

Deep Learning Meets Egyptology: a Hieroglyphic Transformer for Translating Ancient Egyptian

Mattia De Cao^{*†}, Nicola De Cao[‡], Angelo Colonna[†], Alessandro Lenci[†]

[†]Università di Pisa, [‡]Google DeepMind

mattia.dc96@gmail.com, ndecao@google.com

angelo.colonna@unipi.it, alessandro.lenci@unipi.it

Abstract

This work explores the potential of Transformer models focusing on the translation of ancient Egyptian hieroglyphs. We present a novel Hieroglyphic Transformer model, built upon the powerful M2M-100 multilingual translation framework and trained on a dataset we customised from the *Thesaurus Linguae Aegyptiae* database. Our experiments demonstrate promising results, with the model achieving significant accuracy in translating hieroglyphic into both German and English. This work holds significant implications for Egyptology, potentially accelerating the translation process and unlocking new research approaches. Source code at <https://github.com/mattia-decao/hiero-transformer>.

1 Introduction

Egyptology, with its rich trove of texts and inscriptions, has recently begun to embrace the potential of computational linguistics. However, a notable scarcity of publications on the topic is evident, with existing efforts primarily focused on optical recognition of hieroglyphs rather than their translation (Sommerschield et al., 2023). Notably, the development of these resources primarily originates from computer science disciplines and highlights the need for deeper integration with Egyptology field.

We bridge this gap by proposing an Egyptology-driven automatic translation approach, merging

^{*}Mattia De Cao made the most significant contributions to this study, including developing the study conception and its elaboration, designing the experiments, mining/analyzing the data and writing the manuscript. Nicola De Cao supported the implementation of computational models, created the paper layout, and revised the manuscript. Angelo Colonna reviewed the human evaluation process and ancient Egyptian background material. Alessandro Lenci contributed to widening the automatic evaluation phase and provided technical review.

Egyptology with Natural Language Processing (NLP) tools. Our Hieroglyphic Transformer translates ancient Egyptian using an adaptation of M2M-100 multilingual model (Fan et al., 2021) to address hieroglyphic writing’s challenges. We construct a meticulously curated dataset derived from the renowned database project *Thesaurus Linguae Aegyptiae* (TLA; Richter et al., 2023)¹ ensuring its compatibility with the model through rigorous data filtering, cleaning and structuring.

Experiments yield promising results, with the Hieroglyphic Transformer achieving reasonable accuracy in translating hieroglyphs into both German and English. Furthermore, we evaluate the model’s performance on texts of varying grammatical complexity and literary styles, highlighting its capacity to handle diverse linguistic structures.

This work holds significant implications for Egyptology. NLP-powered approaches like ours can potentially accelerate and improve translation accuracy and depth. Furthermore, it paves the way for applying Deep Learning models to decipher and translate other ancient languages.

The main contributions of our work can be summarised as follows:

1. presenting a new dataset extracted from the TLA database;
2. adapting a pretrained model to translate Hieroglyphic;
3. showing an automatic and a human evaluation of the model’s performance.

2 Background

2.1 Machine Translation for Ancient Languages

The linguistic diversity of the world encompasses over 7,000 distinct languages. Of these, English, Chinese, Spanish, Japanese, and other Eu-

¹<https://thesaurus-linguae-aegyptiae.de>

ropean languages represent the most extensive corpora (Summer Institute of Linguistics, 2024; UNESCO, 2024), while languages spoken primarily in Asia and Africa often lack comparable data resources (even thousands of times less). These “low-resource” languages attract research from both humanistic and engineering perspectives, with studies offering novel ideas (Aharoni et al., 2019) or exploring understudied niches (Ahia and Ogueji, 2020).

Ancient languages are also part of this wave, but most of their data remains non-machine-readable (i.e., images of objects with text on them or scans of parchment or papyri). Thus most of the recent attention from the machine learning community was directed to Optical Character Recognition (OCR). Major case of these studies include: (i) Kuzushiji, a Japanese cursive script of 8th-18th centuries (Lamb et al., 2020); (ii) Mayan hieroglyphs (Roman-Rangel et al., 2009); (iii) ancient Chinese character manuscripts (Sun et al., 2022); (iv) Sumerian cuneiform (Ahmed H. et al., 2020); and (v) Akkadian cuneiform (Gutherz et al., 2023).

While ancient Egyptian has a decent amount of data available, a substantial portion remains non-machine-readable, primarily in physical books and articles. Even though these sources are accessible online, they necessitate significant digitization efforts for effective utilization in language processing.²

Fortunately, the Egyptian language benefits from the numerous publications digitized and translated into German and English collected in the monumental project *Thesaurus Linguae Aegyptiae* (TLA; Richter et al., 2023) which we use as the source of data in this work.

2.2 Related Work

The majority of recent research in Egyptology using AI focuses primarily on OCR. Examples of such studies include those conducted by Franken and Van Gemert (2013); Hossam et al. (2018); Barucci et al. (2021); Moustafa et al. (2022); Barucci et al. (2023).

Apart from OCR, to the best of our knowledge, only a single publication addresses the task of translation. This work was undertaken by Wiesenbach and Riezler (2019), who sought to address the

²A significant portion of Egyptological articles and books available online have been digitized as images or in a format that hinders machine data extraction. Thus, the first step in making these data usable would be transcribing them into a machine-readable format.

scarcity of resources by incorporating transliteration and POS tags into the training process. This scarcity of publications highlights the need for further research in the application of AI to Egyptology.

2.3 Ancient Egyptian Language

The ancient Egyptian language is a member of the so-called Afro-Asiatic language family and one of the longest continuously attested, having been used from approximately 3200 BCE to 1100 CE (Allen, 2014). Its historical development is usually articulated in six phases: Archaic Egyptian, Old Egyptian, Middle Egyptian, Late Egyptian, Demotic, and Coptic.


Notably, Middle Egyptian (2100-1600 BC) retained its status as a “classical” language for the production of historical and religious texts even after its decline as a spoken language, persisting until the end of ancient Egyptian history. For this reason, we opted for Middle Egyptian as the reference language to train the models in our study (to which we added Old Egyptian as later explained in Section 3.2).

Throughout its existence, ancient Egyptian employed four primary writing systems: hieroglyphic, hieratic, demotic, and coptic. **Hieroglyphic** consists of pictorial signs mostly carved in stone and used in monumental contexts. **Hieratic**, was a simplified and cursive form of hieroglyphic, used for writing on ostraca and papyri. **Demotic**, a late cursive script developed from hieratic, was exclusively employed during the language phase of the same name. **Coptic** writing was derived from the Greek alphabet, with seven additional letters from Demotic to express sounds absent in Greek, and was solely used to write Coptic.

In this work, we used hieroglyphic (or hieratic transcribed to hieroglyphic) because demotic and coptic scripts were used to write language phases other than the ones we chose to employ, i.e., Old and Middle Egyptian. Therefore we will not expand on the other writing systems. For more information about the ancient Egyptian language system, we redirect the reader to Loprieno (1995).

2.4 Hieroglyphs

A hieroglyph can be classified into three distinct categories: *ideogram*, *phonogram*, and *determinative* (Allen, 2014).

Ideograms indicate the word that they depict. In this way, for example, the hieroglyph  repre-








Sign	Gardiner code	Transliteration	Description
	G1	ʃ	Egyptian vulture
	I9	f	Horned viper
	V24	wḏ	Cord wound on stick
	S12	nbw	Bead collar

Table 1: Example of hieroglyphs and their Gardiner code, Transliteration and Description.





senting a mouth writes the word “mouth”, while the hieroglyph  representing a house’s top view is actually the word “house”.

Phonograms represent the phonetic structure (sounds) of the individual word depicted according to the rebus principle. For example the sign  is used to express the phoneme *r*.

Determinatives are used to indicate the semantic sphere of the preceding words, and so these signs are not meant to be pronounced. For example, the hieroglyph , used as determinative, refers to words belonging in the semantic sphere of enclosed spaces and is not read.




2.5 Gardiner Code

The Gardiner code, also called Gardiner’s Sign List, represents the standard system used to identify hieroglyphic signs through alphanumeric codes. It was compiled by the English egyptologist and philologist Alan H. Gardiner as an integral part of his Egyptian grammar (Gardiner, 1957), which remains a standard reference in the Egyptian language study.

The Gardiner code consists of main categories identified by a capital letter of the English alphabet and a descriptive label (e.g., “A. Human beings, male”). Within these sections, each hieroglyph is assigned a progressive number (e.g.,  = A1,  = A2). For subsequent additions of sign variants, later than the original Gardiner’s list itself, a lower-case letter was added after the number (e.g., in the section “N. Sky, earth, water”, we find  = N35,  = N35a).

2.6 Transliteration

In Egyptology, transliteration is the process of converting hieroglyphs into alphabetical symbols to represent the consonants of ancient Egyptian. It is a convention that makes it possible to organize hieroglyphic signs into dictionaries. The transliteration can also be pronounced, but it should always be remembered that only consonants were written

(not vowels), and in many cases, the phonetic value of the signs is unknown. We can only infer the pronunciation based on the Coptic forms as well as on the spelling of Egyptian words in other ancient languages, and vice versa (Allen, 2014). Phonograms and most ideograms can be transliterated into one, two, or three consonants, depending on the number of sounds they represent. For instance, the sign  represents one consonant *m*, the sign  represents *ms*, and the sign  represents *nbw*. See Table 1 for examples of hieroglyphs with their Gardiner code, transliteration, and description.

3 Dataset Construction

This work is based on a snapshot (Richter et al., 2018)³ collected from the database that also feeds the *Thesaurus Linguae Aegyptiae* (TLA; Richter et al., 2023)⁴ and last updated in 2018.

3.1 Thesaurus Linguae Aegyptiae

The TLA project aims “to document and annotate the Ancient Egyptian language through its entire lifespan” (Richter et al., 2023). This objective manifests in two primary digital outcomes: the text corpus (*corpus dataset*) and the lemma list (*vocabulary dataset*).

The *corpus* encompasses a vast collection of hieroglyphic texts, transliterations, and translations. All entries come enriched with metadata such as production dates, script types and connections among data points. Notably, each corpus word is “lemmatized”, i.e. linked to a specific entry in the lemma list. This allows researchers to access broader information spectrum per data point, including part-of-speech (POS) tags, for each element.

While most texts have German translations, some include English or both, promoting cross-language accessibility and the project’s global reach.

³<https://nbn-resolving.org/urn:nbn:de:kobv:b4-opus4-29190> (CC BY-SA 4.0 Int.).

⁴<https://thesaurus-linguae-aegyptiae.de>

3.2 Data Extraction

One of the major contributions of this study consists in the construction of a new dataset from the data collected by the TLA project. We chose Middle Egyptian as the reference language, as explained in section 2.3. However, limited data availability led us to include Old Egyptian (2700-2100 BC) due to its close linguistic relationship with Middle Egyptian, enriching the language representation. Our dataset includes specific elements for each data point. Unfortunately, not all elements were consistently present, preventing a complete construction. In Figure 1 we outline the structure of a data point in our dataset. Taking into account all the diverse elements, these include:

- **Gardiner code:** A unique identifier for each hieroglyph.
- **Transliteration:** The alphabetical representation of hieroglyphs.
- **Translation:** Either German or English.
- **Lemma IDs:** Numerical identifiers for lemmas (basic forms of words).
- **Token inflection codes:** Information about the inflectional forms of the lemmas morphologically marked in the script, such as gender and number of nouns.
- **Datapoint ID:** A unique identifier for the datapoint (each one is a text⁵ containing several sentences).
- **Sentence ID:** A unique identifier for a single sentence in a text.
- **Part-of-speech tags:** Labels used to classify the lexical category of lemmas (e.g., noun, verb, adjective).
- **Metadata:** Unique IDs for data such as language phase, and historical period.

During the mining process, preliminary cleansing was performed to eliminate inconsistencies and irregularities, including: (i) tabs, (ii) carriage returns, (iii) line separators, (iv) excessive white space, and (v) multiple hyphens within hieroglyphs.

The total number of data points extracted was 103,906. We then focused on selecting Old and Middle Egyptian data points delving into language phase metadata. In cases where this information was absent but reliably inferable, we examined his-

⁵From Richter et al. (2023): “A ‘text’ [...] in the TLA is an entity marked as an independent textual unit by clearly marked text delimiters (beginning and end). An individual text may either consist of writing only, or it may be a multimodal composition of writing and illustrations.”

```
{
  "source": <Source as Gardiner code>,
  "transliteration": <Transliteration>,
  "target": <Translation>,
  "lKey": <Lemma IDs>,
  "wordClass": <Part-of-speech tags>,
  "flexCode": <Inflection codes>,
  "metadata": {
    "target_lang": <Target language>,
    "id_datapoint": <Datapoint ID>,
    "id_sentence": <Sentence ID>,
    "language": <Language phase>,
    "date": <Historical period>,
    "script": <Script type>,
    "id_tree": <Assigned ID"
  }
}
```

Figure 1: Structure of a datapoint.

torical metadata to reconstruct it.⁶ The total number of datapoints after filtering was 61,605.

3.3 Data Cleaning

A crucial aspect of our work was the development of comprehensive cleansing operations. Initially, we meticulously hand-cleaned several texts, enabling the identification of recurring patterns and the formulation of generalizable cleansing procedures. This iterative process resulted in the creation of over 280 distinct cleaning operations (e.g., elimination of brackets ‘(, ’) and their contents in German translation, elimination of brackets ‘[’ and ‘]’ while maintaining the content in the transliteration, elimination of ‘!’ from the hieroglyphs). In particular, *lacunae* were treated differently if they were reconstructed or not. If reconstruction was present, we used it; if not, we discarded the datapoint element (e.g. transliteration) as the training process could be altered. Reconstructions were always used.

An example of a datapoint cleansing process is presented in Table 2. A comprehensive compendium of the cleansing operations, including detailed descriptions, treatment methods, and underlying motivations, is provided in the GitHub repository for our project.⁷

3.4 Validation and Test sets

We randomly selected a validation and a test set comprising 100 distinct sources each. Some

⁶In Appendix A we reported datapoint counts relating to both language and historical phases.

⁷<https://github.com/mattia-decao/hiero-transformer>

Gardiner code	Translation	Transliteration
Raw		
Aa1-:D21 M17-S29 [?-*"?"- *I10-*"?"-*?]-:[?-*"?"-*D46- *"?"-*?] N25-:X1-*Z1 V30	and then every foreign land [says]:	ḥr js ?rdd ¹ ? ḥʒs,t nb(.t)
Cleaned		
Aa1 D21 M17 S29 I10 D46 N25 X1 Z1 V30	and then every foreign land says:	ḥr js dd ḥʒs,t nb.t

Table 2: Example of raw and cleaned datapoint (ID tree: aaew_corpus_sawlit_687_107).

sources had multiple translations (i.e., both in English and German) thus we included both versions in the set to (i) increase its size, and (ii) avoid contamination in the training set. Eventually, the validation set had 125 parallel data points, 25 of which possessed English translation, 75 German translation, and 25 containing only transliteration and hieroglyphic. Similarly, the test set had 150 data points, comprising 50 that possessed English translation, 50 German translation, and 50 containing only transliteration and hieroglyphic.

4 Experimental Design

4.1 Data Pairing

Prior to feeding the data into the model, it was essential to organize the data points into source-target pairs. These represent the input-output pairings employed during training (e.g. Hieroglyphs to German). We used two sources as inputs: *egy*, i.e. Gardiner code of ancient Egyptian hieroglyphs; and τ , i.e., transliteration. Both of them were paired with five targets as outputs: (i) *de*, i.e. German; (ii) *en*, i.e. English; (iii) τ ; (iv) *IKey*, i.e. lemma IDs; and (v) *wordClass*, i.e. part-of-speech tags. We reported in Table 3 the list of all different data pairs employed, together with the count of data points in which each pair is present.

4.2 Training

We did not aim to develop novel machine learning techniques or models but rather to harness the capabilities of an existing one and apply it to the Ancient Egyptian language. We then chose to use the finetune M2M-100 model (Fan et al., 2021) for its versatility and effectiveness in multilingual machine translation. M2M-100, originally designed for translating between 100 modern languages, including English and German, was a compelling choice due to its open-source availability and relative novelty. By utilizing this pre-trained model, we

Source	Target	Datapoints
egy	de	16,075
egy	en	2,105
egy	τ	20,155
egy	IKey	21,036
egy	wordClass	20,045
τ	de	45,760
τ	en	2,174
τ	IKey	56,240
τ	wordClass	54,039

Table 3: Data pairs and their distribution among the datapoints.

effectively employed *transfer learning*, a powerful technique that leverages knowledge acquired from a related task to improve performance on a new task. For each experiment, we trained for between 6 and 20 epochs.⁸

We checked validation loss for model selection every 10% per epoch and employed early stopping if no improvement happened for the past 15-20 evaluations.⁹ We used the Adam optimizer (Kingma and Ba, 2015) with batch size 16 and a fixed learning rate 3e-5.

We experimented with different mixtures of source and target (e.g., some included/ excluded the use of transliteration or POS tags). Overall, 11 models were trained,¹⁰ and we reported a selection in Table 4. The comprehensive table of all experi-

⁸Initial experiments used 20 epochs, subsequently reduced due to: (i) no improvements after the third epoch, (ii) increased data pairs significantly extended execution time, and (iii) the 12-hour execution limit of the experimentation platform (Google Colab) rendered maintaining the same epochs impractical.

⁹This value was dynamically adjusted for each experiment due to variations in the amount of data-pairs.

¹⁰Due to cost constraints, we conducted most of our experiments with one NVIDIA T4 Tensor Core (16 GB), and the last model (ALL) that mixes all the data available, with one NVIDIA A100-SXM4 Tensor Core (40 GB). For ALL we increased the batch size to 180.

Source	SacreBLEU					RougeL				
	egy			τ		egy			τ	
Target	de	en	τ	de	en	de	en	τ	de	en
DE (raw)	4.0	-	-	-	-	18.4	-	-	-	-
DE	<u>54.4</u>	-	-	-	-	62.8	-	-	-	-
DE+EN	52.6	28.4	-	-	-	63.1	33.5	-	-	-
DE+EN ^B	61.5	36.4	-	-	-	67.7	38.1	-	-	-
DE+ τ	43.2	-	57.7	<u>54.0</u>	-	55.4	-	78.9	61.8	-
DE+ τ +EN ^B	47.6	20.1	<u>58.4</u>	47.1	<u>30.3</u>	58.8	27.9	<u>80.2</u>	63.1	<u>37.5</u>
ALL	<u>54.4</u>	<u>31.6</u>	59.9	56.2	35.3	<u>64.5</u>	<u>35.5</u>	82.1	<u>62.7</u>	38.1

Table 4: Results of automatic evaluation (SacreBLEU, RougeL). **Bold** results are best and underlined are second best.

ment metrics results can be found in Tables 8 and 9 in Appendix E.

In the training phase, we gave single data (e.g., transliteration or German translation) to the model by assigning them a special language id token (used as prefix in both the source and target text) already present within the model itself. These were *en* for English, *de* for German, *ar* (Arab) for ancient Egyptian, *th* (Thai) for POS tags, *lo* (Lao) for transliteration and *my* (Burmese) for lemma IDs. Except for German, English, and ancient Egyptian¹¹, the codes were arbitrarily selected from Fan et al. (2021) in order to avoid their duplication in the list where data quantities derived from other languages and language groups are presented (Figure 3 of the same article).

Backtranslation Due to the scarcity of data points containing English translations, we employed the M2M-100 model to translate our entire dataset from German to English and incorporated these translations into training.

4.3 Metrics

To assess the performance of the conducted experiments, we employed two automated evaluation metrics: SacreBLEU (Post, 2018) and RougeL (Lin, 2004).

Automatic metrics do not always correlate with human judgment, so we also employed a human evaluation. For that, we applied the model to a series of examples, 16 in total,¹² exhibiting a variety of grammatical constructions (listed in Appendix

¹¹We hypothesized that using Arab for ancient Egyptian could potentially enhance model performance due to its similarities in sentence construction, i.e. verb-subject-object. Further research is required to corroborate this hypothesis.

¹²Of these, 15 were composed of one to three sentences, 1 of eleven.

B), subsequently comparing the model’s output against our own translations or those derived from established publications (Bresciani, 1969; Allen, 2015; Grapow, 1952; Gardiner, 1969; Vogelsang, 1913). During the comparison, we rigorously examined all the distinct data pairs generated by the model, evaluating both the quantity and quality of its correct and erroneous outputs.

5 Results

5.1 Data Cleaning

To assess the effectiveness of our cleaning operations, we conducted and compared two experiments: (i) DE (raw), with raw data; (ii) DE, after the cleaning. Cleaning the data increased the resulting SacreBLEU from 4.0 to 54.4 and RougeL from 18.4 to 62.8. As evident, results have demonstrated that our cleaning procedure significantly improves the model’s training performance.

5.2 Main Results

As evident in Table 4, for translation and transliteration the ALL model (i.e., trained with all data) exhibits the best or second-best performance. This suggests that the model successfully incorporates signals from different forms (e.g., POS tags and transliteration).

Unsurprisingly, mixing back-translation data (DE+EN v.s. DE+EN^B) significantly increases performance in English (SacreBLEU 52 \rightarrow 61 and RougeL 33 \rightarrow 38). However, it surprisingly increases performance in German as well.

Notably, the DE+EN^B model shows the highest accuracy from hieroglyphic to German and English translation. Moreover, both DE+ τ and DE+ τ +EN^B do not perform better than DE and DE+EN^B in German and English. These results suggest that adding

transliteration during training may have some detrimental effects on accuracy. We reported a comprehensive list of results in Tables 8 and 9 in Appendix E.

5.3 N-fold cross validation analysis

We did a 10-fold cross-validation to DE and ALL experiments.¹³ The M2M-100 model was subjected to the same conditions as the previous DE and ALL experiments, allowing for a direct comparison of their performance under different evaluation methods.

The results for the DE experiment exhibited a significant discrepancy, while the performance metrics for ALL were more consistent with the previous findings. This suggests that the validation and test datasets employed previously may have introduced a selection bias, which was mitigated by the larger and more diverse data submitted to training ALL. We reported the full results of n-fold cross validation analysis in Table 10 of Appendix E.

This finding highlights the importance of employing rigorous evaluation strategies to ensure reliable machine learning models, particularly in the context of low-resource languages like ancient Egyptian.

5.4 Human Evaluation

Following the training phase, the model ALL was identified as the most promising candidate due to its superior performance across all data pairs. In this phase, its effectiveness was assessed through a comprehensive trial procedure.

We divided the evaluation process into three distinct steps: (i) Grammatical Complexity, (ii) Literary Passages, and (iii) Stress Test. For every step our evaluation proceeded to analyze all the data pairs (detailed in Section 4.1).

For each Human Evaluation step, the model was submitted to two separate testing waves. In the first wave, the input was presented to the model as Gardiner code, while in the second wave, it was presented as transliteration.

We assessed the sentences based on specific criteria, including: (i) Morphological accuracy; (ii) Grammatical correctness; (iii) Verb-subject agreement in number and gender; (iv) Adequacy of terminology; (v) Semantic coherence.

This two-pronged approach aimed to assess the model’s performance under both input representa-

tions, i.e. hieroglyphic and transliteration. Through this trial procedure, the effectiveness of the ALL model was thoroughly evaluated, demonstrating its potential for a quite accurate and versatile writing of hieroglyphic into transliteration, and both inputs into German, English, Lemma IDs and POS tags. We reported the list of grammatical forms submitted as input in Appendix B.

5.4.1 Grammatical Complexity

We presented exercises of increasing grammatical complexity to the model to assess its ability to handle diverse grammatical structures. All the exercises were extracted from Gardiner’s grammar (Gardiner, 1957). An excerpt is reported in Table 5. The model exhibits no significant difficulties, but rather, it is more sensitive to variations in sentence construction due to low-resource training.

5.4.2 Literary Passages

Passages taken from literary works, encompassing a wide range of grammatical elements and one to three clauses in length, were fed into the model to examine its performance in natural language contexts. The works selected were the “Story of Sinuhe”, the “Tale of the Shipwrecked Sailor”, the “Admonitions of Ipuwer”, and “The Eloquent Peasant”. We observed that the model performs slightly better than the previous phase. Additionally, we noticed higher translation accuracy with transliteration input compared to the Gardiner code.

5.4.3 Stress Test

We submitted a lengthy passage extracted from the “Story of Sinuhe” to thoroughly evaluate the model’s robustness, testing its ability to handle extended and complex linguistic structures. After that, we submitted the same passage divided into single units. Due to the length of the passage, it has been reported in the GitHub repository for our project.¹⁴ We observed that the model fails with lengthy sentences that exceed three clauses but, when provided with a sentence of one or two clauses, it produces quite accurate results.

5.4.4 Human Evaluation Conclusion

The ALL model performed better with short and medium-length input texts comprising one to two sentences. The generated outputs were effective, but there are occasional inconsistencies in completing the fields of transliteration, POS tags and

¹³This technique was not applied to every experiments due to resource limitations.

¹⁴<https://github.com/mattia-decao/hiero-transformer>


Source			
			
D21 Aa1 Y1 V31 G43 A1 V13 G43 D21 Aa1 Y1 V31 G43 A1 D21 N35 V31			
Target	Prediction (from hieroglyphic)	Prediction (from transliteration)	Reference
DE	Ich kenne dich, ich kenne deinen Namen	Ich kenne dich und ich kenne deinen Namen	Ich kenne dich, ich kenne deinen Namen
EN	You know me, I know your name	I know you, I know your name	I know you, I know your name
τ	r.kwj tw r.kwj rn =k	–	rh.kw tw rh.kw rn =k
lkey	95620 44000 174900 95620 44000 94700 10110	95620 174900 95620 94700 10110	95620 174900 95620 94700 10110
pos	verb_2-lit personal_pronoun personal_pronoun verb_2-lit personal_pronoun personal_pronoun substanti	verb_2-lit personal_pronoun verb_2-lit substantive_masc personal_pronoun	verb_2-lit personal_pronoun verb_2-lit substantive_masc personal_pronoun

Table 5: Example of a grammar complexity exercise manually evaluated.

occasionally lemma IDs. For input texts exceeding three sentences, the model struggles to produce exact predictions, particularly in terms of precision and completeness of writing.

Regarding the choice of input, despite transliteration is more accurate than Gardiner code, we recommend comparing both results to obtain a more comprehensive understanding.

We observed great accuracy in generating lemma IDs, indicating that they could be actively used to extract additional information from the TLA database.

Finally, the model exhibits no significant difficulties when submitted to an increasing grammatical complexity. Conversely, it struggles as the input length grows and the rare terms increase.

6 Conclusions

We publicly released our dataset and source code and designed them for easy utilization and assessment. The AI model produces suitable results for research applications and is user-friendly.

This work opens up avenues for future research, including expanding the dataset by incorporating other language phases (Late Egyptian, Demotic and Coptic), integrating additional modern languages, and conducting more comprehensive and diversified experiments.

These efforts could pave the way for enhanced model precision and contribute significantly to the advancement of research in Egyptology and the

application of NLP to the translation and study of ancient languages.

References

- Roe Aharoni, Melvin Johnson, and Orhan Firat. 2019. [Massively multilingual neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.
- Orevaoghene Ahia and Kelechi Ogueji. 2020. Towards supervised and unsupervised neural machine translation baselines for nigerian pidgin. In *Eight International Conference on Learning Representations (ICLR2020) AfricaNLP Workshop*.
- Al-Noori Ahmed H., Talib Moahaimen, and Saeed Mohammed Alhakimi Jamila. 2020. [The classification of ancient sumerian characters using convolutional neural network](#). *Proceedings of the 1st International Conference on Computing and Emerging Sciences (ICCES)*, 1:31–35.
- James P. Allen. 2014. *Middle Egyptian: An Introduction to the Language and Culture of Hieroglyphs*, 3 edition. Cambridge University Press.
- James P. Allen. 2015. *Middle Egyptian Literature, Eight Literary Works of the Middle Kingdom*. Cambridge University Press, Cambridge.
- Andrea Barucci, Michela Amendola, Fabrizio Argenti, Chiara Canfailla, Costanza Cucci, Tommaso Guidi, Lorenzo Python, and Massimiliano Franci. 2023. *Discovering the ancient Egyptian hieroglyphs with*

- Deep Learning*. Consiglio Nazionale delle Ricerche (CNR), Rome, Italy.
- Andrea Barucci, Costanza Cucci, Massimiliano Franci, Marco Loschiavo, and Fabrizio Argenti. 2021. *A deep learning approach to ancient egyptian hieroglyphs classification*. *IEEE Access*, 09:123.438–123.447.
- Edda Bresciani. 1969. *Letteratura e poesia dell'Antico Egitto*. Giulio Einaudi Editore, Turin.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. *Beyond english-centric multilingual machine translation*. *J. Mach. Learn. Res.*, 22:107:1–107:48.
- Morris Franken and Jan Van Gemert. 2013. *Automatic egyptian hieroglyph recognition by retrieving images as texts*. *MM 2013 - Proceedings of the 2013 ACM Multimedia Conference*, pages 765–768.
- Alan H. Gardiner. 1957. *Egyptian Grammar, Being an Introduction to the Study of Hieroglyphs*, third edition. Griffith Institute, Oxford.
- Alan H. Gardiner. 1969. *The Admonitions of an Egyptian Sage, from a Hieratic Papyrus in Leiden (pap. Leiden 344 recto)*. Georg Olms Verlag, Hildesheim.
- Silke Grallert, Tonio Sebastian Richter, and Daniel A. Werning. 2023. TLA Lemma Lists in: Thesaurus Linguae Aegyptiae ed. by Tonio Sebastian Richter & Daniel A. Werning on behalf of the Berlin-Brandenburgische Akademie der Wissenschaften and Hans-Werner Fischer-Elfert & Peter Dils on behalf of the Sächsische Akademie der Wissenschaften zu Leipzig. <https://thesaurus-linguae-aegyptiae.de/info/lemma-lists>.
- Hermann Grapow. 1952. *Der stilistische Bau der Geschichte des Sinuhe*, volume 1 of *Untersuchungen zur ägyptischen Stilistik*. AkademieVerlag, Berlin.
- Gai Gutherz, Shai Gordin, Luis Sáenz, Omer Levy, and Jonathan Berant. 2023. *Translating Akkadian to English with neural machine translation*. *PNAS Nexus*, 2(5):96–105.
- Reham Hossam, Mohammed Abdel-Megeed Mohammed Salem, and Rimón Elias. 2018. *Image based hieroglyphic character recognition*. In *International Conference on Signal Image Technology & Internet-Based Systems (SITIS)*, pages 32–39, Las Palmas de Gran Canaria, Spain.
- Diederik P. Kingma and Jimmy Ba. 2015. *Adam: A method for stochastic optimization*. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Alex Lamb, Tarin Clanuwat, and Asanobu Kitamoto. 2020. *Kuronet: Regularized residual u-nets for end-to-end kuzushiji character recognition*. *SN Computer Science*, 1.
- Chin-Yew Lin. 2004. *ROUGE: A package for automatic evaluation of summaries*. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Antonio Loprieno. 1995. *Ancient Egyptian: A Linguistic Introduction*. Cambridge University Press.
- Ragaa Moustafa, Farida Hesham, Samiha Hussein, Badr Amr, Samira Refaat, Nada Shorim, and Taraggy Ghanim. 2022. *Hieroglyphs language translator using deep learning techniques (scriba)*. In *International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 125–132, Cairo, Egypt.
- Matt Post. 2018. *A call for clarity in reporting BLEU scores*. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Tonio Sebastian Richter, Ingelore Hafemann, Hans-Werner Fischer-Elfert, and Peter Dils. 2018. Database snapshot of project "Strukturen und Transformationen des Wortschatzes der ägyptischen sprache" (excerpt from January 2018). On behalf of Berlin-Brandenburgische Akademie der Wissenschaften and Sächsische Akademie der Wissenschaften zu Leipzig. <https://nbn-resolving.org/urn:nbn:de:kobv:b4-opus4-29190> (CC BY-SA 4.0 Int.).
- Tonio Sebastian Richter, Daniel A. Werning, Hans-Werner Fischer-Elfert, and Peter Dils. 2023. Thesaurus Linguae Aegyptiae v2.0.2.1 On behalf of Berlin-Brandenburgische Akademie der Wissenschaften and Sächsische Akademie der Wissenschaften zu Leipzig. <https://thesaurus-linguae-aegyptiae.de>. Accessed 17-September-2023.
- Edgar Roman-Rangel, Carlos Pallan, Jean-Marc Odobez, and Daniel Gatica-Perez. 2009. *Retrieving ancient maya glyphs with shape context*. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, pages 988 – 995.
- Thea Sommerschild, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androustopoulos, and Nando de Freitas. 2023. *Machine Learning for Ancient Languages: A Survey*. *Computational Linguistics*, 49(3):703–747.
- Summer Institute of Linguistics. 2024. Ethnologue: Languages of the world. <https://www.ethnologue.com/>. Accessed: 2024-01-18.

Jinhu Sun, Peng Li, and Xiaojun Wu. 2022. Handwritten ancient chinese character recognition algorithm based on improved inception-resnet and attention mechanism. In *2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI)*, pages 31–35.

UNESCO. 2024. The world atlas of languages. <https://en.wal.unesco.org/world-atlas-languages>. Accessed: 2024-01-18.

Friedrich Vogelsang. 1913. *Kommentar zu den Klagen des Bauern*, volume 6 of *Untersuchungen zur Geschichte und Altertumskunde Ägyptens*. Leipzig.

Philip Wiesenbach and Stefan Riezler. 2019. Multitask modeling of phonographic languages: Translating middle egyptian hieroglyphs. In *International Workshop on Spoken Language Translation*.

A Taxonomy Analysis of Data Mining

Language	Datapoints
Absent	70,559
Egyptian	28
Middle Egyptian	23,997
Late Egyptian	8,615
Demotic	707

Table 6: Amount of datapoints for each language phase. Counts done on the datapoints mined from TLA (before filtering) and corresponding to 103.906.

Date	Datapoints
Absent	1,165
Old Kingdom	35,849
First Intermediate Period	571
XI Dynasty	466
Middle Kingdom	7,633
Second Intermediate Period	3,634
New Kingdom	38,078
Third Intermediate Period	3,590
Late Period	2,191
600 to 200 BC	2,977
Hellenistic Period	7,133
Roman Period	619

Table 7: Amount of datapoints for each historical period. Counts done on the datapoints mined from TLA (before filtering) and corresponding to 103.906.

B Grammatical Inputs of Human Evaluation

The examples submitted to the model during the human Evaluation comprised various type of sentences. The Grammatical Complexity included: adverbial, nominal (A B), verbal ($s\dot{d}m = f$), negative verbal ($s\dot{d}m = f$), pseudo-verbal and stative. The Literary passage included: verbal ($s\dot{d}m = f$ and $s\dot{d}m.n = f$), verbal negative ($s\dot{d}m.n = f$), adverbial, nominal (A + pw), infinitive, participle, and two longer sentences. The Stress Test included: infinitive, verbal (causative ($s\dot{d}m = f$), stative, subject-stative, adverbial and containing dates or epithets.

C Data Entry Methods

The approach described below ensures that the model receives a clean and standardized representation of hieroglyphic and transliteration, minimizing potential misinterpretations that could arise from

extraneous elements and enhancing its ability to produce accurate translations.

C.1 Hieroglyphic Input

To input hieroglyphs, it is essential to employ Gardiner code. Each hieroglyph must be meticulously cleansed of any brackets, letters, or graphic symbols that extraneously adhere to it, altering its visual representation (it can be checked using Jshesh¹⁵). To divide hieroglyphs, a single space should be inserted between them, while any other extraneous character should be eliminated.

The model has been trained on Ancient and Middle Egyptian hieroglyphs and may encounter challenges with inputs from later linguistic phases and grammatical structures postdating the Second Intermediate Period.

We recommend utilizing signs list of Gardiner's grammar (Gardiner, 1957), or preferably Allen's (Allen, 2014), for a more accurate use of Gardiner code.

C.2 Transliteration Input

For transliteration input, it is necessary to adhere to conventions similar to the one employed by the TLA.

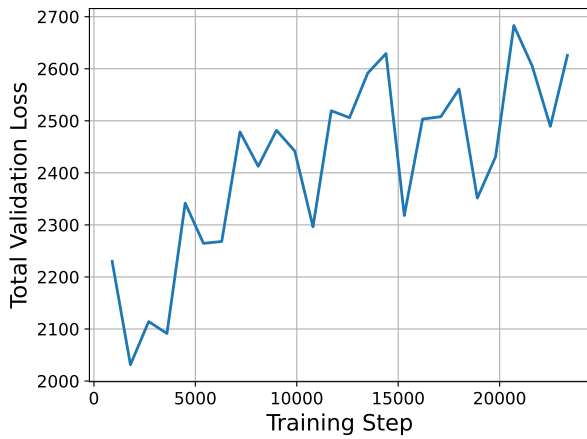
- Proper nouns should have the first letter capitalized.
- It may be beneficial, but not compulsory, to incorporate hyphens between individual lemmas of proper nouns or concepts (e.g., *shṭp-jb-r'* or *w3ḏ-wr*)
- The equal sign (=) to indicate a suffixed pronoun must always be preceded by a space and followed directly by the pronoun, without any additional characters (e.g., *z3 =f m pr*)
- The *j* is utilized for the strong yod while *i* for the weak yod.
- A dot should be employed to distinguish the root of verbs from a suffix other than a pronoun (e.g., *n* in *sḏm.n =f* form) and occasionally for the plural/dual.
- A comma should be employed for the feminine ending and occasionally also for the plural/dual.

¹⁵<https://jshesh.qenherkhopeshef.org>

Transliterated characters can be submitted to the model both as a proper character (e.g., ð) or according to the computer-encoding system (e.g., A for the ð; [Grallert et al., 2023](#)).

To enable the insertion of both upper and lowercase letters, while preserving the computer-encoding, we have implemented a simple mechanism that allows you to capitalize a letter by preceding it with an asterisk. In practice, a straightforward substitution operation has been created in the section where inputs are entered. For instance, since to obtain *d* you must insert *D*, then to get *Ḑ* you have to type **D*; similarly, to attain *D*, you must enter **d*. To input the weak radical *ḑ* simply enter an *i*.

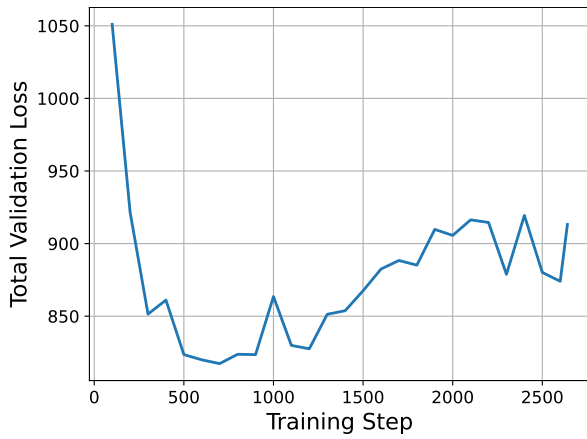
D Experiments Graphs



(a) Model DE (raw). Best loss: step 1,800.



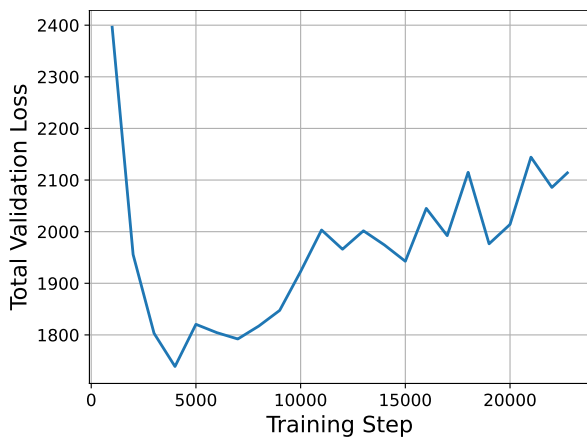
(b) Model DE. Best loss: step 4,500.



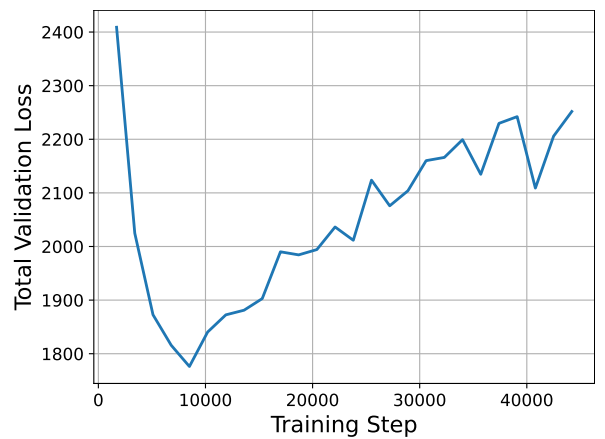
(c) Model EN. Best loss: step 700.



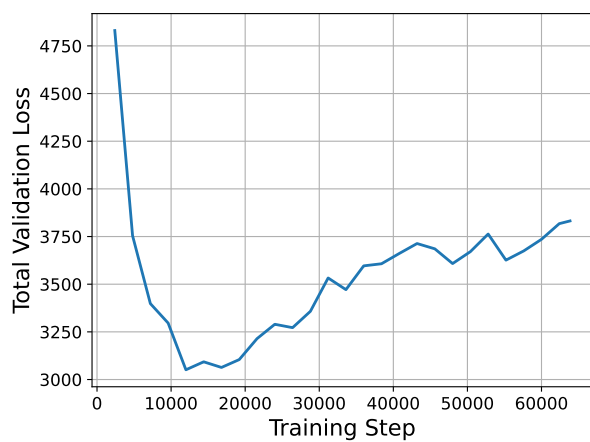
(d) Model DE (lem). Best loss: step 3,600.



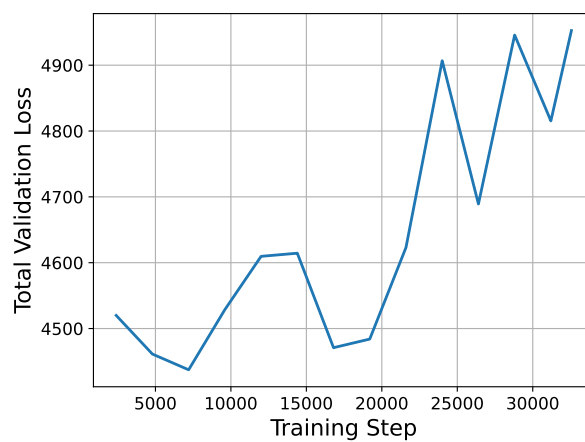
(e) Model DE+EN. Best loss: step 4,000.



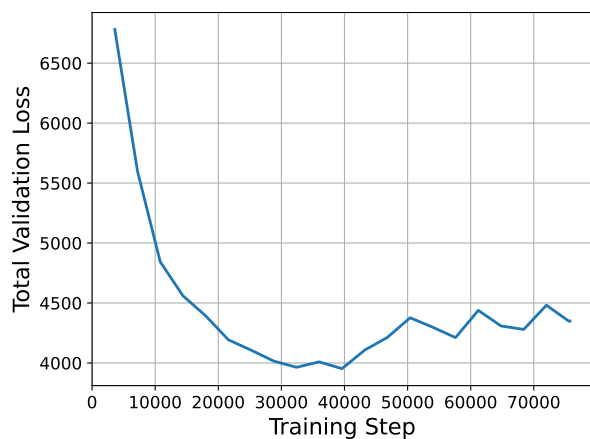
(f) Model DE+EN^B. Best loss: step 8,500.



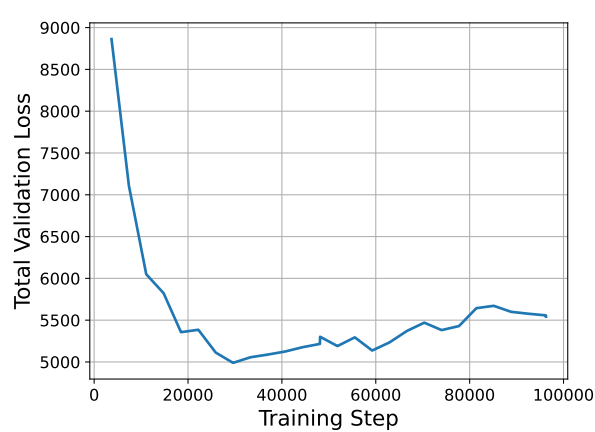
(g) Model $DE+\tau$. Best loss: step 12,000.



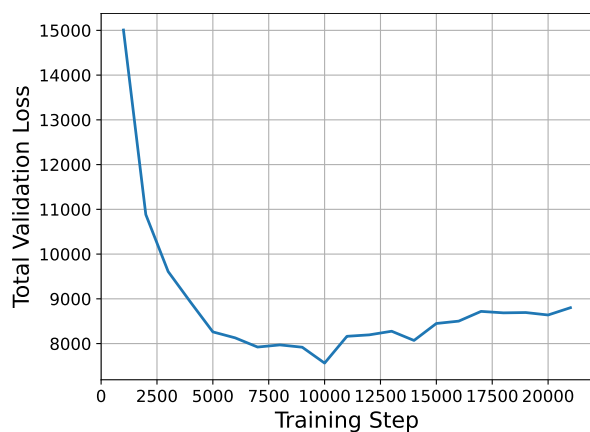
(h) Model $DE+\tau+EN^B$. Best loss: step 7,200.



(i) Model $DE+\tau+POS$. Best loss: step 39,600.



(j) Model $DE+\tau+LKEY$. Best loss: step 29,600.



(k) Model ALL. Best loss: step 10,000.

Figure 2: Validation losses of different models and at which step the loss is at its minimum.

E Taxonomy Analysis of Generated Models: SacreBLEU, RougeL and 10-fold Cross Validation

SacreBLEU									
Source	egy					τ			
Target	de	en	τ	lkey	POS	de	en	lkey	POS
DE (raw)	4.0	-	-	-	-	-	-	-	-
DE	54.4	-	-	-	-	-	-	-	-
EN	-	22.6	-	-	-	-	-	-	-
DE (lem)	25.9	-	-	-	-	-	-	-	-
DE+EN	52.6	28.4	-	-	-	-	-	-	-
DE+EN ^B	61.5	36.4	-	-	-	-	-	-	-
DE+ τ	43.2	-	57.7	-	-	54.0	-	-	-
DE+ τ +EN ^B	47.6	20.1	58.4	-	-	47.1	<u>30.3</u>	-	-
DE+ τ +POS	53.2	-	60.0	-	82.1	49.6	-	-	87.1
DE+ τ +LKEY	<u>55.1</u>	-	59.4	64.4	-	58.9	-	<u>70.9</u>	-
ALL	54.4	<u>31.6</u>	<u>59.9</u>	<u>63.9</u>	<u>79.0</u>	<u>56.2</u>	35.3	74.0	86.4

Table 8: Results of automatic evaluation, in particular SacreBLEU, of all models along with POS tags and lKey. **Bold** results are best and underlined are second best.

RougeL									
Source	egy					τ			
Target	de	en	τ	lkey	POS	de	en	lkey	POS
DE (raw)	18.4	-	-	-	-	-	-	-	-
DE	62.8	-	-	-	-	-	-	-	-
EN	-	25.1	-	-	-	-	-	-	-
DE (lem)	42.0	-	-	-	-	-	-	-	-
DE+EN	63.1	33.5	-	-	-	-	-	-	-
DE+EN ^B	67.7	38.1	-	-	-	-	-	-	-
DE+ τ	55.4	-	78.9	-	-	61.8	-	-	-
DE+ τ +EN ^B	58.8	27.9	80.2	-	-	63.1	<u>37.5</u>	-	-
DE+ τ +POS	62.9	-	83.1	-	83.8	67.3	-	-	<u>87.6</u>
DE+ τ +LKEY	59.6	-	<u>82.6</u>	<u>71.5</u>	-	<u>63.8</u>	-	<u>75.4</u>	-
ALL	<u>64.5</u>	<u>35.5</u>	82.1	71.7	<u>82.6</u>	62.7	38.1	77.7	88.4

Table 9: Results of automatic evaluation, in particular RougeL, of all models along with POS tags and lKey. **Bold** results are best and underlined are second best.

SacreBLEU									
Source	egy					τ			
Target	de	en	τ	lkey	POS	de	en	lkey	POS
DE	32.0 \pm 2.0	10.2 \pm 1.2	-	-	-	5.4 \pm 2.0	0.2 \pm 0.3	-	-
ALL	45.5 \pm 1.4	35.9 \pm 3.7	52.7 \pm 1.3	57.9 \pm 5.1	71.9 \pm 1.3	59.6 \pm 1.4	42.6 \pm 2.9	74.3 \pm 2.4	79.2 \pm 0.7

RougeL									
Source	egy					τ			
Target	de	en	τ	lkey	POS	de	en	lkey	POS
DE	41.1 \pm 0.9	14.7 \pm 1.1	-	-	-	3.9 \pm 1.3	0.3 \pm 0.4	-	-
ALL	53.4 \pm 1.1	40.9 \pm 2.4	78.9 \pm 0.6	65.2 \pm 4.3	81.6 \pm 0.6	68.0 \pm 1.0	47.9 \pm 1.5	79.1 \pm 2.1	88.0 \pm 0.8

Table 10: Results of 10-fold cross validation.