Is linguistically-motivated data augmentation worth it?

Ray Groshan* and **Michael Ginn*** and **Alexis Palmer**
University of Colorado
Correspondence: michael.ginn@colorado.edu
∗ Equal contribution

Abstract

Data augmentation, a widely-employed technique for addressing data scarcity, involves generating synthetic data examples which are then used to augment available training data. Researchers have seen surprising success from simple methods, such as random perturbations from natural examples, where models seem to benefit even from data with nonsense words, or data that doesn't conform to the rules of the language. A second line of research produces synthetic data that does in fact follow all linguistic constraints; these methods require some linguistic expertise and are generally more challenging to implement. No previous work has done a systematic, empirical comparison of both linguistically-naive and linguistically-motivated data augmentation strategies, leaving uncertainty about whether the additional time and effort of linguistically-motivated data augmentation work in fact yields better downstream performance.

In this work, we conduct a careful and comprehensive comparison of augmentation strategies (both linguistically-naive and linguistically-motivated) for two low-resource languages with different morphological properties, Uspanteko and Arapaho. We evaluate the effectiveness of many different strategies and their combinations across two important sequence-to-sequence tasks for low-resource languages: machine translation and interlinear glossing. We find that linguistically-motivated strategies can have benefits over naive approaches, but only when the new examples they produce are not significantly unlike the training data distribution.

1 Introduction

Data augmentation refers to techniques that are used to create additional, artificial examples for training machine learning models in order to increase the total amount of training data. Data augmentation has been well-studied in computer vision, where simple perturbations such as flipping, rotating, or recoloring images are applied on natural data (Lecun et al., 1998). Similar approaches that depend on random perturbation have been used in NLP for tasks such as morphological inflection (Silfverberg et al., 2017; Bergmanis et al., 2017; Anastasopoulos and Neubig, 2019; Yang et al., 2022), classification (Wei and Zou, 2019; Karimi et al., 2021), and machine translation (Wang et al., 2018; Guo et al., 2020).

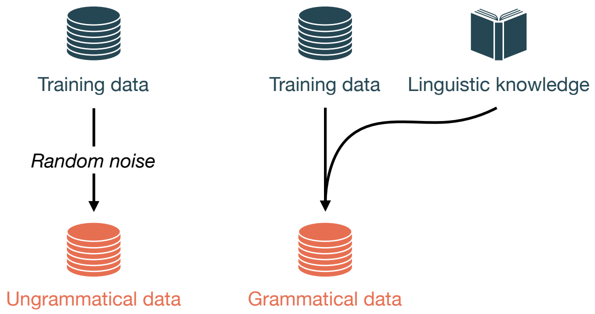


Figure 1: Two types of approach to data augmentation. **Naive augmentation (left)** uses random perturbations to produce new examples which are not necessarily grammatically valid, while **linguistically-informed augmentation (right)** uses linguistic knowledge to constrain synthetic examples to be grammatically valid.

One limitation of such approaches is that they often create new examples which are not linguistically valid. For example, a strategy which randomly inserts words might produce an ungrammatical sentence such as "The dog chases *bird* the cat," where *bird* is the inserted word. To address this issue, some researchers leverage linguistic resources to produce examples that are both novel *and* grammatical (Zhang et al., 2015; Wei and Zou, 2019; Pratapa et al., 2018; Seo et al., 2023).

Designing linguistically-motivated strategies (Figure 1) generally requires an expert with knowledge of the target language and an understanding of the principles underlying data augmentation. For many low-resource and endangered languages,

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26924–26939
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

such experts are rare, and speakers and scholars have many competing obligations. In this work, we examine whether this expert effort is worthwhile by carefully comparing linguistically-motivated strategies with strategies that can be implemented without a language expert.

We study a variety of data augmentation methods across two low-resource languages, Arapaho and Uspanteko. We evaluate on translation (to a high-resource language) in both directions and on the task of *interlinear glossing*, where the model generates a sequence of morphological glosses corresponding to the input sentence (Ginn et al., 2023). To design linguistically-motivated augmentation strategies, our first author, a trained linguist, extensively studied linguistic reference materials for both Arapaho and Uspanteko.

We compare non-linguistic augmentation strategies, such as random word insertion or deletion, with our strategies designed to generate grammatically valid examples. We find that the linguistically-motivated strategies can provide small benefits over the non-linguistic approaches in some cases. However, in cases where the linguistic strategies produce examples which are grammatically valid, but rare or unusual, performance is actually worse for the augmented models. We conclude that while the incorporation of linguistic expert knowledge may be beneficial, it must consider *both* linguistic grammaticality and the target data distribution.

Our specific contributions are the following:

- A systematic and comprehensive comparison of various linguistic and non-linguistic data augmentation strategies for low-resource machine translation and interlinear glossing.
- Analysis of the effect of combining various augmentation strategies.
- Analysis of the interaction between the size of the original training set and the benefits from data augmentation.

Our code is available on GitHub[1] and our results are available on WandB.[2]

2 Related Work

2.1 Non-linguistic Augmentation

Several methods for augmentation have been proposed that do not rely on linguistic knowledge, instead relying on shallow heuristics or statistical methods to produce novel examples, which may or may not be valid utterances in the language.

Backtranslation is a common technique in machine translation, where monolingual data in the target language is translated into the source language (Sennrich et al., 2016), though the resulting examples may not be completely valid. Wang and Yang (2015) generate novel sentences by replacing words with other words that have similar static embeddings. Likewise, Fadaee et al. (2017) seek to produce valid sentences by substituting words that produce high-probability sentences according to a language model. Andreas (2020) perform a similar procedure with sentence fragments, searching for phrases that appear in similar contexts.

Wei and Zou (2019) apply word substitutions, deletions, and insertions, performing perturbations that do not necessarily produce valid sentences. Karimi et al. (2021) use a similar approach but manipulate only punctuation marks. Many additional studies have considered similar heuristics for random perturbation (Silfverberg et al., 2017; Wang et al., 2018; Anastasopoulos and Neubig, 2019; Guo et al., 2020; Liu and Hulden, 2022).

2.2 Linguistic Augmentation

While work in the prior section replaces words or phrases according to statistical patterns, other work proposes the use of linguistic resources to identify valid replacements. This has been done with thesauri (Zhang et al., 2015), WordNet (Wei and Zou, 2019), and (for code-switched text) bilingual lexicons (Pratapa et al., 2018; Winata et al., 2019; Tarunesh et al., 2021). Instead of entire words, some research modifies the linguistic features of selected words in each sentence, such as pronominal gender (Zhao et al., 2018) or verbal inflection (Li and He, 2021). Still other research generates entirely synthetic examples by combining morphemes (Seo et al., 2023) or sampling from formal grammars such as finite-state machines (Lane and Bird, 2020) and context-free grammars (Lucas et al., 2024).

2.3 Our Contributions

Our work is novel by providing a careful comparison of similar linguistic and non-linguistic strategies. Additionally, most previous work uses shallow knowledge about the language in the form of dictionaries and thesauri, while we utilize a trained linguist and full reference grammars.

[1] https://github.com/lecs-lab/is-ling-augmentation-worth-it

[2] https://wandb.ai/augmorph

The closest prior works are Dai and Adel (2020); Kashefi and Hwa (2020), which compare linguistic and non-linguistic augmentation strategies, but their work studies classification tasks, while we experiment with sequence-to-sequence tasks. Sequence-to-sequence tasks present additional difficulties for effective data augmentation. For classification tasks, it is trivial to ensure the labels for the synthetic data adhere to the set of valid labels; however, for sequence-to-sequence datasets, the labels are unrestricted sequences, and thus it is far more difficult to guarantee their validity.

3 Datasets and Tasks

We use the datasets from Ginn et al. (2023). Each example consists of a sentence in the target language, a translation into Spanish (for Uspanteko) or English (for Arapaho), and a line of *interlinear glosses*. Interlinear glosses provide a tag for each morpheme in the original sentence, which may either be a translation (for stem morphemes) or morphological category. Below are examples for Uspanteko item 1 and Arapaho item 2.

(1) wi' neen tb'ank juntir
EXS INT INC-hacer-SC todo
"Tienen que hacer todo"

(2) Nihtooneete3eino' hini' xouu
PAST-almost-run.into-1S that.those skunk
"I almost ran into that skunk"

We use a fixed test set, and dynamically create three different evaluation sets by splitting the training set for each random seed. We report the splits in Table 1.

Language	# train	# eval	# test
Uspanteko	9096	479	1064
Arapaho	41824	2202	4892

Table 1: The number of sentences per dataset split. The test set is fixed across all runs. The eval set is dynamically created across runs by splitting the original training set, to ensure we don't overfit to a particular eval dataset.

The three tasks we study are translation from the target language to a high-resource language (*usp* → *esp*, *arp* → *eng*), translation in the opposite direction (*esp* → *usp*, *eng* → *arp*), and interlinear glossing (*usp* → *igt*, *arp* → *igt*). For the latter, the input is the sentence in the target language (first line) and the desired output is the interlinear gloss line (second line).

Name	Category	# examples
Uspanteko		
UPD-TAM	Linguistic	0.3
INS-CONJ	Linguistic	20.0
INS-NOISE	Non-linguistic	20.0
DEL-ANY	Non-linguistic	0.2
DEL-EXCL	Linguistic	0.2
DUP	Non-linguistic	0.3
Arapaho		
INS-INTJ	Linguistic	20.0
INS-NOISE	Non-linguistic	20.0
PERM	Linguistic	10.0

Table 2: An overview of the data augmentation methods used in our study. We categorize the strategy as either non-linguistic (random perturbation) or linguistic (linguistically-motivated transformations). In addition, we report the average number of new, synthetic examples created *for each original example*.

4 Emulating a Linguistic Expert

Designing linguistically-motivated augmentation strategies requires in-depth knowledge of linguistics and of the target language. We did not have access to an expert for our target languages, but we emulated this by having our first author extensively study the grammars of the Uspanteko and Arapaho languages. The author has formal training in linguistics at the graduate level, but no prior exposure to Uspanteko or Arapaho.

In order to gain a strong understanding of the grammars of these languages, the first author spent over a year and nearly 200 hours studying linguistic materials (primarily reference grammars and bilingual dictionaries) and interlinear gloss datasets. The linguistic materials included Coon (2016) and Vicente Méndez (2007) for Uspanteko and Cowell and Moss (2008) for Arapaho. By the end of this period, the first author–while not a fluent speaker of either language–was able to create fully grammatical sentences, following the reference materials.

5 Augmentation Strategies

We design both linguistic and non-linguistic augmentation strategies and describe them here (summary in Table 2).

5.1 Uspanteko

Uspanteko is an endangered language spoken in Guatemala with fewer than 6000 speakers (Bennett et al., 2022). Uspanteko is an agglutinating language with complex verbs that may include morphemes for TAM (tense-aspect-mood), person, and other suffixes (Coon, 2016).

We design six augmentation strategies for Uspanteko that include three linguistic and three non-linguistic methods. When running augmentation, we modify the original Uspanteko sentence, as well as the corresponding Spanish sentence (for translation) and interlinear glosses (for IGT generation), adding, deleting, or replacing words as needed. When necessary, we use the Spanish-Uspanteko bilingual dictionary of Vicente Méndez (2007) to translate words. Examples of each method are shown in Table 7.

1. UPD-TAM: Uspanteko obligatorily marks aspect on the verb, and completive (COM) and incompletive (INC) are high-frequency aspect markers that are easily mapped to their Spanish equivalents. This strategy updates the TAM marker to change completive verbs into incompletive, and vice versa. We skip examples that don't have a verb beginning with COM or INC. To make sure the Spanish translation matches the updated Uspanteko sentence, we use mlconjug3 (Diao, 2023) to update the Spanish verb conjugations.

2. INS-CONJ: Inserts a random conjunction or adverb at the start of the sentence (which is generally valid in Uspanteko), using twenty common conjunctions and adverbs from the Ginn et al. (2023) dataset.

3. INS-NOISE: Inserts a random word at the start of the sentence, using twenty random words[3] from the training data (which are not conjunctions or adverbs). Unlike the prior strategy, this is not guaranteed to produce a linguistically well-formed sentence, allowing us to directly compare whether linguistically-motivated insertion has any benefits over a purely random strategy.

4. DEL-ANY: Randomly deletes a word from the sentence by index, as well as the corresponding index in the translation and glosses.

5. DEL-EXCL: Randomly deletes a word from the sentence by index, excluding verbs. If the randomly selected index refers to a verb, the example is skipped and not used for data augmentation. Unlike the prior strategy, this approach seeks to avoid producing entirely ungrammatical sentences.[4]

6. DUP: Duplicates the word at a randomly chosen index.[5]

[3]We chose twenty words to match the twenty conjunctions/adverbs in the prior strategy.

5.2 Arapaho

Arapaho is an endangered language spoken in the United States, primarily in Wyoming and Oklahoma, with fewer than 300 fluent speakers (Cowell and Moss, 2008). Arapaho is a polysynthetic language with free word order and highly complex verbs (Cowell and Moss, 2008). Unlike Uspanteko, it is quite difficult to modify verbs in a way that guarantees a valid sentence, so we instead focus on sentence-level augmentation strategies. Examples of each method are shown in Table 8.

1. INS-INTJ: Inserts an interjection at the start of the sentence, using twenty common interjections, greetings, and conjunctions from the original textual data. [6]

2. INS-NOISE: Similar to the Uspanteko version, inserts a random word at the start of the sentence. The word list is composed of twenty words from the training set. The majority of these are nouns, as they were easiest to isolate and confidently identify.

3. PERM: Produces up to 10 permutations of the original word order.[7] The permuted sentences are linguistically valid, but may not be preferred by a native speaker due to pragmatic factors (Cowell and Moss, 2008).

6 Experimental Setup

For each task and language, we run experiments to evaluate the effect of data augmentation on task performance. We train models using the BYT5-SMALL pretrained model (Xue et al., 2022), a 300

[4]Both delete strategies are restricted to examples where all four lines of the gloss have the same number of whitespace-separated words, in order to reduce the likelihood of the wrong word being deleted.

[5]Restricted like the previous strategyy.

[6]There was a limited number of isolatable conjunctions in the data, so we included interjections and greetings in order to have a list of comparable size to the Uspanteko methods. This limitation prevented us from increasing the set size beyond twenty across languages and methods.

[7]We set a limit of 10 to prevent "drowning" the model with a potentially huge number of augmented samples.

million parameter encoder-decoder transformer model that operates over byte sequences.[8] The inputs are formatted with a short prompt, such as the example in Table 4.

Input	Translate into English: Henee3nee-3oonouh'ut niine'eehek nehe' hotii
Label	This car is very cheap.

Table 4: An example prompt used for training, in this case to translate from Arapaho into English. We use different prefixes for each of the tasks, though this is likely not strictly necessary.

As a baseline, we finetune models on the original training set, using the hyperparameters described in Appendix A. We also finetune models on the augmented sets created by each strategy, using the same hyperparameters.

Theoretical perspectives have claimed that the key to successful data augmentation is creating a diverse set of augmented examples that is not too similar to the original data (Feng et al., 2021). Thus, in addition to individual strategies, we finetune models on each possible combination of two or more augmentation strategies, for a total of $2^6 = 64$ experimental settings for Uspanteko and $2^3 = 8$ settings for Arapaho.

In addition, we wish to disentangle the effects of augmentation at different training sizes. We sample a subset of the training data and use the subset to produce the augmented training set. We experiment with samples of 100, 500, 1000, and 5000 examples, in addition to the full training set. For every setting, we train 3 different models with different random seeds and different subsets at that size. For each task, we train a total of $5 \times 3 \times 64 = 960$ models for Uspanteko and $5 \times 3 \times 8 = 120$ models for Arapaho.

[8]This avoids the issues that come with tokenization and rare languages, and has been shown to be beneficial on these specific datasets (He et al., 2023).

Finetuning is performed with a fixed number of training steps across all settings. We use a learning curriculum where the model is first trained on the synthetic data, followed by the original data, resetting the optimizer in between phases. This approach, used in Lucas et al. (2024), essentially treats the augmented data as pretraining data and controls for any effect that might arise with mixing the augmented data into the original dataset.

7 Results

We report the complete set of results for all 1000+ settings on our GitHub,[9] and highlight the key results here. In addition, for all of the visualizations in this section, we report the results in tabular format in Appendix C.

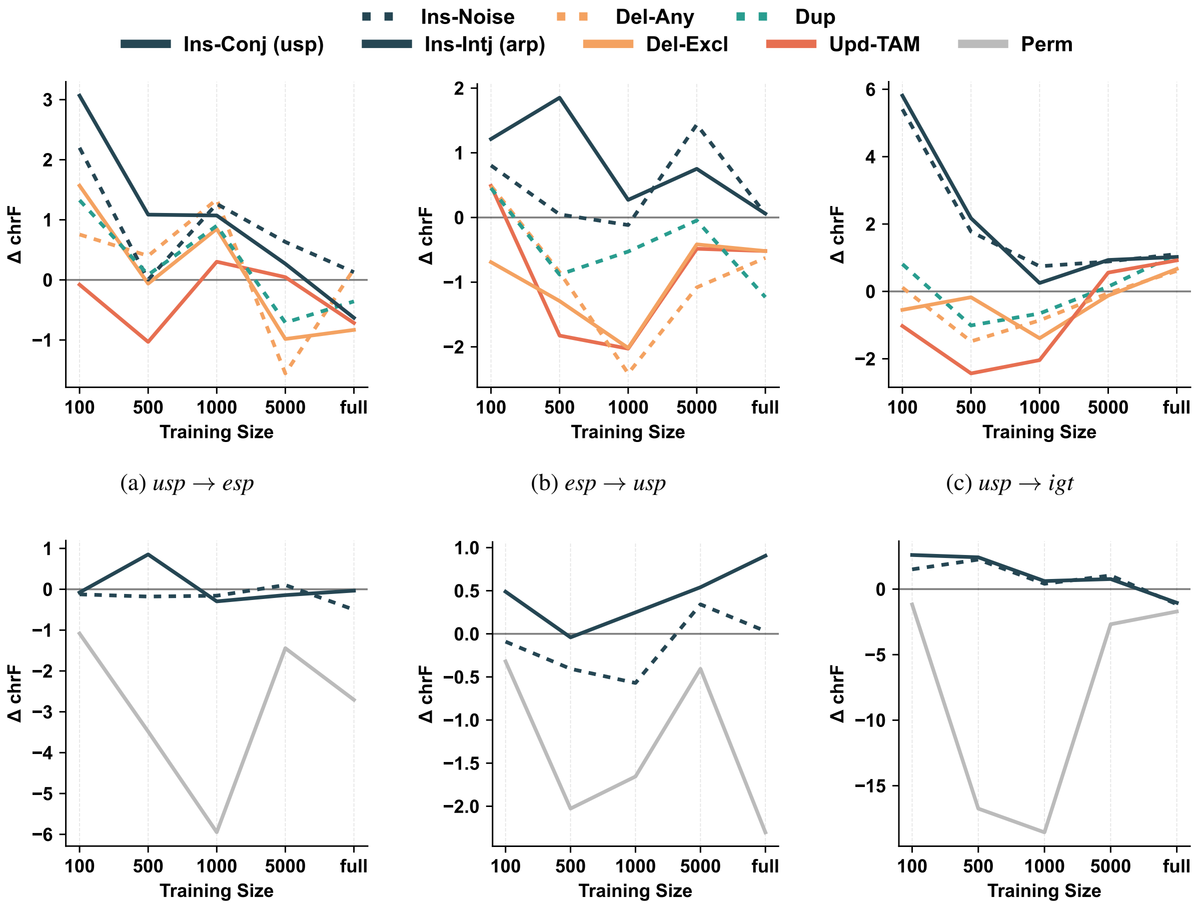
In Table 3, we report the chrF score[10] for the baseline models (no augmentation) across languages and tasks. As expected, the scores are virtually zero for the smallest training size setting, with continual improvements as the amount of training data increases. We also observe that the interlinear glossing task (*usp/arp* $\rightarrow$ *igt*) is far easier than the translation task, likely because the output sequences are restricted to gloss sequences, which is a smaller output space than translated text. We also observe that translation into the higher-resource language (Spanish or English) is easier than the reverse; one possible explanation is that the pretrained ByT5 model has already been trained to output valid text in those languages, but not in the low-resource languages.

[9]https://github.com/lecs-lab/is-ling-augmentation-worth-it

[10]We chose not to use chrF++ or BLEU as many of our examples are polysynthetic sentences with very few words, so word gram-based methods are less informative.

Task	100	500	1000	5000	full
Uspanteko					
usp $\rightarrow$ *esp*	14.6 (0.8)	26.4 (0.1)	31.7 (0.5)	44.1 (0.4)	45.2 (1.9)
esp $\rightarrow$ *usp*	13.7 (0.6)	23.1 (0.3)	29.1 (0.7)	39.6 (0.7)	40.6 (0.6)
usp $\rightarrow$ *igt*	18.4 (2.0)	53.9 (1.9)	65.2 (0.6)	74.5 (0.8)	75.4 (0.1)
Arapaho					
arp $\rightarrow$ *eng*	15.3 (0.6)	18.7 (0.2)	22.2 (0.6)	31.0 (0.4)	38.9 (0.2)
eng $\rightarrow$ *arp*	21.8 (0.7)	27.4 (0.2)	30.7 (0.9)	40.4 (0.6)	46.2 (2.3)
arp $\rightarrow$ *igt*	17.7 (1.0)	38.7 (2.0)	51.2 (0.6)	68.0 (0.3)	76.7 (0.1)

Table 3: **Baseline** chrF scores (without any augmented data) across languages, tasks, and training sizes. Reported as the mean over three runs, with the format $mean(std)$.



(a) *usp* → *esp*

(b) *esp* → *usp*

(c) *usp* → *igt*

(d) *arp* → *eng*

(e) *eng* → *arp*

(f) *arp* → *igt*

Figure 2: Difference in (test set) chrF score for various individual augmentation strategies from the baseline (black) for Uspanteko (top) and Arapaho (bottom). Dashed lines indicate non-linguistic strategies, while solid lines are used for linguistic strategies. Averaged over three runs at each point. Tabular form in Table 9.

In Figure 2, we visualize the performance impact (chrF score) of the individual augmentation strategies as an increase or decrease compared to the baseline performance. We observe that the majority of strategies actually worsen performance somewhat. The only strategies that seem to consistently improve performance are the INS-NOISE and INS-CONJ strategies (in most cases for the latter).

In Figure 3, we visualize the impact of adding each strategy to a combined augmentation strategy. We compute the mean improvement for each strategy by taking the mean difference in chrF score between combinations with and combinations without the particular strategy. For example, for the strategy INS-NOISE, we would take the mean of the following:

$$chrF_{\text{Ins-Noise}} - chrF_{\text{Baseline}}$$

$$chrF_{\text{Ins-Noise, Upd-TAM}} - chrF_{\text{Upd-TAM}}$$

$$chrF_{\text{Ins-Noise, Upd-TAM, Del}} - chrF_{\text{Upd-TAM, Del}}$$

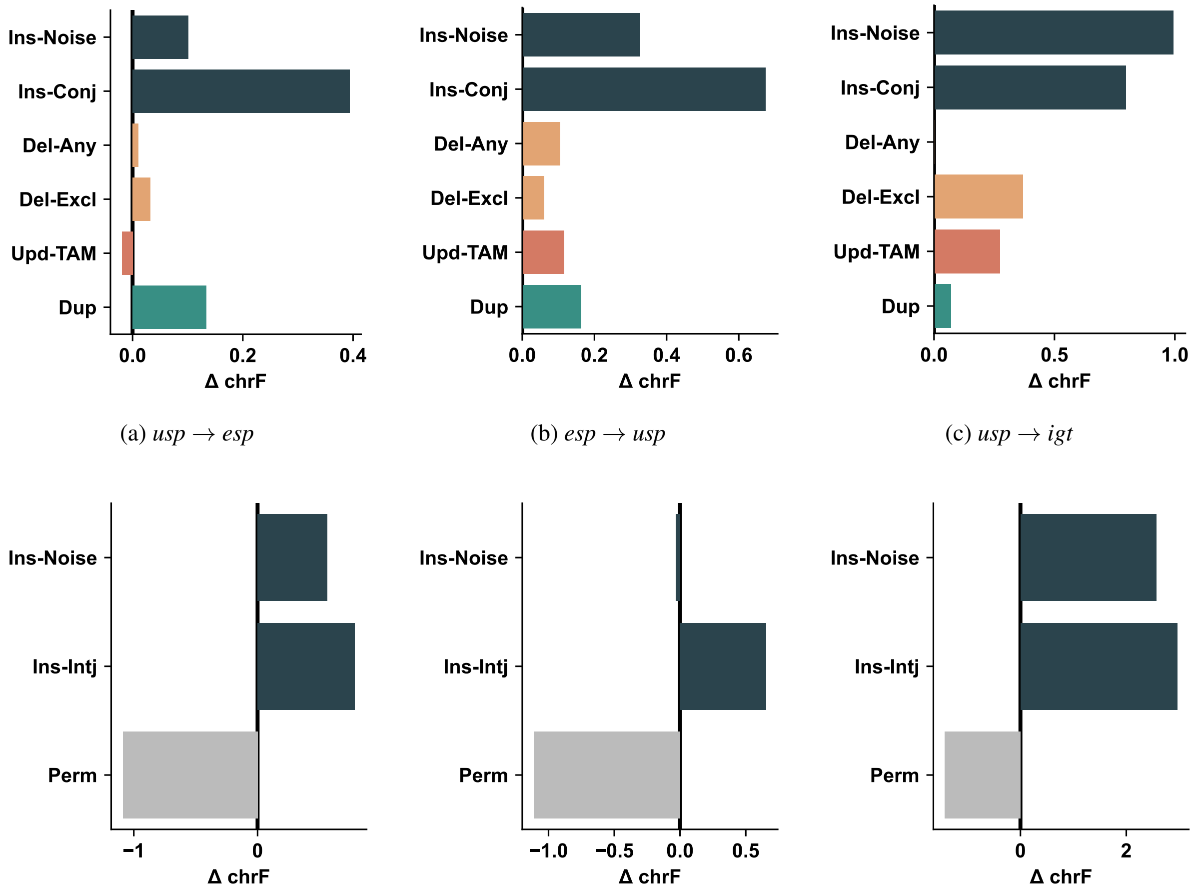
$$\dots$$

This allows us to disentangle the effects of adding a particular strategy from the interactions of the other strategies. We report these differences in Figure 3.

Finally, in Figure 4, we visualize the best overall augmentation strategies for each task, including combined strategies. We compute the mean improvement in chrF scores (on the evaluation set) across the five training set sizes and select the top five augmentation settings.

8 Discussion

All performance improvements are small, with the absolute best strategies achieving an improvement of around +8 (Uspanteko) or +3 (Arapaho) chrF points. This is unsurprising, as these sequence-to-sequence tasks are difficult, and perturbing natural examples may only increase the distributional coverage of our training set by a limited amount. Nonetheless, we do observe improvements on aver-



(a) *usp* → *esp* (b) *esp* → *usp* (c) *usp* → *igt*

(d) *arp* → *eng* (e) *eng* → *arp* (f) *arp* → *igt*

Figure 3: Average difference in (test set) chrF score between combinations including a given strategy and combinations excluding that strategy. Averaged over all runs and training sizes. Tabular form in Table 10.

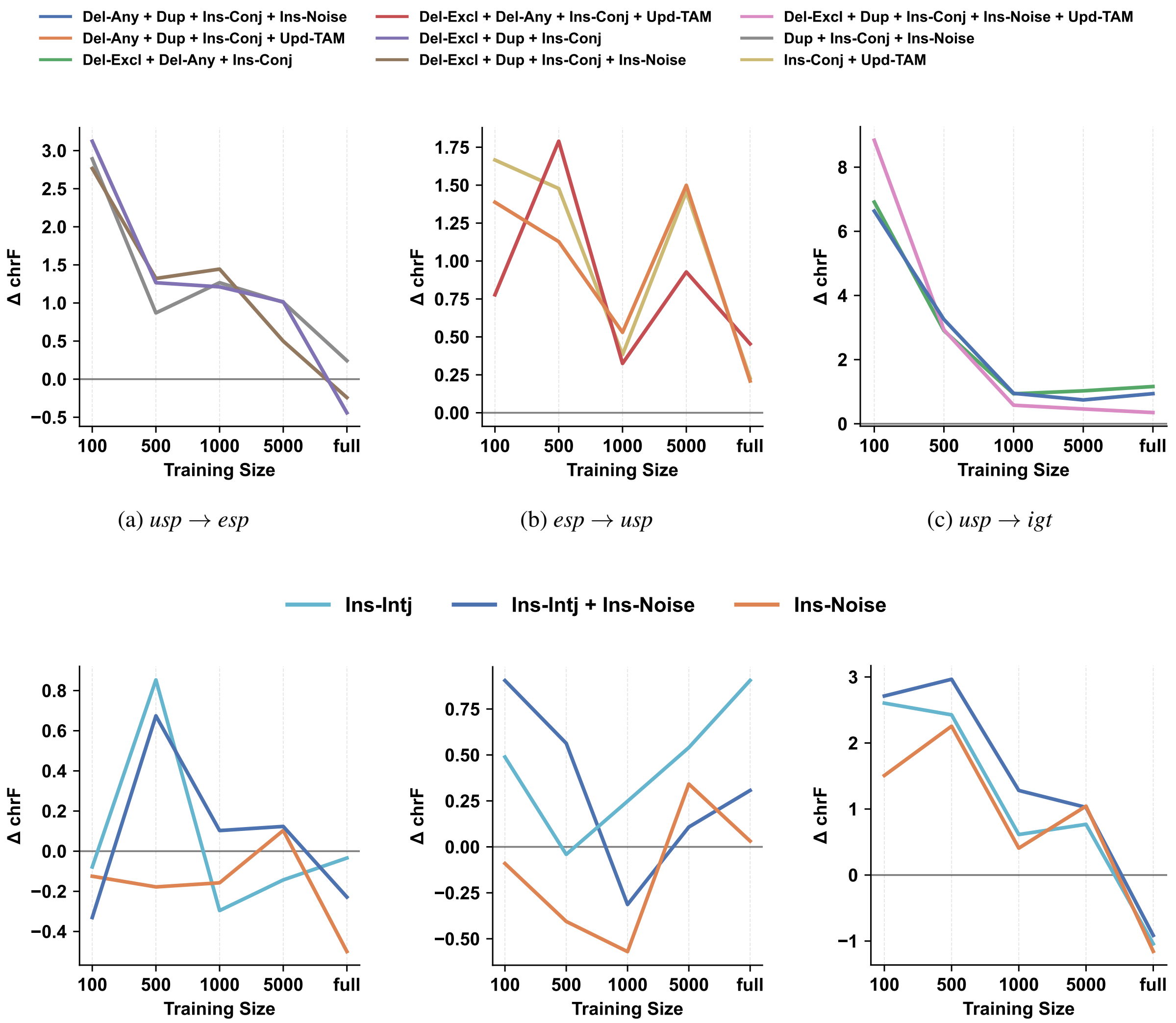
age in Figure 3 over all training sizes and combinations. While the improvements from data augmentation alone may not greatly alter the performance of these models, they can certainly be useful in achieving the best possible performance in combination with other techniques.

Effect of linguistically-motivated strategies We observe mixed effects from our linguistically-motivated strategies. The UPD-TAM strategy appears to provide small improvements on two of three tasks; however, these improvements are smaller than those of DUP, a completely non-linguistic strategy.

The linguistically-motivated DEL-EXCL strategy provides a benefit over the corresonding DEL-ANY for glossing and a small improvement for translation from Uspanteko, but the reverse effect on translation into Uspanteko. It is difficult to interpret this particular result as meaningful evidence one way or another.

On the other hand, we observe a clear improvement of INS-CONJ and INS-INTJ over the corresponding INS-NOISE strategy in both translation tasks (though the IGT task has a smaller or opposite effect). We also observe that INS-CONJ and INS-INTJ are generally the best individual strategies (Figure 2) and included in most of the top combined strategies (Figure 4). In this case, the evidence suggests that the linguistic motivation is beneficial. One possible explanation is that sentences with various conjunctions/interjections as the first word are typical in the actual data distribution, and adding more of such sentences helps the model learn this pattern. On the other hand, sentences with random words inserted in the first position are likely very different from the actual data.

We also observe that the PERM strategy consistently worsens performance, causing over a 1-point drop in chrF for translation (the drop is larger for IGT, which is unsurprising as the gloss sequences depend on the word order of the input sentence).



(a) *usp* → *esp* (b) *esp* → *usp* (c) *usp* → *igt*

(d) *arp* → *eng* (e) *eng* → *arp* (f) *arp* → *igt*

Figure 4: Performance of the best overall strategies, selected by average performance (chrF score on evaluation set) across training sizes. Performance is reported, as in Figure 2, as the difference between the target chrF score and the baseline score. Results are averaged over three runs. Tabular form in Table 11.

This is an interesting result, as we know that shuffling the word order should always produce a valid sentence in Arapaho (due to the language's free word order). However, even in languages with free word orders, speakers typically exhibit preferences towards certain orderings (Dryer, 1995). Thus, the augmented examples created by PERM may be very unlike the data distribution, which could account for the clear detriment to performance.

To control for this, we compute an additional evaluation metric. The metric is a variation of chrF that disregards word order, scoring any permutation of the correct words as correct. We compute this by simply removing the character n-grams which cross word boundaries. We report average results for Arapaho *eng* → *arp* setting with chrF and our modified chrF in Table 5. We observe that there is a similar trend between the two, indicating that the error is not due to producing sentences with grammatically valid but uncommon word orderings.

	Baseline	+PERM
chrF	30.0	29.0
Modified chrF	30.9	29.9

Table 5: Average chrF across training sizes for Arapaho *eng* → *arp*, using the standard chrF and a modified chrF that ignores word order. We observe a similar trend regardless of metric.

These findings point to a key takeaway for linguistically-motivated augmentation: in order to improve performance,[11] it is not sufficient for the augmented examples to be linguistically valid; they must also be similar (but not too similar) to the target data distribution. On the other hand, it is worth considering whether this is an appropriate evaluation for low-resource translation, since producing grammatical–but unusual–translations may be preferable.

Effect of combined augmentation strategies For Uspanteko, we observe that the best overall strategies always include a combination of various augmentation strategies (for Arapaho, there are far fewer possible combinations). One explanation for this is that the use of several strategies produces an augmented dataset with greater diversity, preventing the model from fitting too much to the specific type of augmented example.

Effect of training set size Unsurprisingly we observe that, in most cases, the magnitude of the improvements caused by data augmentation decreases with a larger original training dataset, most dramatically in the *usp* → *esp* and *usp* → *igt* settings. The clear takeaway is that while augmentation can provide benefits in low-resource settings, obtaining additional naturalistic data is more effective.

9 Conclusion

We observe varying performance benefits from different data augmentation strategies on translation and interlinear glossing in two low-resource languages. We consider augmentation strategies which utilize linguistic domain knowledge to produce more linguistically/grammatically valid synthetic examples, and we compare these strategies with approaches that simply utilize random noise and produce potentially ungrammatical examples. We find that the linguistic strategies that match the data distribution most closely (INS-CONJ, INS-INTJ) have clear benefits over the non-linguistic approach. On the other hand, a strategy that produces valid but rare examples (PERM) significantly worsens performance.

Overall, the answer to our primary research question is cautionary. There do appear to be cases where utilizing linguistic expertise for data augmentation can give an edge over general language-agnostic methods, if the strategies take into account the natural distribution of data. However, the improvements are small, and this may not be the most productive use of expert effort. Instead, this effort could be used to facilitate high-quality data collection and annotation, as collecting additional natural data has clear benefits across NLP tasks.

10 Limitations

As several of the methods by their very nature do not target every example in the original dataset (e.g., UPD-TAM is only relevant for sentences containing verbs marked with completive or incompletive aspect), the number of new examples generated varies across strategies. While we control for this effect by using a fixed number of training iterations, it is still possible that having a larger, and thus more diverse, augmented dataset has an effect.

11 Ethical Considerations

When working with endangered languages, it is vital to ensure that language data is used in accordance with the wishes of the language community (Schwartz, 2022). Furthermore, NLP systems for such languages should be used with caution, as low-quality translation/glossing/etc can be harmful to the language.

Acknowledgments

Author RG was supported in part by funding from CLASIC, CU Boulder's professional MS program in Computational Linguistics. Thanks to the CLASIC cohort and to anonymous reviewers for useful feedback. Parts 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. 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.

References

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

[11] At least on a held-out test set which is randomly sampled from the same distribution as the training data.

984–996, Hong Kong, China. Association for Computational Linguistics.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Ryan Bennett, Meg Harvey, Robert Henderson, and Tomás Alberto Méndez López. 2022. The phonetics and phonology of Uspanteko (Mayan). *Language and Linguistics Compass*, 16(9).

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.

Jessica Coon. 2016. Mayan morphosyntax. *Language and Linguistics Compass*, 10(10):515–550.

Andrew Cowell and Alonzo Moss. 2008. *The Arapaho Language*. University Press of Colorado.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sekou Diao. 2023. mlconjug3. *GitHub. Note: https://github.com/Ars-Linguistica/mlconjug3*.

Matthew S Dryer. 1995. Frequency and pragmatically unmarked word order. *Word order in discourse*, 30:105.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. 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.

Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. 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.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Omid Kashefi and Rebecca Hwa. 2020. Quantifying the evaluation of heuristic methods for textual data augmentation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 200–208, Online. Association for Computational Linguistics.

William Lane and Steven Bird. 2020. Bootstrapping techniques for polysynthetic morphological analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6652–6661, Online. Association for Computational Linguistics.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Xia Li and Junyi He. 2021. Data augmentation of incorporating real error patterns and linguistic knowledge for grammatical error correction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 223–233, 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.

Agustín Lucas, Alexis Baladón, Victoria Pardiñas, Marvin Agüero-Torales, Santiago Góngora, and Luis Chiruzzo. 2024. Grammar-based data augmentation for low-resource languages: The case of Guarani-Spanish neural machine translation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6385–6397, Mexico City, Mexico. Association for Computational Linguistics.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jaehyung Seo, Hyeonseok Moon, Jaewook Lee, Sugyeong Eo, Chanjun Park, and Heuiseok Lim. 2023. CHEF in the language kitchen: A generative data augmentation leveraging Korean morpheme ingredients. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6014–6029, Singapore. 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.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online. Association for Computational Linguistics.

Miguel Angel Vicente Méndez. 2007. *Diccionario bilingüe uspanteko-español - Cholaj tzijb'al li Uspanteko*. Cholsamaj, Guatemala.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563, Lisbon, Portugal. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

A Training Details

We did not perform extensive hyperparameter optimization. For each language, we started with the default parameters and made minor adjustments until we achieved relatively low loss on the training and eval set. We use the Adam optimizer with default parameters and the hyperparameters described in Table 6. For the augmented models, we first train on the augmented data for the specified number of steps (500 or 2000 for Uspanteko or Arapaho).

Then, we train on the original training dataset for 1000 or 4000 steps. For the non-augmented models, we train on the original training dataset in both phases, but still reset the optimizer between phases.

Parameter	Usp	Arp
Batch size	32	16
Learning rate	2E-4	2E-4
Weight decay	0.5	0.5
Training steps (aug. data)	500	2000
Training steps (training data)	1000	4000

Table 6: Hyperparameters for all training runs in each language.

The only parameters we specifically tuned were weight decay and the number of training steps, in order to prevent overfitting and ensure convergence. As the Arapaho dataset is roughly four times larger than the Uspanteko dataset, we use four times as many training steps. We train models on several A100 GPUs in the (omitted cluster name for anonymity), and the entire study used around 1000 GPU hours.

B Augmentation Examples

We provide examples for each strategy in Uspanteko (Table 7) and Arapaho (Table 8).

C Table Results

In this section, we provide the corresponding numerical results for all of the visualizations.

Table 7: Augmentation examples for each method in Uspanteko. Modified parts of the examples are highlighted with bold face.

ORIGINAL	wi' neen tb'ank juntir wi' neen t-b'an-k juntiir EXS INT INC-hacer-SC todo Tienen que hacer todo
UPD-TAM	wi' neen **x**b'ank juntir wi' neen **x**-b'an-k juntiir EXS INT **COM**-hacer-SC todo **tuvieron** que hacer todo
INS-CONJ	**Pwes** wi' neen tb'ank juntir **Pwes** wi' neen t-b'an-k juntiir **pues** EXS INT INC-hacer-SC todo **Pues** Tienen que hacer todo
INS-NOISE	**Saneb'** wi' neen tb'ank juntir **Saneb'** wi' neen t-b'an-k juntiir **arena@de@rio** EXS INT INC-hacer-SC todo **Harenas del río** Tienen que hacer todo
DEL-ANY	wi' neen **[–]** juntir wi' neen **[–]** juntiir EXS INT **[–]** todo Tienen que **[–]** todo
DEL-EXCL	wi' neen tb'ank **[–]** wi' neen t-b'an-k **[–]** EXS INT INC-hacer-SC **[–]** Tienen que hacer **[–]**
DUP	wi' neen tb'ank **tb'ank** juntir wi' neen t-b'an-k **t-b'an-k** juntiir EXS INT INC-hacer-SC **INC-hacer-SC** todo Tienen que hacer **hacer** todo

Table 8: Augmentation examples for each method in Arapaho. Modified parts of the examples are highlighted with bold face.

ORIGINAL	Nihtooneete3eino' hini' xouu PAST-almost-run.into-1S that(aforementioned).those skunk I almost ran into that skunk .
INS-INTJ	**Yeheihoo** Nihtooneete3eino' hini' xouu **gee.whiz** PAST-almost-run.into-1S that(aforementioned).those skunk **Gee whiz** I almost ran into that skunk .
INS-NOISE	**Bih'ih** Nihtooneete3eino' hini' xouu **mule.deer** PAST-almost-run.into-1S that(aforementioned).those skunk **Mule deer** I almost ran into that skunk .
PERM	hini' xouu Nihtooneete3eino' **[order changed]** PAST-almost-run.into-1S that(aforementioned).those skunk **[order changed]** I almost ran into that skunk . **[order changed]**

Table 9: Difference in (test set) chrF score for various individual augmentation strategies from the baseline. Reported as the mean over three runs, with the format mean(std).

(a) *usp* → *esp*

	100	500	1000	5000	full
DEL-EXCL	1.57 (1.07)	-0.06 (1.27)	0.85 (0.47)	-0.98 (0.87)	-0.83 (1.99)
INS-CONJ	3.07 (0.89)	1.09 (0.43)	1.07 (1.04)	0.27 (0.47)	-0.63 (2.04)
INS-NOISE	2.20 (1.18)	0.00 (1.28)	1.27 (0.55)	0.63 (0.51)	0.13 (1.81)
DEL-ANY	0.76 (0.94)	0.40 (0.62)	1.34 (0.56)	-1.56 (0.50)	0.18 (1.76)
DUP	1.33 (1.41)	0.08 (1.24)	0.90 (0.54)	-0.71 (0.50)	-0.36 (1.79)
UPD-TAM	-0.08 (0.80)	-1.03 (0.31)	0.30 (1.19)	0.05 (0.79)	-0.72 (1.79)

(b) *esp* → *usp*

	100	500	1000	5000	full
DEL-EXCL	-0.69 (1.01)	-1.29 (0.71)	-2.02 (0.77)	-0.42 (0.59)	-0.52 (1.81)
INS-CONJ	1.21 (0.88)	1.85 (0.73)	0.27 (0.70)	0.75 (0.82)	0.06 (0.95)
INS-NOISE	0.80 (0.81)	0.05 (1.20)	-0.12 (0.79)	1.44 (0.63)	0.05 (0.53)
DEL-ANY	0.49 (0.95)	-0.84 (0.98)	-2.41 (1.38)	-1.08 (0.70)	-0.63 (1.19)
DUP	0.45 (0.96)	-0.88 (0.39)	-0.53 (1.06)	-0.05 (0.79)	-1.23 (0.59)
UPD-TAM	0.49 (0.69)	-1.83 (0.73)	-2.03 (1.24)	-0.48 (0.97)	-0.52 (0.90)

(c) *usp* → *igt*

	100	500	1000	5000	full
DEL-EXCL	-0.55 (2.77)	-0.17 (1.96)	-1.39 (0.70)	-0.13 (0.78)	0.67 (0.68)
INS-CONJ	5.81 (2.74)	2.17 (1.65)	0.25 (0.93)	0.93 (0.77)	1.02 (0.64)
INS-NOISE	5.41 (2.93)	1.78 (1.93)	0.75 (0.71)	0.89 (0.80)	1.11 (0.35)
DEL-ANY	0.12 (1.95)	-1.48 (1.66)	-0.86 (0.75)	-0.05 (0.75)	0.61 (0.84)
DUP	0.81 (1.94)	-1.01 (1.75)	-0.66 (0.58)	0.14 (0.77)	1.11 (0.20)
UPD-TAM	-1.03 (1.78)	-2.43 (1.66)	-2.04 (0.71)	0.56 (0.90)	0.92 (0.51)

(d) *arp* → *eng*

	100	500	1000	5000	full
INS-INTJ	-0.08 (0.97)	0.85 (0.35)	-0.30 (0.70)	-0.14 (0.87)	-0.03 (0.25)
INS-NOISE	-0.12 (0.62)	-0.18 (0.73)	-0.16 (0.84)	0.10 (0.87)	-0.50 (0.42)
PERM	-1.08 (0.53)	-3.48 (0.47)	-5.95 (0.58)	-1.44 (0.58)	-2.71 (0.52)

(e) *eng* → *arp*

	100	500	1000	5000	full
INS-INTJ	0.49 (1.08)	-0.04 (0.79)	0.25 (0.82)	0.54 (0.81)	0.91 (2.20)
INS-NOISE	-0.09 (1.05)	-0.41 (0.87)	-0.57 (0.84)	0.34 (0.54)	0.03 (2.43)
PERM	-0.32 (0.87)	-2.03 (0.93)	-1.66 (0.79)	-0.41 (1.04)	-2.30 (2.87)

(f) *arp* → *igt*

	100	500	1000	5000	full
INS-INTJ	2.60 (0.91)	2.43 (1.89)	0.61 (0.62)	0.77 (0.82)	-1.04 (0.34)
INS-NOISE	1.51 (0.93)	2.25 (2.76)	0.41 (1.22)	1.04 (0.48)	-1.16 (1.89)
PERM	-1.17 (1.13)	-16.75 (1.82)	-18.56 (0.76)	-2.68 (0.40)	-1.71 (0.21)

Table 10: Average difference in (test set) chrF score between combinations including a given strategy and combinations excluding that strategy. Reported as the mean over all runs and training sizes, with the format mean(std).

(a) *usp* → *esp*

INS-NOISE	0.10 (1.08)
INS-CONJ	0.39 (1.17)
DEL-ANY	0.01 (0.94)
DEL-EXCL	0.03 (0.95)
UPD-TAM	-0.02 (1.00)
DUP	0.13 (0.98)

(b) *esp* → *usp*

INS-NOISE	0.33 (1.21)
INS-CONJ	0.67 (1.27)
DEL-ANY	0.11 (1.06)
DEL-EXCL	0.06 (1.10)
UPD-TAM	0.12 (1.08)
DUP	0.16 (1.08)

(c) *usp* → *igt*

INS-NOISE	0.99 (3.59)
INS-CONJ	0.80 (3.71)
DEL-ANY	0.00 (3.30)
DEL-EXCL	0.37 (3.29)
UPD-TAM	0.27 (3.25)
DUP	0.07 (3.31)

(d) *arp*→ *eng*

INS-NOISE	0.56 (1.45)
INS-INTJ	0.79 (1.41)
PERM	-1.09 (1.06)

(e) *eng*→ *arp*

INS-NOISE	-0.03 (1.34)
INS-INTJ	0.65 (1.56)
PERM	-1.11 (1.78)

(f) *arp*→ *igt*

INS-NOISE	2.57 (6.00)
INS-INTJ	2.96 (6.05)
PERM	-1.43 (0.88)

Table 11: Performance of the best overall strategies, selected by average performance (chrF score on evaluation set) across training sizes. Performance is reported as the difference between the target chrF score and the baseline score. The results are reported as the mean over three runs, with the format mean(std).

(a) *usp* → *esp*

	100	500	1000	5000	full
DEL-EXCL + DUP + INS-CONJ	3.12 (0.73)	1.27 (0.86)	1.21 (1.29)	1.02 (0.62)	-0.44 (2.06)
DEL-EXCL + DUP +INS-CONJ + INS-NOISE	2.77 (0.78)	1.32 (0.61)	1.44 (0.50)	0.50 (0.61)	-0.24 (1.70)
DUP + INS-CONJ + INS-NOISE	2.89 (1.45)	0.87 (1.05)	1.27 (0.79)	1.01 (0.42)	0.24 (1.88)

(b) *esp* → *usp*

	100	500	1000	5000	full
DEL-ANY + DUP +INS-CONJ + UPD-TAM	1.39 (1.18)	1.13 (1.47)	0.53 (0.74)	1.50 (0.77)	0.21 (1.00)
DEL-EXCL + DEL-ANY + INS-CONJ + UPD-TAM	0.78 (0.86)	1.79 (0.61)	0.32 (1.07)	0.93 (0.63)	0.45 (0.83)
INS-CONJ + UPD-TAM	1.67 (0.78)	1.48 (0.75)	0.38 (0.88)	1.46 (0.68)	0.23 (0.60)

(c) *usp*→ *igt*

	100	500	1000	5000	full
DEL-ANY + DUP +INS-CONJ + INS-NOISE	6.63 (2.47)	3.25 (1.74)	0.95 (0.75)	0.74 (0.72)	0.94 (0.29)
DEL-EXCL + DEL-ANY +INS-CONJ	6.91 (2.44)	2.90 (1.85)	0.93 (0.58)	1.03 (0.76)	1.16 (0.13)
DEL-EXCL + DUP + INS-CONJ + INS-NOISE + UPD-TAM	8.84 (3.67)	2.93 (1.72)	0.58 (0.59)	0.46 (0.96)	0.35 (0.51)

(d) *arp*→ *eng*

	100	500	1000	5000	full
INS-INTJ	-0.08 (0.97)	0.85 (0.35)	-0.30 (0.70)	-0.14 (0.87)	-0.03 (0.25)
INS-INTJ + INS-NOISE	-0.33 (0.64)	0.67 (0.25)	0.10 (0.50)	0.12 (0.95)	-0.23 (0.21)
INS-NOISE	-0.12 (0.62)	-0.18 (0.73)	-0.16 (0.84)	0.10 (0.87)	-0.50 (0.42)

(e) *eng*→ *arp*

	100	500	1000	5000	full
INS-INTJ	0.49 (1.08)	-0.04 (0.79)	0.25 (0.82)	0.54 (0.81)	0.91 (2.20)
INS-INTJ + INS-NOISE	0.90 (0.96)	0.56 (0.66)	-0.31 (1.26)	0.11 (0.61)	0.31 (2.43)
INS-NOISE	-0.09 (1.05)	-0.41 (0.87)	-0.57 (0.84)	0.34 (0.54)	0.03 (2.43)

(f) *arp*→ *igt*

	100	500	1000	5000	full
INS-INTJ	2.60 (0.91)	2.43 (1.89)	0.61 (0.62)	0.77 (0.82)	-1.04 (0.34)
INS-INTJ + INS-NOISE	2.71 (1.20)	2.97 (2.23)	1.28 (0.67)	1.02 (0.56)	-0.91 (0.67)
INS-NOISE	1.51 (0.93)	2.25 (2.76)	0.41 (1.22)	1.04 (0.48)	-1.16 (1.89)