosen to match the size of the character-level vocabulary: 102 characters found in the training set data, plus the 0 index to indicate the "blank symbol" meaning no character.

Additionally, each convolutional and recurrent layer was followed by a regularization layer, and the BiLSTM layers by dropout layers:

- Each convolutional layer was followed by a group normalization layer with group size 32. Group normalization is a variant of batch normalization adapted to computer vision tasks where small batch sizes are required due to memory constraints. Instead of normalizing across multiple samples in a batch, group normalization normalizes across channels within a single sample. In our case, this grouped channels into groups of 32 and normalized activations within each group. For more details, see Wu and He (2018)
- Each BiLSTM layer was followed by a dropout layer with dropout probability 0.5.

The outputs of the model for each step are interpreted as logits corresponding to the probability that each character in the vocabulary is present at that horizontal location in the line of text. We then used greedy decoding to identify the most likely character at each step.

Interpreting the output of such a model requires an additional merger step. For example, consider the following output of a similar OCR system (Fig. 4):

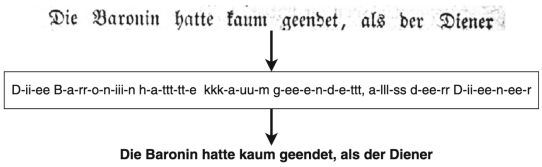


Figure 4: Old German OCR, reproduced from Fig. 10 in Martínek et al. (2020).

Since the model's output represents small displacements in the horizontal direction, the same character will be identified multiple times in a row. Therefore we merge the same label when it appears multiple times in a row, without another label or the blank symbol appearing in between. This is the *connectionist temporal classification (CTC) alignment* introduced by Graves et al. (2006).

2.3 Training

The model was trained in two stages: First, it was trained from scratch on the 194,000 artificially generated textual images from SAAo. Then, it was fine-tuned on the 30 manually labelled paragraph images from SAA books (about 900 lines of text). Although our manually labelled dataset was quite small, we found that the fine-tuning stage was critical for achieving acceptable results.

The objective used was the so-called *connectionist temporal classification (CTC) loss*. Similar to the CTC alignment described above, CTC loss is

used to compare the output of a continuous recognition system like OCR or speech recognition to a desired string of tokens. The motivation for CTC loss is twofold:

- The training data available to us is pairs of images and desired text, without spatial alignments.
- The network's outputs are character scores for each horizontal position, so the same token may be identified in multiple adjacent positions.

As first described in Graves et al. (2006), the CTC loss function solves these issues as follows. First, it takes as input the ground truth text and the network's outputs (probabilities per character for each horizontal position). It then calculates the likelihood of the ground truth text for each possible *path* (possible alignment) and sums them together over all possible paths. This is the objective function we used to train the network.

For both stages of training, we used the recommended settings from Kraken: batch size 1, Adam optimizer (learning rate $1e-3$ and momentum 0.9). In both stages, minimum validation loss was achieved after a single epoch of training, after which the model began to overfit, so we used the results of training on a single epoch.

3 OCR Results on Training and Testing Data

Results in Table 1 were calculated as follows: The **accuracy** we present was measured by computing the edit distance between the output of the OCR and the ground truth text in the test data images, divided by text length (averaged between the ground truth and OCR output texts). Before calculating edit distance, we normalized newlines and whitespace and combined together period (.) and dash (-) characters, since these can always be distinguished in context.

The new CuReD model has a character error rate (CER) of 9% on clean data and 11% on representative scans. We observe that manually fine-tuning the model with real dataset images greatly improves our accuracy, even though the fine-tuning dataset was extremely small. The baseline model, only trained on artificial data, overfit to this type of data and did not generalize well to real scans. Visually observing the baseline model output showed that it regularly had trouble distinguishing certain characters (e.g. "a" vs. "u"), and we hypothesize that this is because of the different appearance of these characters in the artificial training data fonts and the fonts used in the test data. Fine-tuning likely helps the model to quickly adapt to these differences.

Model	Validation Accuracy	Test Accuracy
Baseline	99.8%	77%
Fine-tuned 10 Images	91%	89%
Fine-tuned 30 Images	91%	89%

Table 1: OCR performance when training on artificially generated SAAo data, and finetuned on manually labelled SAA scanned transliterations. Accuracy tested on manually labelled transliterations from RINAP.

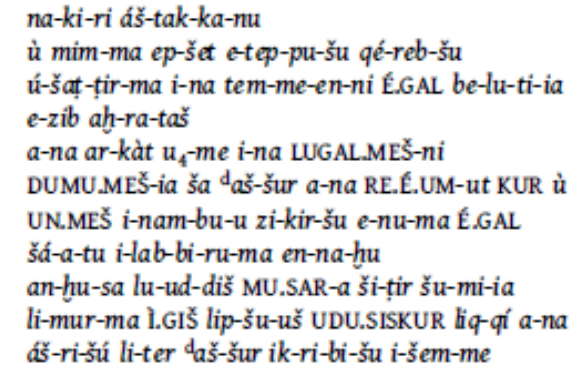
The columns "Validation Accuracy" is the accuracy of OCR prediction on a validation set selected from the training data. For the baseline model this is calculated on artificially generated SAAo transliteration images, while for fine-tuning it is calculated on a validation set of manually-labelled scanned images of SAA books from the fine-tuning set. The column "Test Accuracy" is the final accuracy of OCR predictions on the test dataset of real scanned transliterations from RINAP books (Fig. 5).

We also observe that even after fine-tuning on 10 images, we already reach a plateau in performance, and adding another 20 manually-labelled images to the fine-tuning does not noticeably improve performance. Thus, minimal data is needed to fine-tune the model on previously unseen published transliterations.

4 Real-world Experiments with the CuReD Tool

4.1 A human-in-the-loop pipeline

The OCR model released with this paper on GitHub can continuously improve on new datasets through fine-tuning. Yet, there remains a gap between cuneiform specialists and their ability to fine-tune and improve machine learning (ML) models. A set of *Cuneiform Recognition* tools, abbreviated CuRe, was therefore created. These tools are currently an online interactive platform for cuneiform experts as part of the Babylonian Engine project, but are in the process of becoming a standalone browser application for the sake of long-term upkeep; such as server maintenance costs. The *Cuneiform Recognition Documents* or CuReD tool provides a platform



(a) Sample scan from RINAP test data

```
na-ki-ri áš-tak-ka-m
ut mim-ma ep-šat e-tap-pu-šu qé-reb-šu
i-šag-rir-ma i-na tem-me-en-ni É ṣa be-lu-ri-i
e-zib aḫ-ra-taš
a-na ar-kdt u-me i-na LUGAL-MEŠ-n
UMU-MEŠ-ia ša dšaš-šur a-na 15.É.UM-ur gk ù
UN-MEŠ i-nam-bu-u zi-kir-šu e-nu-ma É SAL
šá-a-tu i-lab-bi-ru-ma en-na-ḫu
an-ḫu-sa lu-ud-diš MU.SAR-a ši-ṭir šu-m-ie
li-mur-ma Ì.GIŠ lip-šu-uš UDU.SISKUR kq-qí a-n
áš-ri-šú li-rer škaš-šur ik-ri-bi-šu i-šem-mq
```

(b) Final (fine-tuned) CuReD model output

Figure 5: Comparison of source image with CuReD OCR output.

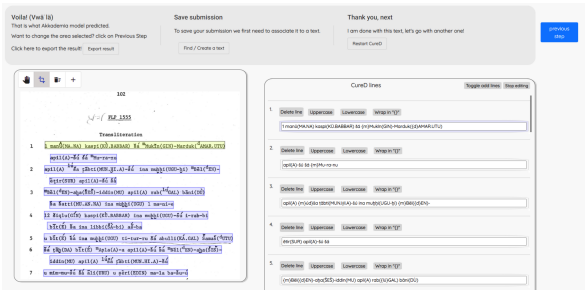


Figure 6: The CuReD tool interface.

for correcting the initial results of the OCR model presented above, and fine-tuning the model on new, unfamiliar types of transliterations.

There, users can upload currently one-by-one a PDF or image, place a bounding box around a text. Then lines of transliteration are automatically identified, and a corresponding line by line digital text is generated that can be manually edited for corrections (Fig. 6). Corrected text is saved for fine-tuning the model at a later stage, and the machine-readable output can be downloaded immediately as plain text. All OCR'ed transliterations are also searchable in the Babylonian Engine gallery.

The ML models are envisioned as "co-workers" which provide likely suggestions to the user, aiding the process of cuneiform scholarly edition publication, and improving as the user corrects them. This way, it is not only the ML models that benefit from the corrections and labeled data created by experts, but also the experts can enjoy a designated work environment for cuneiform studies, and download the results of their work–already advancing cuneiform scholarship.

In what follows we present two real-world scenarios of cuneiform scholarship: text editions published in book form, and legacy materials in the form of lexical cards. Both were created with typewriter in the late 1970's and early 80's of the 20th century.

4.2 Experiment 1: Text editions

We chose to digitize the texts edited in the dissertation of Raymond B. Dillard (1975). Namely, 81 Neo-Babylonian archival and administrative documents from the Free Library of Philadelphia (FLP), purchased on the antiquities market in the early 20th century by John Frederik Lewis.

Why Dillard? First, these texts are not digitized on any of the large online databases, such as CDLI, Achemenet, ORACC, or eBL. Second, it is a diverse corpus chronologically, geographically, and stems from a variety of archives (see metadata file on GitHub).

We initially had quite poor results of 53% accuracy, but after correcting only 10 texts, the OCR model reached 85% accuracy. Additional training on 47 texts increased the model's performance only incrementally to 89%. Thus, similarly to our initial fine-tune phase, the model requires a minimal number of ca. 10 documents in order to be a significant assistant in the digitization process of ancient texts (Fig. 7).

4.3 Experiment 2: Legacy collections

The Sumerian Lexicography collection is housed in the Babylonian Section of the University of Pennsylvania Museum of Archaeology and Anthropology. This collection consists of approximately 200,000 index cards (see Fig. 1) compiled by Å. W. Sjöberg in the late 1970's and early 1980's. These cards serve as the foundation for the intended Pennsylvania Sumerian Dictionary (PSD). No other collection of lexicographic cards in the field of Sumerian Lexicography matches its scale.

The PSD was never completed. From 1984 to 1992, only the letters A-B were published. In May 2004, the project transitioned to a digital format, evolving into the electronic Pennsylvania Sumerian

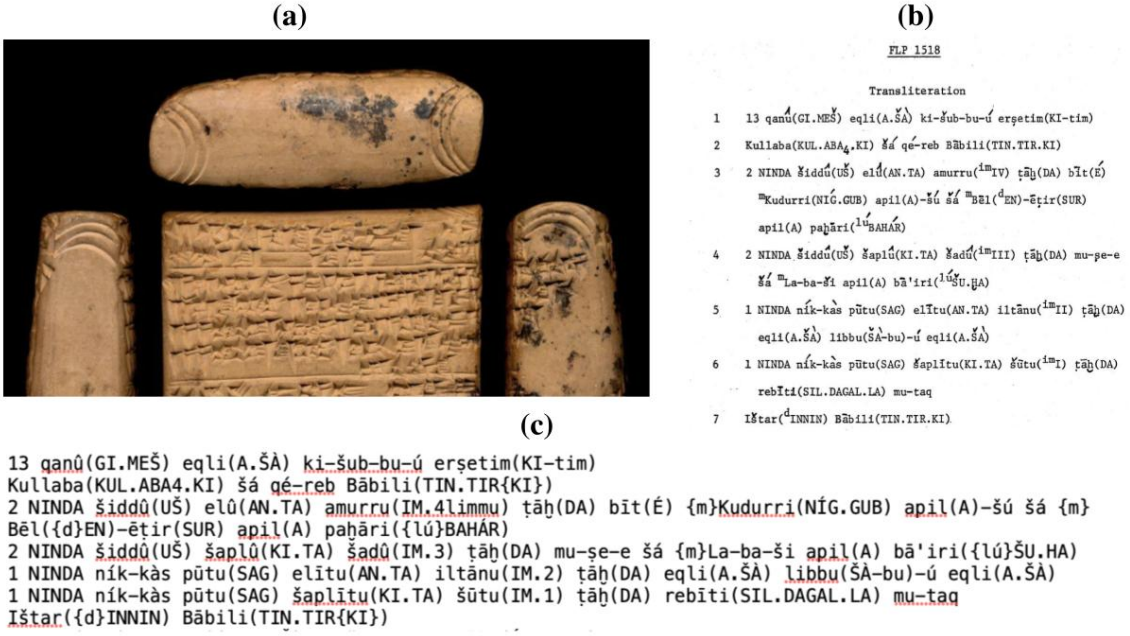


Figure 7: (a) Neo-Babylonian cuneiform tablet from the Free Library of Philadelphia collection; (b) Its transliteration in Dillard (1975); (c) Its plain text output from CuReD.

Dictionary (ePSD). It has undergone significant changes in editorial principles, replacing the manually compiled catalog with a much larger digital corpus.

The index card collection was digitized in May 2023 by Dr. Anna Glenn on behalf of the Institute for Assyriology und Hittitology of the LMU Munich. Hosted by the university library, this digitized collection forms the dataset for the study case presented here (Sjöberg, 2023).

The project plans to convert these scans into machine-readable text, and to link the results with other lexical collections, as part of the eBL platform. For test purposes, we uploaded to CuReD in the first step 30 cards, a little bit more than 60 lines, and corrected the results for fine-tuning. Prior to the training, the accuracy level stood at 87%. Note that the model at this time was already fine-tuned on the texts published by Dillard (see above). Following the first training session on this corpus, the accuracy improved to 94%.

Although the results improved significantly, a new issue emerged: the OCR fails to recognize a line when it consists of only a single word (compare to other lexical index cards digitized by Idziak et al. (2021)). This is particularly critical because many lemmas, i.e., lexemes in the Sumerian language, are made up of a single phoneme, that is, one letter.

5 Related Literature

To the best of our knowledge, this is the first custom-trained OCR model for transliterated cuneiform documents, trained initially on transliterations. See, however, the Tesseract-based model used for OCR'ing secondary literature in assyriology, which includes text editions (Anderson, 2023).

Human-in-the-loop pipelines for transcribing historical and epigraphical documents from other periods, however, are revolutionizing how those are being recorded and studied in the humanities and the galleries, libraries, archives, and museums (GLAM) sector. Some of the most impactful tools in this regard are Transkribus and eScriptorium, each of which has produced hundreds of studies based on their OCR/HTR engine, and several more are on the rise (Idziak et al., 2021; Nockels et al., 2022, 2024; Calvelli et al., 2023).

It is important to separate, however, the OCR/HTR efforts from Latin transliterations and OCR/HTR of cuneiform signs themselves on clay tablets, stone inscriptions, etc. Identifying cuneiform signs requires other designated models, and the several advances in recent years are summarized in Bogacz and Mara (2022); see also the newly published contribution by Yugay et al. (2024).

The high-performance of our model on minimal ground-truth data was possible due to the rel-

ative simplicity of generating representative artificial training data. Improvements in recent years in the generation of data that is similar enough to ground truth is proving more and more vital in aiding the digitization of low-resource languages, such as cuneiform (Rusakov et al., 2019) and Aramaic inscriptions (Aioanei et al., 2024), to name a few. These methods are probably to be vital in the upcoming years to push forward the digitization of ancient languages (Sommerschield et al., 2023).

Although our work only covers OCR digitization of transliterations as printed in published sources, there has also been work on automatic conversion of such transliterations to phonological transcriptions representing how texts were pronounced in the Akkadian language. See Sahala et al. (2020) for an example of a such deep-learning based model.

6 Conclusion

To aid the community of cuneiform experts in digitizing published records of cuneiform texts, we developed an OCR system for recognizing Akkadian Latin transliterations written using standard scholarly conventions. Because of a lack of natural labelled training data, we bootstrapped an OCR model using the Kraken open-source framework by generating artificial training data, rendering text from the SAA corpus using various fonts and text styles. After fine-tuning the resulting model on a small set of manually-labelled scans, we achieved 89% accuracy on a representative set of scans.

We integrated this model in a human-in-the-loop tool called CuReD (Cuneiform Recognition Documents), to allow scholars and students to OCR various scanned or photographed materials, and help continuously improve their model. We further showed this tool in practice, by performing two real-world experiments OCR'ing text editions and legacy lexical materials in machine typeface, both of which included handwritten notation. The fine-tuning of the two experiments was integrated into our model, which is also on the CuReD online tool, making it already highly effective for OCR'ing machine typed transliterations. Minimal fine-tuning is needed to improve its results on unseen texts, and the same should hold true for transliterations of other languages using the cuneiform script.

We provide this model as an open-source contribution to researchers of the ancient Near East and the general public, in hopes that it will make cuneiform inscriptions more accessible in machine-readable form.

Limitations

Our current OCR system has been tested only on Latin transliterations of Akkadian and Sumerian cuneiform texts, but not on the other languages of the ancient Near East using the cuneiform script. While we assume this transfer learning would be easy for the model given the similarities in the transliteration practices (see Appendix A), that remains to be seen.

Additionally, both experiments show how the model can be effectively fine-tuned with few examples to drastically improve performance. However, the results are never perfect. A common challenge in both experiments is the presence of many handwritten notes, such as accents, subscripts, diacritics, special characters, square brackets, or simply marginalia scribbled around the text. These factors lead to inaccuracies in the OCR results, particularly creating errors in the line segmentation.

The CuReD model, with its human-friendly interface, permits users to quickly correct the remaining errors. The fine-tuning process makes the correction phase extremely efficient. It may not completely make typing of editions a thing of the past, but it reduces the time by at least 90%. In addition, further manual improvements can be considered, such as validating the OCR'ed results against known cuneiform sign readings, or combining CuReD with Handwritten Text Recognition (HTR) (Nockels et al., 2022) to identify marginalia etc.

Furthermore, the continual fine-tuning of the model makes it familiar with additional typefaces and editorial conventions. The significant uptick in accuracies before fine-tuning between the experiments (from 53% on the Dillard texts to 87% on the PSD card catalogue) shows this quality, as both experiments share a similar typeface. Initial results on unseen texts will thus continue to improve as more corpora are added for training, and fewer and fewer examples will be required for fine-tuning.

Ethics Statement

The training data used in this work consists of publicly available scholarly publications and does not contain any sensitive personal information. The resulting OCR system is intended as a tool to aid scholarly research and all code and data is made freely available under a CC-BY 4.0 license. We do

not anticipate any major ethical concerns stemming from this work.

Acknowledgements

We would like to thank the Babylonian Section of the University of Pennsylvania Museum of Archaeology and Anthropology for providing access to their Sumerian Lexicography card collection, to Enrique Jiménez for suggesting this case study for CuReD, and to Anna Glenn for digitizing the cards.

References

Andrei C. Aioanei, Regine R. Hunziker-Rodewald, Konstantin M. Klein, and Dominik L. Michels. 2024. Deep Aramaic: Towards a synthetic data paradigm enabling machine learning in epigraphy. *PLOS ONE*, 19(4):1–29. Publisher: Public Library of Science.

Adam Anderson. 2023. FactGrid Cuneiform Discovery Project: Building Linked Open Data Repositories.

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

Lorenzo Calvelli, Federico Boschetti, and Tatiana Tommasi. 2023. EpiSearch. Identifying Ancient Inscriptions in Epigraphic Manuscripts. *Journal of Data Mining & Digital Humanities*, Historical Documents and...(Sciences of Antiquity and...):10417.

Raymond B. Dillard. 1975. *Neo-babylonian texts from the john frederick lewis collection of the free library of philadelphia*. Doctoral Dissertation, The Dropsie University, Philadelphia, PA.

Ignace J. Gelb. 1970. Comments on the akkadian syllabary. *Orientalia*, 39:516–546.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks.

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.

Timo Homburg, Tim Brandes, Eva-Maria Huber, and Michael A. Hedderich. 2023. From an Analog to a Digital Workflow: An Introductory Approach to Digital Editions in Assyriology. *Cuneiform Digital Library Bulletin*, 2023(4). Publisher: Cuneiform Digital Library Initiative.

Jan Idziak, Artjoms Šeļa, Michał Woźniak, Albert Leśniak, Joanna Byszuk, and Maciej Eder. 2021. Scalable Handwritten Text Recognition System for Lexicographic Sources of Under-Resourced Languages and Alphabets. In *Computational Science – ICCS 2021*, pages 137–150, Cham. Springer International Publishing.

Jiří Martínek, Ladislav Lenc, and Pavel Král. 2020. Building an efficient OCR system for historical documents with little training data. *Neural Computing and Applications*, 32(23):17209–17227.

Joe Nockels, Paul Gooding, Sarah Ames, and Melissa Terras. 2022. Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of Transkribus in published research. *Archival Science*, 22(3):367–392.

Joseph Nockels, Paul Gooding, and Melissa Terras. 2024. The implications of handwritten text recognition for accessing the past at scale. *Journal of Documentation*, 80(7):148–167. Publisher: Emerald Publishing Limited.

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.

Simo Parpola. 2015. *The correspondence of Sargon II, Part I: Letters from Assyria and the West*, reprinted edition. Number 1 in State Archives of Assyria. Eisenbrauns, Winona Lake, Indiana.

Karen Radner, Jamie Novotny, and et al. 2015. State Archives of Assyria Online (SAAo). Publisher: The SAAo Project.

Maxim Romanov, Matthew Thomas Miller, Sarah Bowen Savant, and Benjamin Kiessling. 2017. Important new developments in arabographic optical character recognition.

Eugen Rusakov, Kai Brandenbusch, Denis Fisseler, Turna Somel, Gernot A Fink, Frank Weichert, and Gerfrid GW Müller. 2019. Generating cuneiform signs with cycle-consistent adversarial networks. In *Proceedings of the 5th international workshop on historical document imaging and processing*, pages 19–24.

Aleksi Sahala and Krister Lindén. 2023. A neural pipeline for POS-tagging and lemmatizing cuneiform languages. In *Proceedings of the ancient language processing workshop*, pages 203–212, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Aleksi Sahala, Miikka Silfverberg, Antti Arppe, and Krister Linden. 2020. Automated phonological transcription of akkadian cuneiform text.

Åke W. Sjöberg. 2023. The university of pennsylvania collection of sumerian lexicography.

Gustav Ryberg Smidt, Katrien De Graef, and Els Lefever. 2024. At the Crossroad of Cuneiform and NLP: Challenges for Fine-grained Part-of-Speech Tagging. European Language Resources Association.

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.

Juan-Pablo Vita. 2021. *History of the Akkadian language*. Number 152 in Handbook of Oriental studies. Brill, Leiden.

Wolfram von Soden. 1995. *Grundriss der akkadischen grammatik*, 3 edition. Editrice Pontificio Instituto Biblico, Rome.

Yuxin Wu and Kaiming He. 2018. Group normalization.

Vasiliy Yugay, Kartik Paliwal, Yunus Cobanoglu, Fabian Simonjetz, Luis Sáenz, Ekaterine Gogokhia, Shai Gordin, and Enrique Jiménez. 2024. Stylistic Classification of Cuneiform Signs Using Convolutional Neural Networks. *it - Information Technology*. Publisher: De Gruyter Oldenbourg.

A Appendix: Akkadian Latin Transliteration

We include here a short description of the main features of the conventions used for Akkadian Latin transliteration. This standard system was described in Gelb (1970) and von Soden (1995), each with some modifications. It is in large part also used for other languages written in the cuneiform writing system, most notably Sumerian, but also with necessary modifications for Eblaite, Elamite, Hurrian, Urartian, Hittite, Luwian and several minor Anatolian languages written in cuneiform (like Hattian).

Besides the usual characters of the Latin alphabet, cuneiform transliterations can contain the following special characters used to represent particular sounds:

- Š š, equivalent to the English *sh*-sound
- □ □, equivalent to the *ts*-sound
- □ □, an emphatic *t*-sound (e.g. **th**eatre)
- □ □, the voiceless uvular fricative (e.g. German a**ch**t)
- □ and □, aleph (glottal stop) and ayin (pharyngeal fricative), respectively
- G̃ g̃, nasal g (*ng*-sound)
- Ř ř, alveolar trills (see řeka)

Cuneiform symbols may be used phonetically to represent syllables with structure V, VC, CV, or CVC. When used in this way, the transliteration of these signs is written in italic lowercase letters with dashes separating syllables of the same word. For example, the word *iddin* 'he gave' may be written phonetically as *id-din*, *id-di-in*, or using other variants.

Uppercase, normal-style (i.e. non-italic) letters are used to represent logograms; cuneiform symbols representing words or morphemes rather than phonetic values. Some editions represent the logographic values in small caps instead. The text in uppercase represents the reading of the logogram in Sumerian, from which it was borrowed, although the Akkadian speaker would have probably read it in their native language. For example, the transliteration DINGIR represents a cuneiform sign that would have been read in context as Akkadian *ilu* ("god"). Logogram compounds are separated with periods in transliterations; for example, DUMU.MUNUS-*ia* "my daughter".

The Sumerian language for which cuneiform was originally developed had a large number of homonymic symbols (symbols with the same phonetic value). In order to distinguish these in transliteration, scholars use accents and subscript digits. For example, *gu, gú, gù* represent three different cuneiform symbols with the same pronunciation *gu*; the fourth such symbol and onwards would be notated as gu_4, the fifth as gu_5, and so on. Newer resources may only use superscript numbers instead of accents (gu_2, gu_3). Many homonymic readings are used simultaneously in cuneiform languages.

Superscript symbols are used to represent determinatives, also known as classifiers, which are cuneiform signs that do not have an independent reading but rather clarify the meaning of following or preceding sign(s). For example, superscript *d* represents the determinative indicating a divine name, and superscript *m* indicates a male name.

Since cuneiform inscriptions are often broken or not fully legible, a number of special symbols are used to indicate textual anomalies. The most common of these are:

- Square brackets [ ] - used to indicate missing signs, such as when there is a hole in the text. May contain editorial guesses as to the missing contents, or X to indicate a missing sign.
- Half brackets ⌜ ⌝ - indicate fragmentary but legible signs

- Superscript ! - indicates a scribal error
- Superscript ? - indicates an uncertain sign
- angle brackets < > - used to add signs that the modern editor thinks the ancient scribe has omitted.
- double angle brackets « » - indicate signs which the modern editor thinks the ancient scribe has erroneously added, and believes should be ignored for phonetic and linguistic reconstruction.

The notation system for homonymic signs and editorial marks for textual anomalies are shared across the transliteration conventions of texts written in the cuneiform script, as well as combinations of lowercase, uppercase, and italics. Furthermore, the Sumerian readings of logograms are shared across the many languages written in the cuneiform script. Thus, CuRed is likely to be an efficient baseline model of transliterations from other cuneiform languages.