Multiple Sources are Better Than One: Incorporating External Knowledge in Low-Resource Glossing

Anonymous ACL submission

Abstract

In this paper, we address the data scarcity problem in automatic data-driven glossing for low-resource languages by coordinating multiple sources of linguistic expertise. We supplement models with translations at both the token and sentence level as well as leverage the extensive linguistic capability of modern LLMs. Our enhancements lead to an average absolute improvement of 5%-points in word-level accuracy over the previous state of the art on a typologically diverse dataset spanning six low-resource languages. The improvements are particularly noticeable for the lowest-resourced language Gitksan, where we achieve a 10%-point improvement. Furthermore, in a simulated ultra-low resource setting for the same six languages, training on fewer than 100 glossed sentences, we establish an average 10%-point improvement in word-level accuracy over the previous state-of-the-art system.

1 Introduction

The extinction rate of languages is alarmingly high, with an estimated 90% of the world's languages at risk of disappearing within the next century (Krauss, 1992). As speech communities dwindle, linguists are urgently prioritizing the documentation of these languages. This is a multi-step process involving: 1. phonetic and orthographic transcription, 2. translation into a so-called *matrix language* like English or Spanish, which provides a common frame of reference for all annotations, 3. morpheme segmentation, and 4. grammatical annotation (Crowley, 2007). The end-result is represented as Interlinear Glossed Text (IGT) like the Gitksan example below (see Appendix A for additional details):

Orthography:	Ii hahla'lsdi'y goohl IBM
Segmentation:	ii hahla'lst-'y goo-hl IBM
Gloss:	CCNJ work-1SG.II LOC-CN IBM
Translation:	And I worked for IBM.

The traditional manual approach to language documentation, while thorough, is notably labor-intensive. This has spurred the development of automated tools leveraging machine learning for tasks such as word segmentation and glossing. For example, Moeller and Hulden (2018) train neural models for automatic glossing of Lezgi, a Nakh-Daghestanian language. Their models deliver reasonable performance when trained on a small training set of 3,000 glossed tokens of Lezgi text. However, neural models are data-hungry and the small training set prevents the models from reaching their full potential. The most straightforward way to improve model performance would be to manually gloss more training data. However, as stated above, manual glossing is a very time-consuming process. Therefore, additional data sources should be considered.

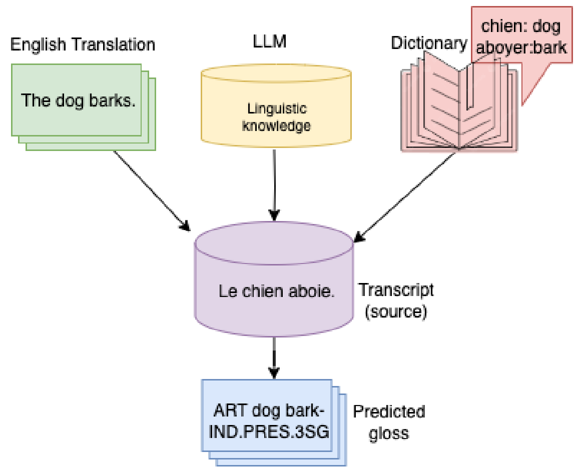

Figure 1: When glossing input such as the French sentence *Le chien aboie*, our system utilizes multiple information sources: an English sentence-level translation, general linguistic knowledge provided by an LLM and dictionary definitions for the input tokens.

Many recent glossing approaches (Girrbach, 2023; Moeller and Hulden, 2018) exclusively train on glossed source language transcripts. However, we often have access to additional helpful knowledge sources. One option is to augment the data using translations of the training examples into the matrix language.[1] These provide an important source of lexical information because the gloss of nouns and verbs can often be found within the translation.[2] Because translation is a part of the language documentation process, these are often readily available and, thus, represent a quick and cost-effective way to provide an additional source of supervision. Our system incorporates translations as an added information source.

Unfortunately, the availability of translations for IGT data is necessarily limited simply because the quantity of IGT data itself is limited. As an additional source of lexical information, our system incorporates external dictionaries which provide word-level translations of target language lexemes into the matrix language. This helps the system generalize to words missing from the training data.

Recently, powerful pretrained models have emerged as a viable approach to strengthen and supplement the training signal for NLP tasks in low-resource settings (Ogueji et al., 2021; Bhattacharjee et al., 2021; Hangya et al., 2022). Advancements in large language models (LLMs) also present new opportunities for enhancing the language documentation process. Pretrained language models such as BERT (Devlin et al., 2018) and LLMs like GPT-4 (Achiam et al., 2023), trained on billions of tokens of text, encode extensive lexical and linguistic knowledge in the matrix language, and their incorporation has improved the benchmarks in many natural language tasks (Zhao et al., 2023; Bommasani et al., 2021; Zhou et al., 2023). We integrate LLMs into our glossing pipeline as a post-correction step through in-context learning. It is worth noting that our approach does not require fine-tuning and is, therefore, appropriate in low-resource settings where compute capacity is limited.

By leveraging three external sources of information (see Figure 1): utterance translations, external dictionaries and LLMs, our glossing pipeline achieves an average absolute improvement of 5%-points over the previous state-of-the-art on datasets from the SIGMORPHON 2023 Shared Task on Interlinear Glossing (Ginn et al., 2023). In particular, the incorporation of dictionaries leads to significant advancements for ultra-low resource languages such as Gitksan, resulting in a 10%-points increase in word-level accuracy. Our key contributions are:

1. We enhance the training of glossing systems—in addition to plain glossed training examples, we introduce additional supervision in the form of input translations which are encoded using a pre-trained language model.
2. We utilize external dictionaries which improve glossing performance, particularly for the lowest-resourced languages.
3. We pioneer the use of LLM prompting and in-context learning techniques as a post-correction step in the glossing pipeline. To our knowledge, this is the first time LLMs have been applied to the automatic glossing task. Our findings show that in-context prompting results in substantial improvements, especially when very limited training data is available.

2 Related Work

Interlinear Glossing Research into automatic glossing starts with rule-based analysis (Bender et al., 2014; Snoek et al., 2014) followed by data-driven neural models (Moeller and Hulden, 2018; Girrbach, 2023; Ginn and Palmer, 2023; Zhao et al., 2020). More recently, the integration of pre-trained multilingual models (Ginn et al., 2024; Sheikh et al., 2024) has shown great potential to aid documentation projects. Our work is inspired by the success of these powerful models and aims to build upon their strengths.

Integrating Translation into the Glossing Task We are not unique in incorporating translation information into a glossing system in the presence of small training datasets. The system presented by Okabe and Yvon (2023) is based on CRFs (Sutton et al., 2012), and also employs translations. However, in contrast to our approach, they heavily rely on source and target word alignments derived from an unsupervised alignment system (Jalili Sabet et al., 2020). In low-resource settings, it is hard to learn an accurate alignment model. [3]

[1] Frequently, the matrix language will be English but can also be another language like Spanish or Russian.

[2] For the French sentence *Le chien aboie*, the correct gloss of both *chien* 'dog' and *aboyer* 'bark' can be found in its English translation: *The dog barks*.

[3] Moreover, Okabe and Yvon (2023) assume morphologically-segmented input, which considerably simplifies the glossing task. We instead address the much harder task of predicting glosses without segmentation information.

Pioneering studies by Zoph and Knight (2016), Anastasopoulos and Chiang (2018) and Zhao et al. (2020), show that leveraging translations can enhance the performance of a neural glossing system. A notable limitation in all of these approaches is the scarcity of available English translations for training models. Therefore, only modest improvements in glossing accuracy are observed. Our work, in contrast, incorporates translation information through large pre-trained language models, which leads to greater improvements in glossing performance. This strategy has lately become increasingly popular in low-resource NLP and shows promise across various language processing tasks (Ogueji et al., 2021; Hangya et al., 2022).

Similarly to our approach, Okabe and Yvon (2023) also take advantage of the BERT model in their study, but only utilize BERT representations for translation alignment. In contrast, we directly incorporate encoded translations into our glossing model. He et al. (2023) also use pre-trained language models, namely, XLM-Roberta (Conneau et al., 2020), mT5 (Xue et al., 2021) and ByT5 (Xue et al., 2022), as part of their glossing model. However, they do not incorporate IGT translation information.[4] Instead, they directly fine-tune the pre-trained models for glossing.

LLM Prompting In recent years, the application of LLMs for various NLP tasks has expanded significantly, demonstrating remarkable potential in few-shot and in-context learning. This approach leverages the inherent knowledge and adaptability of LLMs like GPT-4 (Achiam et al., 2023) and LLaMA-3 (Touvron et al., 2023), allowing them to perform tasks based on a few examples provided as context, without requiring further fine-tuning. Margatina et al. (2023) introduce a novel perspective by applying active learning (AL) principles to in-context learning with LLMs. Their study frames the selection of in-context examples as a pool-based AL problem conducted over a single iteration. Various AL algorithms, including uncertainty, diversity, and similarity-based sampling, is explored to identify the most informative examples for in-context learning. The findings consistently indicate that selecting examples semantically similar to the test instances significantly outperforms other methods, including random sampling and traditional uncertainty-based approaches .

[4]Though He et al. (2023) do use external dictionary information for post-correction of glosses.

Language	Train(num)	Dev(num)	Test(num)	Matrix lang.
Arapaho (arp)	39,501	4,938	4,892	(eng)
Gitksan (git)	31	42	37	(eng)
Lezgi (lez)	701	88	87	(eng)
Natügu (ntu)	791	99	99	(eng)
Tsez (ddo)	3,558	445	445	(eng)
Uspanteko (usp)	9,774	232	633	(spa)

Table 1: 2023 Sigmorphon Shared Task Dataset Information (Ginn et al., 2023)

Building on these insights, our proposed work aims to enhance the task of automatic glossing in low-resource settings by integrating LLM prompting and active learning principles. Our approach applies the strategies outlined by (Margatina et al., 2023) by focusing on similarity-based methods for selecting in-context examples. This ensures that the most relevant and informative examples are utilized, enhancing the model's ability to generate accurate glosses. Additionally, we explore the effectiveness of various active learning methods such as BERT-similarity, word overlapping, longest common subsequence, and random sampling, tailoring these approaches to the specific needs of the glossing task.

3 Data

We conduct experiments on data from the 2023 SIGMORPHON shared task on interlinear glossing (Ginn et al., 2023). The shared task provides two distinct tracks: an open track, where the input is morphologically segmented, and a closed track, where no segmentations are provided. Our analysis focuses on data from the closed track. This setting is substantially more challenging because morphological segmentation now, effectively, becomes a part of the glossing task. The closed-track languages are Arapaho (arp), Gitksan (git), Lezgi (lez), Natügu (ntu), Tsez (ddo), and Uspanteko (usp).[5] Data details are shown as in Table 1. With most languages, except Arapaho, comprising fewer than training 10,000 sentences, our datasets can be called low-resourced. For all languages, the data includes translations in a matrix language which is English, except from Uspanteko, where it is Spanish.

4 Baseline Model

Our glossing system is based upon a neural glossing model developed by Girrbach (2023). This is

[5]We exclude one language Nyangbo, because its dataset lacks translations.

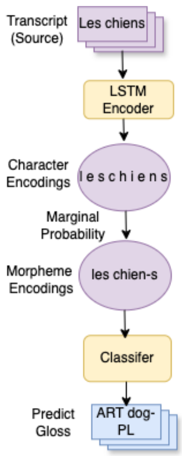

Figure 2: Pipeline of Girrbach (2023)'s model.

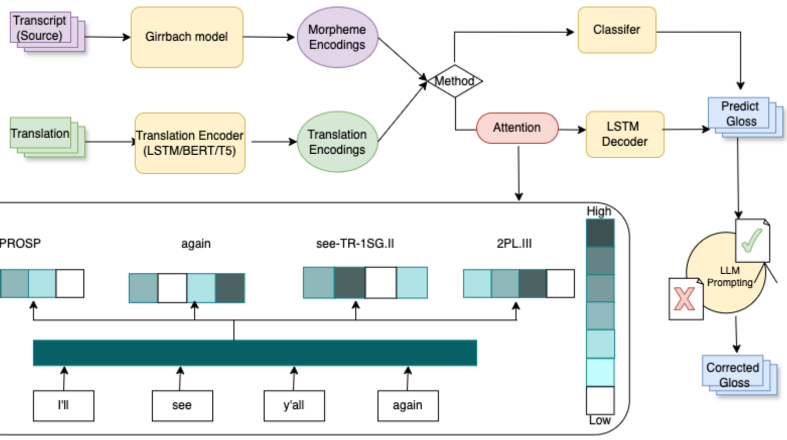

Figure 3: Pipeline of the proposed work. The lower portion of the diagram demonstrates how attention weights inform the model when predicting the glossing targets.

the winning system of the 2023 SIGMORPHON shared task on internlinear glossing. As shown in Figure 2, the model accomplishes glossing of morphological segments through a three-stage process: input encoding, unsupervised morpheme segmentation, and morpheme classification.

Input encoder The model input consists of a character-sequence $s = s_1, \ ..., \ s_N$, representing a sentence. A bidirectional long short-term memory network (BiLSTM) encodes the input into a sequence of contextualized embeddings $\mathbf{h}_i$, one for every character in s.

Morpheme Segmenter Next, the model performs unsupervised morphological segmentation using the forward-backward algorithm (Kim et al., 2016). In a first step, an MLP is used to predict the number of morphemes J_w for each word w in input sentence s. For each character s_i, the model applies a linear layer with Sigmoid activation function to its character encoding $\mathbf{h}_i$ to get the probability p_i^{seg} that indicates whether s_i is the last character of the morpheme segment. Then the forward and backward scores (α and β, respectively) for each input position i and target morpheme j can be computed as follows:

$$\alpha_{i,j} = \alpha_{i-1,j} \cdot (1 - p_{i-1}^{\text{seg}}) + \alpha_{i-1,j-1} \cdot p_{i-1}^{\text{seg}}$$
$$\beta_{i,j} = \beta_{i+1,j} \cdot (1 - p_{i}^{\text{seg}}) + \beta_{i+1,j+1} \cdot p_{i}^{\text{seg}}$$

Finally, the marginal probability of a morpheme boundary at position i relating to morpheme j is given by:

$$\xi_{i,j} = \frac{\alpha_{i,j} \cdot \beta_{i,j}}{\alpha_{N,J_w}}$$

where N is the sequence length, and J_w is the number of morphemes in the word w.

Morpheme classifier After segmentation, we get each morpheme encoding $\mathbf{e_j}$ through averaging its corresponding character encodings. An MLP is then used to predict the gloss for each morpheme based on its morpheme encoding. Model training optimizes the cross-entropy loss between the predicted and ground-truth gloss labels.

5 Our Methods

Our glossing system enhances the baseline model by incorporating utterance translations (both at the sentence level and token level) and a character-based decoder.[6] Model and training details are provided in Appendix B. Additionally, we implement a gloss post-correction component using LLM-powered in-context learning. Figure 3 presents an overview of the system.

5.1 Character-Based Gloss Decoder

Our first addition to the Girrbach (2023) model is a character-based decoder. The baseline model is unable to predict glosses which were not observed in the training data, because it treats glossing as a morpheme classification task with a closed set of potential gloss labels. This deficiency is particularly harmful when predicting glosses for lexical morphemes (i.e. word stems) which represent a much larger inventory than grammatical morphemes (i.e. inflectional and derivational affixes). A character-based decoder can enhance the model's capability

[6] Our code is publicly available: https://link/to/our/repo

to use words from a translation of the input example. Following Kann and Schütze (2016), we implemented a LSTM decoder. However, we adapt it to function at the character level for lexical morphemes and at the morpheme level for grammatical morphemes. [7]

5.2 Translation Encoder

We then extend the model of Girrbach (2023) by incorporating matrix-language translations. We encode the English or Spanish (in the case of Uspanteko) translations in the shared task datasets using a deep encoder. We experiment with three different encoders: a character-based BiLSTM (Hochreiter and Schmidhuber, 1997) and pre-trained transformers BERT-base (Kenton and Toutanova, 2019) and T5-large (Raffel et al., 2020).[8] To represent translations, we then either use the final hidden state from the translation encoder, or attend over the translation hidden states.

When attending over the hidden states, we apply Bahdanau attention (Bahdanau et al., 2014) scoring the association between each encoder hidden states and the previous decoder state $\mathbf{d}_{i-1}$. We separately attend to the encoded morpheme representations $\mathbf{e_j}$ in the input example (morphemes are discovered by our baseline model in an unsupervised manner as explained above) and the encoded subword-tokens $\mathbf{t_k}$ in the translation. This gives us a morpheme representation $\mathbf{e}_i = \sum_{j=1}^{J} w_j^e \mathbf{e_j}$ and a translation representation $\mathbf{t}_i = \sum_{k=1}^{K} w_k^t \mathbf{t_k}$ at time-step i. We then use the concatenated representation $[\mathbf{e}_i; \mathbf{t}_i]$ to compute the next gloss decoder state $\mathbf{d}_i$.

5.3 Post-correction through in-context learning

Preliminary experiments revealed that the glossing system sometimes generates typos and non-sensical glosses such as *stoply* instead of *story*. To mitigate this issue, we introduce a post-correction step leveraging LLM prompting. We enhance the accuracy and reliability of glosses through an in-context learning approach.

For each language, we generate conservative silver glosses (requiring correction) using a BERT-based model with attention (BERT+attn+chr) to prevent excessive corrections, as the baseline model (Girrbach, 2023) already provides a reasonably accurate starting point. We use one-quarter of the training data to produce silver glosses for the remaining training data, fine-tuning the model on the original development split. To reduce noise, we apply an edit distance constraint, retaining examples where the gloss edit distance from the gold gloss is limited to 4-8 characters.[9] The initial one-quarter of data is then reintroduced into the training set, ensuring completeness and accuracy, as these glosses match the original training data.

Here we prepare a prompt which asks the LLM to correct the lexical morphemes in a glossed input sentence. A prompt is generated by selecting two training examples as in-context learning examples for each test example. Each in-context learning example includes the source language transcript, morpheme/word translations based on the training data, the English translation of the sentence, the silver gloss, and the gold gloss. The test example is structured similarly but omits the gold gloss, prompting the language model to generate the corrected gloss. The prompting pipeline is illustrated in Figure 4. When using an external dictionary, we additionally provide word translations in the prompt. Following the in-context paradigm, we do not perform any further training or fine-tuning of the LLM. The template used for the prompting is detailed in Appendix F. We experiment with two models in this scenario: GPT-4 (Achiam et al., 2023) and LLaMA-3 (Touvron et al., 2023).

In-context Learning Examples Selection Techniques In our experiment, we compare three techniques to optimize the selection of in-context learning examples. We evaluate these techniques against random selection. **BERT Similarity (BERT-Sim)** We first embed the translated test sentence from the IGT using BERT (we use multilingual BERT for Uspanteko). We then find the two training sentences with the lowest embedded cosine distance from the test case, and use them as our in-context examples. **Overlapping Words (Overlap)** We calculate the number of overlapping words between source sentences in the test and training datasets. In-context examples are selected to maximize the number of overlapping words between the test case and the training sentences. **Longest Common Substrings (LCS)** We select in-context examples from the training sentences that maximize the LCS with the test case.

[7] For instance, if the word gloss is "dog-FOC", the decoder will generate it as "d-o-g-FOC".

[8] See Appendix B for details concerning the encoders.

[9] The character number is determined by half the length of the word glosses, depending on the language.

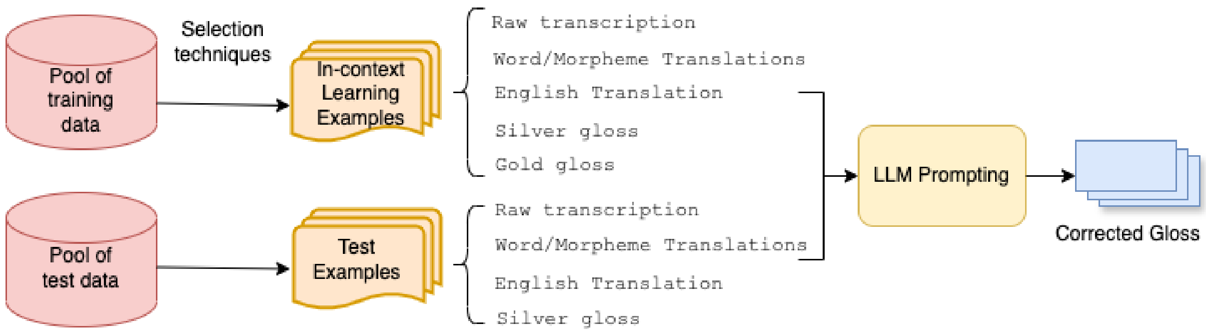

Figure 4: The procedure of selecting in-context learning examples to generate components for LLM prompting.

6 Experiments and Results

In all experiments, we evaluate based on word-level glossing accuracy.

6.1 Translation Enriched Model Results

Table 2 shows the glossing accuracy across different model settings and languages.[10] We report performance separately for original shared task datasets and our simulated ultra low-resource datasets spanning 100 training sentences. We group the Gitksan shared task dataset in the ultra low-resource category because it only has 30 training examples.[11]

Shared Task Data When only integrating translations through the final state of a bidirectional LSTM, we observe an improvement in average glossing accuracy, but performance is reduced for two languages (Arapaho and Uspanteko).

Augmenting translations via an attentional mechanism (LSTM+attn) does not confer consistent improvements. In contrast, translation information incorporated via a pre-trained model (BERT+attn) renders consistent improvements in glossing accuracy across all languages and we see notable gains in average glossing accuracy over the baseline. Incorporating a character-based decoder leads to further improvements in average glossing accuracy and for all individual languages. The T5 model (T5+attn+chr) attains the highest average performance: 82.56%, which represents a 3.97%-points improvement over the baseline. It also delivers the highest performance for three out of our five test languages (Arapaho, Lezgi and Tsez), while the BERT-based model with attention (BERT+attn+chr) delivers the best performance for the remaining two (Natügu and Uspanteko). Among all languages, we see improvements over the baseline model ranging from 2.32%-points to 5.95%-points.[12]

Ultra Low-Resource Data In order to investigate the performance of our model in ultra low-resource settings, we additionally form smaller training sets by sampling 100 sentences from the original shared task training data. We use the original shared task development and test sets for validation and testing, respectively.

Translations integrated through the final state of a randomly initialized bidirectional LSTM (LSTM and LSTM+attn), lead to an average 6%-points improvement in accuracy over the baseline. We achieve particularly impressive gains for Uspanteko, surpassing the baseline accuracy by over 15%-points. Incorporating pre-trained models (BERT+attn) exhibits a slight increase in accuracy for certain languages. However, when we incorporate both pre-trained models and the character-based decoder (BERT+attn+chr and T5+attn+chr), we see larger gains in accuracy across the board. Here, BERT achieves the highest average accuracy of 42.04%, which represents a 9.78%-points improvement over the baseline. It achieves the highest performance for three languages (Arapaho, Gitksan and Uspanteko), while T5 delivers the best performance for two of the languages (Lezgi and Natügu). The plain LSTM model attains the best performance for Tsez.

6.2 Prompting Model Results

The prompting experiments aim to further improve the output of the T5/BERT+attn+chr model by post-

[10]We additionally present edit distance in Appendix C.

[11]Apart from the baseline, all systems apply majority voting from 10 independently trained models. Its impact is discussed in Appendix D.

[12]We visualize the attention patterns over the English translation representations. The visualizations are shown in Appendix E

Model setting	arp	lez	ntu	ddo	usp	ave	arp-low	git-low	lez-low	ntu-low	ddo-low	usp-low	ave
Girrbach (2023)	78.79	78.78	81.04	80.96	73.39	78.59	19.12	21.09	48.84	51.08	36.12	17.32	32.26
LSTM	77.04	81.42	83.55	84.99	73.01	80.00	18.67	20.71	54.29	59.56	**44.5**	32.92	38.44
LSTM+attn	79.31	76.19	83.01	85.12	76.24	79.97	24.38	18.49	55.75	58.48	42.37	29.52	38.17
BERT+attn	78.98	81.87	84.57	85.84	77.63	81.78	27.33	20.31	55.86	60.13	41.85	33.04	39.75
BERT+attn+chr	80.79	82.19	**85.41**	84.13	**79.34**	82.37	**28.82**	**28.11**	56.99	62.73	39.72	**35.84**	**42.04**
T5+attn+chr	**81.11**	**82.37**	84.68	**85.91**	78.72	**82.56**	27.31	24.23	**57.33**	**62.82**	39.97	33.59	40.88

Table 2: Word-level accuracy of languages in the 2023 Sigmorphon Shared Task (Ginn et al., 2023) (left) and ultra low-resource settings (right). Model specifics are elaborated in Section 5.

Model setting	arp	lez	ntu	ddo	usp	git
T5/BERT+attn+chr	81.11	82.37	85.41	**85.91**	**79.34**	28.11
+GPT4-random	81.12	83.52	85.79	84.76	70.62	28.58
+GPT4-BERT-Sim	81.17	**84.70**	86.07	85.32	72.44	29.02
+GPT4-Overlap	**81.57**	84.47	86.11	85.53	73.64	29.14
+GPT4-LCS	81.25	83.86	**86.38**	84.98	72.78	28.77
+LLaMA3-Overlap	81.23	83.01	86.09	83.77	70.99	**30.11**

Table 3: Word-level accuracy of all languages. We incorporate prompts using different selection techniques for in-context examples, which add into the information enriched models (T5/BERT+attn+chr).

Model setting	arp	lez	ntu	ddo	usp	git	ave
Girrbach (2023)	78.79	78.78	81.04	80.96	73.39	21.09	69.01
T5/BERT+attn+chr	81.11	82.37	85.41	**85.91**	**79.34**	28.11	**73.88**
T5/BERT+attn+chr+Prmpt	**81.57**	**84.70**	**86.38**	85.53	73.64	**30.11**	73.66

Table 4: Word-level accuracy of all languages. We compare the performance of models that incorporate prompts from our optimal in-context example selection techniques with other models.

correcting its glossed output using an LLM. We only allow the LLM to change the gloss of lexical morphemes because preliminary experiments demonstrated that post-processing tends to worsen performance on grammatical morphemes. The word-level accuracy shown in Table 3 highlights the performance of various training data selection techniques across multiple languages.[13] We further select the best setting to compare with the baseline model and translation enriched models. The comparison demonstrates that using in-context learning continues to boost glossing accuracy. This approach delivers further improvements for Arapaho, Lezgi, Natügu, and Gitksan. It presents the highest accuracy for Lezgi , showing a 2.33%-points increase over the highest-performing translation enriched model T5/BERT+attn+chr.

When applying GPT-4 for post-correction, the Overlapping Words selection technique emerges as the most effective, achieving the highest accuracy for Arapaho at 81.57% and maintaining strong performance across other languages. The BERT similarity and LCS techniques also provide substantial improvements over random selection, with notable improvements for Lezgi at 84.70% and Natügu at 86.38% accuracy, respectively. Additionally, the LLaMA-3 model using the Overlapping Words method shows competitive results, particularly excelling in the low-resource language Gitksan at 30.11%, indicating its potential utility in such challenging settings.

We further examine predictions from the prompting model. One such example in Lezgi includes a sentence whose translation is "*She was lonely*". The pre-corrected gloss from our encoder-decoder model (T5/BERT+attn+chr) contains incorrect lexical morpheme glosses, including “pie” and “he”. It is evident that the prompting model successfully changed these lexical morphemes according to the words in the translation line of the IGT[14]. Results are as shown below:

Silver Gloss:	pie old.woman was he
Prompt Gloss:	alone old.woman was still.be
Gold Gloss:	alone old.woman was still.be,.remain

Interestingly, both the GPT-4 and LLaMA-3 in-context learning setups perform worse when the translations are in Spanish than in English, as evidenced by the accuracy drop in Uspanteko. The reasons behind this require further investigation.

6.3 External Dictionaries

We also assess the impact of introducing additional word translations into the in-context prompts to enhance accuracy. We expand the word translations in the prompt using word translations from an external dictionary for Arapaho, Lezgi, and Gitksan. The source and detailed information about the dic-

[13] We additionally present lexical morpheme accuracy in Appendix G.

[14] We observe that the prompting results can contain synonyms. To gain a better understanding of our model’s performance, we use BERT score as an alternative evaluation metric to evaluate the lexical morphemes. Results are shown in Appendix H.

Model setting	arp	lez	git	ave
Girrbach (2023)	78.79	78.78	21.09	59.55
T5/BERT+attn+chr	81.11	82.37	28.11	63.86
T5/BERT+attn+chr+Prmpt	81.57	84.70	30.11	65.46
T5/BERT+attn+chr+Prmpt+Dict	**81.61**	**85.30**	**31.32**	**66.08**

Table 5: Word-level accuracy of all languages. We compare the model performance among the accumulated effort of incorporating external dictionaries with other models.

tionaries are shown in Appendix I. The word-level results, as presented in Table 5 illustrate that the integration of out-of-domain dictionary resources is highly beneficial, especially for languages with limited training data like Gitksan. Dictionary translations consistently boost the performance of our best models, enhancing benefits obtained solely through prompting. The dictionary-supplemented models achieve the best results in all three languages, with an overall average accuracy of 66.08%, surpassing the baseline model by 6.53%-points and the plain prompting model by 0.62%-points.

6.4 Learning Curves

The learning curves in Figure 5 illustrate the impact of prompting on model performance when using varying amounts of IGT training data. This comparison includes models with and without prompting, focusing on both word-level and lexical morpheme accuracy. We focuse on the Arapaho language, which has the largest number of manually glossed training examples: 39,501 training sentences, in total.

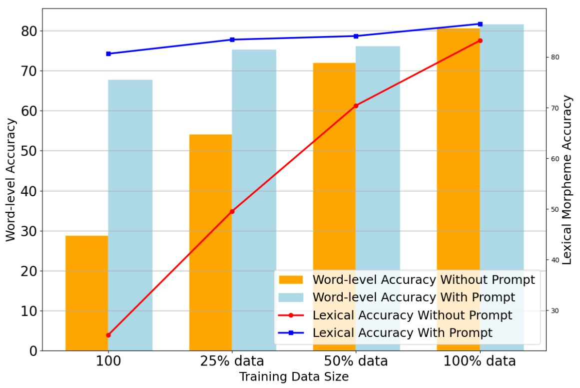

Figure 5: Lexical morpheme and word-level accuracy on Arapaho. We incorporate prompting with the encoder-decoder model which is enriched with translation.

The bar chart represents the word-level accuracy for models trained with varying amounts of data (100 sentences, 25% data, 50% data, and 100% data). The results clearly demonstrate that in-context post-correction greatly improves glossing accuracy. In ultra-low data conditions, the post-corrected model is more than twice as accurate as the uncorrected model. As the amount of training data increases, the benefits gained through prompting diminish.

The line chart maps the accuracy of lexical morphemes prior- and post-correction. Similarly to the word-level accuracy, the accuracy of lexical morphemes benefits greatly from in-context post-correction. The most significant improvements are again observed when training data is restricted. With only 100 training sentences, the post-corrective model achieves a lexical morpheme accuracy that is nearly as high as that obtained using the full dataset.

7 Conclusions

This paper offers a promising and efficient solution by introducing multiple resources to aid in the glossing task, particularly in linguistically diverse and data-sparse environments. The current study demonstrates the effectiveness of incorporating translation information at both the token and sentence level, alongside LLM prompting in automatic glossing for low-resource languages. The proposed system, based on a modified version of Girrbach's model (Girrbach, 2023), shows significant performance enhancements, particularly in low-resource settings. By leveraging translation data and integrating a character-based decoder, our approach provides a robust solution for unobserved lexical morphemes (stems).

This research pioneers the application of LLM prompting to the glossing task. By employing various in-context example selection strategies and adding extra dictionary words as a resource, we have shown that LLM prompting can substantially refine lexical morpheme glosses, leading to higher word-level accuracy. This approach is also particularly beneficial in scenarios with limited training data, as it maximizes the potential of minimal data resources.

In all, the integration of translation information, additional dictionary resources, along with LLM prompting, sets a new benchmark in automatic glossing.

8 Limitations

The limitations of our study primarily pertain to the extent of our experimentation and the models we have chosen. Firstly, our investigation relies solely on an LSTM decoder. This decision was influenced by time constraints, which limited our ability to explore more complex decoders. Additionally, our experimentation is confined to the T5-large model. While this model has shown promising results in our study, we acknowledge the existence of other large language models in the field of natural language processing. Although we did explore other large language models such as LLaMA-2 (Touvron et al., 2023), our preliminary experiments yielded unsatisfactory results compared to T5. Consequently, we made the decision not to include LLaMA-2 in our paper due to its inferior performance. These limitations underscore the need for future research to explore a wider range of decoding architectures and incorporate various large language models to enhance our understanding of the subject matter. However, using large language models requires significant computational resources, which can have an environmental impact due to increased energy consumption.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Antonis Anastasopoulos and David Chiang. 2018. Leveraging translations for speech transcription in low-resource settings. *arXiv preprint arXiv:1803.08991*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Emily M. Bender, Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. 2014. Learning grammar specifications from IGT: A case study of chintang. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 43–53, Baltimore, Maryland, USA. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Terry Crowley. 2007. *Field linguistics: A beginner's guide*. OUP Oxford.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Michael Ginn, Sarah Moeller, Alexis Palmer, Anna Stacey, Garrett Nicolai, Mans Hulden, and Miikka Silfverberg. 2023. Findings of the SIGMORPHON 2023 shared task on interlinear glossing. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 186–201.

Michael Ginn and Alexis Palmer. 2023. Taxonomic loss for morphological glossing of low-resource languages. *arXiv preprint arXiv:2308.15055*.

Michael Ginn, Lindia Tjuatja, Taiqi He, Enora Rice, Graham Neubig, Alexis Palmer, and Lori Levin. 2024. Glosslm: Multilingual pretraining for low-resource interlinear glossing. *arXiv preprint arXiv:2403.06399*.

Leander Girrbach. 2023. Tü-CL at SIGMORPHON 2023: Straight-through gradient estimation for hard attention. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–165, Toronto, Canada. Association for Computational Linguistics.

Viktor Hangya, Hossain Shaikh Saadi, and Alexander Fraser. 2022. Improving low-resource languages in pre-trained multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11993–12006.

Taiqi He, Lindia Tjuatja, Nathaniel Robinson, Shinji Watanabe, David R Mortensen, Graham Neubig, and Lori Levin. 2023. SigMoreFun submission to the SIGMORPHON shared task on interlinear glossing. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 209–216.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. 2016. Structured attention networks. In *International Conference on Learning Representations*.

Michael Krauss. 1992. The world's languages in crisis. *Language*, 68(1):4–10.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034, Singapore. Association for Computational Linguistics.

Sarah Moeller and Mans Hulden. 2018. Automatic glossing in a low-resource setting for language documentation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 84–93, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126.

Shu Okabe and François Yvon. 2023. Towards multilingual interlinear morphological glossing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5958–5971, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Zaid Sheikh, Antonios Anastasopoulos, Shruti Rijhwani, Lindia Tjuatja, Robbie Jimerson, and Graham Neubig. 2024. Cmulab: An open-source framework for training and deployment of natural language processing models. *arXiv preprint arXiv:2404.02408*.

Conor Snoek, Dorothy Thunder, Kaidi Lõo, Antti Arppe, Jordan Lachler, Sjur Moshagen, and Trond Trosterud. 2014. Modeling the noun morphology of Plains Cree. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 34–42, Baltimore, Maryland, USA. Association for Computational Linguistics.

Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Xingyuan Zhao, Satoru Ozaki, Antonios Anastasopoulos, Graham Neubig, and Lori Levin. 2020. Automatic interlinear glossing for under-resourced languages leveraging translations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5397–5408, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.

A IGT Information

In the IGT data, the second line includes segmentations with morphemes normalized to a canonical orthographic form. The third line has an abbreviated gloss for each segmented morpheme. Lexical morphemes typically correspond to the stems of words. The morpheme glosses usually have two categories: Lexical and Grammatical morphemes. For example, in glossing labels such as work-1SG.II, "work" would be considered a Lexical morpheme, representing the core semantic unit. On the other hand, Grammatical morphemes like '1SG.II" are often denoted by uppercase glosses and generally signify grammatical functions, such as tense, aspect, or case, rather than specific lexical content.

B Model Settings

Our experimental framework and hyperparameters draw inspiration from Girrbach's methodology, with a focus on organizing and optimizing the technical setup. For model optimization, we employ the AdamW optimizer (Loshchilov and Hutter, 2017), excluding weight decay, and set the learning rate at 0.001. Except for this specific adjustment, we maintain PyTorch's default settings for all other parameters.

Our configuration is structured to allow a range of experiments, varying from 1 to 2 LSTM layers, with hidden sizes spanning from 64 to 512, and dropout rates fluctuating between 0.0 and 0.5. The scheduler γ is adjusted within a range of 0.9 to 1.0, and batch sizes are diversified, ranging from 2 to 64. This versatile approach is designed to thoroughly evaluate the model's performance across a spectrum of hyperparameter configurations.

Departing from the original model which was trained for 25 epochs, our approach extends the training duration to 300 epochs when using large pretrained models. In cases where the BERT model is utilized, we sometime apply a 0.5 dropout rate during the BERT training phase. We exclusively employ the multilingual BERT model for Uspanteko, while we utilize the standard BERT model for all other languages. This comprehensive and meticulously organized setup is aimed at enhancing the effectiveness and efficiency of our model training process.

To prevent coincidences, for each proposed model configuration, we train the model for 10 iterations, and the final prediction is determined through majority voting.

C Edit Distance

Results are shown in Table 6.

D Influence of Majority Voting

Average accuracy across 10 models and results utilized majority voting are shown in Table 7. Improvements in performance can be achieved even without resorting to voting, particularly accentuated in ultra low-resource datasets as opposed to the Shared Task datasets.

E Attention Distribution

To assess whether our model is able to successfully incorporate translation information, we visualize attention patterns (from the BERT+attn+chr model) over the English translation representations. Figure 6 presents an example for Natügu. Attention weights are displayed in a heat map, where each cell indicates difference from mean attention: $a - 1/(n+2)$. Here n is the length of the translation in tokens ($+2$ here because of the start-of-sequence and end-of-sequence tokens [CLS] and [SEP] which are concatenated to the translation). Positive red cells inidicate high attention and negative blue cells low attention. The visualization clearly indicates that the model attends to the relevant tokens in the translation when predicting the stems *people*, *mankind* and *kill*. Figure 7-Figure 12 shows randomly picked heat maps for the rest of the languages. We can see that attention weights for the larger shared task datasets tend to express relevant associations, while attention weights for the ultra low-resource training sets largely represent noise. Figure 7-Figure 12 also displays attention distributions when translations are incorporated using a randomly initialized LSTM instead of a pre-trained language model. These distributions also largely represent noise indicating that pre-trained models confer an advantage.

F Prompt template

You are a linguistic annotator for the Gitksan language, tasked with correcting errors in glossing based on translation details and morpheme translations. Your task is to adjust errors in the stems (in lowercase) without changing the total number of morphemes or words in the gloss. Each gloss element is separated by hyphens within morphemes and spaces between words.

Here are two examples:

Model setting	ara	git(-low)	lez	ntu	ddo	usp	ara-low	lez-low	ntu-low	ddo-low	usp-low
Girrbach (2023)	-	-	-	-	-	-	6.59	3.64	4.78	4.92	3.79
LSTM	1.52	5.65	1.22	**1.17**	0.72	0.88	6.50	3.28	4.12	**3.93**	2.84
LSTM+attn	**1.31**	6.27	1.62	1.34	0.72	0.86	6.04	3.26	3.81	4.25	3.21
BERT+attn	1.39	5.57	1.24	1.23	0.69	**0.70**	5.97	3.20	3.81	4.1	2.88
BERT+attn+chr	1.50	**5.30**	1.20	1.25	0.53	0.81	**5.54**	3.04	**3.55**	4.27	2.78
T5+attn+chr	1.40	5.51	**1.18**	1.27	**0.52**	0.78	5.62	**3.00**	**3.55**	4.36	**2.74**

Table 6: Word-level edit distance of languages in the 2023 Sigmorphon Shared Task (Ginn et al., 2023) (left) and low-resource settings (right), with ‘arp’ representing Arapaho, ‘git’ for Gitksan, ‘lez’ for Lezgi, ‘ntu’ for Natügu, ‘ddo’ for Tsez, and ‘usp’ for Uspanteko. Model specifics are elaborated in Section 2.

Model setting	arp	lez	ntu	ddo	usp	ave	arp-low	git-low	lez-low	ntu-low	ddo-low	usp-low	ave
Girrbach (2023)	78.79	78.78	81.04	80.96	73.39	78.59	19.12	21.09	48.84	51.08	36.12	17.32	32.26
BERT/T5+attn+chr-average	79.32	79.49	80.76	81.00	74.92	79.10	25.43	23.95	54.28	57.18	32.41	28.77	37.00
BERT/T5+attn+chr-majority	**81.11**	**82.37**	**85.41**	**85.91**	**79.34**	**82.83**	**28.82**	**28.11**	**57.33**	**62.82**	**39.97**	**35.84**	**42.14**

Table 7: Word-level accuracy of languages in the 2023 Sigmorphon Shared Task (Ginn et al., 2023) and low-resource settings. We compute the average across 10 models and also utilized majority voting accuracy results. Language abbreviations were used, with ‘arp’ representing Arapaho, ‘git’ for Gitksan, ‘lez’ for Lezgi, ‘ntu’ for Natügu, ‘ddo’ for Tsez, and ‘usp’ for Uspanteko. Model specifics are elaborated in Section 2.

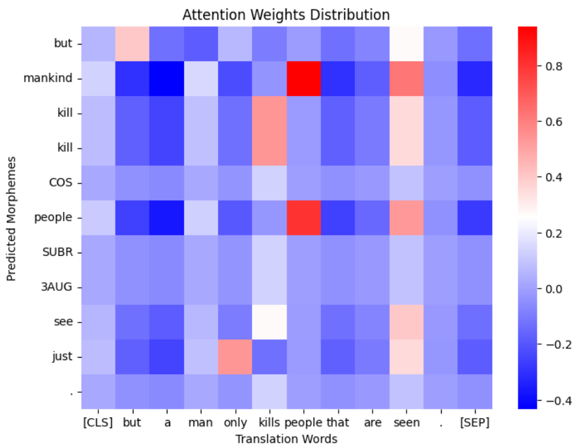

Figure 6: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for a Natügu example (attention weights are derived from the model BERT+attn+chr).

Example 1: Gitksan sentence is {example['train1-raw-sentence']}. You are provided with morpheme translations according to the dictionary: {example['train1-word/morpheme-translation']}. The English translation for this sentence is: {example['train1-sentence-translation']}. The glossing pending to be revised is: {example['train1-silver-gloss']}. The corrected gloss is {example['train1-gold-gloss']}.

Example 2: Gitksan sentence is {example['train2-raw-sentence']}. You are provided with morpheme translations according to the dictionary: {example['train2-word/morpheme-translation']}. The English translation for this sentence is: {example['train2-sentence-translation']}. The glossing pending to be revised is: {example['train2-silver-gloss']}. The corrected gloss is {example['train2-gold-gloss']}.

Now, here's the gloss you need to correct: Gitksan sentence is {example['test-raw-sentence']}. You are provided with morpheme translations according to the dictionary: {example['test-word/morpheme-gloss']}. The English translation for this sentence is: {example['test-translation']}. The glossing pending to be revised is: {example['test-silver-gloss']}. What is the corrected gloss for this sentence? You should answer in this format: **The corrected gloss is:** (your generated answer). Note, don't change the total number of words or morphemes in the gloss.

G Lexical Morpheme Accuracy

Here we only evaluate the lexical morpheme accuracy. Results are shown in Table 8.

H BERT score

Specifically, we compare tokens using BERT embeddings and calculate similarity scores with the BERT model. The results are shown in Table 9. As we do not have access to the results from Girrbach

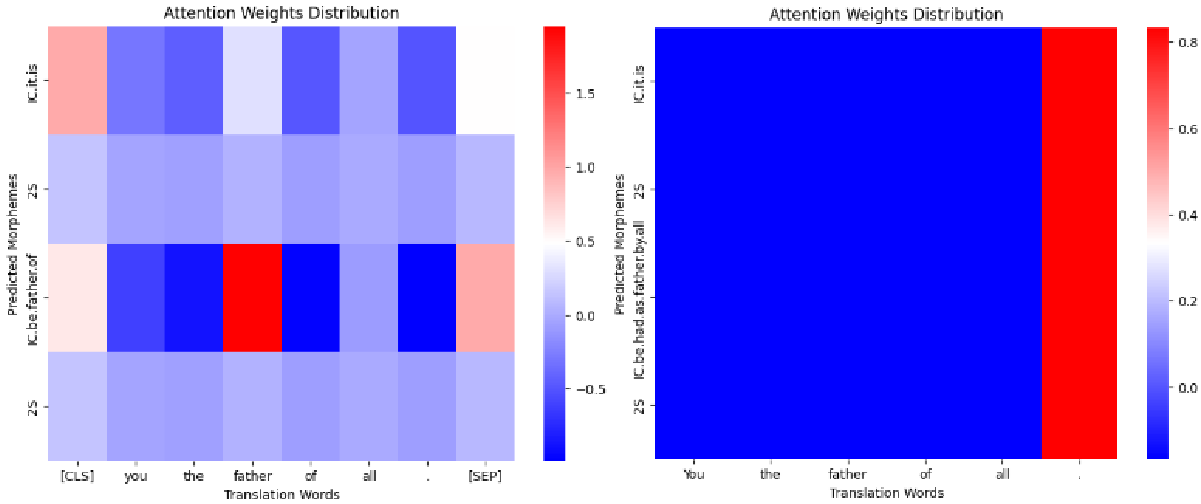

Figure 7: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for an Arapaho example (attention weights are derived from the model BERT+attn+chr (left) and the model LSTM+attm (right)). The gold-standard glosses for this sentence: IC.it.is-2S IC.be.had.as.father.by.all-2S.

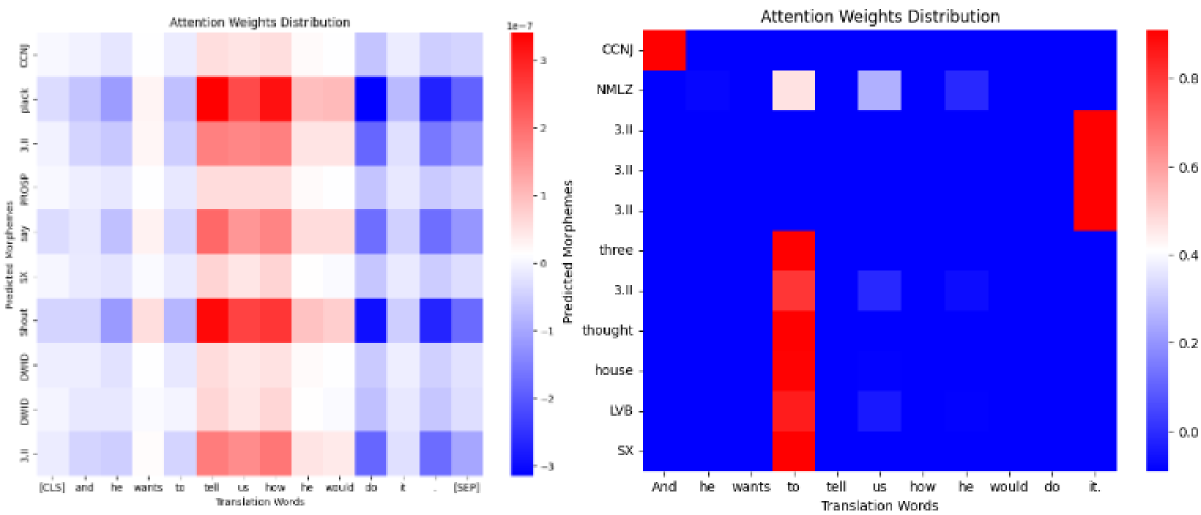

Figure 8: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for a Gitksan example (attention weights are derived from the model BERT+attn+chr (left) and the model LSTM+attm (right)). The gold-standard glosses for this sentence: CCNJ want-3.II PROSP-3.I tell-T-3.II OBL-1PL.II MANR LVB-3.II.

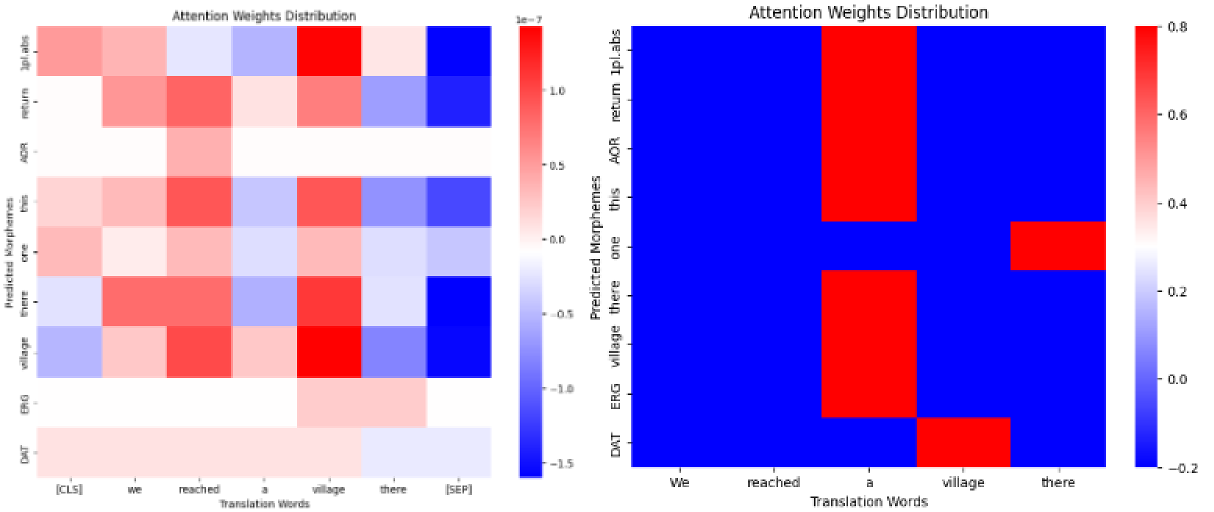

Figure 9: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for a Lezgi example (attention weights are derived from the model BERT+attn+chr (left) and the model LSTM+attm (right)). The gold-standard glosses for this sentence: 1pl.abs return-AOR this one there village-ERG-DAT.

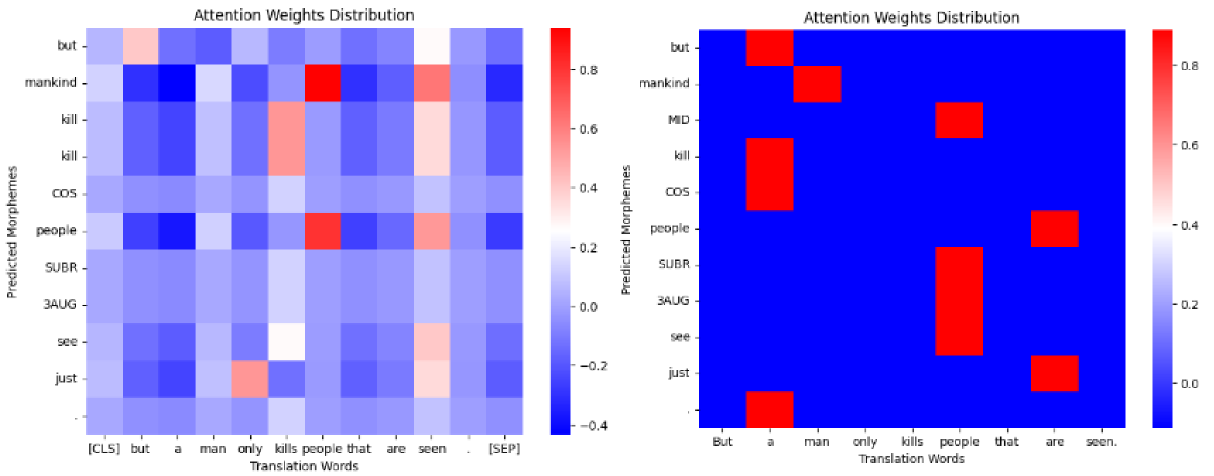

Figure 10: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for a Natügu example (attention weights are derived from the model BERT+attn+chr (left) and the model LSTM+attm (right)). The gold-standard glosses for this sentence: but mankind MID-kill-COS-3MINIS people SUBR PAS-see-INTS-just.

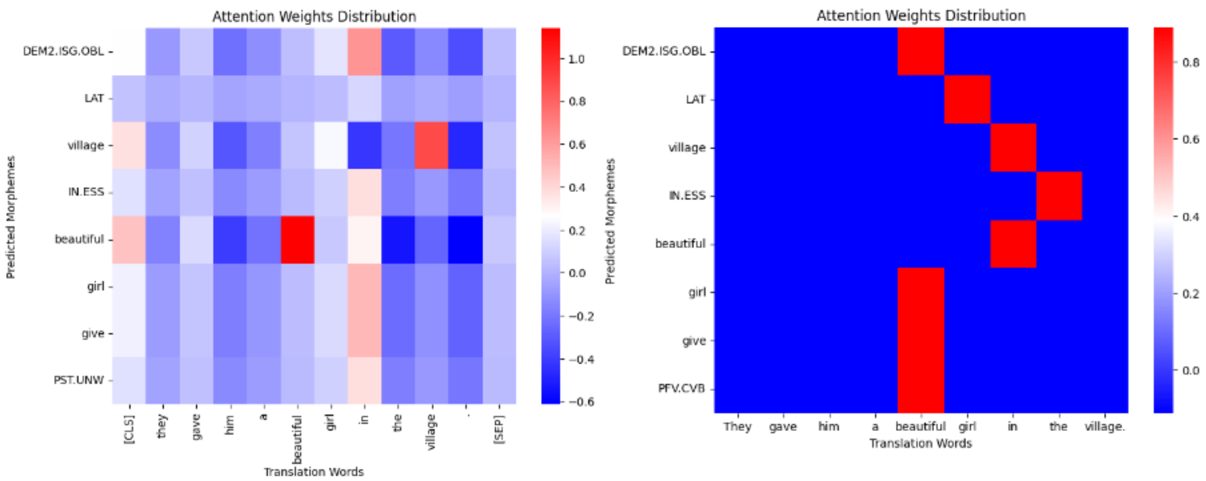

Figure 11: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for a Tsez example (attention weights are derived from the model BERT+attn+chr (left) and the model LSTM+attm (right)). The gold-standard glosses for this sentence: DEM2.ISG.OBL-LAT village-IN.ESS beautiful girl give-PST.UNW

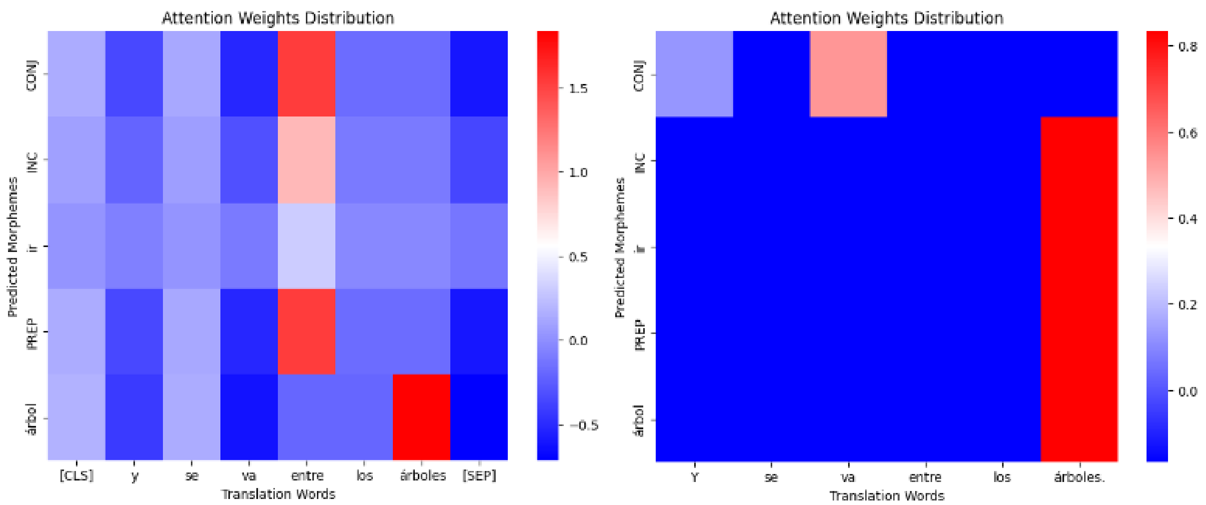

Figure 12: Difference from mean attention weights of glossed output tokens (y-axis) with respect to encoded translation tokens (x-axis) for a Uspanteko example (attention weights are derived from the model BERT+attn+chr (left) and the model LSTM+attm (right)). The gold-standard glosses for this sentence: CONJ INC-ir PREP árbol.

Model setting	arp	lez	ntu	ddo	usp	git
T5/BERT+attn+chr	83.68	81.29	81.51	**92.79**	**82.75**	12.83
+GPT4-random	84.78	85.12	83.19	90.52	70.54	26.79
+GPT4-BERT-Sim	85.13	**86.35**	83.33	91.23	73.28	27.13
+GPT4-Overlap	**86.54**	86.20	84.17	91.76	74.91	27.17
+GPT4-LCS	85.97	85.86	**84.87**	90.87	73.65	26.98
+LLaMA3-Overlap	85.23	84.05	83.88	89.54	71.43	**29.81**

Table 8: Lexical morpheme accuracy across languages in the 2023 Sigmorphon Shared Task (Ginn et al., 2023) with 'arp' representing Arapaho, 'git' for Gitksan, 'lez' for Lezgi, 'ntu' for Natügu, 'ddo' for Tsez, and 'usp' for Uspanteko. Model specifics are elaborated in Section 2.

Model setting	arp	lez	ntu	ddo	usp	git
LSTM	0.889	0.873	0.826	0.925	0.783	0.434
T5/BERT+attn+chr	0.895	0.913	0.860	**0.942**	**0.864**	0.468
T5/BERT+attn+chr+Prmpt	**0.896**	**0.922**	**0.862**	0.940	0.807	**0.526**

Table 9: BERT score of lexical morphemes of languages in the 2023 Sigmorphon Shared Task (Ginn et al., 2023), with 'arp' representing Arapaho, 'git' for Gitksan, 'lez' for Lezgi, 'ntu' for Natügu, 'ddo' for Tsez, and 'usp' for Uspanteko. Model specifics are elaborated in Section 2.

(2023), we use the LSTM-encoder classifier model as our baseline instead. The BERT score results align closely with the word-level accuracy.

I Dictionary Information

The Arapaho dictionary was accessed from https://homewitharapaho.wordpress.com/wp-content/uploads/2015/03/arapaho-dictionary1.pdf.

The Gitksan dictionary is downloaded from http://www.gitxsansimalgyax.com/dictionaries.html.

Lezgi data is unpublished and obtained through personal communication with a linguist.

Word number information of these dictionaries are in Table 10.

Language	total words(num)	new words(num)
Arapaho	2436	2155
Lezgi	2081	1299
Gitksan	2034	2019

Table 10: The table details the dictionary information for Arapaho, Lezgi, and Gitksan, including the number of total words and the number of new words compared with the training data.