On the Robustness of Neural Models for Full Sentence Transformation

Michael Ginn, Ali Marashian, Bhargav Shandilya, Claire Benét Post, Enora Rice, Juan Vásquez, Marie McGregor, Matt Buchholz, Mans Hulden, Alexis Palmer
University of Colorado, Depts. of Linguistics and Computer Science
first.last@colorado.edu,
{benet.post,juan-vasquez-1}@colorado.edu

Abstract

This paper describes the LECS LAB submission to the AmericasNLP 2024 Shared Task on the Creation of Educational Materials for Indigenous Languages (Chiruzzo et al., 2024). The task requires transforming a base sentence with regards to one or more linguistic properties (such as negation or tense). We observe that this task shares many similarities with the well-studied task of word-level morphological inflection, and we explore whether the findings from inflection research are applicable to this task. In particular, we experiment with a number of augmentation strategies, finding that they can significantly benefit performance, but that not all augmented data is necessarily beneficial. Furthermore, we find that our character-level neural models show high variability with regards to performance on unseen data, and may not be the best choice when training data is limited.

1 Introduction

Morphological inflection is an NLP task with a rich history of rule-based, statistical, and neural methods (Clark 2002; Durrett and DeNero 2013; Nicolai et al. 2015; Cotterell et al. 2016; Faruqui et al. 2016; Wu et al. 2021; inter alia). Typically, systems must predict an inflected form of a word (such as *"cats"*) given a lemma form (*"cat"*) and an inflectional change (plural).

In the AmericasNLP 2024 Shared Task on Creation of Educational Materials for Indigenous Languages (Chiruzzo et al., 2024), systems must convert a base sentence into a target sentence by changing one or more linguistic properties (example in Table 1). Generally, this transformation involves inserting or deleting helper words, modifying the inflection of words in the source sentence, or both, and we observe many similarities (and some differences) between this task and the morphological inflection task.

Source	Ko'one'ex ich kool
Change	PERSON:1_PL
Target	Ko'ox ich kool

Table 1: Example from the Yukatek Maya training data.

In approaching this task, we apply lessons from research on inflection models. The shared task poses particular difficulties due to the limited amount of available training data. To alleviate this issue, we utilize sequence-to-sequence (seq2seq) neural models and explore various techniques, focusing in particular on exploring various **data augmentation** strategies. We present results for all three task languages: Bribri, Yukatek Maya,[1] and Guaraní. Our code is available on GitHub.[2]

2 Background

In 2021, the first edition of the workshop (and shared task) on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP) was proposed. For this edition, the task of machine translation was presented to the participants. The goal of this shared task was to learn machine translation models for ten indigenous languages. The participants were given ten sets of language pairs: Quechua–Spanish, Wixarika–Spanish, Shipibo-Konibo–Spanish, Asháninka–Spanish, Raramuri–Spanish, Nahuatl–Spanish, Otomí– Spanish, Aymara–Spanish, Guarani–Spanish, and Bribri–Spanish (Mager et al., 2021). For the 2022 edition, the participants were asked to present novel speech-to-text translation systems for Bribri–Spanish, Guaraní–Spanish, Kotiria–Portuguese, Wa'ikhana–Portuguese, and Quechua–Spanish (Ebrahimi et al., 2022). Finally, in 2023, the task was machine translation for the

[1]This language is referred to by task organizers (and many speakers) simply as 'Maya' - we also use this shorter form.
[2]https://github.com/lecs-lab/americasnlp2024

Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP), pages 159–173
June 21, 2024 ©2024 Association for Computational Linguistics

ten pairs mentioned above, plus a new language pair, Chatino-Spanish (Ebrahimi et al., 2023).

3 Related Work

Many of our strategies are inspired by research in morphological inflection. Morphological (re)inflection is the task of predicting an inflected form given a lemma or wordform and one or more target morphological features, and has been studied extensively through several shared tasks (Cotterell et al., 2016, 2017, 2018; Vylomova et al., 2020; Pimentel et al., 2021; Kodner et al., 2022; Goldman et al., 2023).

Morphological inflection has been studied with neural models such as RNNs (Kann and Schütze, 2016), convolutional neural networks (Östling, 2016), variational autoencoders (Zhou and Neubig, 2017), and transformers (Wu et al., 2021).

Data augmentation has been proposed as a strategy to address the challenges of training neural models on inflection tasks, particularly with limited data. Approaches have included creating artificial examples that copy the inputs directly to the outputs (Kann and Schütze, 2017; Bergmanis et al., 2017; Liu and Hulden, 2022; Yang et al., 2022), creating synthetic examples using morphological analyzers (Nicolai et al., 2017), and editing substrings using various methods to identify candidate stems (Silfverberg et al., 2017; Anastasopoulos and Neubig, 2019).

4 Models

We explore a number of models, including small sequence-to-sequence models, pretrained multilingual models, and large language models. For most models, input for a given instance consists of the source sentence plus the expected set of linguistic changes (e.g. PERSON:1_PL in Table 1).

4.1 Character-level neural models

We compare several different small character-level sequence-to-sequence models, using the `Yoyodyne` library for implementation.[3]

LSTM. We use a standard encoder-decoder LSTM with cross-attention. LSTMs have proven effective at inflection tasks (Cotterell et al., 2018), outperforming transformers under certain conditions (Wu et al., 2021). The expected linguistic changes are concatenated with the source sentence.

Transformer. Wu et al. (2021) also finds that in many cases, the transformer can outperform recurrent networks at character-level tasks. Thus, we also compare with an encoder-decoder transformer. Linguistic changes are treated as in the LSTM.

Pointer-generator. For tasks such as summarization (and the current task!) where the output sequences may share many tokens with the input sequence, the pointer-generator mechanism (See et al., 2017) has proven effective. The mechanism is a modification of an encoder-decoder architecture that introduces a pointing mechanism, where the model can copy a token from the input sequence rather than generating a novel token. Unlike the prior models, linguistic changes are encoded and attended to separately, so that they cannot be "pointed to" by the pointer-generator mechanism. We explore both LSTMs and Transformers with pointer-generator mechanisms.

We performed a hyperparameter search to determine the optimal hyperparameters for both the attentive-LSTM and pointer generator. The results of our search, our final hyperparameters, are given in Table 2. The full hyperparameter space we explored is reported in Appendix A. We train all models on a NVIDIA A100 GPU, with Adam optimization, a linear scheduler, a learning rate of 0.001, and a dropout of 0.2. We also explored using a larger architecture with the parameters described in Yang et al. (2022), however, we find these models nearly always underperform by a wide margin.[4]

4.2 Pretrained multilingual models

Transfer learning is a common strategy used to overcome limited data in lower-resource languages. To this end, we utilize mBART (Liu et al., 2020), which has shown a promising capability of generalization in the case of unseen languages (Liu et al., 2021). The desired linguistic change is appended to the source sentence, separated by the model separation token.

4.3 Large language models

Large language models (LLMs) generally struggle on rare, low-resource languages that are not well-represented in their training corpora (Robinson et al., 2023; Ahuja et al., 2023). However,

[3]https://github.com/CUNY-CL/yoyodyne

[4]Results are given in Appendix B. We observe that the larger models tend to overfit the training data, with much higher validation loss than their smaller counterparts.

Model	Language	Hyperparameters					
		Batch Size	**Embedding Size**	**Hid. Size**	**Attn Heads**	**Enc. Layers**	**Dec. Layers**
LSTM	Bribri	32	512	448	1	1	1
	Maya	32	256	896	1	2	1
	Guaraní	16	256	1152	1	1	1
PG	Bribri	32	256	1280	2	1	1
	Maya	64	448	1728	1	1	1
	Guaraní	16	192	1152	1	1	1

Table 2: Hyperparameters for LSTM and Pointer Generator models for three languages

LLMs may be able to achieve better performance on these languages through **in-context learning** (also known as *few-shot prompting*), where a small number of examples for a novel task are provided in the prompt at inference time (Brown et al., 2020). With ever-increasing context lengths, LLMs have even been able to learn completely novel languages using comprehensive linguistic resources provided in the context (Tanzer et al., 2024).

We utilize the ChatGPT API and the GPT-4 model to study in-context learning for our sentence transformation task (OpenAI et al., 2024). Since the provided training splits are very small, we provide the entire training set as context in our prompts. We also experiment with attempting to provide a more focused, relevant context, by filtering training examples to only those that have a linguistic change in common with the test sentences.

We utilize the `gpt-4-0125-preview` model, with temperature of 0 and a fixed random seed of 430. Full details about our prompting strategy are provided in Appendix C. As making an API call for every unique test example is fairly expensive, we prompt the model to make predictions on chunks consisting of multiple examples. We experiment with chunks of 20 and 80 examples.

Split	# examples Bribri	Guaraní	Maya
Train	309	178	594
Dev	212	79	149
SENTENCE COPYING	331	226	749
TRANSITIVE TRANSFORM.	3392	195	1671
STEM PERTURB.	200	200	200
CONCATENATION	500	500	500
EMBEDDINGS	300	250	-

Table 3: The number of examples in the train and dev split (top) and the number of artificial examples created by each augmentation strategy (bottom).

5 Data Augmentation

In very low-resource settings, data augmentation can be highly effective at improving output quality and performance. We employ a number of strategies for augmentation. Table 3 summarizes the training splits and number of artificial examples created by each strategy. Examples of each augmentation strategy appear in Table 10 (Appendix D).

Sentence copying (COPY). A major challenge in this task is that the sentences in the evaluation set include lemmas and words which are not present in the training set. To address this, we use a variation of the *lemma copying* technique described in Liu and Hulden (2022); Yang et al. (2022), which we designate *sentence copying*.

In this technique, we create additional training examples where the source and target sentence are identical and the *Change* field is blank. We create examples for every source sentence and target sentence in the training set (COPY$_{tr}$). We also experiment with creating examples for every source sentence in the dataset being used for evaluation, and add these to the former to create COPY$_{all}$. This technique, a form of domain adaptation, provides the model with a bias towards copying and aids the decoder in producing coherent sentences in the language. COPY$_{all}$ was not an allowable strategy for our final shared task submission, but we include the results here for comparison.

External sentence copying (COPY$_{ext}$). As external resources are valid for the shared task, we can extend the coverage provided by the sentence copying technique by using data from outside the provided datasets, similar to the approach used in Kann and Schütze (2017). We find existing unlabeled text corpora in the languages and create additional COPY rows for every sentence.

For Maya, we extract transcriptions from the

ELAN[5] (Sloetjes and Wittenburg, 2008) data in the Yucatec Maya DoReCo dataset (Skopeteas, 2022). We discard non-utterance transcriptions (such as pauses) but keep the same segmentation as the original transcription (which may not be grammatically complete sentences). For Bribri, we leverage the dataset provided by the AmericasNLP 2024 Shared Task 1;[6] we also use the provided orthographic conversion tool.[7] Finally, for Guaraní, we use a portion of the CC-100 corpus (Conneau et al., 2020).

All datasets were sanity-checked to ensure they used orthographies comparable to the training data for a given language, but no comprehensive analysis was performed for orthographic alignment. We also filter the datasets by excluding utterances which are significantly longer than those in the shared task training or dev sets.[8]

Transitive transformations (TRANS). In the standard inflection task, inputs are lemmas and outputs are inflected word forms. In this task, however, the inputs are grammatical sentences (as there is no clear equivalent for a lemma form of a sentence) and have non-null linguistic features already.

For example, there are instances in the datasets which transform a sentence to carry second person inflection. Presumably, the source sentence in these instances is either first or third person; the linguistic features of the source sentences are not specified. If there is *also* an instance in the dataset where the same source sentence is transformed to carry third person inflection, then we know there is a relationship between the two target sentences (in addition to their relationships to the common source sentence).

In these cases, we can create an additional example that takes one of the target sentences as input and produces the other target sentence, using the linguistic change from the latter instance (and vice versa). We can use this strategy for any pair of examples where the source sentence is identical and the linguistic change of the latter sentence replaces all of the feature values of the former. We describe this strategy as *transitive transformations*.

Stem perturbation (PER). We follow the insights of Silfverberg et al. (2017) and Anastasopoulos and Neubig (2019), which seek to replace stems with random character sequences from the language. Different approaches have been used to identify stems: Silfverberg et al. (2017) uses the longest common substring, while Anastasopoulos and Neubig (2019) uses character alignment to select substrings that are aligned between the lemma and inflected form.

We use an alternate strategy based on edit trees. Starting with a source sentence, we randomly change one or two characters (via deletion, or via insertion of or replacement with a random character from the domain character set); if the edit trees which could be applied to the original source can be also applied to the altered sentence, the latter is considered valid and added to the pool of possible augmentations. We repeat this process ten times per original source sentence (with each altered sentence serving as the new ‘source’ sentence), then randomly sample from the pool of possible augmentations for training.

Concatenation (CON). For this strategy, we select sentence pairs that have exactly the same set of linguistic transformations. We then produce a new training example by concatenating the two source sentences to be the new source, and concatenating the two target sentences to be the new target output.

Embedding-based augmentation. A more structured approach to augmentation is to replace words with their synonyms whenever possible while keeping the sentence structure and type of transformation constant. To find synonyms in our vocabulary, we first train language-specific static embeddings over external datasets for Guaraní and Bribri. For this purpose, we simply use the data provided as part of the first shared task of AmericasNLP 2024.

Deviating from our previous character-based approach, we use byte-pair encoding to tokenize our data. We then train a word2vec model and use these vectors as subword representations. Words that are not inflected in the training data[9] are replaced with a randomly sampled word from its top 3 most similar words in the embedding space. This allows us to create duplicates of both source and target sentences with minimal, targeted alteration

[5] https://archive.mpi.nl/tla/elan

[6] https://turing.iimas.unam.mx/americasnlp/2024_st_1.html

[7] https://github.com/AmericasNLP/americasnlp2024/

[8] Arbitrarily defined, per language, as 1.5 times the max length in characters of a sentence in the training or dev set.

[9] After byte-pair encoding, we create a list of standalone tokens and use them as candidates for synonym replacement. Our BPE encoder uses underscores to denote that a token is inflected or acts as an inflection. We assume that these standalone tokens that frequently appear without underscores can be replaced with a synonym.

to the semantic and morpho-syntactic content of the data.

6 Results and Discussion

6.1 Evaluation

We report results on the evaluation split provided for the shared task. Models are evaluated with per-sentence accuracy, BLEU score (Papineni et al., 2002), and CHRF score (Popović, 2015).

6.2 Models

We compare the various architectures described in section 4 and report results in Table 4.

Character-level neural models. Our character-level models strongly outperform the baselines on Maya, are competitve on Bribri, and underperform on Guaraní. Within the character-level architectures, the LSTM models perform best in nearly all cases. For the smaller datasets (which have roughly 200-300 training examples), the standard LSTM model achieves the best performance, while on Maya ($\sim$ 600 examples) the pointer-generator LSTM outperforms. This may indicate that the pointer-generator model needs a certain amount of training data to effectively utilize the pointing mechanism and outperform a standard LSTM, and only the Maya dataset meets that threshold.

For Guaraní, all of the sequence-to-sequence models perform very poorly. Qualitative analysis of the results shows that the models struggle to repeat back valid sentences in the language at all.

Pretrained multilingual models. mBART achieves our second best performance on Maya (second to pointer-generator LSTM), and the results for Guaraní and Bribri are also competitive with those of ChatGPT models. Unlike the character-level models, mBART tokenizes the source into subwords; hinting at the possible advantages of using subwords and the information they could carry from the model being pretrained on other languages.

Large language models. The ChatGPT-based approach achieves competitive performance, providing evidence that the model is able to capture some patterns correctly through in-context learning. The approach outperforms all other models on Guaraní (the language with the least training data), demonstrating that the LLM is able to leverage its vast training knowledge as a strong prior on the task at hand, and to make robust generalizations from the available data.

We observe minimal differences based on the chunk size, except for Maya where the smaller chunk size performs significantly better. The system using smart retrieval (SR) is able to achieve close performance for Guaraní and Maya, but underperforms on Bribri; SR is potentially a viable way to reduce prompt size and thereby cost.

LLMs offer a promising approach to building NLP systems for under-resourced languages, particular when using in-context learning for rare languages, as here. However, the high cost of inference, lack of control (due to the closed-source nature of the models), and privacy concerns are major considerations for practical usage in an endangered language context.

6.3 Data augmentation

Based on the results of the previous section, we select the LSTM and pointer-generator LSTM for our experiments with various augmentation strategies. Noting that the three shared task metrics do not always align in their assessment of best-performing model, we primarily focus on chrF, as accuracy and BLEU score tend to have high variability.[10]

We present results for models trained using each of the data augmentation strategies in Figure 1. The copying strategies tend to be the strongest, followed by the stem perturbation strategy. The other strategies show mixed results, and in some cases underperform the baseline.

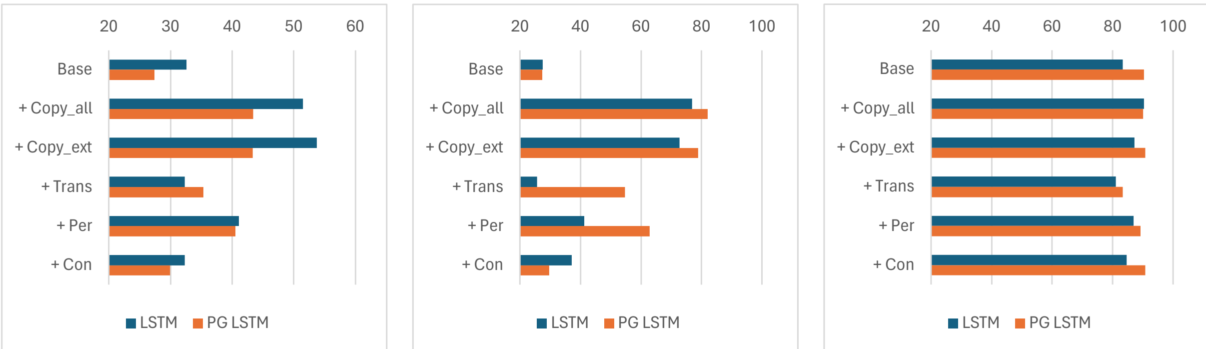
Sentence copying. We focus on a number of variations and combinations of the copy strategy and report results in Figure 2, finding that all of our strategies generally improve over the baseline. Unsurprisingly, the models trained on data including the source sentences of the evaluation set outperformed those without by an average of 14.46 chrF points. This strategy, in which the model is retrained before running inference and the target outputs are neither required nor exposed, provides clear benefits in this highly low-resource scenario.

The COPY_{ext} strategies show mixed results, sometimes matching or outperforming the COPY_{all} strategies (as in Bribri) but sometimes underperforming (as in Maya, LSTM). Combining strategies shows mixed results, and we suspect that after

[10]We observe these metrics jump wildly during training. Furthermore, having even a single incorrect output character can affect the accuracy and BLEU metrics significantly.

Architecture	Bribri Acc.	Bribri BLEU	Bribri chrF	Guaraní Acc.	Guaraní BLEU	Guaraní chrF	Maya Acc.	Maya BLEU	Maya chrF
Naive Copy	0.00	10.59	38.42	0.00	23.33	71.47	0.00	33.67	69.15
Edit Trees	5.66	20.35	45.56	22.78	34.99	77.14	26.17	52.38	78.72
LSTM	0	**19.73**	**32.57**	0	**1.95**	27.43	40.94	61.24	83.33
PG-LSTM	0	17.38	27.36	0	1.64	27.34	**51.68**	**75.51**	**90.37**
TRANSFORMER	0	13.29	29.17	0	1.27	**27.90**	16.11	42.33	70.33
PG-TRANSFORMER	0	7.9	23.09	0	0.64	22.16	10.74	36.45	64.74
MBART	5.66	40.13	60.43	**32.91**	35.12	77.62	50.34	74.12	88.70
ChatGPT									
chunksize = 20	12.26	43.43	**63.31**	**32.91**	45.63	79.21	**48.99**	74.46	**89.54**
chunksize = 80	**12.74**	**43.87**	62.39	**32.91**	**48.70**	80.32	32.89	51.36	69.84
chunksize = 1, SR	6.13	39.42	57.67	30.38	45.55	**81.80**	48.32	**74.50**	88.47

Table 4: Results for different models on development data, with no data augmentation. We **bold** the best results overall and the best results within each section. PG = pointer-generator. SR = smart retrieval.



(a) Bribri (b) Guaraní (c) Maya

Figure 1: chrF results for various data augmentation strategies.

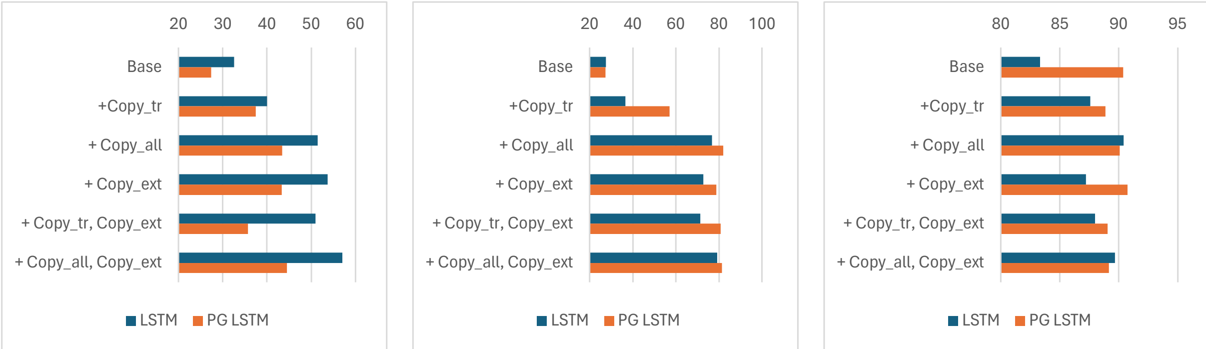
a certain number of synthetic examples, the utility of this strategy declines.

Combined strategies. Finally, we experiment with combinations of augmentation strategies, directly concatenating the synthetic datasets, with results in Figure 3. We observe mixed results—for Guaraní and Maya, none of the combined strategies show significant improvements over individual strategies, and in some cases performance degrades somewhat. We do see improvements in Bribri with the combined COPY_{all} + PER strategy and the COPY_{all} + PER + CON strategy over any of the individual strategies. Broadly, we find that synthetic data of this sort can only help up to a certain amount, and creating more synthetic data does not necessarily continue to improve performance.

7 Shared Task Submission

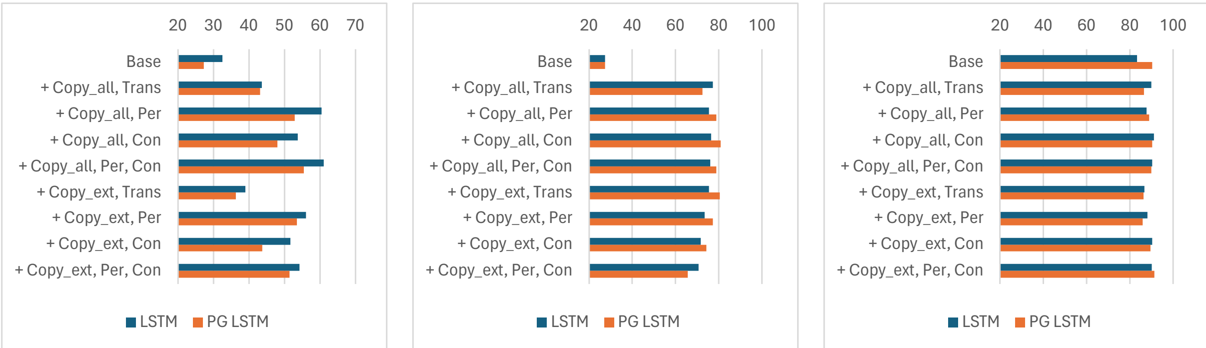
We selected a number of systems for final submission to the shared task, based on our evaluation results. We use the ChatGPT system with a chunk size of 20, the MBART system, and several of the augmented character-level neural systems. We aim to select a diverse set of augmented systems, so we select the COPY_{ext}, COPY_{tr} + COPY_{ext}, and COPY_{ext} + PER systems for the LSTM model and the COPY_{ext}, COPY_{ext}+TRANS, and COPY_{ext}+PER+CON systems for the pointer-generator model.

We train final models using the training data and specified synthetic dataset. We perform hyperparameter search and select the optimal model architecture for each language and model, which we report in appendix A. We train models for 1000 epochs, selecting the best model according to vali-



(a) Bribri (b) Guaraní (c) Maya

Figure 2: chrF results for strategies incorporating sentence copying using various sources. COPY_{tr} uses only the training data. COPY_{all} uses training data and source sentences from the evaluation data. COPY_{ext} uses sentences from external corpora.



(a) Bribri (b) Guaraní (c) Maya

Figure 3: chrF results for combinations of strategies.

dation accuracy.

We report results from the covered test set in Table 5. Disappointingly, we observe significant performance discrepancies from our dev set results, with only the ChatGPT-based system maintaining similar scores. We propose three possible factors that could have caused this.

First, all of the datasets involved are quite small, and it is possible that through random variability, the test set was meaningfully different in distribution from the evaluation set. Neural models can be vulnerable to distributional shift, particularly when training data is scarce (Linzen, 2020), which may explain why the non-neural baseline model fared better.

We briefly investigate whether this is the case by examining the types of linguistic changes in each data split. Specifically, for each desired linguistic change in the evaluation and test datasets (which might include multiple changes from a single example), we compute the number of times that change occurs in the training dataset, and average over all changes. This gives us a rough estimate of how common the linguistic changes are in the model's training data.

We report these results in Table 6. We find that for Bribri and Guaraní, the distribution is very similar between the dev and test sets, while for Maya, the test set contains changes that are far more rare (-23.6 points) on average. As Maya was the language where we observed the greatest discrepancy in performance, this could be a contributing factor, and represents an important consideration for neural models.

The other potential contributing factor is that due to the small datasets and difficult nature of the task, the performance of our models was highly variable. For augmentation strategies such as synonym replacement, the base assumption that synonyms are even present in a dataset of this size might not be accurate. During training, we often observed dev accuracy curves that swung wildly, sometimes jumping up or down by 10 points in a single epoch. Furthermore, since we performed a large number

#	Architecture	Bribri Acc.	Bribri BLEU	Bribri chrF	Guaraní Acc.	Guaraní BLEU	Guaraní chrF	Maya Acc.	Maya BLEU	Maya chrF
	Baseline (Edit Trees)	8.75	22.11	52.73	14.84	25.03	76.10	25.81	53.69	80.23
1	ChatGPT	12.08	36.95	66.75	30.77	45.18	82.33	51.61	76.82	90.29
	LSTM models									
2	+COPY$_{ext}$	3.96	16.45	47.74	7.69	17.80	70.54	19.35	57.60	78.29
3	+COPY$_{ext}$ + COPY$_{tr}$	5.00	19.77	48.26	9.34	13.15	67.20	18.71	50.21	76.19
4	+COPY$_{ext}$ + PER	4.17	16.34	51.81	8.24	15.34	66.82	16.77	59.19	79.34
	PG models									
5	+COPY$_{ext}$	0.62	13.52	34.75	7.69	20.74	71.18	30.32	60.14	79.70
6	+COPY$_{ext}$ + TRANS	0.21	7.73	31.29	11.81	17.55	69.13	15.81	43.75	71.75
7	+COPY$_{ext}$ + PER + CON	5.21	27.72	56.81	12.09	22.54	71.85	34.84	69.18	85.89
8	MBART	0.83	9.90	36.47	3.30	13.84	61.46	35.16	68.11	86.04
9	EMBEDDING AUG. + LSTM	0.83	7.91	47.76	0.55	3.80	56.21	-	-	-

Table 5: Test results for our submitted models.

Language	Dev	Test
Bribri	71.9	77.5
Guaraní	12.8	12.1
Maya	71.4	47.8

Table 6: Average frequency in the *training* data of each linguistic change observed in the dev and test set.

of experiments and selected our final models using the same evaluation set, we may have unintentionally overfit to the specific evaluation set and chosen systems that did not generalize well to the new data. In the future, this could be avoided by using many-fold cross-validation to select models rather than a single dev set.[11]

Finally, we saw significant performance benefits to including sentence copying in Figure 2, and we employed this in all of our submitted character-level systems. However, this strategy is most beneficial when it includes the sentences and lemmas that appear in the data being evaluated. It is possible that our external corpora happened to contain more overlap with the dev set examples than those in the test set, which could significantly impact performance. We suspect the strategy of retraining including the test inputs as synthetic examples could alleviate this.

Overall, these results serve as a cautionary example of the risks of selecting final systems based on limited evaluation metrics in extremely low-resource scenarios.

[11]We considered this, but it was ultimately too resource-intensive for the number of experiments we wished to run.

8 Conclusion

We describe our systems for the 2024 AmericasNLP Shared Task on the Creation of Educational Materials for Indigenous Languages, which include LLM-based systems, character-level neural networks, and finetuned multilingual models. We observe potential benefits from augmentation strategies for character-level models, particularly the *sentence copying* strategy, which helps a model adapt to new examples.

However, we find that nearly all of our systems, with the exception of the LLM system, do not generalize well to the covered test set, resulting in poor performance on the shared task. These results reaffirm the difficulty of training robust neural models in low-resource scenarios and the importance of thorough validation.

Acknowledgements

Thanks to Adam Wiemerslage for assistance with Yoyodyne. This work utilized the Blanca condo computing resource at the University of Colorado Boulder. Blanca is jointly funded by computing users and the University of Colorado Boulder.

Portions of this work were supported by the National Science Foundation under Grant No. 2149404, “CAREER: From One Language to Another”. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China. Association for Computational Linguistics.

Toms Bergmanis, Katharina Kann, Hinrich Schütze, and Sharon Goldwater. 2017. Training data augmentation for low-resource morphological inflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 31–39, Vancouver. Association for Computational Linguistics.

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.

Luis Chiruzzo, Pavel Denisov, Samuel Canul Yah, Lorena Hau Ucán, Marvin Agüero-Torales, Aldo Alvarez, Silvia Fernandez Sabido, Alejandro Molina Villegas, Abteen Ebrahimi, Robert Pugh, Arturo Oncevay, Shruti Rijhwani, Rolando Coto-Solano, Katharina von der Wense, and Manuel Mager. 2024. Findings of the AmericasNLP 2024 shared task on the creation of educational materials for indigenous languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*. Association for Computational Linguistics.

Alexander Clark. 2002. Memory-based learning of morphology with stochastic transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 513–520.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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, Online. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Shruti Rijhwani, Enora Rice, Arturo Oncevay, Claudia Baltazar, María Cortés, Cynthia Montaño, John E. Ortega, Rolando Coto-solano, Hilaria Cruz, Alexis Palmer, and Katharina Kann. 2023. Findings of the AmericasNLP 2023 shared task on machine translation into indigenous languages. In *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*, pages 206–219, Toronto, Canada. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Adam Wiemerslage, Pavel Denisov, Arturo Oncevay, Danni Liu, Sai Koneru, Enes Yavuz Ugan, Zhaolin Li, Jan Niehues, Monica Romero, Ivan G Torre, Tanel Alumäe, Jiaming Kong, Sergey Polezhaev, Yury Belousov, Wei-Rui Chen, Peter Sullivan, Ife Adebara, Bashar Talafha, Alcides Alcoba Inciarte, Muhammad Abdul-Mageed, Luis Chiruzzo, Rolando Coto-Solano, Hilaria Cruz, Sofía Flores-Solórzano, Aldo Andrés Alvarez López, Ivan Meza-Ruiz, John E. Ortega, Alexis Palmer, Rodolfo Joel Zevallos Salazar, Kristine Stenzel, Thang Vu, and Katharina Kann. 2022. Findings of the second americasnlp competition on speech-to-text translation. In *Proceedings of the NeurIPS 2022 Competitions Track*, volume 220 of *Proceedings of Machine Learning Research*, pages 217–232. PMLR.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 634–643, San Diego, California. Association for Computational Linguistics.

Omer Goldman, Khuyagbaatar Batsuren, Salam Khalifa, Aryaman Arora, Garrett Nicolai, Reut Tsarfaty, and Ekaterina Vylomova. 2023. SIGMORPHON–UniMorph 2023 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–125, Toronto, Canada. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2017. Unlabeled data for morphological generation with character-based sequence-to-sequence models. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 76–81, Copenhagen, Denmark. Association for Computational Linguistics.

Jordan Kodner, Salam Khalifa, Khuyagbaatar Batsuren, Hossep Dolatian, Ryan Cotterell, Faruk Akkus, Antonios Anastasopoulos, Taras Andrushko, Aryaman Arora, Nona Atanalov, Gábor Bella, Elena Budianskaya, Yustinus Ghanggo Ate, Omer Goldman, David Guriel, Simon Guriel, Silvia Guriel-Agiashvili, Witold Kieraś, Andrew Krizhanovsky, Natalia Krizhanovsky, Igor Marchenko, Magdalena Markowska, Polina Mashkovtseva, Maria Nepomniashchaya, Daria Rodionova, Karina Scheifer, Alexandra Sorova, Anastasia Yemelina, Jeremiah Young, and Ekaterina Vylomova. 2022. SIGMORPHON–UniMorph 2022 shared task 0: Generalization and typologically diverse morphological inflection. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 176–203, Seattle, Washington. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Ling Liu and Mans Hulden. 2022. Can a transformer pass the wug test? tuning copying bias in neural morphological inflection models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 739–749, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2021. Continual mixed-language pre-training for extremely low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2706–2718, Online. Association for Computational Linguistics.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, Ivan Vladimir Meza Ruiz, Rolando Coto-Solano, Alexis Palmer, Elisabeth Mager-Hois, Vishrav Chaudhary, Graham Neubig, Ngoc Thang Vu, and Katharina Kann. 2021. Findings of the AmericasNLP 2021 shared task on open machine translation for indigenous languages of the Americas. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 202–217, Online. Association for Computational Linguistics.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado. Association for Computational Linguistics.

Garrett Nicolai, Bradley Hauer, Mohammad Motallebi, Saeed Najafi, and Grzegorz Kondrak. 2017. If you can't beat them, join them: the University of Alberta system description. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 79–84, Vancouver. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Robert Östling. 2016. Morphological reinflection with convolutional neural networks. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 23–26, Berlin, Germany. Association for Computational Linguistics.

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

Tiago Pimentel, Maria Ryskina, Sabrina J. Mielke, Shijie Wu, Eleanor Chodroff, Brian Leonard, Garrett Nicolai, Yustinus Ghanggo Ate, Salam Khalifa, Nizar Habash, Charbel El-Khaissi, Omer Goldman, Michael Gasser, William Lane, Matt Coler, Arturo Oncevay, Jaime Rafael Montoya Samame, Gema Celeste Silva Villegas, Adam Ek, Jean-Philippe Bernardy, Andrey Shcherbakov, Aziyana Bayyr-ool, Karina Sheifer, Sofya Ganieva, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Andrew Krizhanovsky, Natalia Krizhanovsky, Clara Vania, Sardana Ivanova, Aelita Salchak, Christopher Straughn, Zoey Liu, Jonathan North Washington, Duygu Ataman, Witold Kieraś, Marcin Woliński, Totok Suhardijanto, Niklas Stoehr, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Richard J. Hatcher, Emily Prud'hommeaux, Ritesh Kumar, Mans Hulden, Botond Barta, Dorina Lakatos, Gábor Szolnok, Judit Ács, Mohit Raj, David Yarowsky, Ryan Cotterell, Ben Ambridge, and Ekaterina Vylomova. 2021. SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259, Online. 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.

Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. ChatGPT MT: Competitive for high- (but not low-) resource languages. In *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.

Stavros Skopeteas. 2022. Yucatec maya doreco dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 1.2*. Leibniz-Zentrum Allgemeine Sprachwissenschaft & laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Berlin & Lyon.

Han Sloetjes and Peter Wittenburg. 2008. Annotation by category: ELAN and ISO DCR. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

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

Ekaterina Vylomova, Jennifer White, Elizabeth Salesky, Sabrina J. Mielke, Shijie Wu, Edoardo Maria Ponti, Rowan Hall Maudslay, Ran Zmigrod, Josef Valvoda, Svetlana Toldova, Francis Tyers, Elena Klyachko, Ilya Yegorov, Natalia Krizhanovsky, Paula Czarnowska, Irene Nikkarinen, Andrew Krizhanovsky, Tiago Pimentel, Lucas Torroba Hennigen, Christo Kirov, Garrett Nicolai, Adina Williams, Antonios Anastasopoulos, Hilaria Cruz, Eleanor Chodroff, Ryan Cotterell, Miikka Silfverberg, and Mans Hulden. 2020. SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–39, Online. Association for Computational Linguistics.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2021. Applying the transformer to character-level transduction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1901–1907, Online. Association for Computational Linguistics.

Changbing Yang, Ruixin (Ray) Yang, Garrett Nicolai, and Miikka Silfverberg. 2022. Generalizing morphological inflection systems to unseen lemmas. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 226–235, Seattle, Washington. Association for Computational Linguistics.

Chunting Zhou and Graham Neubig. 2017. Morphological inflection generation with multi-space variational encoder-decoders. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 58–65, Vancouver. Association for Computational Linguistics.

A Hyperparameter Search Space

We performed a hyperparameter search for the attentive-LSTM and pointer-generator models using the **sentence copying** data augmentation strategy. We used random search with the goal of maximizing validation accuracy. We report the search space we considered in Table 7.

Hyperparameter	Distribution	Values
Batch Size	categorical	16, 32, 64
Embedding Size	q_uniform	128 to 1024; q=64
Hidden Size	q_uniform	128 to 2048; q=64
Attention Heads	values	1, 2
Encoder Layers	values	1, 2, 3
Decoder Layers	values	1, 2, 3

Table 7: Hyperparameter Search Space

B Larger Architectures

For thoroughness, we also compare architectures using the architecture size described in Yang et al. (2022). We report these results in Table 8.

Except for the transformer models, these larger models well underperform their smaller counterparts, in many cases overfitting the training data and completely failing to generalize. The transformer models perform more robustly, and seem to benefit from deeper and larger architectures.

C LLM Prompting

We attempted two different prompting strategies for our Chat-GPT implementation.

In the first strategy, we used a full-context approach, using the entire language's training split as the context. We tried these two different chunk size settings, calling the API with chunks of 20 or 80 test sentences at a time.

In the second strategy, we tried a smart-retrieval approach with a chunk size of one to only provide relevant examples as context. Relevant examples were those with the same changes as the test sentences within the language's training split.

In Table 9, an example of the prompt we provided using the smart-retrieval approach for a sentence in Bribri is shown. Note that this prompt provides just one training instance; in our experiments we provided multiple instances per prompt.

D Augmentation Examples

We provide examples of the rows created by each augmentation strategy in Table 10

Architecture	Bribri Acc.	Bribri BLEU	Bribri chrF	Guarani Acc.	Guarani BLEU	Guarani chrF	Maya Acc.	Maya BLEU	Maya chrF
LSTM	0	9.44	26.21	0	0.59	18.38	0	5.53	27.13
PG-LSTM	0	8.45	25.54	0	0.85	18.32	24.16	49.66	**76.77**
TRANSFORMER	0	**18.19**	**32.93**	0	1.42	**29.96**	**27.52**	**53.14**	74.18
PG-TRANSFORMER	0	0	0.26	0	0	0.33	0	0	1.61

Table 8: Results for different architectures, using larger model sizes of Yang et al. (2022). PG = pointer-generator.

Prompt

Below is an example of a sentence in Bribri, the linguistic change, and the target sentence after applying the change.

ID:	Bribri0303
Source:	Ye’ shka’
Change:	TYPE:NEG, TENSE:PRF_PROG
Target:	Ye’ kë̀ ku̱’bak shkö́k

Below is a similar example, where the source sentence and linguistic change are given, and the output sentence is not known. For this example, please output only the id and target sentence values, as in:

ID:	Some ID
Target:	Sentence after applying the change

Do not output any additional text, and do not output the Source or Change fields. This is very important, take your time and do not mess up or I will lose my job.

Example Input:

ID:	Bribri0367
Source:	Pûs kapë’wa̱
Change:	TYPE:NEG, TENSE:PRF_PROG
Target:	*

Model Response:

ID:	Bribri0367
Target:	Pûs kë̀ ku̱’bakapë’wa̱

Table 9: Example prompt given while LLM prompting.

Strategy		Source	Change	Target
COPY	(original)	Ko po ojupi	TENSE:FUT_SIM	Ko po ojupíta
	(augmented)	Ko po ojupi	NOCHANGE	Ko po ojupi
COPY$_{ext}$	(original)	-	-	-
	(augmented)	Nde ruvichápe	NOCHANGE	Nde ruvichápe
TRANS	(originals)	Che rasy Che rasy	PERSON:2_SI PERSON:1_PL_EXC	Nde nderasy Ore rorasy
	(augmented)	Nde nderasy	PERSON:1_PL_EXC	Ore rorasy
PER	(original)	Ha'e oguapy	PERSON:3_PL	Hikuái oguapy
	(augmented)	Ha'e ocguapy	PERSON:3_PL	Hikuái ocguapy
CON	(originals)	Nde nderejapói Apurahéi kuri	PERSON:3_PL PERSON:3_PL	Ha'ekuéra ndojapói Ha'ekuéra opurahéikuri
	(augmented)	Nde nderejapói apurahéi kuri	PERSON:3_PL	Ha'ekuéra ndojapói ha'ekuéra opurahéikuri
EMBED	(original)	Mombe'ukuéra omboty kuri pende arete	ASPECT:IPFV	Mombe'ukuéra omboty kuri hína pende arete
	(augmented)	Sombezlkuéra omboty-kuri pende arete	ASPECT:IPFV	ombeãrkuéra omboty kurir hína pende arete

Table 10: Example applications of our augmentation strategies. All examples are Guaraní.