

HieroLM: Egyptian Hieroglyph Recovery with Next Word Prediction Language Model

Stanford CS224N Custom Project

Xuheng Cai

Department of Computer Science
Stanford University
xuheng@stanford.edu

Erica Zhang

Department of Management Science
and Engineering
Stanford University
yz4232@stanford.edu

Abstract

Egyptian hieroglyphs can be found on numerous unearthed ancient Egyptian artifacts, but it is common that they are blurry or even missing due to natural erosion. Most existing efforts to restore blurry hieroglyphs adopt computer vision techniques and model hieroglyph recovery as an image classification task. Unfortunately, the CV-based approaches suffer from two major limitations: (i) They cannot handle severely damaged or completely missing hieroglyphs. (ii) They make predictions based on a single hieroglyph without considering contextual and grammatical information. This project proposes a novel approach to model hieroglyph recovery as a next word prediction task and use language models to assist the recovery process. Leveraging the strong local affinity of semantics in Egyptian hieroglyph texts, we propose a hieroglyph language model HieroLM based on LSTM-enhanced recurrent neural networks. Extensive experiments show that HieroLM not only achieves over 44% accuracy on next word prediction, but also maintains notable accuracy on multi-shot predictions. With its strong language modeling capability, HieroLM is a useful tool to assist scholars in inferring missing hieroglyphs. It can also complement CV models to significantly reduce perplexity in recognizing blurry hieroglyphs.

1 Team Information


- TA mentor: Anna Goldie
- External collaborators: No
- External mentor: No
- Sharing project: No

2 Introduction

As the formal written language and an important medium for religious and funerary practices in Ancient Egypt, the Egyptian hieroglyphs can be found on numerous ancient Egyptian artifacts. The process of decoding hieroglyphs involves first converting them into transliterations and then translating the transliterations into modern languages (Gardiner, 1927). Table 1 presents an illustration of this process.

Due to natural erosion, it is common that the hieroglyphs on the surface of the unearthed artifacts are blurry or even missing. Efforts have been made to assist the process of recognizing blurry hieroglyphs with computer vision (CV) techniques (Barucci et al., 2021, 2022; Aneesh et al., 2024). Specifically, these works formulate hieroglyph recognition as an image classification task and use state-of-the-art CV models such as Convolutional Neural Networks (CNNs) to classify the blurry signs. However, there are two major limitations in the CV-based approaches: (i) They cannot handle severely damaged or completely missing

Table 1: Example transliteration and translation of a hieroglyphic sentence.

Hieroglyphs	
Transliteration	wbn r ^c m 3ht
Transliteration (MdC)	wbn ra m Axt
English Translation	Re (the Sun God) rises in the horizon.

hieroglyphs because they rely on the visual characteristics of the signs. **(ii)** They make predictions based on a single hieroglyph, without considering the contextual and grammatical information contained in surrounding words that could help significantly reduce perplexity.

As an example, the blurry hieroglyph A in the blue box in Figure 1 would confuse a CV model, because it could be either \dagger (nhb) or \ddagger (sw) based on its vague shape, but from the surrounding words we know that this sentence describes an offering by the king to the god Osiris, so it is likely that this blurry sign is \ddagger (sw), which means "the king". Moreover, for the red box in Figure 1, the signs are almost entirely missing, and the CV models will become useless, but from the words before it, we know that it should be a title of Osiris, which indicates that the missing word is probably \ddagger (ddw), because \cup \ddagger (nb ddw; "lord of Djedu") is a common title for Osiris in the offering formula.

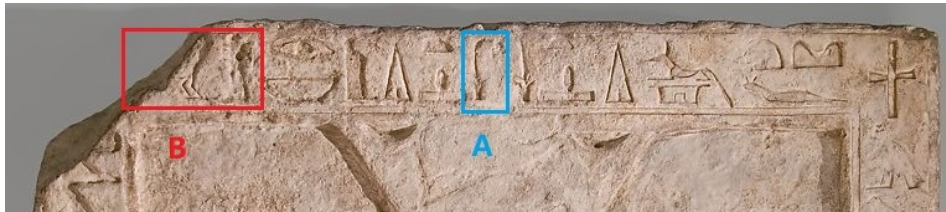


Figure 1: A Middle Kingdom tablet at The Metropolitan Museum of Art.[†] Hieroglyph A in the blue box is an example of blurry hieroglyphs. Hieroglyph B in the red box is an example of (nearly) missing hieroglyphs. [†] Source: <https://www.metmuseum.org/art/collection/search/545055>.

In light of the abovementioned limitations, we propose a novel approach where we model hieroglyph recovery as a next word prediction problem, which can be addressed effectively with language models. Currently, there are two major architectures for language models: (i) Recurrent architectures. Representative models include Recurrent Neural Network (RNN) (Medsker and Jain, 1999) and its variants enhanced with Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). These models demonstrate strong ability in extracting short-range dependencies and are naturally biased towards local information. Thus, they are often preferred when the sequences are short (Zhao et al., 2024) or when the process is near-Markov, such as stock price prediction (Bilokon and Qiu, 2023) and climate forecasting (Buestán-Andrade et al., 2023). (ii) Transformer-based architectures. With self-attention, Transformers (Vaswani et al., 2017) gain superiority in processing long sequences, modeling long-range dependencies, and scaling up with parallel computing (Al-Selwi et al., 2024). However, Transformer models require larger-scale datasets (Ezen-Can, 2020), are more difficult to train (Popel and Bojar, 2018) and sometimes less robust (Liu et al., 2022).

To select the best architecture for our task, we consider the following characteristics of Egyptian hieroglyphs (Allen, 2000): (i) Egyptian hieroglyphs is a dead language, and thus the available data is very limited. (ii) It is mostly used in limited scenarios, such as funerals, religious rituals, and monumental inscriptions. (iii) Due to its limited scope of usage, its sentence structure is less diverse and has strong local affinity (e.g., a large portion of a sentence could be titles following names of gods or kings). Based on these characteristics, we build our proposed HieroLM based on an LSTM-enhanced RNN architecture. To validate our choice, we compare the performance of a Transformer-based model with our model in Section 5.5, and analyze the reasons behind the failure of Transformer in Section 6.4.

Our contributions can be summarized as follows:

- We introduce a novel approach to hieroglyph recovery by modeling it as a next word prediction task and addressing it with state-of-the-art language models.
- We build a hieroglyph language model HieroLM based on LSTM, which achieves over 44% prediction accuracy. The trained model is released to benefit the community.
- Extensive experiments show that HieroLM demonstrate notable performance on both next word and multi-shot predictions, making it a useful tool in inferring missing words and complementing CV models to reduce perplexity in blurry hieroglyph recognition.

3 Related Work

Most existing works on Egyptian hieroglyph recognition and recovery with machine learning adopt computer vision techniques like CNNs. To the best of our knowledge, this is the first attempt to model hieroglyph recovery as a next word prediction task using language models. For comprehensiveness, we will first review related research in CV for hieroglyph recognition in Section 3.1, and then we introduce state-of-the-art language models in Section 3.2.

3.1 Hieroglyph Recognition with Computer Vision

Modeling hieroglyph recognition as an image classification task is a well-explored direction. Franken and van Gemert (2013) proposed to use the Histogram of Oriented Gradients (HOG) and the Shape-Context (SC) descriptors to extract the shapes of the hieroglyphs and compare them with labeled data. The HOG method was later enhanced with Region of Interest (ROI) extraction by Elnabawy et al. (2021). Moustafa et al. (2022) explored the performance of ShuffleNet, MobileNet, and EfficientNet on hieroglyphs recognition. Aneesh et al. (2024) evaluated ResNet, VGG, DenseNet, and Inception v3 on the same task. With an architecture designed specifically for hieroglyph recognition, Glyphnet (Barucci et al., 2021) achieved the state-of-the-art performance in classification accuracy. However, all of these computer vision models rely heavily on the visual quality of the signs, and fail to incorporate contextual and grammatical information from surrounding words.

3.2 Language Models for Next Word Prediction

Next-word prediction involves predicting the subsequent word in a sequence given the preceding context. Originating from pioneering work on information theory by Shannon (1948, 1951), it is foundational for applications like text generation, auto-completion, and machine translation. Early approaches use n-gram models that, despite their simplicity, suffer from data sparsity and limited context understanding.

The introduction of Neural Probabilistic Language Models (NPLM) (Bengio et al., 2000) addresses the limitations of n-gram models by learning distributed word representations and using neural networks to model the probability distribution of the next word. Subsequent enhancements included continuous space language models (CSLM) (Schwenk, 2007), which incorporates recurrent structures to handle variable-length contexts. This paved the way for Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), which improve next-word prediction by maintaining memory and capturing longer dependencies. The effectiveness of RNNs for next-word prediction is widely demonstrated (Mikolov et al., 2010; Sutskever et al., 2011).

The Transformer architecture (Vaswani et al., 2017) revolutionizes language modeling by employing self-attention mechanisms, allowing the model to consider the entire input context simultaneously. This leads to substantial improvements in performance and scalability. Building on the Transformer architecture, large-scale pre-trained models like Generative Pre-trained Transformer (GPT) (Radford et al., 2018, 2019) demonstrated the power of pre-training on vast corpora followed by fine-tuning. Due to the restricted data availability for Egyptian hieroglyphs, we are unable to leverage the large-scale pre-trained models for our task. Instead, we experiment with state-of-the-art non-pretrained models such as NPLM, LSTM, RNN, as well as a simple Transformer Encoder model.

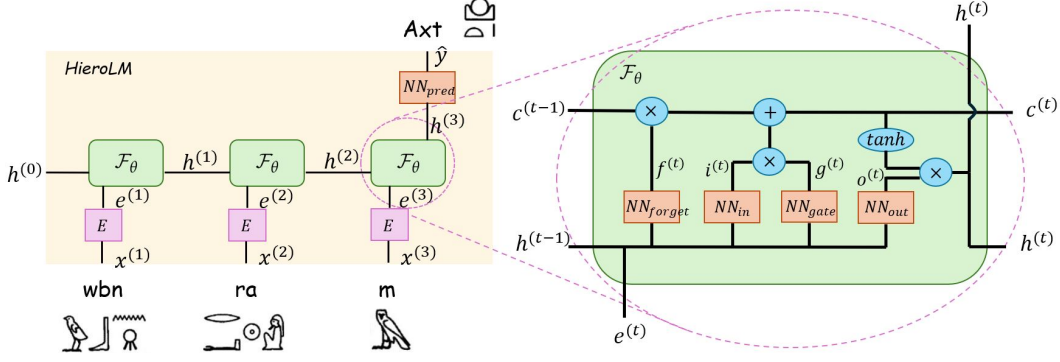


Figure 2: Model structure of HieroLM.

4 Methodology

In this section, we describe in detail our proposed HieroLM model, which adopts a recurrent architecture enhanced by LSTM. The overall model structure is illustrated in Figure 2.

4.1 Recurrent Structure for Local Affinity Modeling

Assume that the input sentence has T words. Let $x^{(t)} \in \{0, 1\}^{|V|}$ be the one-hot encoding of the t -th word ($1 \leq t \leq T$) in the sentence. Then, its embedding $e^{(t)} \in \mathbb{R}^s$, where s is the embedding size, is computed through an embedding layer: $e^{(t)} = Ex^{(t)}$. Following the common practice of RNN-based models (Medsker and Jain, 1999), the hidden state $h^{(t)} \in \mathbb{R}^d$, where d is the hidden dimension size, at step t is computed as:

$$h^{(t)} = \mathcal{F}_\theta(h^{(t-1)}, e^{(t)})$$

where \mathcal{F}_θ is a parameterized transformation, and $h^{(0)}$ is the initial hidden state. Then, the predicted output is calculated by:

$$\hat{y} = NN_{pred}(h^{(T)})$$

where NN_{pred} is a single neural layer plus a softmax layer, which projects the final hidden state from the hidden dimension d to the size of the vocabulary $|V|$.

4.2 Long Short-Term Memory for Long-range Perception

Recurrent models are often overly biased towards the last few words. To mitigate this issue, we incorporate Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) in our parameterized transformation \mathcal{F}_θ . Specifically, given $h^{(t-1)}$ and $e^{(t)}$, we compute the following interim states with single layer NNs:

$$\begin{aligned} f^{(t)} &= NN_{forget}(h^{(t-1)}, e^{(t)}) & i^{(t)} &= NN_{in}(h^{(t-1)}, e^{(t)}) \\ g^{(t)} &= NN_{gate}(h^{(t-1)}, e^{(t)}) & o^{(t)} &= NN_{out}(h^{(t-1)}, e^{(t)}) \end{aligned}$$

The cell state $c^{(t)} \in \mathbb{R}^d$ is computed as:

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot g^{(t)}$$

where $c^{(0)}$ is the initial cell state. Finally, the hidden state $h^{(t)}$ is calculated as:

$$h^{(t)} = o^{(t)} \odot \tanh(c^{(t)})$$

5 Experiments

5.1 Baselines

We compare our LSTM-based HieroLM model with the following baselines:

- *Neural Probabilistic Language Model (NPLM)* (Bengio et al., 2000). We use a trigram NPLM as the naivest baseline.
- *Recurrent Neural Network (RNN)* (Medsker and Jain, 1999). We adopt a unidirectional, single-layer RNN, with the same form of input and output as the above in LSTM.
- *Transformer Encoder (TE)* (Vaswani et al., 2017). Since our task requires outputting a single word for a sequence of input, We adopt the encoder part of the Transformer architecture. Specifically, we employ a single-layer encoder with `nheads=16` and `dropout = 0`.

5.2 Data

We evaluate the language models on three datasets, with statistics summarized in Table 2:

- *Ancient Egyptian Sentences (AES)* (Jauhiainen and Jauhiainen, 2023): It is a collection of over 100,000 ancient Egyptian sentences across dynasties.
- *The Ramses Transliteration Corpus* (Rosmorduc, 2020): It contains transliterations converted from a large corpus of Late Egyptian sentences.
- *Mixed*: Since AES contains sentences from different eras while texts in Ramses come from Late Egypt, their distributions are different due to language evolution. To evaluate cross-distribution modeling ability, we synthesize AES and Ramses into a mixed dataset.

We use MdC transliteration throughout our experiments because it replaces irregular letters (e.g., `ⲥ` and `ⲑ`) in the common transliteration with English letters for convenient processing. The datasets are processed as follows: Firstly, the MdC transliterations of the sentences are extracted. Then, sentences with only one word are filtered out because they cannot be used for next word prediction. Finally, the sentences are split into training, validation, and test sets by an 8:1:1 ratio. During model training, for each sentence of length n , the first $n - 1$ words will be used as the source sequence, while the last $n - 1$ words are the target sequence.

Table 2: Dataset statistics.

Dataset	Sentence #	Vocab #	Training #	Validation #	Test #
AES	98,375	7,058	78,801	9,800	9,774
Ramses	61,069	3,499	48,848	6,116	6,105
Mixed	159,444	8,436	127,649	15,916	15,879

5.3 Evaluation Metrics

We evaluate the models on three metrics:

- *Perplexity*. It measures the model’s probability of predicting the correct word. A lower perplexity score indicates better predictive performance.
- *Accuracy*. It is the ratio between the number of correct predictions and the total predictions. It reflects the practical efficacy of our models in real-world application.
- *F1 Score*. This metric harmonizes precision and recall, providing a balanced view of performance across all classes. We use the "macro" averaging method in calculation.

5.4 Experimental Details

For fair comparison, we adopt embedding size 1024 and hidden dimension size 1024 for HieroLM and all the baselines, based on the hyperparameter analysis in Section 6.2. The dropout rate is searched specifically for each dataset. We employ a learning rate decay and early stopping strategy, such that when the validation perplexity stops decreasing for 5 epochs, the learning rate decays by half, and when the learning rate has been decayed five

times before the maximum number of epochs, the training will stop early. We utilized the PyTorch versions of RNN and LSTM, and reused some pipeline code in Assignment 3.

5.5 Results

We summarize the experimental results in Table 3, with the following observations:

- **Recurrent architecture dominates.** As the table shows, models with recurrent architecture (RNN and HieroLM) exhibit consistent superiority over the others. This demonstrates the recurrent models’ ability to capture local affinity of semantics in hieroglyphs.
- **Long Short-Term Memory enhances performance.** The comparison between HieroLM and RNN is a natural ablation study. The outperformance of HieroLM w.r.t. RNN proves that LSTM can enhance the model’s ability by long-range perception.
- **Transformer is ill-suited for this task.** We can see that TE underperforms both RNN and HieroLM, which is discussed in more details in Section 6.4.

Table 3: Performance results.

Dataset	Metric	NPLM	TE	RNN	HieroLM
AES	Perplexity	41.57	52.21	42.25	26.50
	Accuracy	0.3075	0.3143	0.3828	0.4525
	F1 Score	0.0485	0.0488	0.1201	0.1420
Ramses	Perplexity	28.75	38.59	31.89	21.59
	Accuracy	0.3553	0.3727	0.4387	0.4895
	F1 Score	0.0775	0.0905	0.1933	0.2074
Mixed	Perplexity	42.14	53.78	43.34	26.48
	Accuracy	0.3022	0.3151	0.3801	0.4450
	F1 Score	0.0481	0.0466	0.1377	0.1421

6 Analysis

6.1 Multi-shot Prediction Performance

In reality, it is common for a number of contiguous hieroglyphic words to be missing together. Thus, it is important to evaluate the model’s ability to predict a series of words accurately without teacher forcing. Figure 3 presents the prediction accuracy of HieroLM for multiple following words. We can observe a favorable diminishing decrease in accuracy with the increase of prediction range. The model maintains an accuracy of over 13% on predicting 4 words in a row. This robust multi-shot prediction performance proves the pragmatic value of HieroLM in assisting the recovery of missing hieroglyphs.

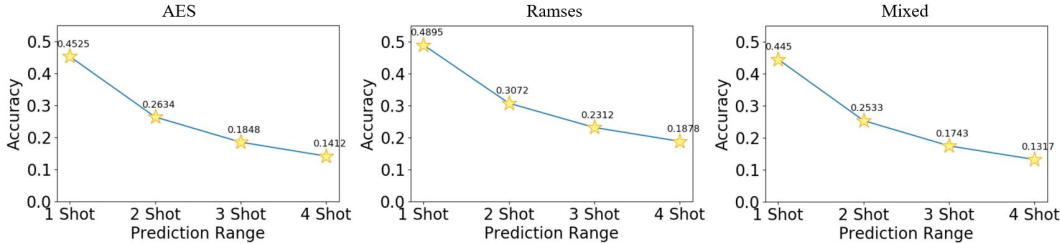


Figure 3: Multi-shot prediction accuracy on three datasets.

6.2 Hyperparameter Analysis

In this section, we present the sensitivity of HieroLM with respect to key hyperparameters including embedding size, hidden dimension size, and dropout rate. The results, as summarized in Figure 4, also provides ground for our selection of model hyperparameters.

As expected, larger embedding size and hidden dimension size generally lead to better performance. However, the performance gain diminishes for embedding size and hidden dimension larger than 1024. To keep the models in reasonable complexity, we choose 1024 as the final embedding size and hidden dimension. The right column of Figure 4 indicates that the performance of HieroLM is relatively stable across different dropout rate.

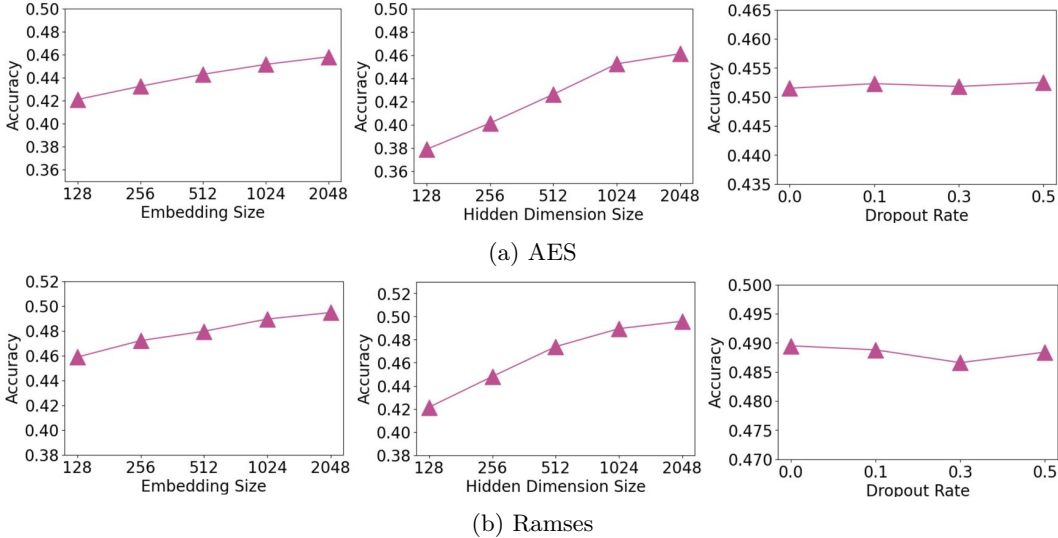


Figure 4: Test accuracy w.r.t. embedding size, hidden dim size, and dropout rate.

6.3 Case Study

We demonstrate HieroLM’s ability to learn semantic patterns in hieroglyphs by two cases.

Case 1: Offering formula. Below is the #1563 sentence in the test set of the Mixed dataset. The different sense groups in the sentence have been highlighted with different colors.

Processed MdC: n kA n wr swN w pnTw mAa xrw

Transliteration: n k3 n wr-swn.w pnṯw m3ꜥ ḥrw

English Translation: For the ka of the great physician Pentu , the true of voice.¹

This sentence is a common component of the offering formula, usually appearing at the end of the offering inscriptions. It has a fixed format: $[n\ k3\ n] + [Title\ and\ name\ of\ the\ deceased] + [m3ꜥ\ ḥrw]$, where $m3ꜥ\ ḥrw$ (“the true of voice”) is a universal title for the deceased. Therefore, upon seeing $n\ k3\ n$ and the title and name of the deceased, HieroLM is capable of predicting that the following words should be $m3ꜥ\ ḥrw$. When we input the sequence “ $n\ kA\ n\ wr\ swN\ w\ pnTw$ ”, the model outputs “ mAA ”, and when appending “ mAA ” to the input, it outputs “ xrw ”. Note that this is also a successful case of 2-shot prediction.

Case 2: Titles of kings. Below are the first few words of #8779 sentence in test set of Mixed.

Processed MdC: nswt bj tj nb tA du wsr mAa t raw stp n jmn zA ra ...

Transliteration: nswt-bity nb t3.du wsr-m3ꜥt-rꜥ stp.n-imm s3 rꜥ ...

English Translation: King of Upper and Lower Egypt, Lord of the Two Lands, Ramesses IV, Son of Re ...

This part of the sentence is the name and titles of the king Ramesses IV. Titles of kings in ancient Egypt have rigorous formats. $nswt-bity$ (“King of Upper and Lower Egypt”) is the title preceding the coronation name of the king, and $s3\ rꜥ$ (“Son of Re”) is a title commonly following the king’s name. After seeing $nswt-bity$ and the name of the king, HieroLM can infer that the following words are likely to be $s3\ rꜥ$. When we feed in the sequence “ $nswt\ bj$

¹In ancient Egypt, *ka* refers to a part of human soul that leaves the body upon death.

tj nb tA du wsr mAa t raw stp n jmn”, the model responds with ”zA”, and when appending ”zA” to the input, it outputs ”ra”, which is again a 2-shot prediction example.

6.4 Limitations of Transformer-based Models with Small-scale Dataset

In this section, We analyze the surprising under-performance of Transformer-based models in our use case, which can be ascribed to three factors: (i) underfitting due to the mismatch between model size and data; (ii) lack of locality biases; and (iii) instability of gradient flow.

Transformers often suffer from underfitting on small-scale datasets. Kaplan et al. (2020) show that the number N of non-embedding parameters in a Transformer is approximately:

$$N \propto n_{\text{layer}} d_{\text{model}}^2,$$

where n_{layer} is the number of layers and d_{model} is the dimension of the residual stream. This quadratic dependence on the dimension of the residual stream means that even a moderately sized Transformer can have numerous parameters, leading to significant model complexity. On small datasets, this complexity results in insufficient learning of the underlying data distribution. To mitigate this issue, we set `n_layer` = 1 and `dropout` = 0 in the Transformer Encoder (TE) model we adopt. While this improves the performance, TE still exhibits more signs of underfitting than recurrent models. With close inspections at the predictions given by TE, we find that it tends to predict words with frequent occurrence in training data (i.e., prepositions such as `n` and `m`), which is apparently a sub-optimal solution. This blind preference towards prepositions also results in the low F1 score for TE in Table 3.

Transformers’ limitations on small datasets are widely discussed, especially on visual Transformers (ViT’s) (Shao and Bi, 2022; Li et al., 2021; Dai et al., 2021). While having superior global connectivity, Transformers lack locality biases, which are crucial for capturing local patterns in small datasets. The self-attention in Transformer is calculated as $Z = \text{softmax}(QK^T/\sqrt{d})V$, where Q, K, V are the projected query, key, and value matrices. Since Z is a weighted sum of value vectors in V , the self-attention mechanism inherently captures information from all tokens. Even with positional encoding, this mechanism does not prioritize local information, which is important in hieroglyphs due to local semantic affinity. This is in contrast with CNNs or RNNs whose architectures naturally incorporate local context. The convolution operation for a 1D CNN can be represented as:

$$y_i = \sum_{j=1}^k w_j x_{i+j-1}.$$

Here, CNNs admit local context through the convolution operation by focusing on a small window of input data. Shao and Bi (2022) show that integrating convolutional layers can improve Transformer’s ability to learn from limited data by local feature extraction.

The instability of gradient flow in deep models is also a significant factor for Transformers’ poor performance on small datasets (Xu et al., 2021). For a transformer with L layers, let δ_L be the gradient at layer l , and the gradient at the input layer δ_0 can be expressed as:

$$\delta_0 = \delta_L \Pi_{i=1}^L W_i.$$

where δ_L is the gradients at the final layer, and W_i is the weight matrix of the i -th layer. Hence, the magnitudes of the gradients can grow or shrink exponentially with the number of layers, causing significant instability in learning, especially on small-scale datasets.

7 Conclusion

In this project, we propose to model Egyptian hieroglyph recovery as a next word prediction task that can be addressed by language models. Specifically, we construct a hieroglyph language model with recurrent architecture enhanced with LSTM. Extensive experiments show that our model can achieve remarkable performance on both next word and multi-shot predictions, which makes it useful in archaeological practices to infer missing hieroglyphs and complement CV models to reduce perplexity in blurry hieroglyph recognition. However, the effective way of integrating CV models and language models into a unified hieroglyph recovery system remains largely unexplored, which is left for future work.

8 Ethics Statement

Our next-word prediction language model for ancient Egyptian hieroglyphs presents unique ethical challenges and societal implications. We discuss potential concerns and propose strategies for mitigation.

- *Cultural Sensitivity and Appropriation.* The interpretation and use of cultural heritage data, such as hieroglyphic texts, must be handled with sensitivity. There is a risk of cultural appropriation or misrepresentation when modern technologies are applied to ancient artifacts. To mitigate this, our project respects cultural significance and historical accuracy by testing our model in real-time on its predictions, monitoring its behavior to ensure that no culturally inappropriate misinterpretations are made.
- *Accuracy and Misinterpretation.* Given the ancient and often sacred nature of the texts being analyzed, inaccuracies in translation or prediction could lead to misinterpretations of historical facts. Currently, we mitigate this by monitoring evaluation metrics like accuracy rate, perplexity, and F1 score. Furthermore, we clearly communicate the probabilistic nature of our model’s predictions, emphasizing that they should be verified by human experts. In the future, as we make our model more robust and effective, we will implement rigorous validation of our model outputs with expert reviewers in the field of Egyptology to more properly address this concern. As Stanford does not have an Egyptology lab, we hope to reach out to Egyptologists such as Professor Carol Redmount and Dr. Rita Lucarelli at Berkeley’s Archaeological Research Facility (ARF).²
- *Impact on Archaeological and Linguistic Communities.* The automation of text prediction could impact traditional roles in archaeology and linguistics, possibly reducing opportunities for manual translation and analysis. While our project aims to support and enhance scholarly work, we recognize the importance of maintaining a balanced relationship with these fields. For current work, we plan to release a playground for the archaeological community to experiment with our language model and also for us to collect critical feedbacks. In the future, we will explore ways our technology can complement rather than replace traditional methods. This includes, for instance, providing tools for preliminary analysis that must be refined and verified by human experts. We will do this by engaging with the egyptologists mentioned in the bullet point above.

References

- Safwan Mahmood Al-Selwi, Mohd Fadzil Hassan, Said Jadid Abdulkadir, Amgad Muneer, Ebrahim Hamid Sumiea, Alawi Alqushaibi, and Mohammed Gamal Ragab. 2024. Rnnlstm: From applications to modeling techniques and beyond—systematic review. *Journal of King Saud University-Computer and Information Sciences*, page 102068.
- James P Allen. 2000. *Middle Egyptian: An introduction to the language and culture of hieroglyphs*. Cambridge University Press.
- NA Aneesh, Anush Somasundaram, Azhar Ameen, Govind Sreekar Garimella, and R Jayashree. 2024. Exploring hieroglyph recognition: A deep learning approach. In *2024 2nd International Conference on Computer, Communication and Control (IC4)*, pages 1–5. IEEE.
- Andrea Barucci, Chiara Canfailla, Costanza Cucci, Matteo Forasassi, Massimiliano Franci, Guido Guarducci, Tommaso Guidi, Marco Loschiavo, Marcello Piccolo, Roberto Pini, et al. 2022. Ancient egyptian hieroglyphs segmentation and classification with convolutional neural networks. In *International Conference Florence Heri-Tech: the Future of Heritage Science and Technologies*, pages 126–139. Springer.
- Andrea Barucci, Costanza Cucci, Massimiliano Franci, Marco Loschiavo, and Fabrizio Argenti. 2021. A deep learning approach to ancient egyptian hieroglyphs classification. *Ieee Access*, 9:123438–123447.

²See the official website at: <https://arf.berkeley.edu/region/egypt>.

- Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.
- Paul Bilokon and Yitao Qiu. 2023. Transformers versus lstms for electronic trading. *arXiv preprint arXiv:2309.11400*.
- Pablo-Andrés Buestán-Andrade, Matilde Santos, Jesús-Enrique Sierra-García, and Juan-Pablo Pazmiño-Piedra. 2023. Comparison of lstm, gru and transformer neural network architecture for prediction of wind turbine variables. In *International Conference on Soft Computing Models in Industrial and Environmental Applications*, pages 334–343. Springer.
- Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. 2021. Coatnet: Marrying convolution and attention for all data sizes.
- Reham Elnabawy, Rimón Elias, Mohammed A-M Salem, and Slim Abdennadher. 2021. Extending gardiner’s code for hieroglyphic recognition and english mapping. *Multimedia Tools and Applications*, 80:3391–3408.
- Aysu Ezen-Can. 2020. A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*.
- Morris Franken and Jan C van Gemert. 2013. Automatic egyptian hieroglyph recognition by retrieving images as texts. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 765–768.
- Alan Henderson Gardiner. 1927. *Egyptian grammar: being an introduction to the study of hieroglyphs*. Clarendon Press.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Heidi Jauhiainen and Tommi Jauhiainen. 2023. Transliteration model for egyptian words. *Digital Humanities in the Nordic and Baltic Countries Publications*, 5(1):149–164.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.
- Yawei Li, Kai Zhang, Jiezhong Cao, Radu Timofte, and Luc Van Gool. 2021. Localvit: Bringing locality to vision transformers.
- Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2022. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*.
- Larry Medsker and Lakhmi C Jain. 1999. *Recurrent neural networks: design and applications*. CRC press.
- Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, pages 1045–1048.
- Ragaa Moustafa, Farida Hesham, Samiha Hussein, Badr Amr, Samira Refaat, Nada Shorim, and Taraggy M Ghanim. 2022. Hieroglyphs language translator using deep learning techniques (scriba). In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 125–132. IEEE.
- Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

- Serge Rosmorduc. 2020. Automated transliteration of late egyptian using neural networks. *Lingua Aegyptia-Journal of Egyptian Language Studies*, 28:233–257.
- Holger Schwenk. 2007. Continuous space language models. In *Computer Speech Language*, volume 21, pages 492–518.
- Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.
- Claude E. Shannon. 1951. Prediction and entropy of printed english. *Bell System Technical Journal*, 30(1):50–64.
- Ran Shao and Xiao-Jun Bi. 2022. Transformers meet small datasets. *IEEE Access*, 10:118454–118464.
- Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In *ICML*, pages 1017–1024.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J.D. Prince, and Yanshuai Cao. 2021. Optimizing deeper transformers on small datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2089–2102, Online. Association for Computational Linguistics.
- Enyu Zhao, Ning Zhou, Chanjuan Liu, Houfu Su, Yang Liu, and Jinmiao Cong. 2024. Time-aware maddpg with lstm for multi-agent obstacle avoidance: a comparative study. *Complex & Intelligent Systems*, pages 1–15.