Syntax as a Rosetta Stone: Universal Dependencies for In-Context Coptic Translation

Abhishek Purushothama* **Emma Thronson*** **Alexia Guo** **Amir Zeldes**
Corpling Lab
Georgetown University
{ap2089, et726, qg65, amir.zeldes}@georgetown.edu

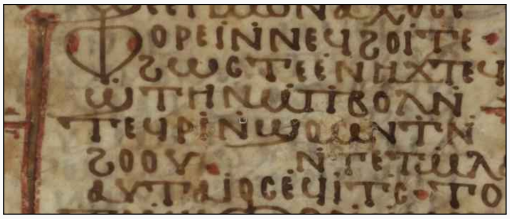
Abstract

Low-resource machine translation requires methods that differ from those used for high-resource languages. This paper proposes a novel in-context learning approach to support low-resource machine translation of the Coptic language to English, with syntactic augmentation from Universal Dependencies parses of input sentences. Building on existing work using bilingual dictionaries to support inference for vocabulary items, we add several representations of syntactic analyses to our inputs, specifically exploring the inclusion of raw parser outputs, verbalizations of parses in plain English, and targeted instructions of difficult constructions identified in subtrees and how they can be translated. Our results show that while syntactic information alone is not as useful as dictionary-based glosses, combining retrieved dictionary items with syntactic information achieves significant gains across model sizes, achieving new state-of-the-art translation results for Coptic.

1 Introduction

Recent advances in LLMs have raised the quality of baseline results of machine translation (MT) in high-resource languages (HRLs) to the point where they can be used in user facing and downstream application contexts (Zhu et al., 2024). At the same time, low-resource languages (LRLs) have seen limited benefits from baseline prompting approaches (Frontull and Ströhle, 2025; Pava et al., 2025), since models have little or no language modeling capabilities for low-resource languages.

However, prompt augmentation using in-context learning (ICL), by integrating bilingual glosses for vocabulary items (Ghazvininejad et al., 2023), has offered a promising direction. This is notably useful to overcome the limitations in languages where we have no hope of obtaining sufficient amounts of raw text data. We can leverage LLMs' fluency, especially when there is a dictionary or glossary for the source language, and the target language is high-resource, i.e. for translation from the LRL to the HRL. At the same time, simple glosses are inherently limited to single, or in some cases few-word listed expressions, and cannot inform models about the grammar and specific constructions of the LRL, leaving models to generate target sentences based on plausible configurations of target lexical item senses.



An excerpt from Apophthegmata Patrum 25, MS MONB.EG 67 (K 0321), (courtesy Österreichische Nationalbibliothek). The corresponding text is featured below.

Coptic: ϩⲱⲥⲧⲉ ⲉⲛⲉⲭⲧⲉϥⲱⲧϩⲙ ⲙⲙⲡⲃⲟⲗ ⲛⲧⲉϥⲣⲓ ⲛⲱⲟⲙⲙⲛⲧ ⲛϩⲟⲟⲩ ⲛⲧⲉⲧⲙⲙ-ⲗⲁⲁⲩ ⲧⲁⲓⲟⲥ ⲉϥⲓⲧⲥ

Reference: *such that if he throws his tunic out of his cell for three days no one will pick it up to wear it*

GPT-4.1 Translation: *So that his light would not shine before people. You should not do this at all.*

Figure 1: Reference translation and the baseline translation for Coptic text, (corresponds to the excerpt at the top). Even large models such as GPT-4.1 provide fluent yet fundamentally incorrect translation without augmentation.

*Equal contribution

Findings of the Association for Computational Linguistics: ACL 2026, pages 36176–36195
July 2-7, 2026

In this paper, we target **Coptic** (specifically the Sahidic dialect),[1] a low-resource language[2] for which LLMs have little or no coverage out of the box, and in which grammatical constructions often convey differences in meaning that are not obvious from content words alone.

Coptic, illustrated in Figure 1, is the last phase of the indigenous language of Egypt, spoken and written primarily in the first millennium CE. It forms part of the Egyptian branch of the Afro-Asiatic language family. It is an agglutinative, head initial language with a complex system of auxiliaries, grammatical gender and number, aspect and tense distinctions. As the language of Christian Egypt in the Hellenistic period, it is crucial for our understanding of the history of religion and of the Mediterranean in Late Antiquity, and it remains in use today as the heritage language of Egyptian Christian Copts in Egypt and the diaspora.

However, data for Coptic is scarce, as are experts in the language, leading to many cataloged manuscripts in museum libraries remaining undigitized, or digitized but untranslated. This motivates our work to strive for high quality machine translation outputs, which, even if not sufficiently accurate to be used as-is, could reduce the effort for experts to correct.

Testing current LLMs on out-of-the-box translation quality for Coptic quickly reveals their inadequacy. For example, the sentence taken from the manuscript in Figure 1 refers to an ascetic monk's rags being so tattered that, if left outside unattended for three days, no one would bother to steal them. Even very large models, such as GPT-4.1, produce a fundamentally incorrect translation, indicating the model's inability to process the language as-is.

For languages such as Coptic, exploring translation based on the methodologies and resources that are specifically available is important (Bapna et al., 2022). Although dictionary-based prompt augmentation already improves machine translation for LRLs, and has even recently been applied to Coptic (Miyagawa, 2025), the inability to encode grammatical relations by simply listing lexical items creates a ceiling on translation quality, which this work aims to address.

In particular, this paper is the first to explore whether the addition of syntactic information, such as Universal Dependencies or UD (de Marneffe et al., 2021), can augment in-context MT for LRLs such as Coptic when provided with lexical information. We compare various strategies: baseline translation with no augmentation, with dictionary augmentation, with syntactic augmentation, and with both dictionary and syntactic information. We test both open-weight and closed-source models (§3).

We find that while lexical information remains more important than syntactic information, adding syntactic information based on UD parses significantly improves performance over lexicon integration alone across all model sizes (§4). We additionally provide qualitative and quantitative analyses examining the effects of different operationalizations, the impact of automatic parses (compared to gold parses), and the differences between biblical and non-biblical samples (§5). All code is made available on the GitHub repository for this paper.[3]

2 Background

LLMs have become the standard for MT in many settings (Kocmi et al., 2025b). However, methods and models vary based on the languages, resources, and domains. This paper specifically focuses on using LLMs in an ICL setting, utilizing lexicon and syntactic analysis. Supervised training methods have shown success when parallel training data can be collected or built, but this does not extend to LRLs such as Coptic. In these scenarios, both expert and non-parallel data have been used for augmentation to support low-resource MT.

Lexicon Bilingual lexicons are a common resource used for low-resource MT, including in LLM-based approaches, especially for translating from a LRL to HRL. Ghazvininejad et al. (2023) showed how simple textual demonstrations can be added to prompts to improve performance. Lu et al. (2024) showed how a chain of bilingual lexicons can be used to provide a bridge between source and target languages. A range of parameters can influence ICL formulations, especially for LRLs (Court and Elsner, 2024), partly due to the context-use challenges of LLMs (Liu et al., 2024).

Grammar ICL MT provides a unique opportunity to provide information about the grammar of a

[1] ISO 639-2 code: 'cop', Glottolog: https://glottolog.org/resource/languoid/id/copt1239. Sahidic: https://glottolog.org/resource/languoid/id/sahi1241

[2] It is considered an endangered language with no L1 speakers (Eberhard et al., 2026).

[3] The code is available at https://github.com/gucorpling/in-context-coptic-translation

language to be used in-context to help with translation. This has been studied in various settings, such as retrieval from a grammar with extraction of excerpts (Tanzer et al., 2024) or constructions of an expert set of grammar rules (Zhang et al., 2025). Pei et al. (2025) investigated ICL MT for the LRL of Manchu (mnc), using dictionary entries (with morphological analysis), parallel examples, and extracted grammar excerpts. They found grammar excerpts to be less useful in the presence of the other two resources. Our paper utilizes dictionary entries and morphological analysis (although not as high-level components). It additionally focuses on using grammatical information derived from the input, rather than relying on the explicit grammar of language itself as in Pei et al. (2025).

Grammatical information Unlike grammar books and excerpts, grammatical information is relatively less explored as a resource for augmentation for ICL. Such information can be broad, from morphological analysis (Elsner and Needle, 2023; Pei et al., 2025) to annotated linguistic data such as UD. Additional challenges are the variations in the frameworks and the nature of the data (such as genre) annotated. The heart of our work explores this scenario, specifically with syntactic information, using Universal Dependencies.

Universal Dependencies treebanks for MT UD treebanks provide annotations for over 150 languages using a consistent grammatical framework. While UD treebanks are used predominantly for the study of language and the improvement of monolingual and multilingual parsing, they have also been used in conjunction with other tasks, including MT. For example, Nagy et al. (2023) used UD trees as a substrate to create additional parallel data for MT. To our knowledge, our paper is the first to utilize UD as a source of grammatical information for MT.

Coptic MT Work targeting Coptic in particular has only recently emerged, with papers using English (Wannaz and Miyagawa, 2024; Saeed et al., 2024; Miyagawa, 2025), French (Chaoui and Khoury, 2025) and Arabic (Saeed et al., 2024) as HRL targets. The source data in all of these papers is in Sahidic Coptic, the classical dialect of the language which we also target, except for Saeed et al. (2024), who use the later attested Bohairic dialect of Coptic. Although these studies have shown that fine-tuning models can somewhat improve results (Chaoui and Khoury, 2025), translation outputs are still far from ready for public use, and therefore do not yet satisfy our aspirations to make more Coptic texts available to the public via translation.

Model (Size)	Technical Report
Gemma3(-it) (12B)	Team et al. (2025)
Gemma3(-it) (27B)	Team et al. (2025)
GPT4.1 (NA)	OpenAI et al. (2024)
GPT4.1 Mini (NA)	OpenAI et al. (2024)
Llama3.1(-Inst) (8B)	Grattafiori et al. (2024)
Aya-Expanse (8B)	Üstün et al. (2024)
Aya-Expanse (32B)	Üstün et al. (2024)

Table 1: Models used for our experiments with dev data. We report results for test and ostraca from Wannaz and Miyagawa (2024) only for the first three models. We do not use the base variant of models, so will not explicitly refer to those with the (-it) suffix for the rest of the paper.

3 In-Context Coptic Translation

In-context translation approaches for LRLs, such as Coptic, augment input with examples and other forms of language or linguistic data, rather than relying solely on supervised training data. These methods focus on selecting and presenting auxiliary information, such as lexical or syntactic details, that can guide and improve translation behavior.

Within this framework, translation prompts are explicitly constructed to incorporate relevant resource-derived information alongside the source sentence. This design allows linguistic knowledge to be selectively included and evaluated, providing a flexible means of improving translation quality in low-resource settings. For our translation task, Coptic is the source language, English is the target language, and it is a segment-level translation. This is mainly a function of the resources (discussed further in §3.2), but it also allows us to focus on both designing expert based methods (CON in §3.3), and providing more detailed analyses (§5).

We experiment with several open models, but we will focus on Gemma models for our analysis, which showed the best performance during development. We also use GPT-4.1 as the closed reference models (see Table 1 for the full list).

3.1 Data

The largest publicly available dataset of Coptic sentences and translations comes from Coptic Scriptorium (Schroeder and Zeldes, 2016). This includes 2.3M tokens of Coptic, of which ~1.43M are in the Sahidic dialect, though translations exist for only 1.21M tokens of this data, including most books of

the Old and New Testament. This is also the dataset used for translation by Chaoui and Khoury (2025), who selected four books of the Bible (1 Corinthians, Mark, Galatians, and Hebrews) as their test set. Since we already have translations of the Bible and suspect that LLMs can more easily produce translations of Bible verses provided they can identify the target verse (for example, through the occurrence of proper names), in this paper we additionally analyze MT results between Biblical and non-Biblical texts.

UD Treebank: Translation dataset and parses Since our method is the first to use UD parses for MT, we require data for which gold standard syntax trees exist, which allows us to assess the impact of cascading parser errors by comparing predicted and gold parse inputs. We therefore focus on the manually treebanked subset of Coptic Scriptorium available in the Sahidic UD Coptic treebank (Zeldes and Abrams, 2018). This contains over ~60K tokens (2,387 sentences) with translations and covers a range of genres including Bible translations, indigenous Coptic hagiography, sermons, and documentary materials. We use the UD treebank's standard splits of `dev` (380 sentences, of which 182 are from the Bible) and `test` (405 sentences) as our core data.

Ostraca: Out-of-domain translation dataset We also report results on out-of domain data from Wannaz and Miyagawa (2024), which contains 4 ostraca (21 sentences), with previous MT results.[4]

3.2 Resources

Broadly, we utilize two sets of information: a fixed dictionary and a sample-specific syntactic analysis NLP pipeline.

Dictionary The Coptic Dictionary (hereafter the dictionary)[5] from Feder et al. (2018) and updated with the lemma list from Burns et al. (2020) acts as the as our sole dictionary resource. The dictionary was constructed by integrating multiple bilingual lexicons, providing extensive and elaborate information for lexical items. The dictionary covers over 10K entries, but multiple entries can exist for a single surface form.

Syntactic analysis To add grammatical information to our input sample, we utilize CopticNLP (Zeldes and Schroeder, 2016), which performs automatic segmentation of agglutinative Coptic word forms into tokens and produces full UD parses. These parses include both Google Universal POS tags (Petrov et al., 2012, upos), language-specific tags (Zeldes and Schroeder, 2015, Coptic xpos tags), morphological features, and language-of-origin for each word (for example, identifying Greek loan words). We use this information in multiple ways below, including raw parser output, templated verbalization into English (e.g. 'The subject of the verb X is the noun Y'), and verbalized translation instructions for special constructions (see below).

3.3 Components

To make the best use of these resources in the ICL setting, we design specific 'components' that handle processing and representing information from the resources. We define four components: `LEX`, `DEP`, `CON`, and `CoNLLU` . Each draws on the resources and provides complementary information about the source sentence.

Lexicon `LEX` Our `LEX` component is designed to map the structured information from the dictionary to the input context. We use syntactic analysis for the input sentence to inform this. We use POS and morphological information (lemma and segmentation) to search the lexicon for dialect specific translations. To control prompt length and relevance, we further filter retrieved entries, retaining the most relevant entry- and sense-level hierarchical information for inclusion in the instruction context. Further details about the `LEX` component are provided in appendix B.2.

Syntax - Dependency `DEP` Our `DEP` component verbalizes the syntactic structure from the parses for inclusion into the instruction. For each input sentence, we extract the head–dependent relations and verbalize them as short, plain English statements (e.g. 'ⲛ is the case marking of ϣⲟϣⲟⲣ'). This representation provides explicit syntactic relations between tokens, which are not necessarily inferrable from surface word order alone. Multiple parameters control the granularity and content of the syntactic information added to the instruction, such as the selected UD label set, selected parts of speech (POS), and disambiguation for repeated tokens. In all configurations, the dependency descriptions are rendered in plain English and the

[4] We do not compare results with Saeed et al. (2024) since their data targets Bohairic (`https://glottolog.org/resource/languoid/id/boha1242`), a substantially different dialect.

[5] `https://coptic-dictionary.org/`

dependency section as a whole can be added to the instruction similarly to the lexicon component. Further details about the DEP component are provided in appendix B.3.

Syntax - Construction CON CON is the most bespoke component in the paper. We perform a manual analysis of model errors on the development set,[6] and in conjunction with the syntactic analysis, identify specific grammatical constructions for which the pilot model demonstrates error.[7] In particular, we identify 26 tree configurations and formalize their characteristic dependency subgraphs using DepEdit (Peng and Zeldes, 2018), a Python templating library for UD trees. These constructions range from simple and general to highly specific.

For example, imperatives are trivially recognizable using UD morphological features. Like English, Coptic verbs with no subject have an imperative meaning, but merely glossing words for translation can easily miss the need for an imperative translation, which a targeted instruction can clarify. On the complex end, one construction involving postponed subjects (roughly, examples like 'he went to the desert, that is, the monk') triggers an explanation of how Coptic uses this configuration.

This module also uses a dedicated subroutine to transliterate any words tagged as proper nouns (PROPN) into Latin characters, creating a sub-instruction which states that the word is a name, accompanied by the transliteration. For more detailed examples see appendix B.4.

Syntax - CoNLLU Since we use UD and Coptic NLP for syntactic analysis, a low effort option is to effectively dump the raw output (CoNLL-U format) in the instruction. Given the prevalence of the CoNLL formats and similar structured formats on the internet, it is necessary to explore this low-effort representation operationalization.[8]

3.4 Instruction Design

Our instruction (or prompt) consists of a base instruction with additional information derived from our components, and closed out with consistency cues (see Figure 2). We adapt the base instruction from Kocmi et al. (2025a). For dictionary-based information, we adapt the instructional framework from Pei et al. (2025). We add information from each of the components for the setting (§3.6) with a small textual header indicating the section. We additionally added some consistency cues similar to Pei et al. (2025) to help improve model responsivity.

> You are a professional Coptic-to-English translator tasked with providing translations suitable for use in United States (en_US). ⋯ Please translate the following Coptic text into English (en_US): ⋯
> (From LEX) For the translation task, you are given dictionary entries for Coptic. ⋯
> (From CoNLLU) The raw conllu data for the sentence is in the CONLL-U format:
> 1 ⲁⲛⲟⲕ ADP PREP _ 3 case _ _⋯
> (From DEP) The dependency information for the sentence is: ⲡⲣⲉⲥⲃⲩⲧⲉⲣⲟⲥ is the root. ⋯
> (From CON) The information about specific constructions ⋯ The dislocated element ⲡⲁⲓ is a repeated reference to the pronoun dependent of the predicate ϫⲱ. Using all the information provided above, now please translate the sentence into English(en_US). Remember your source sentence is: {*source*} .The English translation is:

Figure 2: A condensed example of how the different information is added to the instruction. Information added from each component is based on the experimental setting (§3.6). LEX+SYN would include information from all components. More details of different parts are provided in appendix B.5.

3.5 Metrics

MT has a wide variety of metrics targeting different aspects of evaluation (Lavie et al., 2025). We use BERTScore (Zhang et al., 2020), specifically the Avg. F1 of BERTScore, as our primary metric for our development work and analyses in this paper. It was chosen based on the previously seen correlations with human evaluation, even in comparison to LLM-as-judge settings for MT (Lavie et al., 2025).

We additionally report BLEU (Papineni et al., 2002) for comparison to Wannaz and Miyagawa (2024) on the ostraca. For more effective comparison with their reported results, we further include METEOR (Banerjee and Lavie, 2005) for this dataset.

Additional discussion and details about the metrics for completeness and reproducibility are provided in appendix B.1

[6] One of the authors has training in Coptic.

[7] We used results from GPT4.1 mini for this analysis.

[8] An example of CoNLL-U is given in the appendix Figure 4.

3.6 Settings

We evaluate a range of prompting configurations that incrementally augment the baseline translation setting with additional components. Our baseline setting consists of a simple translation instruction without any auxiliary information. We then run experiments that add individual linguistic components on their own, as well as combinations of multiple components.

We consider settings with a single component LEX, CoNLLU, CON, or DEP. We additionally examine two combination settings, DEP+CON and LEX+SYN (which is effectively LEX+CoNLLU+CON+DEP). The DEP+CON setting groups the designed syntactic components but not the raw parse component CoNLLU to isolate the effect of this distinction.

LEX and DEP grid search Both LEX and DEP have parameters that determine the kind and amount of information incorporated into the instruction context. We perform a targeted grid search to determine our best lexicon and dependency parameter values using our pilot model (GPT-4.1 mini) and a diagnostic subset of dev data consisting of 20 samples drawn from an initial baseline run. We used 10 sentences with the highest translation quality scores (the 10 easiest) and the 10 with the lowest scores (the 10 hardest). We then shortlisted four parameter configurations and choose the final configuration based on the dev split using BERTScore F1 (see §3.5) as the primary criterion. Further information on this is in appendix B.

With gold parses To analyze the use of automatic parses compared to gold-standard annotations, we additionally conduct experiments only on the dev split using gold-standard UD parses in place of automatically generated parses. We conduct and report this experiment only for the Gemma models.

4 Results

Our strategy of adding syntactic information leads to performance improvements in both open and closed models, and across test (shown in Table 2)[9] and ostraca (shown in Table 11).

Across the three models, adding the LEX component yields statistically significant improvements over the baseline (for all of dev, test, and ostraca) as expected. Likewise, adding SYN also results in significant gains in BERTScore (see appendix D for significance test details). This indicates that syntactic information provides complementary benefits (see Table 2) beyond what we see from just lexicon. These trends are consistent across both Gemma models and GPT-4.1.

The highest augmentation setting LEX+SYN shows the best performance across all our models. This suggests that combining both lexical and syntactic information provides cumulative improvements to the model. We disambiguate the contribution of each syntactic component when added on top of LEX in the next section (§5).

The same trends are seen in dev (see Table 2) and across other models. However, this was not consistent for all metrics.

For the ostraca, our best open-models results are comparable to the closed models reported by Wannaz and Miyagawa (2024), and GPT-4.1 versions perform better than any reported system on METEOR but not BLEU (see Table 3). The LEX+SYN setting, which was shown to be the most useful in dev and test, also has the best performance on this set for both open models and closed models.[10]

5 Analysis and Findings

To better understand the impact of different components, and nature of the data, we perform additional analyses on dev.

Error analysis Table 4 provides some examples for qualitative error analysis, which illustrate how models leverage LEX and SYN. In the baseline setting, open models can generate totally unrelated translations, since they have almost no support for Coptic. Lexicon augmentation induces relevant vocabulary in outputs yielding lexically accurate outputs even in the 12b model, and near-perfect lexical choices in translations from the 27b model (e.g., 'Give to the poor' and 'wishes to kill'). Adding syntax helps both in connecting the subject and verb for 'He gave to the poor' and in complex nesting for 'he said that the abbot Pambo said', a relatively unlikely nested reported speech, which the lexicon-only augmentation fails to capture.

[9] BLEU is also reported for consistency with a relaxed setting on test. It is unstable with the test data due to variation in LLM translations and drastic LLM errors, making it unreliable (more details in appendix B.1). For dev and ostraca it is reported with default signature.

[10] Both papers used the default settings from sacrebleu python package for BLEU.

	Gemma-12b		Gemma-27b		GPT-4.1	
Setting	**BLEU** (Δ)	**BertScore** (Δ)	**BLEU** (Δ)	**BertScore** (Δ)	**BLEU** (Δ)	**BertScore** (Δ)
			test split			
Baseline	60.65 (0.00)	0.8363 (0.0000)	19.18 (0.00)	0.8385 (0.0000)	13.56 (0.00)	0.9012 (0.0000)
LEX	36.84 (-23.81)	0.8551 (0.0187)	9.41 (-9.77)	0.8565 (0.0181)	13.56 (0.00)	0.9152 (0.0140)
CoNLLU	13.56 (-47.09)	0.8489 (0.0126)	13.56 (-5.62)	0.8547 (0.0162)	13.56 (0.00)	0.9056 (0.0044)
CON	5.50 (-55.15)	0.8511 (0.0148)	19.18 (0.00)	0.8518 (0.0133)	13.56 (0.00)	0.8998 (-0.0014)
DEP	60.65 (0.00)	0.8420 (0.0057)	13.56 (-5.62)	0.8417 (0.0033)	13.56 (0.00)	0.9014 (0.0002)
DEP+CON	5.50 (-55.15)	0.8502 (0.0139)	13.56 (-5.62)	0.8530 (0.0146)	13.56 (0.00)	0.9030 (0.0018)
LEX+SYN	13.56 (-47.09)	0.8707 (0.0344)	9.41 (-9.77)	0.8746 (0.0361)	13.56 (0.00)	0.9195 (0.0183)
			dev split			
Baseline	16.45 (0.00)	0.8342 (0.0000)	10.30 (0.00)	0.8375 (0.0000)	14.58 (0.00)	0.9015 (0.0000)
LEX	5.46 (−10.99)	0.8593 (0.0250)	14.46 (4.16)	0.8611 (0.0235)	10.72 (−3.86)	0.9139 (0.0124)
CoNLLU	27.49 (11.04)	0.8453 (0.0111)	14.86 (4.56)	0.8522 (0.0147)	28.21 (13.62)	0.9048 (0.0033)
CON	20.26 (3.81)	0.8499 (0.0157)	15.43 (5.14)	0.8509 (0.0133)	12.77 (−1.81)	0.9000 (−0.0016)
DEP	28.66 (12.21)	0.8405 (0.0063)	15.28 (4.98)	0.8414 (0.0039)	16.30 (1.72)	0.9020 (0.0004)
DEP+CON	21.65 (5.20)	0.8490 (0.0147)	10.38 (0.09)	0.8509 (0.0134)	12.79 (−1.79)	0.9018 (0.0003)
LEX+SYN	7.71 (−8.74)	0.8722 (0.0380)	18.84 (8.54)	0.8736 (0.0360)	12.69 (−1.89)	0.9169 (0.0154)

Table 2: Results for the `test` split and `dev` split Gemma and GPT-4.1 model across different settings (differences in performance are significant for $p < 0.0001$.). Syntactic information provides complementary benefits beyond what we see from just lexicon. The `LEX+SYN` setting (bolded) combining `LEX`, `CoNLLU`, and `DEP+CON` performs the best for all models. The `LEX` setting, known to be effective, is underlined for ease of comparison. BLEU is reported for completeness (see §3.5).

Model	**BLEU**	**BertScore**	**MET.**
From this paper			
Gemma			
12b LEX+SYN	7.82	0.8850	0.32
27b LEX+SYN	16.19	0.8781	0.34
GPT-4.1			
LEX+SYN	17.99	0.9046	0.53
DEP	18.15	0.8859	0.38
From Wannaz and Miyagawa (2024)			
Claude Opus	20.02	-	0.46
Claude Haiku	11.52	-	0.35
CopticTrans	8.43	-	0.30

Table 3: The best Gemma and GPT-4.1 settings for ostraca (§3.1). The BLEU and METEOR scores for Claude Opus and Haiku (Anthropic, 2020), and CopticTranslator are from Wannaz and Miyagawa (2024). Our open models are comparable, and GPT-4.1 exceeds previous model on METEOR but not BLEU. Full results are in the appendix (Table 11).

Effectiveness of transliteration in `CON` for nouns tagged `PROPN` is also apparent, leading to correct renditions of 'Herodias' and 'Pambo'. GPT-4.1 follows a similar pattern: baseline translations seem almost random, and `LEX` augmentation supplies some relevant words; however `SYN` augmentation allows for recognition of relations such as relative clauses ('mindless ones who are...') and disambiguating senses – the negation in 'there was no place' is likely due to the Coptic negation 'n' being a homonym with the preposition 'of', a distinction the UD tree makes clear by explicitly identifying prepositions.

Differing forms of syntax The improvements from syntactic information varies across models and depends on how it is incorporated. We evaluate four syntax-only settings: `CoNLLU`, `CON`, `DEP`, and `DEP+CON`. When added to our baseline, as shown in Table 2, GPT-4.1 and Gemma-27b achieve their largest F1 improvements even with just the `CoNLLU` component. In contrast, Gemma-12b improves the most from `CON`, actually seeing the least improvement from the `CoNLLU` component.

When added on top of `LEX`, as seen in Table 5, all syntactic components yield gains for the Gemma models. `CON` and `CoNLLU` provide larger improvements than `DEP` individually. In GPT-4.1, the simple use of `CoNLLU` outshines other syntactic components and this model sees limited or even negative gains from other components.

Improvements from `CoNLLU` alone show the potential for a low-effort way of improving performance, given high-quality NLP tools for the language.

Setting	Translation	Δ F1
Gemma-12b		
Reference	*He also said that Abba Pambo said:"This is the way that it is fitting for the monk to wear his clothe"*	–
Baseline	*And with all the powers through Jesus Christ our Lord, to give life to those who are sick.*	0
LEX	*Again, they say that the abbot has garments that it is fitting to bear, indeed, for the monks.*	+0.0790
LEX+SYN	*Again, he said that the abbot Pambo said that he is fitting for the monks to bear the garments.*	+0.1004
Reference	*Why do we now turn around and entangle us in them?*	–
Baseline	*And on the flesh of his body.*	0
LEX	*Again, let it turn around, and let it be wrapped in the upper part.*	+0.0220
LEX+SYN	*Why do we turn around to embrace the upper parts of them?*	+0.0912
Gemma-27b		
Reference	*Herodias set herself against him, and desired to kill him, but she couldn't*	–
Baseline	*The knowledge of God is the beginning of life. [. . .]*	0
LEX	*He is angry with the anger of the people. He wishes to kill with his strength*	+0.0190
LEX+SYN	*Herodias is angry, she wishes to kill him, but she is not able.*	+0.0981
Reference	*he gave them to the poor*	–
Baseline	*Into the water*	0
LEX	*Give to the poor*	+0.0392
LEX+SYN	*He gave to the poor*	+0.0898
GPT-4.1		
Reference	*And the mindless, whose acts of stupidity abound, are like bats.*	–
Baseline	*And they will weep bitterly in great lamentation; they will stay up through the nights.*	0
LEX	*And hearts that are greatly exalted among measures of grain; they are likened to stars.*	+0.0079
LEX+SYN	*And the mindless ones, who are exceedingly great in foolishness, are like bats.*	+0.0776
Reference	*Now on the mountainside there was a great herd of pigs feeding.*	–
Baseline	*But she did not die, for a great multitude of angels appeared to her.*	0
LEX	*But there was no place there on the mountain for the great herd of swine to feed.*	+0.0553
LEX+SYN	*But there was a place near him on the mountain, where a large herd of pigs was feeding.*	+0.0753

Table 4: Sample outputs with no augmentation (Baseline), with lexicon information added (LEX), and LEX+SYN (also includes all syntax components), as well as BERT-Score F1 improvement Δ for Gemma and GPT-4.1 models.

	Gemma		GPT-4.1
	12b	27b	
Setting	**BertScore**		
LEX	0.8593	0.8611	0.9139
+CoNLLU	+0.0053	+0.0076	**+0.0042**
+CON	+0.0107	+0.0094	−0.005
+DEP	+0.0005	−0.0009	+0.0001
+DEP+CoNLLU	+0.0053	+0.0076	+0.0036
+DEP+CON	+0.0112	+0.0105	−0.0010
LEX+SYN	**+0.0129**	**+0.0125**	+0.0030

Table 5: LEX + ablation results on dev. Each row displays the change from the baseline of LEX. We underline the best single-component score and bold the best overall score. Syntactic augmentation consistently improves performance, with strong score improvements from CON, CoNLLU, and SYN.

We generally see that while syntax alone is not as useful, it can provide significant complementary gains when layered on top of LEX.

Automatic parsing is good enough Coptic-NLP provides high quality parses (§3.2) with a labeled attachment score of 89.7 (Zeldes et al., 2025). These high-quality automatic parses are useful and effective. But are they as useful as gold standard parses?

	Gemma-12b		Gemma-27b	
	Auto	Gold	Auto	Gold
Setting	**BertScore**			
Baseline	0.8342	-	0.8375	-
CoNLLU	0.8453	0.8499	0.8522	0.8545
CON	0.8499	0.8502	0.8509	0.8512
DEP	0.8405	0.8411	0.8414	0.8413
DEP+CON	0.8490	0.8483	0.8509	0.8518

Table 6: The performance of different syntax related settings with the automatic parses (all other results), and explicit use of gold parses. The different in performance between the two settings exists, but is not drastic. Results of all the different settings, with multiple metrics can be seen in appendix (Table 12).

In general, we observe performance improvements with both gold and automatic parses, with no consistent difference between their benefits. Although we assume that parsing accuracy should

	Gemma-12b		Gemma-27b	
	Bible	**Other**	**Bible**	**Other**
Setting	**BertScore**			
Baseline	0.8323	0.8297	0.8378	0.8361
CoNLLU	0.8458	0.8436	0.8519	0.8482
CON	0.8537	0.8448	0.8544	0.8453
DEP	0.8422	0.8373	0.8423	0.8390
LEX	0.8585	0.8582	0.8635	0.8577
DEP+CON	0.8496	0.8467	0.8538	0.8455
LEX+SYN	0.8738	0.8685	0.8792	0.8665

Table 7: Results of Gemma model for dev reported for based on whether they are from the Bible (182 of 380). The BertScore in each setting (including baseline) is higher for the Biblical text. Results with BertScore and BLEU are reported in appendix (Table 13).

make a difference, for our data and samples we find no systematic impact on downstream translation quality, suggesting that automatic parsing errors from a ~90% accurate parse may not cascade into drastic translation errors. Table 6 shows the performance of the syntax settings of the open models (for dev) with automatic and gold parses.

Biblical text fares better Models may perform better in translating Biblical text than non-Biblical text (Liu et al., 2021). We see that to be the case for our dev set (Table 7). All settings do better for the Bible subset, to varying degrees.

It is also not hard to find qualitative examples in which baseline GPT-4.1 results strongly suggest recognition of Bible content. For example, for the reference target in (1) (Mark 7:34), the baseline GPT prediction in (2) is very good and gets 'be opened' right, likely because the Greek letters rendering the foreign term 'Ephphatha' give away the related Bible verse, which the model has memorized.

(1) Looking up to heaven, he sighed, and said to him, 'Ephphatha!' that is, 'Be opened!' (Reference)

(2) And when he had come near, he touched him and said to him,"Be opened" And immediately it was opened. (GPT-4.1)

6 Conclusion

We propose augmenting in-context translation for low-resource languages with syntactic information. Such augmentation could support better translation compared to just dictionary-based augmentation.

We build components for dictionary and multiple operationalizations of syntactic augmentation based on Universal Dependencies parses, for in-context translation of the low-resource language of Coptic. We validate these components with multiple open-weight (mainly Gemma) and reference closed-source (mainly GPT-4.1) models, and report results for both in-domain and out-of-domain evaluation sets.

Inclusion of syntactic information in addition to dictionary information provides higher quality translations than just supplying dictionary information, leading to statistically significant improvements in BERTScore F1 and observable improvements in translation quality. Gains in out-of-domain data for can also be seen in scores compared to previously reported results on Coptic ostraca.

We show that even limited linguistic resources can meaningfully improve in-context translation quality for low-resource languages, with improvements seen in both open and closed models.

The type of added syntactic information that is useful can vary by model: while larger models such as GPT-4.1 can make more effective use of UD parses in a raw, CoNLL-U format, smaller models benefit more from verbalizations. All models exhibit improvements from a dedicated construction module explaining constructions specific to the Coptic language and pointing out tricky configurations, as well as indicating and transliterating proper names. For GPT-4.1 this benefit emerges primarily when combined with raw parses.

Additionally, we also find minimal differences between gold and automatic UD parses, showing that high-quality automatic parses are sufficient for in-context translation, at least given parser performance of ~90%. Our error analysis highlights how different information can provide support for the models to identify both structural and relational information to improve translations.

We are hopeful that these results will lead to more work on incorporating abstract syntactic or even semantic information in general (for example WordNet or similar resources, Slaughter et al. 2019, or semantic analysis graphs such as UMR, Van Gysel et al. 2021), and outputs from automatic UD parsing in particular, into ICL approaches to low-resource machine translation.

Limitations

Small dataset Our experiments are on a dataset of less than 800 translation samples. This limits the generalizability of our findings.

Single operationalization of lexicon We consider dictionary augmentation with a single source, and one operationalization, based on the resource we use (although aligned with TEI standard). However, dictionaries can differ and consequently augmentation may not be adaptable.

Single formalism While Universal Dependencies is both the most popular and diverse (across languages) collection of treebanks, it is not the only syntactic or grammatical formalism available. However, the verbalizations are more generic, and could help mitigate this limitation.

Non exhaustive settings combination We selected a subset of experimental settings, and some combinations of linguistic components were not included, such as `DEP+CoNLLU`.

Prompt length Our settings incorporate multiple linguistic components, producing long prompts in some cases. The length of these prompts can provide additional challenges for LLMs, as they are sensitive to context length, possibly affecting translation quality for more complex experimental settings.

Unattested metrics Coptic, being a low-resource language, has not had extensive investigation of MT metrics and their relation to human judgment. Although our target language is English, there is no attestation of metrics for Coptic-English translation. Hence the reported results may not represent human judgments of translation quality.

Lack of human evaluation We do not perform any human evaluation, but rather only use automatic metrics based on human-translated references.

Acknowledgments

The work in this paper was supported by the Georgetown University Massive Data Institute (MDI) through funding for an MDI Scholar. Compute and inference resources provided through the Department of Linguistics and the Massive Data Institute at Georgetown University were used for this work.

References

Mathew Almond, Joost Hagen, Katrin John, Tonio Sebastian Richter, and Vincent Walter. 2013. Kontaktinduzierter sprachwandel des ägyptisch-koptischen: Lehnwort-lexikographie im projekt database and dictionary of greek loanwords in coptic (ddglc). *Perspektiven einer corpusbasierten historischen Linguistik und Philologie*, pages 283–315.

Anthropic. 2020. Claude 3.5 sonnet model card addendum.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, and 1 others. 2022. Building machine translation systems for the next thousand languages.

Dylan M. Burns, Frank Feder, Katrin John, and Maxim Kupreyev. 2020. Comprehensive Coptic lexicon: Including loanwords from Ancient Greek.

Nasma Chaoui and Richard Khoury. 2025. Neural machine translation for Coptic-French: Strategies for low-resource ancient languages. *Preprint*, arXiv:2508.10683.

Sara Court and Micha Elsner. 2024. Shortcomings of LLMs for low-resource translation: Retrieval and understanding are both the problem. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1332–1354, Miami, Florida, USA. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

David M. Eberhard, Gary F. Simons, and Alison J. Robinson, editors. 2026. *Ethnologue: Languages of the World. Twenty-ninth edition*. SIL International, Dallas, Texas.

Micha Elsner and Jordan Needle. 2023. Translating a low-resource language using GPT-3 and a human-readable dictionary. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–13, Toronto, Canada. Association for Computational Linguistics.

Frank Feder, Maxim Kupreyev, Emma Manning, Caroline T. Schroeder, and Amir Zeldes. 2018. A linked

Coptic dictionary online. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 12–21, Santa Fe, New Mexico. Association for Computational Linguistics.

Samuel Frontull and Thomas Ströhle. 2025. Compensating for data with reasoning: Low-resource machine translation with LLMs. *Preprint*, arXiv:2505.22293.

Marjan Ghazvininejad, Hila Gonen, and Luke Zettlemoyer. 2023. Dictionary-based phrase-level prompting of large language models for machine translation. *Preprint*, arXiv:2302.07856.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Tom Kocmi, Sweta Agrawal, Ekaterina Artemova, Eleftherios Avramidis, Eleftheria Briakou, Pinzhen Chen, Marzieh Fadaee, Markus Freitag, Roman Grundkiewicz, Yupeng Hou, Philipp Koehn, Julia Kreutzer, Saab Mansour, Stefano Perrella, Lorenzo Proietti, Parker Riley, Eduardo Sánchez, Patricia Schmidtova, Mariya Shmatova, and Vilém Zouhar. 2025a. Findings of the WMT25 multilingual instruction shared task: Persistent hurdles in reasoning, generation, and evaluation. In *Proceedings of the Tenth Conference on Machine Translation*, pages 414–435, Suzhou, China. Association for Computational Linguistics.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakougna, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025b. Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets. In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413, Suzhou, China. Association for Computational Linguistics.

Alon Lavie, Greg Hanneman, Sweta Agrawal, Diptesh Kanojia, Chi-Kiu Lo, Vilém Zouhar, Frederic Blain, Chrysoula Zerva, Eleftherios Avramidis, Sourabh Deoghare, Archchana Sindhujan, Jiayi Wang, David Ifeoluwa Adelani, Brian Thompson, Tom Kocmi, Markus Freitag, and Daniel Deutsch. 2025. Findings of the WMT25 shared task on automated translation evaluation systems: Linguistic diversity is challenging and references still help. In *Proceedings of the Tenth Conference on Machine Translation*, pages 436–483, Suzhou, China. Association for Computational Linguistics.

Ling Liu, Zach Ryan, and Mans Hulden. 2021. The usefulness of Bibles in low-resource machine translation. In *Proceedings of the 4th Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 44–50, Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Hongyuan Lu, Haoran Yang, Haoyang Huang, Dongdong Zhang, Wai Lam, and Furu Wei. 2024. Chain-of-dictionary prompting elicits translation in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 958–976, Miami, Florida, USA. Association for Computational Linguistics.

So Miyagawa. 2025. RAG-enhanced neural machine translation of Ancient Egyptian text: A case study of THOTH AI. In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 33–40, Albuquerque, USA. Association for Computational Linguistics.

Attila Nagy, Dorina Lakatos, Botond Barta, and Judit Ács. 2023. TreeSwap: Data augmentation for machine translation via dependency subtree swapping. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 759–768, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Juan N. Pava, Caroline Meinhardt, Haifa Badi Uz Zaman, Toni Friedman, Sang T. Truong, Daniel Zhang, Elena Cryst, Vukosi Marivate, and Sanmi Koyejo. 2025. Mind the (language) gap: Mapping the challenges of LLM development in low-resource language contexts.

Renhao Pei, Yihong Liu, Peiqin Lin, François Yvon, and Hinrich Schuetze. 2025. Understanding in-context machine translation for low-resource languages: A case study on Manchu. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8767–8788, Vienna, Austria. Association for Computational Linguistics.

Siyao Peng and Amir Zeldes. 2018. All roads lead to UD: Converting Stanford and Penn parses to English Universal Dependencies with multilayer annotations. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 167–177, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Muhammed Saeed, Asim Mohamed, Mukhtar Mohamed, Shady Shehata, and Muhammad Abdul-Mageed. 2024. From Nile sands to digital hands: Machine translation of Coptic texts. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 298–308, Bangkok, Thailand. Association for Computational Linguistics.

Caroline T. Schroeder and Amir Zeldes. 2016. Raiders of the lost corpus. *Digital Humanities Quarterly*, 10(2).

Laura Slaughter, Luis Morgado Da Costa, So Miyagawa, Marco Büchler, Amir Zeldes, and Heike Behlmer. 2019. The making of Coptic Wordnet. In *Proceedings of the 10th Global Wordnet Conference*, pages 166–175, Wroclaw, Poland. Global Wordnet Association.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A benchmark for learning to translate a new language from one grammar book. In *The Twelfth International Conference on Learning Representations*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Jens Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Timothy J. O'Gorman, Andrew Cowell, W. Bruce Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a uniform meaning representation for natural language processing. *Künstliche Intelligenz*, 35:343–360.

Audric-Charles Wannaz and So Miyagawa. 2024. Assessing large language models in translating Coptic and Ancient Greek ostraca. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 463–471, Miami, USA. Association for Computational Linguistics.

Amir Zeldes and Mitchell Abrams. 2018. The Coptic Universal Dependency treebank. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 192–201, Brussels, Belgium. Association for Computational Linguistics.

Amir Zeldes and Caroline T. Schroeder. 2015. Computational methods for Coptic: Developing and using part-of-speech tagging for digital scholarship in the humanities. *Digital Scholarship in the Humanities*, 30(1):164–176.

Amir Zeldes and Caroline T. Schroeder. 2016. An NLP pipeline for Coptic. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 146–155, Berlin, Germany. Association for Computational Linguistics.

Amir Zeldes, Nina Speransky, Nicholas E. Wagner, and Caroline T. Schroeder. 2025. A UD treebank for bohairic Coptic. In *Proceedings of the Eighth Workshop on Universal Dependencies (UDW, SyntaxFest 2025)*, pages 59–69, Ljubljana, Slovenia. Association for Computational Linguistics.

Chen Zhang, Jiuheng Lin, Xiao Liu, Zekai Zhang, and Yansong Feng. 2025. Read it in two steps: Translating extremely low-resource languages with code-augmented grammar books. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3977–3997, Vienna, Austria. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Data and Resources

Resource	Reference
UD Treebank	Zeldes and Abrams, 2018
Ostraca	Wannaz and Miyagawa, 2024
Coptic Dictionary	Feder et al., 2018
Coptic NLP	Coptic Scriptorium

Table 8: The different data and resources used in this paper.

Table 8 collates the core resources used in this paper.

Details about the Dictionary

The lexicon contains multiple entries for a single form.

(3) ⲉⲓⲣⲉϩⲧⲓ[11]

a. except
b. nevertheless

A specific entry can contain elaborate information, including translations in multiple languages, and for multiple dialects of Coptic. We focus on and use the Sahidic dialect.

(4) except[12]

a. except (for), if not
b. except (for), without that, unless
c. but

B System and Experiments

B.1 Discussion on BLEU and chrF++

We also report chrF++ (Popović, 2017) for many of the experiments as an additional reference metric.

We note that, as discussed in Post (2018), BLEU can be parameterized in multiple ways. We use the implementation from the sacrebleu package,[13] and use default settings for dev and ostraca.[14] Since BLEU is designed as a corpus level metric, the default settings of BLEU lead to some 0 scored outputs for the test split, so we used a relaxed setting with max 3-gram, effective order, and 'floor' smoothing with 0.1.[15]

BLEU is specifically not useful for the test set since, although the floor smoothing allows for non-zero BLEU scores, but leads to the common value 13.56 for many settings. BLEU is not the primary metric in the paper, but is kept for completness.

BLEU is unreliable in test BLEU as a metric was somewhat representative if not correlated with BertScore for dev and ostraca, but with default signature the BLEU scores for test collapsed to 0 for many settings. We have reported a smoothed BLEU for test to be consistent.

This is mainly a function of the model output utilizing a wider variety of surface forms in conflict with an n-gram overlap mechanism that BLEU is based upon, but is not distinct from size of the reference or the output sentence.

Consider the following translation sentence: 'For you have taken the things of the poor and the widows and the orphans, and you have placed them in your window.' Gemma-3-12b with the DEP+CON setting responds simply with 'You did.' This leads to a 0 BLEU score without smoothing. Overall there are only 29 of the 405 sentences that show this behavior across the three test models we report in Table 2. This highlights for Coptic the need for further work in validating the representativeness and quality of the wide-ranging metrics that exist for MT (Lavie et al., 2025).

BLEU is inconsistent in dev Unlike with test, BLEU is not unstable on dev; however it is inconsistent. In Table 2, for Gemma-12b we can see that BLEU does not follow the same pattern as BertScore. This is partly due to the difference in the two class of metrics. Since BLEU relies on n-gram precision, it is more sensitive to lexical choice.

Compare the output translation 'And the remainder of the great ones in the palace.' with the reference translation 'as did also the other nobles who were attached to the Palace.' The translation did capture some semantic elements such as 'great ones' in comparison to 'nobles' but by design, BLEU would consider 0-precision, while BertScore would catch some quotient of similar-

[11] https://coptic-dictionary.org/results.py?quick_search=C8880

[12] https://coptic-dictionary.org/entry.py?tla=C8880

[13] https://github.com/mjpost/sacrebleu

[14] The sacrebleu metric signature is nrefs:379|case:mixed|eff:no|tok:13a|smooth:none|version:2.5.1

[15] The metric signature is nrefs:405|case:mixed|eff:yes|tok:13a|smooth:floor[0.10]|version:2.5.1.

ity. Such patterns are more common in LLM based generation, compared to older controlled or limited vocabulary methods.

B.2 `LEX` parameters and grid search

Our `LEX` component has multiple configurable parameters, one controlling the target languages we include from dictionary and the rest for controlling the information added.

Two parameters control the first-k entries that we include after retrieval and the first-k senses for each entry that we include. The final parameter is for a deduplication feature to specifically handle the verbose enumeration for the DDGLC (Almond et al., 2013) portion of the dictionary.

The grid search for `LEX` was performed with the best setting for `DEP` as decided by the previous grid search. Then the `LEX` grid search was performed on the sub-selected 10 each of the easiest and hardest to translate for our pilot model.

Fixed setting The final parameters were to use only entries from English, no more than 100 entries per sample, with no more than 10 senses per entry, and without deduplicating the information from DDGLC.

B.3 `DEP` parameters and grid Search

Our `DEP` component is configurable with three different parameters controlling how duplicate tokens are handled, the robustness of verbalized dependency relations, and the inclusion of various ‘tiers’ of POS tags.

When handling duplicates, we choose subscript notation (’ⲉϩⲣⲁⲓ$_2$ is the case marking of ⲱⲟⲱⲟⲩ’) or verbalized nominalizations (’the second ⲉϩⲣⲁⲓ is the case marking of ⲱⲟⲱⲟⲩ).

When choosing which dependency relations to verbalize in this component, we either ‘collapse’ the relations or maintain the full list. For example, the non-collapsed setting verbalizes each dependency relation as they are defined in UD: *ccomp* is expressed as a clausal complement, and *xcomp* as an open clausal complement. In the collapsed setting, both *ccomp* and *xcomp* are verbalized as simply ‘complement’.

Finally, we choose from three tiers of POS tags to verbalize in this `DEP` component: content, participants, or all. We excluded punctuation, symbols, and ‘other’ tags across the board. The content tier is the most restrictive, allowing only for nouns, verbs, proper nouns, adpositions, and adverbs. This setting results in the shortest final `DEP` component as it filters out the most relations from the final verbalization. The participants tier expands upon the content tier, adding pronouns, auxiliaries, determinants, and numerals. The final tier is the most inclusive, adding conjunctions to the previous list.

Fixed setting Following our grid search, we determined that our best setting is to verbalize duplicates as *subscripts, not collapse* the dependency relations into more general categories, and to use the *participants tier* of POS tags.

B.4 Constructions `CON`

The `CON` component provides information and instruction targeting specific source language constructions. Construction prompts are triggered by 26 DepEdit rules which identify syntax subtrees using a declaration of nodes to be found and subgraph relations which must hold between them.

For example, when a verb is accompanied by both the future and preterit auxiliaries, whose meanings individually correspond to future tense (‘will VERB’) and continuous past (‘was VERBing’), the combination results in a counterfactual conditional (‘would have VERBed’). This is captured by declaring three nodes to match: the VERB (via POS tag) and the two dependents (the auxiliaries, via their lemmas and POS tags) in a particular order (preterite, then future, then the VERB).

The DepEdit module rule can then extract annotations attached to these nodes, such as their form (since both the VERB and auxiliaries can inflect), lemma, or morphological categories such as person or gender. The template for the counterfactual construction is verbalized as:

(5) The combination of the future auxiliary {{FUT. AUX FORM}} with the counterfactual preterit {{PRET. AUX FORM}} is used together with the predicate {{VERB FORM}} to express a counterfactual meaning (would have VERBed).

The list of construction triggers and their verbalizations is available with the full code in the paper’s GitHub repository.

B.5 Instruction Design

Figure 3 shows how different parts of the instruction appear. Figure 4 shows the CoNLL-U separate from the rest of the instructions. Figure 2 shows how these are ordered and provided in our prompt in a condensed manner.

C Full results

Model (Size)	Model Card
Gemma3(-it) (12B)	HF Hub : link
Gemma3(-it) (27B)	HF Hub : link
GPT-4.1 (NA)	OpenAI Platform : link
GPT-4.1 Mini (NA)	OpenAI Platform : link
Llama3.1(-Inst) (8B)	HF Hub : link
Aya-Expanse (8B)	HF Hub : link
Aya-Expanse (32B)	HF Hub : link

Table 9: Models used for our experiments with `dev` data and corresponding model cards.

We reported the results of the Gemma models and GPT-4.1 for `dev` and `test` in the main body. Table 10 contains the `dev` results for all the open models we experimented with. The `DEP+CON` setting is reported only for the Gemma models, and not for Aya or Llama models.

Settings using `CoNLLU` and `LEX` exceed the context window of models such as Aya-Expanse, which have smaller context windows (8K compared to the 128K of the other models we used), nonetheless we report it for completeness. This is visible in their performance under those settings, since much of the queue is truncated.

Ostraca Table 11 shows the performance of our Gemma and GPT models across the various settings for the ostraca set.

Automatic parses vs gold parses Table 12 shows the results for the default usage of automatic parses versus the use of gold parses. Note that our baseline does not use parses in any form, hence performance is equal.

Bible Texts Table 13 shows the difference in performance between Biblical text and other text in `dev`.

D Significance testing

For LMs (`dev`, and `test`), contrasts were compared using mixed effects models implemented in R using the library `lme4`. Because the sentences translated are identical in all settings, we treat sentence ID as a random effect (repeated measures) and the addition of the lexicon and syntax augmentations as independent fixed effects, predicting the quality metric (BERT-Score F1). Single term deletions with likelihood ratio tests demonstrate that both lexicon and syntax augmentations improve translation quality as assessed by the metric for $p < 0.0001$.

E Implementation and resources

We ran our experiments with open-weighted models on Nvidia H100 GPUs. We maintain explicit requirements for reproducibility, and used a fixed random seed of 42 in all our local runs for replicability.

With fixed random seed, we only need single run each for our model, setting, split that we report. Especially since there is no random effect commonly found in other training or augmentation. This was tested by running some of the experiments multiple times and checked for consistency.

Table 9 lists the specific model cards for each of the models used.

Hyperparameters We use max tokens of 128 for inference with both local and API models and use a fixed random seed. We used greedy decoding to avoid variance due to temperature. Data results and code are all documented in the repository for replicability and reproducibility.

Gridsearch parameters of the components are documented and available in paper (appendices B.2 and B.3) and available in the repository.

F AI Tool Use

We used GitHub Copilot and Gemini when developing the code for this paper.

Base

You are a professional Coptic-to-English translator tasked with providing translations suitable for use in United States (en_US). Your goal is to accurately convey the meaning and nuances of the original Coptic text while adhering to English grammar, vocabulary, and cultural sensitivities.Produce only the English translation, without any additional explanations or commentary.Please translate the following Coptic text into English (en_US): {*source*}.

Closing cue for consistency.

Using all the information provided above, now please translate the sentence into English(en_US). Remember your source sentence is: {*source*}.The English translation is:

Lexicon

For the translation task, you are given dictionary entries for Coptic. Some words can be polysemous and there might be multiple entries. Each entry can contain multiple senses with translations in ['English']. In such a case, please choose the most appropriate one. Note that for some words, they might be derived from a more basic form, some entries will be for such lemma.
Here are the entries for collected for individual words in the sentence:

Dictionary entry Verb ⲭⲱ has 2 senses.
Sense 1:
- In English, ⲭⲱ means say, speak, tell
Sense 2:
- In English, ⲭⲱ means sing ...

Dependency

The dependency information for the sentence is: ⲡⲣⲣⲏ is the root. ⲉϩⲣⲁⲓ$_1$ is the case marking of ⲃⲁⲧⲕⲁⲗⲟⲛ. ⲉ$_1$ is the fixed multiword expression of ⲉϩⲣⲁⲓ$_1$. ⲃⲁⲧⲕⲁⲗⲟⲛ is the oblique nominal of ⲡⲣⲣⲏ. ⲏ is the coordinating conjunction of ⲱⲟⲱⲟⲩ. ⲉϩⲣⲁⲓ$_2$ is the case marking of ⲱⲟⲱⲟⲩ. ⲉ$_2$ is the fixed multiword expression of ⲉϩⲣⲁⲓ$_2$. ⲱⲟⲱⲟⲩ is the conjunct of ⲃⲁⲧⲕⲁⲗⲟⲛ. ⲉϩⲣⲁⲓ$_3$ is the case marking of ⲗⲁⲁⲩ ...

Construction

The information about specific constructions in the sentence is: The dislocated element ⲡⲁⲓ is a repeated reference to the pronoun dependent of the predicate ⲭⲱ. There is often no need to translate the pronominal mention of the same argument. ...

Figure 3: An example of the content in differrent sections of the Instruction to LM. The CONLL-U is separately shown in Figure 4.

CoNLL-U

1	ⲙⲙ	ⲡ	ADP	PREP	_	3	case	_	_
2	ⲡ	ⲡ	DET	ART	_	3	det	_	_
3	ⲍⲁⲣⲓⲟⲥ	ⲍⲁⲣⲓⲟ	NOUN	N	_	0	root	_	_
4	ⲃⲓⲕⲧⲱⲣ	ⲃⲓⲕⲧⲱⲣ	PROPN	NPROP	_	3	appos	_	_
5	ⲡⲉ	ⲡ	DET	ART	_	6	det	_	_
6	ⲥⲣⲁⲧⲏⲗⲁⲧⲏⲥ	ⲥⲧⲣⲁⲧⲏⲗⲁⲧⲏⲥ	NOUN	N	_	3	appos	_	_
7	ⲁⲧⲱ	ⲁⲧⲱ	CCONJ	CONJ	_	9	cc	_	_
8	ⲡ	ⲡ	DET	ART	_	9	det	_	_
9	ⲙⲙⲁⲣⲧⲩⲣⲟⲥ	ⲙⲙⲁⲣⲧⲩⲣⲟⲥ	NOUN	N	_	6	conj	_	_
10	ⲉⲧ	ⲉⲧⲉⲣⲉ	SCONJ	CREL	_	11	mark	_	_
11	ⲧⲁⲓϩⲩ	ⲧⲁⲉⲓⲟ	VERB	VSTAT	_	9	acl:relcl	_	_
12	ⲙⲙ	ⲡ	ADP	PREP	_	14	case	_	_
...									

Figure 4: An example CoNLL-U data format, which would also be included into the instruction.

Model	Setting	BLEU	BertScore	chrF++
Aya-Expanse-32b	Baseline	3.95	0.7939	14.78
Aya-Expanse-32b	LEX	8.60	0.8085	17.26
Aya-Expanse-32b	CoNLLU	8.25	0.7841	14.35
Aya-Expanse-32b	CON	4.42	0.8017	15.84
Aya-Expanse-32b	DEP	1.63	0.7997	15.06
Aya-Expanse-32b	LEX+SYN	9.48	0.8303	22.30
Aya-Expanse-8b	Baseline	6.99	0.8132	15.34
Aya-Expanse-8b	LEX	4.40	0.8404	21.60
Aya-Expanse-8b	CoNLLU	2.85	0.8265	16.51
Aya-Expanse-8b	CON	8.16	0.8250	15.89
Aya-Expanse-8b	DEP	6.00	0.8073	15.11
Aya-Expanse-8b	LEX+SYN	2.39	0.8239	19.65
Gemma-3-12b	Baseline	16.45	0.8342	16.29
Gemma-3-12b	LEX	5.46	0.8593	23.19
Gemma-3-12b	CoNLLU	27.49	0.8453	18.17
Gemma-3-12b	CON	20.26	0.8499	17.30
Gemma-3-12b	DEP	28.66	0.8405	16.34
Gemma-3-12b	DEP+CON	21.65	0.8490	18.15
Gemma-3-12b	LEX+SYN	7.71	0.8722	27.29
Gemma-3-27b	Baseline	10.30	0.8375	17.88
Gemma-3-27b	LEX	14.46	0.8611	23.52
Gemma-3-27b	CoNLLU	14.86	0.8522	21.65
Gemma-3-27b	CON	15.43	0.8509	19.27
Gemma-3-27b	DEP	15.28	0.8414	18.83
Gemma-3-27b	DEP+CON	10.38	0.8509	20.05
Gemma-3-27b	LEX+SYN	18.84	0.8736	28.67
Llama-3.1-8B-Instruct	Baseline	8.42	0.7843	12.55
Llama-3.1-8B-Instruct	LEX	3.59	0.8004	16.30
Llama-3.1-8B-Instruct	CoNLLU	6.63	0.8009	13.30
Llama-3.1-8B-Instruct	CON	4.58	0.7965	14.16
Llama-3.1-8B-Instruct	DEP	3.60	0.7885	12.06
Llama-3.1-8B-Instruct	LEX+SYN	1.94	0.8108	19.36

Table 10: The results from dev for all the open-weight models we considered. Note DEP+CON was run only reported for Gemma models for analysis focused on disambigation.

Model	Setting	BLEU	BertScore	METEOR
Gemma-3-12b	Baseline	5.72	0.8380	0.11
Gemma-3-12b	LEX+SYN	7.82	0.8850	0.31
Gemma-3-12b	CoNLLU	4.88	0.8536	0.18
Gemma-3-12b	CON	3.84	0.8520	0.10
Gemma-3-12b	DEP	7.50	0.8470	0.11
Gemma-3-12b	LEX	5.48	0.8640	0.26
Gemma-3-12b	DEP+CON	6.42	0.8591	0.12
Gemma-3-27b	Baseline	5.61	0.8490	0.13
Gemma-3-27b	LEX+SYN	16.18	0.8781	0.33
Gemma-3-27b	CoNLLU	15.22	0.8622	0.21
Gemma-3-27b	CON	13.32	0.8550	0.13
Gemma-3-27b	DEP	2.39	0.8525	0.13
Gemma-3-27b	LEX	9.44	0.8621	0.22
Gemma-3-27b	DEP+CON	9.69	0.8599	0.16
GPT-4.1	Baseline	16.25	0.8792	0.31
GPT-4.1	LEX+SYN	17.99	0.90	**0.53**
GPT-4.1	CoNLLU	6.18	0.8877	0.39
GPT-4.1	CON	9.42	0.8888	0.31
GPT-4.1	DEP	<u>18.15</u>	0.8859	0.37
GPT-4.1	LEX	14.27	0.8973	0.42
GPT-4.1	DEP+CON	9.21	0.8862	0.32
From Wannaz and Miyagawa (2024)				
Claude Opus		20.02	-	0.46
Claude Haiku		11.52	-	0.35
CopticTranslator		8.43	-	0.30

Table 11: Results for ostraca §3.1 for Gemma models and GPT-4.1. The BLEU and METEOR scores for Claude (Opus and Haiku) and CopticTranslator are from Wannaz and Miyagawa (2024).

		Automatic			Gold		
Model	**Setting**	**BLEU**	**BertScore**	**chrF++**	**BLEU**	**BertScore**	**chrF++**
Gemma-3-12b	Baseline	16.45	0.8342	16.29	-	-	-
Gemma-3-12b	DEP+CON	21.65	0.8490	18.15	14.56	0.8483	18.09
Gemma-3-12b	CON	20.26	0.8499	17.30	21.74	0.8502	17.37
Gemma-3-12b	DEP	28.66	0.8405	16.34	28.66	0.8411	16.18
Gemma-3-12b	CoNLLU	27.49	0.8453	18.17	13.50	0.8499	19.43
Gemma-3-12b	LEX	5.46	0.8593	23.19	6.45	0.8581	22.84
Gemma-3-12b	LEX+SYN	7.71	0.8722	27.29	14.22	0.8727	27.08
Gemma-3-27b	Baseline	10.30	0.8375	17.88	-	-	-
Gemma-3-27b	DEP+CON	10.38	0.8509	20.05	20.08	0.8518	20.15
Gemma-3-27b	CON	15.43	0.8509	19.27	12.87	0.8512	19.30
Gemma-3-27b	CoNLLU	14.86	0.8522	21.65	10.90	0.8545	22.68
Gemma-3-27b	LEX	14.46	0.8611	23.52	9.56	0.8604	23.22
Gemma-3-27b	DEP	15.28	0.8414	18.83	21.39	0.8413	18.74
Gemma-3-27b	LEX+SYN	18.84	0.8736	28.67	17.11	0.8770	29.97

Table 12: Results for the open-weight Gemma models for dev using Automatic and Gold UD parses (§3.6). The higher performance (BertScore) among the two for each setting is underlined. We don't report specific significance for this comparison, although the lower variability across the Automatic and Gold than among the different settings is encouraging.

		Bible		Other	
Model	**Setting**	**BLEU**	**BertScore**	**BLEU**	**BertScore**
Gemma-3-12b	Baseline	9.55	0.8323	9.73	0.8297
Gemma-3-12b	LEX+SYN	19.30	0.8738	6.17	0.8685
Gemma-3-12b	CoNLLU	10.70	0.8458	18.77	0.8436
Gemma-3-12b	CON	17.29	0.8537	12.55	0.8448
Gemma-3-12b	DEP	20.86	0.8422	18.68	0.8373
Gemma-3-12b	LEX	26.08	0.8585	5.03	0.8582
Gemma-3-12b	DEP+CON	17.75	0.8496	20.01	0.8467
Gemma-3-27b	Baseline	9.78	0.8378	8.68	0.8361
Gemma-3-27b	LEX+SYN	23.87	0.8792	14.72	0.8665
Gemma-3-27b	CoNLLU	21.87	0.8519	10.83	0.8482
Gemma-3-27b	CON	17.29	0.8544	13.22	0.8453
Gemma-3-27b	DEP	15.46	0.8423	12.57	0.8390
Gemma-3-27b	LEX	19.67	0.8635	13.27	0.8577
Gemma-3-27b	DEP+CON	13.13	0.8538	9.15	0.8455

Table 13: Results of Gemma model for dev reported as Bible and Not Bible text. The data are distinct but not drastically disproportinate (182 Bible, rest 198 Other). Performance for Bible data (by BERTScore) is consistently better than Other, which is to interesting, but not suprising for such models w.r.t to low-resource language.