Can we teach language models to gloss endangered languages?

Anonymous ACL submission

Abstract

Interlinear glossed text (IGT) is a popular format in language documentation projects, where each morpheme is labeled with a descriptive annotation. Automating the creation of interlinear glossed text can be desirable to reduce annotator effort and maintain consistency across annotated corpora. Prior research (Ginn et al., 2023; Zhao et al., 2020; Moeller and Hulden, 2018) has explored a number of statistical and neural methods for automatically producing IGT.

As large language models (LLMs) have showed promising results across multilingual tasks, even for rare, endangered languages (Zhang et al., 2024), it is natural to wonder whether they can be utilized for the task of generating IGT. We explore whether LLMs can be effective at the task of interlinear glossing with in-context learning, without any traditional training. We propose new approaches for selecting examples to provide in-context, observing that targeted selection can significantly improve performance. We find that LLM-based methods beat standard transformer baselines, despite requiring no training at all. These approaches still underperform state-of-the-art supervised systems for the task, but are highly practical for researchers outside of the NLP community, requiring minimal effort to use.

1 Introduction

With thousands of endangered languages at risk of extinction, language documentation has become a major area of linguistic research (Himmelmann, 2006; Woodbury, 1999), aiming to produce permanent artifacts such as annotated corpora, reference grammars, and dictionaries. Furthermore, research has explored the potential for computational methods to aid in language documentation and revitalization (Palmer et al., 2009; Moeller and Hulden, 2018; Wiemerslage et al., 2022; Kann et al., 2022; Gessler, 2022; Zariquiey et al., 2022; Zhang et al., 2022; Flavelle and Lachler, 2023).

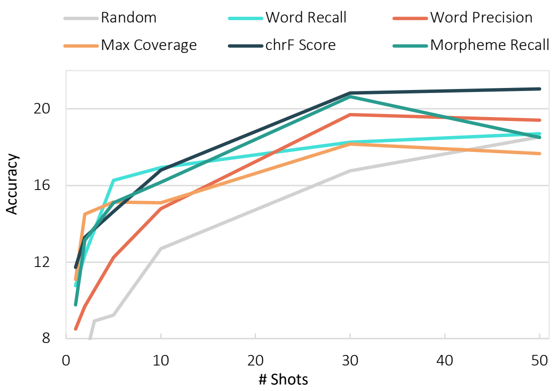


Figure 1: Accuracy of an LLM-based glossing method on Gitksan data, varying the number of provided examples and the strategy for selecting examples.

In particular, we study the task of generating interlinear glossed text (IGT), a line-by-line format for annotated text corpora that is commonly used in documentation projects. IGT generation has been studied using statistical (Palmer et al., 2009; Samardžić et al., 2015; McMillan-Major, 2020) and neural (Moeller and Hulden, 2018; Zhao et al., 2020; Barriga Martínez et al., 2021) methods.

A key challenge when working with endangered languages is that, in nearly all cases,[1] there is very little labeled or unlabeled data available. This is particularly challenging for large neural models which depend on large, representative training data sets. Research has explored methods to overcome this challenge for IGT generation systems, such as crosslingual transfer (He et al., 2023; Okabe and Yvon, 2023; Ginn et al., 2024) and architectural modifications (Girrbach, 2023a), but these approaches struggle in very low-resource scenarios. In addition, previous approaches generally require expertise in model training, implementation, and deployment, as well as the computational resources

[1] As Liu et al. (2022) notes, not all endangered languages are low-resource (and vice versa), and such languages bear different concerns when developing language technology.

needed to serve large neural models.

As large language models (LLMs) have demonstrated impressive performance on various natural language tasks, the question arises whether they can benefit language documentation. We seek to evaluate the ability of current LLMs to generate interlinear glossed text, compared with earlier state-of-the-art methods. This research can also shed light on the language-agnostic capabilities of LLMs, requiring the model to learn patterns in very rare languages which are unlikely to have significant presence in their training data.

We study strategies for selecting in-context examples, finding significant impacts to performance. Our best-performing systems outperform transformer model baselines, despite involving no training whatsoever. They still underperform SOTA systems that induce morphological segmentation, but at the same time hold promise for offering a new approach to interlinear glossing for language documentation practitioners.

2 Background

2.1 Interlinear Glosed Text

A typical example of IGT is shown in item 1.

(1)

nuhu'	tih-'eeneti-3i'	heneenei3oobei-3i'
this	when.PAST-speak-3PL	IC.tell.the.truth-3PL

"When they speak, they tell the truth." (Cowell, 2020)

The first line (transcription line) contains the text in the language being documented, and may be segmented into morphemes (as here). The second line (gloss line) provides a *gloss* for each morpheme in the transcription. Glosses may indicate grammatical function or a translation of the morpheme (for stems). The third line contains a translation into a high-resource language such as English. Producing each of these lines requires knowledge of the language and/or skilled linguistic analysis.

Generally, automated IGT systems are trained to predict the gloss line given the transcription line (and sometimes the translation as in Zhao et al., 2020; Rice et al., 2024). The primary aim of such systems is to assist a human annotator, providing suggestions for common morphemes that are often glossed with the same label. These systems are not intended to replace human annotators, who are vital to the documentation process, annotating novel morphemes and interesting linguistic phenomena, as well as verifying automatically-produced labels.

2.2 LLMs for Rare Languages

Though LLMs generally have limited understanding of rare and low-resource languages (Ebrahimi et al., 2022), they can often achieve significantly better performance through **crosslingual in-context learning** (X-ICL), where a number of examples in the target language are provided directly in the prompt to a multilingual model (Winata et al., 2021; Lin et al., 2022; Cahyawijaya et al., 2024).

We study X-ICL methods for using LLMs for the task of IGT generation, including complete IGT examples in the prompt. We hypothesize that this approach will leverage both the set of labeled training examples and the robust multilingual knowledge of the language model. In particular, we explore the effects of including an increasing number of examples in context (section 4) and using different strategies to select relevant examples (section 5).

2.3 Related Work

A number of approaches have been used for IGT generation. Palmer et al. (2009) uses a maximum entropy classifier and represents the earliest work describing benefits of using automated glossing systems. A number of papers (Samardžić et al., 2015; Moeller and Hulden, 2018; McMillan-Major, 2020) use statistical classifiers such as conditional random fields. Recent research explores neural models such as recurrent neural networks and transformers (Moeller and Hulden, 2018; Zhao et al., 2020; Barriga Martínez et al., 2021). Other approaches improve glossing performance using crosslingual transfer (He et al., 2023; Okabe and Yvon, 2023; Ginn et al., 2024), hard attention (Girrbach, 2023a), and pseudolabeling (Ginn and Palmer, 2023).

IGT data is not only useful for preservation and revitalization projects, but also for downstream tasks such as machine translation (Zhou et al., 2019), developing linguistic resources like dictionaries (Beermann et al., 2020) and UMR (Uniform Meaning Representation) graphs (Buchholz et al., 2024), studying syntax and morphology (Bender et al., 2013; Zamaraeva, 2016; Moeller et al., 2020), and dependency parsing (Georgi et al., 2012).

Given the cost and difficulty of obtaining IGT data, research has explored methods to scrape it from Latex documents (Schenner and Nordhoff, 2016; Nordhoff and Krämer, 2022) and even images (Round et al., 2020). Finally, work has attempted to standardize IGT conventions and formats, balancing consistency and expressiveness across languages (Lehmann, 1982; Hughes et al., 2003; Nordhoff, 2020; Mortensen et al., 2023).

3 Methodology

We study the IGT generation task described in Ginn et al. (2023). Given a transcription line and translation line, systems must predict the gloss line. We focus on the *closed track* setting, where the input words are not segmented into morphemes. This task is strictly more difficult than the setting where words are already segmented, as models must jointly learn segmentation and gloss prediction. As reported in Ginn et al. (2023), the SOTA on this task remains far weaker than the setting with segmented inputs, with up to a 40 point discrepency in SOTA performance.

3.1 Data

We use the IGT corpora and splits from the 2023 SIGMORPHON Shared Task (Ginn et al., 2023), allowing us to directly compare to several other systems. We use the languages described in Table 1.

	# IGT Examples		
Language	Train	Dev	Test
Gitskan [git]	74	42	31
Lezgi [lez]	705	88	87
Natugu [ntu]	791	99	99
Uspanteko [usp]	9774	232	633

Table 1: Languages and data splits, originally from Ginn et al. (2023)

We primarily focus on the lower-resource languages from the shared task, where neural methods tended to struggle due to limited training data. We use the data as formatted by Ginn et al. (2024).

3.2 Evaluation

We evaluate using the same metrics as the shared task. We primarily report *morpheme accuracy*, which measures how many morpheme glosses match between the predicted and true glosses. Any predicted glosses beyond the length of the true gloss string are ignored.

3.3 Models

We run preliminary experiments using Cohere's Command R+ model,[2] a 104B parameter instruction-tuned language model with 128K token context that is designed for multilingual tasks.

3.4 Prompting

Though the exact prompt varies from experiment to experiment, all runs use the same base prompt, included in Appendix A. In the system prompt, we define the IGT generation task and desired output format and provide additional information such as the language and list of possible glosses. In the user prompt, we provide few-shot examples (if any) and the target example to be glossed.

We run each experiment three times with temperature 0 and a different random seed, ensuring both the retrieval strategy and model API calls are reproducible. We report the average and standard deviation for performance.

4 Many-Shot Prompting

Few-shot prompting, where a model is provided with a small number of examples in the context, has proven very effective at a variety of tasks (Brown et al., 2020; Winata et al., 2021; Lin et al., 2022; Cahyawijaya et al., 2024). Furthermore, as model context lengths have continued to increase, it has become possible to provide hundreds or even thousands of examples, and performance typically continues to improve (Bertsch et al., 2024). On the other hand, increasingly long prompts bear a high cost, and strategies to retrieve relevant examples can often achieve similar performance at a fraction of the cost (see section 5).

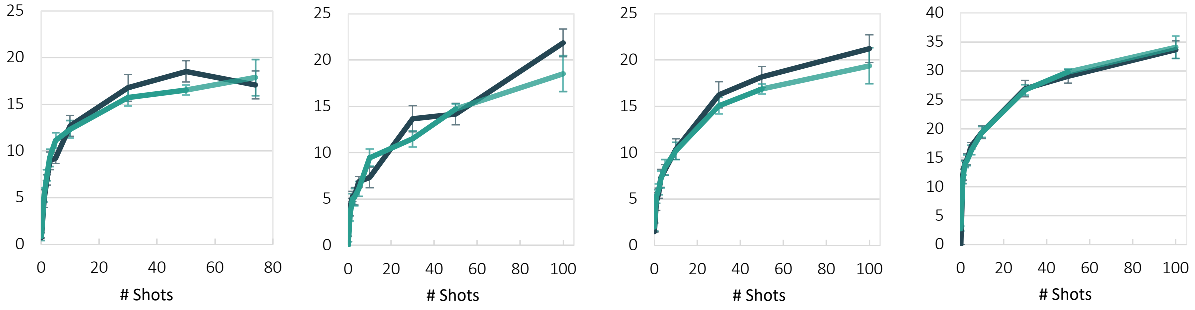
4.1 Experimental Settings

For all experiments, we run two settings, one with just the base task description, and one where we include a list of possible glosses for functional morphemes. We scrape this list of glosses from all of the seen glosses in the training set. We instruct the model to only use these glosses for functional morphemes (while stem morphemes should still be glossed with their translation). We refer to this setting as [+ GLOSSLIST], with an example gloss list in Appendix B.

For each language, we experiment with varying number of examples. For all languages except Gitksan, we run experiments providing no examples (zero-shot) and 1, 2, 3, 5, 10, 30, 50, and 100 examples. Gitksan has fewer than 100 training examples, so we use all 74 for the final setting.

For each example in our eval set, we randomly sample examples from the training set to be included in the prompt. In section 5, we compare this strategy to more intentional retrieval strategies that

[2] https://docs.cohere.com/docs/command-r-plus



(a) Gitksan (b) Lezgi (c) Natugu (d) Uspanteko

Figure 2: Morpheme accuracy of LLM-based glossing on Gitksan, Lezgi, Natugu, and Uspanteko, varying the number of provided examples. Reported values are averages over three runs; error bars indicate standard deviation.

aim to select relevant examples.

4.2 Results

We report results for our languages in Figure 2, with a full table of results provided in Appendix C. Generally, we see that the model has very weak performance in the zero-shot setting, indicating that the model has little knowledge of our chosen languages. In some cases, the zeroshot experiments produce results that are not even in the desired output format.

Performance improves drastically for the first few shots added, showing smaller improvements as the number of shots increases. For Gitksan, performance levels up as the number of provided examples approaches the full training set. For the other languages with much larger training sets, performance shows continued improvement even around 100 shots, supporting the findings of Bertsch et al. (2024). We suspect that this trend would continue to some extent, but the cost of providing hundreds of examples quickly becomes infeasible.

Relationship between Shots and Accuracy What sort of shape is formed by the curve in Figure 1 and Figure 2? The relationship appears to be roughly logarithmic, starting steep and leveling off. To quantify this relationship, we take the $\log(\#shots + 1)$ for each setting.[3] Figure 3 shows the transformed curve for Gitksan, which now shows a strong linear relationship.

We compute the R^2 value over all settings and report it in Table 2.

We observe extremely strong correlation values across all settings. This indicates that the logarithmic model is a good fit for the data, and predicts that maintaining steady performance improvements requires exponentially more examples.

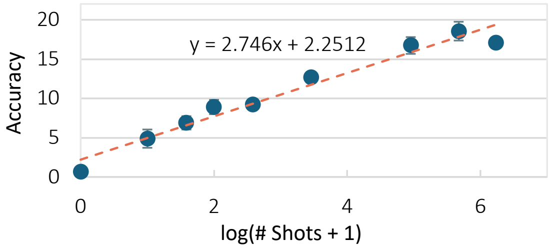


Figure 3: Morpheme accuracy for Gitksan, where the predictor variable is the logarithm of the number of provided examples (plus one).

Language	Base	+ Glosslist
Gitksan	0.962	0.958
Lezgi	0.934	0.981
Natugu	0.993	0.996
Uspanteko	0.952	0.983

Table 2: Coefficient of determination (R^2) computed between morpheme accuracy and $\log(\#shots + 1)$

.

Effect of Gloss List We initially hypothesized that providing a complete list of possible glosses in the prompt could help the model better adhere to the desired glossing conventions. We report a summary plot of the difference in accuracy between the two settings across languages in Figure 4.

The average difference is close to 0, well within a standard deviation in all cases, and thus there is little evidence to suggest that including the gloss list meaningfully affects performance. A possible explanation is that since the model has very limited prior knowledge of these languages, providing a list of glosses without any explanation or examples does not provide any useful information.

[3] Adding 1 so the zero-shot setting is defined.

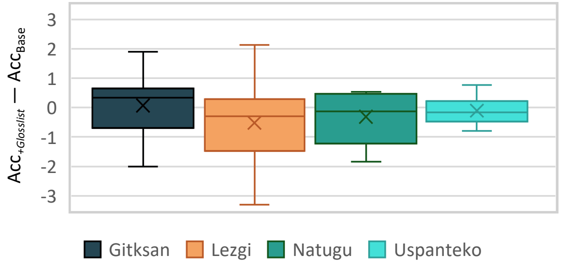


Figure 4: Difference in averaged accuracy between settings with and without a complete gloss list provided in the prompt. We observe minimal differences.

To investigate whether including a gloss list changes the predictions at all, we measure the *adherence percentage*. This metric is computed by dividing the number of predicted (functional) glosses that adhere to the gloss list by the total number of predicted glosses. We report the distribution over languages and settings in Figure 5.

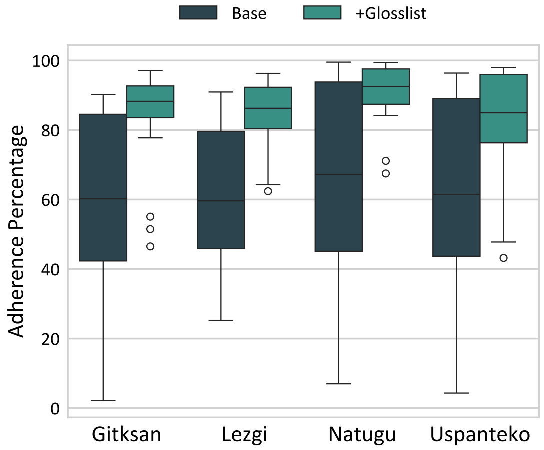


Figure 5: Distribution of adherence percentages, across languages, comparing with and without the glosslist.

We observe that including the gloss list in the prompt is effective for increasing adherence compared to the base setting. While the experiments without the gloss list vary widely, the experiments with it nearly always use glosses from the list. On the other hand, we have observed no evidence that the gloss list improves performance, suggesting that the model may be predicting glosses from the list randomly.

Furthermore, including a gloss list in the prompt carries a fixed cost of several hundred tokens for every prompt (e.g. for Uspanteko, the cost is 124 tokens). Since it provides negligible benefit, we opt to omit the glosslist for future experiments in order to reduce cost.

5 Retrieval Strategies

While including a large number of in-context examples can certainly improve performance, long prompts carry a high cost that may be infeasible for real-world documentation projects. For example, running prompts with a thousand examples in Uspanteko costs roughly 10 cents per inference call, which can quickly add up over thousands of examples. Many LLMs still have limited context length, particularly among open-source models, and including many examples may not even be possible. Finally, Bertsch et al. (2024) suggests that the effectiveness of many-shot prompting is mainly due to the model seeing relevant examples, and ignoring many irrelevant ones.

With this in mind, we consider a method inspired by **retrieval-augmented generation** (RAG, Lewis et al. 2020). RAG was originally used for knowledge-intensive tasks, using document embeddings to search for relevant documents to a given query and include them in prompt context. We apply a similar strategy in order to search for relevant IGT examples from our training corpus to include in our prompt.

5.1 Experimental Settings

We consider several strategies for selecting examples that are relevant for the target sentence.

Random As a baseline, we use the random strategy from the prior section, which simply samples n examples randomly from the training corpus.

Word Recall and Word Precision We hypothesize that a straightforward way to improve performance is by providing examples which have the same morphemes as the target sentence. Since our data is not segmented into morphemes, we instead look for matching words (which will nearly always be composed of the same morphemes). We split each example into words using whitespace, and compute the *word recall* for a target sentence T and candidate training sentence S.

$$\textsc{WordRecall} = \frac{|\text{unique}(S) \cap \text{unique}(T)|}{|\text{unique}(T)|} \tag{1}$$

This computes the fraction of unique words in the target sentence that appear in the candidate sentence. We can also compute the *word precision* with a slightly modified formula:

$$\textsc{WordPrecision} = \frac{|S \cap \text{unique}(T)|}{|S|} \quad (2)$$

This metric rewards examples where the majority of words in the candidate are in the target sentence. Notice that we do not use the unique words of S, instead weighting an example that uses the same word from T several times more heavily. We select the examples with the highest word recall or precision, considering each example independently and breaking ties randomly.

Aggregate Word Recall One limitation of the prior approach is that by considering each candidate individually, we can potentially select several redundent examples in few-shot scenarios. Instead, we can compute the *aggregate word recall* over a candidate sample of n examples.

$$S_{agg} = \bigcup_{i=1}^{n} \text{unique}(S_i) \quad (3)$$

$$\textsc{AggWordRec} = \frac{|S_{agg} \cap \text{unique}(T)|}{|\text{unique}(T)|} \quad (4)$$

This metric rewards samples that jointly cover more of the words in the target. This is equivalent to the *Maximum Coverage Problem*, and as such is NP-Hard (Nemhauser et al., 1978). We use the greedy algorithm, which runs in polynomial time (Hochbaum, 1996).

chrF A limitation of the previous strategies is that, by only considering atomic words, there is no way to select examples that may contain the same morphological units. One way we can attempt to capture morphological similarity is through using substring similarity metrics such as `chrF` (Popović, 2015) and `chrF++` (Popović, 2017). These metrics compute the F-score of character n-gram matches (`chrF++` also incorporates word n-grams), and have been shown to correspond more closely to human judgements for machine translation.

Morpheme Recall Although we do not have segmented data, much research has explored methods to induce morphological segmentations from data in an unsupervised manner. In particular, we use Morfessor (Creutz and Lagus, 2005), a popular statistical method that seeks to find a segmentation that maximizes overall the probability of segmented words.

We create silver segmentations using Morfessor and compute the recall metric as described earlier, but using morphemes rather than words. We train the segmentation model use the default parameters on the training data and use Viterbi inference to segment test examples. We use the Morfessor 2.0 library (Virpioja et al., 2013).

5.2 Results

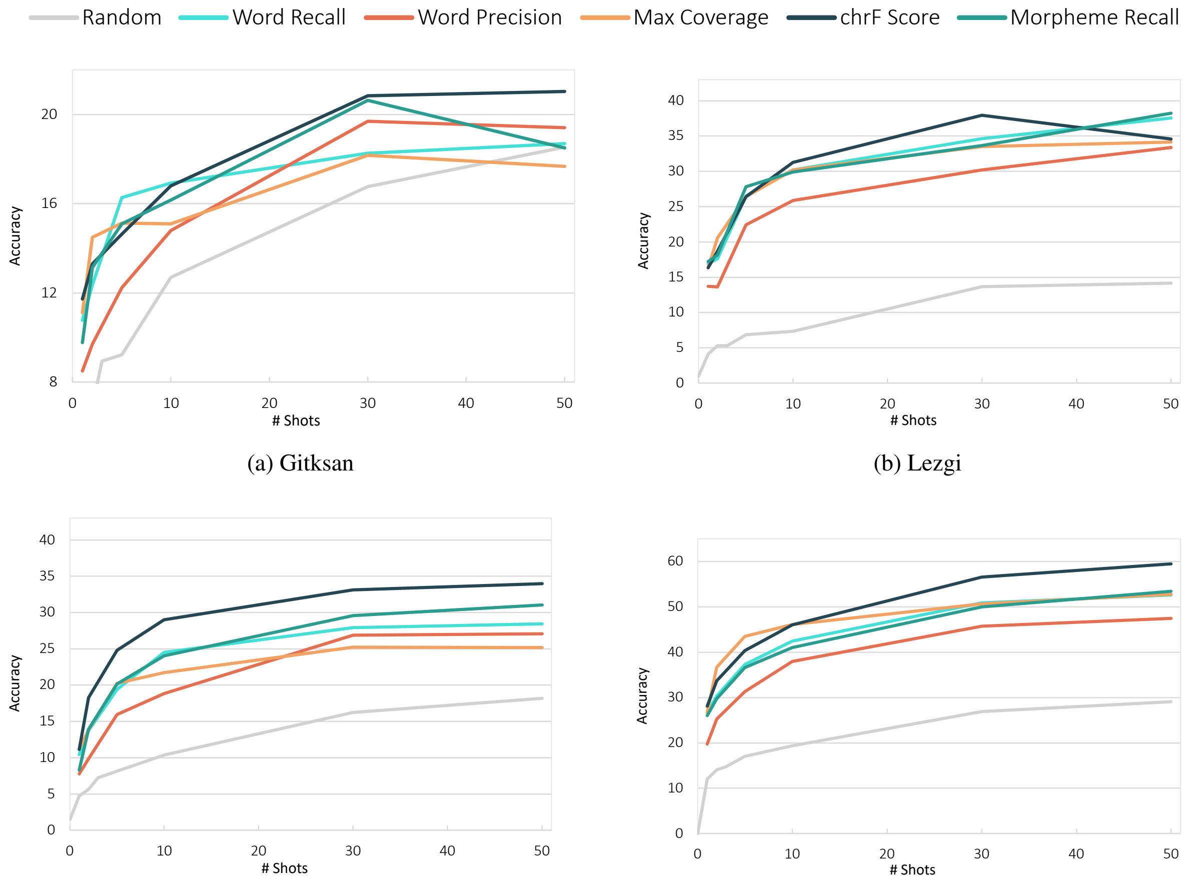
We report results across our four languages and six retrieval strategies in Figure 6. We run tests using 1, 2, 5, 10, 30, and 50 examples in each prompt.

Comparison with Random Retrieval Across all languages, we observe clear and significant improvements over the random selection method described in the prior section (here indicated with a gray line). This is the case both with a small number of fewshot examples and as the number grows large. The only exception is the 50 example setting for Gitksan, at which point the provided examples make up a large fraction of the training corpus.

This is an intuitive result, as the IGT generation task requires, at minimum, knowledge about the words of a language and their potential glosses. Even a simple baseline that glosses tokens with their most common gloss from the training set is often fairly effective (Ginn et al., 2023). This is particularly important since the LLM used seems to have very limited prior knowledge of the language, as evidenced by the poor zero-shot performance.

Relationship between Shots and Accuracy As before, we generally see consistently improving performance as additional examples are added. However, there are several cases where performance drops going from 30 to 50 shots, as in Gitksan (Word Precision, Max Coverage, and Morpheme Recall) and Lezgi (chrF Score). Both of these languages have fairly small corpora, and it is possible that after a point these strategies run out of beneficial examples, and any additional examples simply contribute noise to the prompt.

Effect of Different Granularities Many of the strategies perform very similarly, but there are some observable trends across granularity levels (word, morpheme, and substring). We observe that the chrF strategy is nearly always the most effective, outperforming the word- and morpheme-based strategies in most cases. We hypothesize that this strategy strikes a balance by selecting examples



(a) Gitksan

(b) Lezgi

(c) Natugu

(d) Uspanteko

Figure 6: Morpheme accuracy of LLM-based glossing on four languages, varying the number of provided examples and using different strategies to select relevant examples. Reported values are averages over three runs.

with subword similarity, but not introducing error due to noisy morpheme segmentations.

Word Recall vs Morpheme Recall We observe mixed results across the Word Recall and Morpheme Recall strategies. We observe a few settings where there appears to be a significant gap between the two (Gitksan at 30 shots; Lezgi at 50 shots), but generally the strategies are close. It is possible that the words in our evaluation examples often either are monomorphemic, or contain a combination of morphemes already observed in the training data, and thus selecting relevant examples according to morphemes has little benefit.

Word Recall vs Word Precision While the Word Recall and Word Precision strategies both seek to quantify the word-level similarity between the target and candidate sentences, they are computed slightly differently and produce different results. The Word Recall strategy prioritizes candidate sentences that contain a large fraction of the word *types* in the target sentence, ignoring repeated words. Meanwhile, the Word Precision strategy selects candidates based on the fraction of words within the candidate that are also in the target.

The Word Recall strategy consistently outperforms Word Precision, except for the two largest settings in Gitksan. This indicates that it is more important to provide examples which cover the words in the target than it is to provide several examples for a single word.

Word Recall vs Max Word Coverage We experimented with the Max Word Coverage setting, where we consider the recall of the selected set of candidates as whole, rather than individually. We observe minimal benefits, in fact underperforming the Word Recall setting in many cases.

6 Comparison with SOTA

Finally, we compare our best-performing strategies from the prior section with several previous baseline methods:

- The **token classification** transformer model of Ginn et al. (2023), which uses an encoder model to predict glosses word-by-word

- **Tü-CL** from Girrbach (2023b), which uses hard attention to induce latent segmentations and predict glosses on segmented words

For the LLM-based method, we select the chrF strategy and test with 30 examples for Gitksan and 100 examples for the other languages. We make some small prompt optimizations described in Appendix A, and raise the temperature to 0.2. We use the following language models:

- Cohere's **Command R+**, which was used for preliminary experiments.
- OpenAI's **GPT-4o**, specifically the `gpt-4o-2024-05-13` checkpoint (OpenAI, 2024)
- Google's **Gemini 1.5 Pro** (Gemini Team, 2024)

We run evaluation on the held out test set and report results in Figure 7.

6.1 Discussion

We observe that the LLM based glossing strategies outperform a simple transformer in all languages, despite using no training whatsoever and using a small fraction of the training set as examples. Of the LLM models, Gemini performs best on three languages. However, we note that Gemini refuses to produce answers for many examples, which we count as completely wrong. If we omit such examples, Gemini's performance is even higher, achieving 55.9%, 50.8%, and 63.9% accuracy on Lezgi, Natugu, and Uspanteko respectively.

On the other hand, the LLM methods typically underperform the SOTA method of Girrbach (2023b), except for Gitksan, where the best LLM (Gemini) outperforms by 6.5 points. The Girrbach (2023b) approach explicitly models segmentation through a learned latent representation, which our strategy does not utilize. Future work with LLM-based methods could explore an analogous process, explicitly prompting the LLM to generate segmentations before producing final glosses.

Furthermore, these methods will likely continue to improve as LLMs become more capable for rare (or even completely unseen) languages, as measured by benchmarks such as Tanzer et al. (2024). Most trivially, as LLMs with increasingly long contexts are developed, we can provide more examples in-context, which our results indicate will continue to provide benefits.

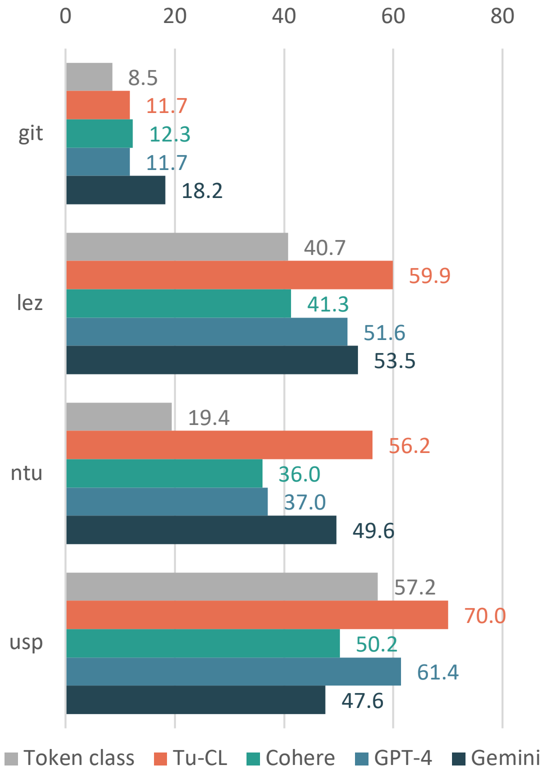


Figure 7: Morpheme accuracy results on test splits, comparing several LLMs and baseline systems.

7 Conclusion

We find that SOTA large language models struggle to produce interlinear glosses for the endangered languages used in our research. However, by selecting relevant examples from a training corpus and providing them as part of the context for each example to be glossed, we can significantly improve performance. We find that the relationship between performance and the number of few-shot examples is roughly logarithmic. Performance improves by a wide margin when we select examples with a high chrF++ score relative to the target sentence.

Our best systems outperform a standard transformer model, despite involving no explicit training and using a fraction of the training data. However, they still underperform the SOTA system for the glossing task on three out of four languages. Thus, for documentary linguists hoping to use automated glossing solutions, the use of LLMs may not achieve ideal accuracy. At the same time, LLMs may still be a preferrable choice for languages with very limited data comparable to Gitksan, and the use of an API is often far more accessible than training and hosting a neural model. Our results encourage further exploration of this approach.

Limitations

While we have selected a small set of languages that we believe give insight into the performance of automated glossing systems, they are certainly not representative of all the world's languages. In particular, LLMs may struggle more with languages that use non-Latin writing scripts (Zhang et al., 2023).

We use a single prompt template for the majority of experiments and do not conduct extensive prompt engineering. Frameworks such as DSPy (Khattab et al., 2024) have shown that prompt optimization can often greatly improve performance, so it is entirely possible that we could achieve better performance on this problem with the same models and strategies.

We evaluate three popular closed-source LLMs, but results may vary across other models. In particular, we have not yet considered open-source, local LLMs due to resource constraints.

Ethics Statement

As our work involves documentation data produced through the combined efforts of documentary linguists and speakers of endangered languages, we strive to respect their desires and avoid treating data as merely a resource to train models with (Schwartz, 2022).

We do not intend for automated glossing systems to replace human annotators, which would drastically impact the quality, novelty, and utility of annotated corpora, but rather to serve as a tool available to support documenters.

Finally, we acknowledge that the use of large language models carries a high environmental cost, and make efforts to minimize unnecessary API calls and to track our usage.

References

Diego Barriga Martínez, Victor Mijangos, and Ximena Gutierrez-Vasques. 2021. Automatic interlinear glossing for Otomi language. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 34–43, Online. Association for Computational Linguistics.

Dorothee Beermann, Lars Hellan, Pavel Mihaylov, and Anna Struck. 2020. Developing a Twi (Asante) dictionary from Akan interlinear glossed texts. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 294–297, Marseille, France. European Language Resources association.

Emily M. Bender, Michael Wayne Goodman, Joshua Crowgey, and Fei Xia. 2013. Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 74–83, Sofia, Bulgaria. Association for Computational Linguistics.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. 2024. In-context learning with long-context models: An in-depth exploration. *Preprint*, arXiv:2405.00200.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Matthew J. Buchholz, Julia Bonn, Claire Benet Post, Andrew Cowell, and Alexis Palmer. 2024. Bootstrapping UMR annotations for Arapaho from language documentation resources. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2447–2457, Torino, Italia. ELRA and ICCL.

Samuel Cahyawijaya, Holy Lovenia, and Pascale Fung. 2024. Llms are few-shot in-context low-resource language learners. *Preprint*, arXiv:2403.16512.

Andrew Cowell. 2020. The arapaho lexical and text database. Department of Linguistics, University of Colorado. Boulder, CO.

Mathias Creutz and Krista Lagus. 2005. Unsupervised morpheme segmentation and morphology induction from text corpora using morfessor 1.0. Technical Report A81, Helsinki University of Technology.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the*

60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Darren Flavelle and Jordan Lachler. 2023. Strengthening relationships between indigenous communities, documentary linguists, and computational linguists in the era of NLP-assisted language revitalization. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 25–34, Dubrovnik, Croatia. Association for Computational Linguistics.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Ryan Georgi, Fei Xia, and William Lewis. 2012. Improving dependency parsing with interlinear glossed text and syntactic projection. In *Proceedings of COLING 2012: Posters*, pages 371–380, Mumbai, India. The COLING 2012 Organizing Committee.

Luke Gessler. 2022. Closing the NLP gap: Documentary linguistics and NLP need a shared software infrastructure. In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 119–126, Dublin, Ireland. Association for Computational Linguistics.

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, Toronto, Canada. Association for Computational Linguistics.

Michael Ginn and Alexis Palmer. 2023. Robust generalization strategies for morpheme glossing in an endangered language documentation context. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 89–98, Singapore. Association for Computational Linguistics.

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

Leander Girrbach. 2023a. 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.

Leander Girrbach. 2023b. 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 171–185, Toronto, Canada. Association for Computational Linguistics.

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, Toronto, Canada. Association for Computational Linguistics.

Nikolaus P Himmelmann. 2006. Language documentation: What is it and what is it good for. *Essentials of Language Documentation*, 178(1).

Dorit S. Hochbaum. 1996. *Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems*, page 94–143. PWS Publishing Co., USA.

Baden Hughes, Steven Bird, and Catherine Bow. 2003. Encoding and presenting interlinear text using XML technologies. In *Proceedings of the Australasian Language Technology Workshop 2003*, pages 61–69, Melbourne, Australia.

Katharina Kann, Abteen Ebrahimi, Kristine Stenzel, and Alexis Palmer. 2022. Machine translation between high-resource languages in a language documentation setting. In *Proceedings of the first workshop on NLP applications to field linguistics*, pages 26–33, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Christian Lehmann. 1982. Directions for interlinear morphemic translations. *Folia Linguistica*, 16(1-4):199–224.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zoey Liu, Crystal Richardson, Richard Hatcher, and Emily Prud'hommeaux. 2022. Not always about you: Prioritizing community needs when developing endangered language technology. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3933–3944, Dublin, Ireland. Association for Computational Linguistics.

Angelina McMillan-Major. 2020. Automating gloss generation in interlinear glossed text. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 355–366, New York, New York. 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.

Sarah Moeller, Ling Liu, Changbing Yang, Katharina Kann, and Mans Hulden. 2020. IGT2P: From interlinear glossed texts to paradigms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5251–5262, Online. Association for Computational Linguistics.

David R. Mortensen, Ela Gulsen, Taiqi He, Nathaniel Robinson, Jonathan Amith, Lindia Tjuatja, and Lori Levin. 2023. Generalized glossing guidelines: An explicit, human- and machine-readable, item-and-process convention for morphological annotation. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 58–67, Toronto, Canada. Association for Computational Linguistics.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14:265–294.

Sebastian Nordhoff. 2020. Modelling and annotating interlinear glossed text from 280 different endangered languages as linked data with LIGT. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 93–104, Barcelona, Spain. Association for Computational Linguistics.

Sebastian Nordhoff and Thomas Krämer. 2022. IMT-Vault: Extracting and enriching low-resource language interlinear glossed text from grammatical descriptions and typological survey articles. In *Proceedings of the 8th Workshop on Linked Data in Linguistics within the 13th Language Resources and Evaluation Conference*, pages 17–25, Marseille, France. European Language Resources Association.

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.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Alexis Palmer, Taesun Moon, and Jason Baldridge. 2009. Evaluating automation strategies in language documentation. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 36–44, Boulder, Colorado. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

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.

Enora Rice, Ali Marashian, Luke Gessler, Alexis Palmer, and Katharina von der Wense. 2024. Tams: Translation-assisted morphological segmentation. ArXiv Preprint.

Erich Round, Mark Ellison, Jayden Macklin-Cordes, and Sacha Beniamine. 2020. Automated parsing of interlinear glossed text from page images of grammatical descriptions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2878–2883, Marseille, France. European Language Resources Association.

Tanja Samardžić, Robert Schikowski, and Sabine Stoll. 2015. Automatic interlinear glossing as two-level sequence classification. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 68–72, Beijing, China. Association for Computational Linguistics.

Mathias Schenner and Sebastian Nordhoff. 2016. Extracting interlinear glossed text from LaTeX documents. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4044–4048, Portorož, Slovenia. European Language Resources Association (ELRA).

Lane Schwartz. 2022. Primum Non Nocere: Before working with Indigenous data, the ACL must confront ongoing colonialism. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 724–731, Dublin, Ireland. Association for Computational Linguistics.

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. *Preprint*, arXiv:2309.16575.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Adam Wiemerslage, Miikka Silfverberg, Changbing Yang, Arya McCarthy, Garrett Nicolai, Eliana Colunga, and Katharina Kann. 2022. Morphological processing of low-resource languages: Where we are and what's next. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 988–1007, Dublin, Ireland. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anthony C Woodbury. 1999. Language documentation. *The Cambridge Handbook of Endangered Languages*, pages 159–186.

Olga Zamaraeva. 2016. Inferring morphotactics from interlinear glossed text: Combining clustering and precision grammars. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 141–150, Berlin, Germany. Association for Computational Linguistics.

Roberto Zariquiey, Arturo Oncevay, and Javier Vera. 2022. CLD^2 language documentation meets natural language processing for revitalising endangered languages. In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 20–30, Dublin, Ireland. Association for Computational Linguistics.

Kexun Zhang, Yee Man Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages with in-context linguistic descriptions. *arXiv preprint arXiv:2402.18025*.

Shiyue Zhang, Ben Frey, and Mohit Bansal. 2022. How can NLP help revitalize endangered languages? a case study and roadmap for the Cherokee language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1529–1541, Dublin, Ireland. Association for Computational Linguistics.

Wenxuan Zhang, Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 5484–5505. Curran Associates, Inc.

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.

Zhong Zhou, Lori Levin, David R Mortensen, and Alex Waibel. 2019. Using interlinear glosses as pivot in low-resource multilingual machine translation. *arXiv preprint arXiv:1911.02709*.

A Prompt Format

We use the following prompts for our preliminary experiments. The blue placeholders are replaced with the appropriate values. The system prompt is as follows.

```
You are an expert documentary linguist,
    specializing in $language. You are
    working on a documentation project
    for $language text, where you are
    creating annotated text corpora
    using the interlinear glossed text (
    IGT) and following the Leipzig
    glossing conventions.

Specifically, you will be provided with
    a line of text in $language as well
    as a translation of the text into
    $metalang, in the following format.

Transcription: some text in $language
Translation: translation of the
    transcription line in $metalang

You are to output the gloss line of IGT.
     You should gloss stem/lexical
    morphemes with their translation in
    $metalang, and gloss gram/functional
     morphemes with a label indicating
    their function. Please output the
    gloss line in the following format:

Glosses: the gloss line for the
    transcribed text

Glosses should use all caps lettering
    for functional morphemes and
    standard lettering for stem
    translations. Glosses for morphemes
    in a word should be separated by
    dashes, and words should be
    separated by spaces.
```

The main prompt is as follows:

```
Here are some complete glossed examples:
$fewshot_examples

Please gloss the following example in
    $metalang.

Transcription: $transcription
Translation: $translation
```

For zero-shot prompts, we remove the first sentence of the main prompt. Furthermore, from qualitative analysis, we observe that the LLM sometimes pulls words from the translation to use as glosses, resulting in incorrect examples. Thus, for the final test, we omit the translation lines from both prompts.

B Example Gloss List

We provide an example list of glosses for Gitksan. There are some formatting artificats, due to the automatic extraction of glosses.

```
#(PROSP), (#COMP), (#PROSP), 1.I, 1.SG
    .=, 1PL.II, 1SG, 1SG.II, 2SG, 3.I, 3.
    II, 3.III, 3PL, 3PL.II, 3PL.INDP, 3
    SG.II, ANTIP, AX, CAUS1, CAUS2, CCNJ
    , CN, CNTR, COMP, CONNN, DEM.PROX,
    DES, DISTR, DM, DWID, EPIS, FOC, FUT
    , FUT=3, IBM, INCEP, INS, IPFV, IPFV
    =EPIS=CN, IRR, IRR=3, LOC, LOC=CN,
    LVB, MANR, NEG, NEG=FOC, NEG=FOC=3,
    NMLZ, OBL, PART, PASS, PCNJ, PN, PR.
    EVID, PREP, PREP=CN, PROG=CN, PROG[=
    CN], PROSP, PROSP=3, PROSP=3.I, REAS
    , SELF, SG, SPT, SX, T, T=PN, TR, TR
    =CN, TR=PN, VAL, VER, VERUM, [#(
    PROSP), [(#COMP), [(PROSP), [PROG=CN
    , [PROSP
```

C Full Results

We present full results across all of our experimental settings in Table 3.

	# In-Context Examples								
Strategy	0	1	2	3	5	10	30	50	100
GITKSAN									
Random	0.7±0.0	4.9±1.2	6.9±0.9	8.9±0.9	9.2±0.4	12.7±0.7	16.8±1.1	18.5±1.2	17.1±0.3
Rand +GLOSSLIST	0.8±0.2	5.3±0.9	7.4±1.0	9.3±1.1	11.1±1.8	12.3±0.8	15.7±1.3	16.5±1.7	17.9±0.7
Word Rec.		10.8±0.3	12.4±0.9		16.3±4.6	16.9±1.1	18.3±1.4	18.7±1.3	
Word Prec.		8.5±0.3	9.7±0.5		12.2±0.6	14.8±1.1	19.7±0.3	19.4±0.5	
Max Word Cov.		11.1±1.9	14.5±1.7		15.1±0.8	15.1±0.5	18.2±1.5	17.7±0.3	
chrF		11.7±0.4	13.3±0.3		14.6±1.1	16.8±0.8	20.8±0.4	21.0±0.6	
Morph. Rec.		9.8±0.2	13.1±0.5		15.1±0.7	16.2±1.2	20.6±2.2	18.5±0.7	
LEZGI									
Random	1.0±0.2	4.1±0.6	5.3±0.6	5.3±0.8	6.9±1.6	7.3±0.6	13.7±1.2	14.2±1.4	21.8±6.0
Rand +GLOSSLIST	1.0±0.1	3.4±0.1	5.0±0.7	5.2±1.0	6.1±0.7	9.5±0.7	11.5±1.6	14.7±3.8	18.5±0.1
Word Rec.		17.0±0.7	17.6±2.8		26.5±1.5	30.2±2.1	34.6±1.6	37.6±1.5	
Word Prec.		13.7±1.3	13.6±0.8		22.4±1.6	25.9±1.4	30.2±1.7	33.4±1.9	
Max Word Cov.		16.3±0.4	20.6±2.6		26.4±0.9	30.2±1.3	33.5±1.2	34.1±1.4	
chrF		16.4±1.6	18.7±0.5		26.4±0.8	31.3±0.7	37.9±0.4	34.6±1.1	
Morph. Rec.		17.2±0.9	18.1±0.5		27.8±0.1	29.9±3.4	33.6±1.3	38.2±1.9	
NATUGU									
Random	1.5±0.3	4.7±0.4	5.6±0.3	7.2±0.7	8.1±0.7	10.4±0.3	16.2±1.3	18.2±1.4	21.2±0.3
Rand +GLOSSLIST	2.0±0.2	5.3±0.4	6.1±0.4	7.1±1.0	8.4±0.3	10.2±0.7	15.1±1.4	16.9±1.0	19.4±0.6
Word Rec.		10.4±0.4	13.7±0.6		19.4±1.0	24.5±1.8	27.9±1.6	28.4±2.1	
Word Prec.		7.8±0.2	9.9±0.5		16.0±0.2	18.8±1.5	26.9±0.8	27.0±1.0	
Max Word Cov.		11.2±0.3	13.8±0.3		20.2±0.3	21.7±1.0	25.2±2.2	25.2±1.0	
chrF		11.1±0.4	18.2±0.7		24.8±0.5	29.0±1.4	33.1±0.9	34.0±0.5	
Morph. Rec.		8.3±0.5	13.9±0.3		20.2±2.0	24.0±1.9	29.6±1.9	31.0±1.4	
USPANTEKO									
Random	2.7±0.3	12.1±0.9	14.1±0.6	14.7±1.0	17.1±0.6	19.4±1.1	26.9±1.4	29.1±1.2	33.7±1.5
Rand +GLOSSLIST	2.8±0.4	11.3±0.8	13.9±0.6	14.6±0.9	16.3±0.8	19.4±0.9	26.7±0.9	29.8±0.5	34.1±1.9
Word Rec.		26.7±1.4	30.4±1.6		37.3±1.3	42.4±0.8	50.9±0.2	52.6±0.7	
Word Prec.		19.7±0.2	25.3±0.4		31.3±1.0	37.9±0.6	45.7±0.4	47.5±0.8	
Max Word Cov.		26.7±1.2	36.7±1.0		43.5±1.7	46.1±1.0	50.7±2.2	52.8±2.0	
chrF		28.1±0.7	33.7±0.7		40.4±0.1	46.0±0.2	56.5±0.7	59.5±0.7	
Morph. Rec.		26.1±0.7	29.8±0.8		36.6±0.1	41.0±1.3	50.0±0.4	53.4±0.3	

Table 3: Full morpheme accuracy results across languages, selection strategies, and number of examples.