Improving Low-Resource Morphological Inflection via Self-Supervised Objectives

Adam Wiemerslage[1,3] and **Katharina von der Wense**[1,2]
[1]University of Colorado Boulder
[2]Johannes Gutenberg University Mainz
[3]Kensho Technologies
{adwi9965,katharina.kann}@colorado.edu

Abstract

Self-supervised objectives have driven major advances in NLP by leveraging large-scale unlabeled data, but such resources are scarce for many of the world's languages. Surprisingly, they have not been explored much for character-level tasks, where smaller amounts of data have the potential to be beneficial. We investigate the effectiveness of self-supervised auxiliary tasks for morphological inflection – a character-level task highly relevant for language documentation – in extremely low-resource settings, training encoder-decoder transformers for 19 languages and 13 auxiliary objectives. Autoencoding yields the best performance when unlabeled data is very limited, while character masked language modeling (CMLM) becomes more effective as data availability increases. Though objectives with stronger inductive biases influence model predictions intuitively, they rarely outperform standard CMLM. However, sampling masks based on known morpheme boundaries consistently improves performance, highlighting a promising direction for low-resource morphological modeling.

1 Introduction

Rapid progress in natural language processing (NLP) has largely been driven by training transformer models on massive amounts of unlabeled data. However, for many of the world's languages, such large datasets do not exist. Still there is great potential for NLP to contribute to areas like language documentation or indigenous language educational technology, where both labeled and unlabeled data is extremely sparse, e.g., via character-level tasks like text normalization, grapheme-to-phoneme conversion, or morphological inflection.

Overcoming low-resource data challenges typically involves biasing an NLP model towards a particular language, task or domain. For instance, in the morphology community, prior work has focused on implementing neural architectures with an inherent inductive bias towards particular tasks like inflection (Aharoni and Goldberg, 2017; Makarov and Clematide, 2018; Wu and Cotterell, 2019).

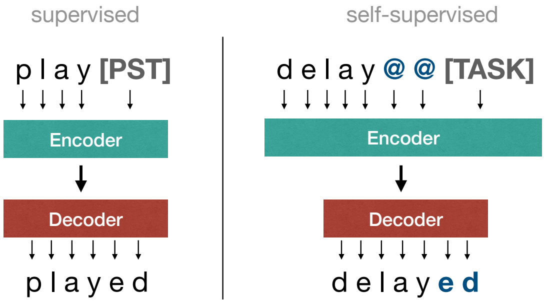


Figure 1: Example of the multitask training; *left:* inflection with the past tense tag; *right:* self-supervised masked language modeling. @ represents the mask token; *[TASK]* is a special tag for the self-supervised task.

Instead, we take inspiration from high-resource NLP where progress has been largely data-driven, focused on training NLP models with self-supervised objectives like language modeling. We explore self-supervised objectives, based on masked language modeling (Devlin et al., 2019, MLM), that have more inductive bias towards a particular character-level task for extremely small datasets in a multitask training setup. The goal is to change the MLM objective in order to optimally leverage unlabeled data to bias the model towards a particular task. This is more interpretable and simpler to implement than adjusting the inductive bias of an architecture.

Prior work has shown that auxiliary training on unlabeled data benefits character-level tasks, but has only investigated a small subset of standard objectives. We hypothesize that this is because tasks like MLM at a typical scale are often associated with high computational cost. We highlight that for many low-resource character-level tasks efficiency is not actually a challenge. Those tasks allow us to quickly and cheaply explore variations of auxiliary tasks. We propose several novel self-supervised

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24494–24510

task objectives with two goals: (1) to improve downstream task performance, and (2) gain a better understanding of how and why self-supervised auxiliary tasks work. We focus on morphological inflection, which has been featured in recent work on auxiliary task training (Kann and Schütze, 2017; Purushothama et al., 2024). Furthermore, it is a character-level task with small models that maintains relevance even in the age of LLMs, due to its potential to aid language documentation efforts (Moeller et al., 2020; Moeller and Hulden, 2018).

We conduct experiments with an encoder-decoder transformer (Vaswani et al., 2017) featuring 19 languages, 13 different auxiliary objectives, and 6 different unlabeled datasets. On a popular inflection benchmark, we find that simply autoencoding unlabeled words, with no masking leads to the best performance when the unlabeled dataset is very small, but MLM performs best as we attain more unlabeled data. While we find evidence that MLM objectives with more inductive bias influence model predictions in intuitive ways, they typically do not outperform the typical MLM setup on average. However, we find that sampling masks according to known morpheme boundaries tends to produce the best performance, indicating a promising direction.

2 Related Work

Inductive Bias of Auxiliary Tasks Following Devlin et al. (2019), several variations of pretraining tasks have been proposed. Joshi et al. (2020) propose masking spans of text in order to train models with an inductive bias towards span selection. Raffel et al. (2020) explore pretraining objectives in an encoder-decoder setup and find that a span-denoising objective performs best. Lewis et al. (2020) propose additionally shuffling the order of sentences, leading to models particularly strong at machine translation. More recent work typically assumes a model has been pretrained with autoregressive language modeling, but explore further training like instruction tuning (Mishra et al., 2022; Wei et al., 2021), or reinforcement learning (Christiano et al., 2017; Ouyang et al., 2022) to align large language models to human preferences.

Auxiliary Tasks for Character-Level Tasks Some work has focused specifically on character-level models in the low-resource setting like we do. Xue et al. (2022) pretrain an encoder-decoder following the objective in Raffel et al. (2020), but at the *byte* rather than subword level. They perform well on many character-level tasks including morphological inflection. Closely related to our work, Kann and Schütze (2017) train inflection models in a mutlitask training setup with autoencoding as an additional objective. Purushothama et al. (2024) also explore multitask training for inflection, comparing autoencoding to denoising, and find large performance gains even when unlabeled data is sampled exclusively from the supervised training set. This implies that models benefit from the task itself, rather than simply the additional data distribution. Our work builds closely upon this idea. Whereas (Purushothama et al., 2024) shows that the self-supervised objective is impactful beyond the incorporation of new data, we explore which self-supervised objective—among several heuristics—is best on various datasets comprising different properties. Other work has specifically investigated the inductive bias of autoencoding for inflection as beneficial for *copy bias* (Liu and Hulden, 2022; Yang et al., 2022).

3 Morphological Inflection

Morphological inflection is a character-level sequence-to-sequence task. Such tasks are characterized by small vocabularies Σ as well as short source and target strings. Given a source lemma $\ell \in \Sigma^*$, target word $y \in \Sigma^*$, and a bundle of morphological feature tags $t \in \tau$, the goal of morphological inflection is to learn a mapping

$$f(\ell, t) \rightarrow y,$$

for instance, an inflection example for generating the past tense of the verb *try* would look like:

$$f(\textit{try}, \text{PST}) \rightarrow \textit{tried}.$$

4 Self-Supervised Training Heuristics

We compare several heuristics for self-supervised training, which are optimized in a multitask learning setup along with inflection. Each heuristic is implemented as an MLM method, which we vary in order to change the inductive bias of the learning problem. We assume a small lexicon $\mathcal{L}$ of words, and an even smaller supervised inflection dataset $\mathcal{D}$ to model with an encoder-decoder.

Baselines We compare all heuristics to two baselines. First, we train exclusively on $\mathcal{D}$, ignoring $\mathcal{L}$ entirely. We refer to this model as `baseline`.

We also implement an auxiliary baseline, which in addition to the supervised task, trains on $\mathcal{L}$ with autoencoding (`AE`). A self-supervised learning sample with autoencoding would be presented to the model, for instance, as:

$$tried + \text{[TASK]} \rightarrow tried, \quad (1)$$

where [TASK] is a special symbol for the self-supervised task that takes the role of the inflection tags in the supervised task. Each heuristic noising method combines an *objective* with a sampling *strategy*, and either *masks* or *deletes* characters.

Objective We compare two objectives for MLM. The first, CMLM (Wiemerslage et al., 2023), follows the MLM method from Liu et al. (2019)—which samples 15% of tokens dynamically and 80% of the time replaces them with a special ⟨MASK⟩ symbol, 10% replaces them with another token from the vocabulary, or 10% leaves the token as is—except it is performed at the character level and the mask sampling rate is increased to 25%. For instance, representing the mask token with @:

$$tr@@ed + \text{[TASK]} \rightarrow tried \quad (2)$$

We additionally implement the span based variant for the T5 objective from Raffel et al. (2020). This differs from CMLM in that the sampled token is *always* replaced with a mask, adjacent masks are merged together into a single mask, and each mask token is unique. For instance

$$t{<}X{>}e{<}Y{>} + \text{[TASK]} \rightarrow tried, \quad (3)$$

where $<X>$ and $<Y>$ are mask tokens. Notice that this differs from the details of Raffel et al. (2020), who additionally mask the output sequence, keeping only the dropped out tokens intact:

$$t{<}X{>}e{<}Y{>} + \text{[TASK]} \rightarrow {<}X{>}ri{<}Y{>}d. \quad (4)$$

In initial experiments, we found that this method in Example 4 severely underperformed masking only the source side as in Example 3, so we exclude it from the experiments presented in this work.

Strategy We compare three masking strategies: `iid`, `suffix`, and `prefix`. For each strategy, we sample 25% of character positions in a word. Each strategy varies in the distribution over characters from which we sample characters to be masked. **`iid`** sampling is implemented with a uniform distribution over all characters. The **`suffix`** strategy explores the hypothesis that masking with a bias towards the end of a string should create training samples that resemble concatenative, suffixing inflection, which is typologically pervasive. Here, we sample from a distribution that is skewed toward the end of the sequence. We distribute 95% of the probability mass uniformly over the final 1/3 of characters, and the other 5% uniformly over the first 2/3 of characters. Finally, for the **`prefix`** strategy, which is typologically less common for inflection and not well-represented in our dataset, we do the same thing as the suffix strategy, but skew the distribution to the start of the sequence.

Character Deletion We additionally explore deleting rather than masking characters. Intuitively, given that inflection is a sequence-to-sequence task where in typical concatenative inflection some substring is appended to the input sequence, deleting characters of a word and then predicting them simulates the target task more closely. For instance, if we sample the word ‘baked’, and apply the `suffix` strategy, we might produce the training sample

$$bake \rightarrow baked. \quad (5)$$

which exactly resembles a supervised sample for producing the past tense without the inflection tag.

Dataset	Samples	Types	Med Len	N-grams
ud-1k	5836.26	3260.68	12.66	5725.79
ud-200	5278.37	2659.11	11.74	4839.58
ud-vnadj	5073.00	3297.58	11.63	4744.63
ud-wl	5263.68	3155.84	12.79	5479.89
ud-vnadj-NR	4976.74	4902.68	12.24	6273.21
ud-wl-NR	5132.74	5059.74	12.92	7562.37

Table 1: Statistics for all datasets, averaged over all languages. This includes the supervised and unlabeled data. Types is the number of unique words, Med Len is the median number of characters in each word, and N-grams is the count of all unique char. bi and trigrams.

Dataset	Samples	Types	Med Len	N-grams
ud-1k	6000.0	3139.2	11.7	4779.8
ud-200	5360.0	2475.4	10.5	4136.0
ud-vnadj	5360.0	3264.6	11.8	3861.6
ud-wl	5360.0	3013.4	11.5	4494.6
ud-wl-NR	5360.0	5320.4	13.2	6829.6
ud-vnadj-NR	5360.0	5320.0	13.1	5467.0
seg	5360.0	5320.4	13.3	5645.0

Table 2: Statistics for all datasets, averaged over the 5 languages with morphological segmentation. This includes the supervised and external data.

Note that, with deletion, T5 and CMLM result in *almost* the same denoising method, since merging masks into spans is irrelevant. The only difference is that CMLM sometimes masks with another character in the vocabulary (10%) or applies the identity function to a character sampled for masking (10%).

Segment Masking We further explore if masking subword segments according to oracle morpheme boundaries results in an objective more suitable for morphological inflection. For a subset of languages, we attain gold morpheme boundaries for the unlabeled words, and query this as an oracle. For this dataset, we produce self-supervised data that samples segments, rather than characters, for masking. All details are otherwise identical to character-level masking. For instance, given the word 'walk - ing', the CMLM objective and suffix strategy, we will produce with 95% probability

$$walk@@@ \to walking, \tag{6}$$

where the segment -ing is masked.

5 Experiments

5.1 Data

We follow Purushothama et al. (2024) in evaluating on SIGMORPHON 2023 shared task data (Goldman et al., 2023) with a subset of training data, and unlabeled data sampled from universal dependencies (UD) treebanks (Zeman et al., 2023). First, we produce results on the exact dataset from Purushothama et al. (2024), comprising 19 languages, 1k supervised samples, and 5k unlabeled words, which we refer to as **ud-1k**. The 19 languages are the subset of SIGMORPHON 2023 data for which UD corpora are also available. In all other experiments, we simulate a lower-resource setting than that work. For each language we sample 200 supervised samples from the SIGMORPHON 2023 training set for each part of speech. This means that each dataset contains 200 – 600 supervised samples depending on which parts of speech are present in the shared task data (nouns, adjectives, or verbs). Sampling is done one paradigm at a time, ensuring a broad variety of inflection tags are attested. This results in supervised datasets similar in size to the low-resource setting studied in the SIGMORPHON 2022 shared task on inflection (Kodner et al., 2022), which sampled 700 inflection pairs. The upper bound on our supervised datasets is 600 inflection pairs if all three of verbs, nouns, and adjectives are available. We augment these supervised samples with 5k unlabeled samples from UD. The unlabeled data is either copied from Purushothama et al. (2024) (**ud-200**), or resampled with additional constraints. In one dataset we constrain word-length: ensuring that all unlabeled words contain at least three characters (two for Japanese) to increase the chance that the dataset contains morphologically rich words (**ud-wl**). We build another dataset that samples from words that are additionally tagged with a verb, noun, or adjective part of speech in the UD treebank, to further increase the likelihood of morphological productivity (**ud-vnadj**).

All unlabeled data so far has been sampled *with* replacement. Given the size of the UD corpora, this leads to several languages with many duplicate words in the 5k unlabeled samples (See Table 1). We thus also sample datasets without replacement (**ud-wl-NR**, **ud-vnadj-NR**) to increase the diversity of the dataset, leading to 5k unique words in most cases. This means that the ***-NR** datasets effectively increase the size of the unlabeled dataset. We sample from additional UD corpora where necessary in order to ensure that we can sample 5k unique words. This was not possible for Amharic (amh) nor Sanskrit (san), so those datasets continue to contain fewer than 5k words. Finally, we build a dataset where all unlabeled words are segmented into morphemes (**seg**). For this, we use the data from the SIGMORPHON 2022 morpheme segmentation shared task (Batsuren et al., 2022). This dataset contains *canonical* segmentation, but we require a *surface* segmentation. We produce surface segmentation boundaries with a heuristic algorithm presented in Appendix B. Because the overlap in languages between the two shared tasks is small, this dataset contains only 5 languages: English, Hungarian, Italian, Russian, and Spanish. This dataset is very morphologically rich, because the shared task sampled words contain many morphemes. Statistics in Table 2 show a high median word length, and a large number of unique character n-grams. The notably high number of n-grams in the ud-wl-NR dataset is likely due to noisy web-scraped data—which is pervasive in the English data—that results in rare characters and character n-grams. We expect the seg dataset to represent many inflection variations, but also many non-inflectional processes, like derivation, or, e.g., Spanish pronominal participles.

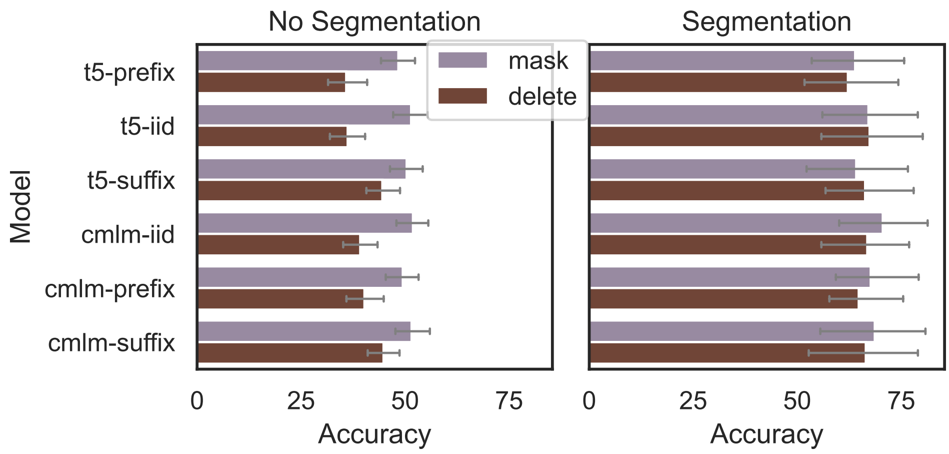


Figure 2: Results for masking v.s. deleting for all models averaged over all datasets. On the right are the segmentation variants of all models, for which there is only one dataset comprising only 5 languages.

	`baseline`	`t5-pref`	`cmlm-pref`	`t5-suff`	`cmlm-suff`	`t5-iid`	`cmlm-iid`	`AE`
ud-1k	64.39*	73.4	69.91	73.43	74.07	74.39	<u>74.67</u>	**75.83**
ud-200	5.16	38.31	41.82	41.34	42.76	41.04	<u>42.92</u>	**47.48**
ud-wl	5.16	38.91	43.16	43.07	<u>44.98</u>	43.05	43.98	**49.55**
ud-vnadj	5.16	39.51	41.46	41.86	<u>43.39</u>	42.86	42.91	**47.8**
ud-wl-NR	5.16	48.04	48.76	49.19	50.51	<u>51.26</u>	**51.68**	50.49
ud-vnadj-NR	5.16	47.67	47.88	49.2	49.42	**51.48**	<u>50.46</u>	50.06

Table 3: Accuracy for all main experiments averaged over all languages. Note that baseline results are the same because each dataset shares the exact same supervised training samples. The best performing self-supervised setup per dataset is **bolded**, and the 2nd best is <u>underlined</u>. *Taken from published results in Purushothama et al. (2024)

5.2 Model

For all experiments, we train an encoder-decoder transformer (Vaswani et al., 2017) following Wu et al. (2021). We use a multitask training setup, where both the supervised and self-supervised task can occur in the same batch, and their losses are summed. Inflection tags are included as additional tokens in the input sequence. For the self-supervised task, these are replaced with a special task tag. Exact hyperparamters are available in Appendix C. Each model is 7.4 million parameters, and trains in about 1 hour on an NVIDIA A100.

6 Results

We first investigate the impact of masking vs. deleting characters for the denoising objectives. In Figure 2 masking outperforms deleting for all training setups on average over all datasets. However, when the training heuristic has knowledge of morpheme boundaries, this is not strictly true. In this case, for both `t5-suffix` and `t5-iid` deletion is stronger. In all subsequent experiments we focus on masking, which is typically the superior strategy.

Next we present the main results in Table 3 averaged over all languages. For per-language results see the Appendix. We first reproduce the results from Purushothama et al. (2024, ud-1k) showing that autoencoding outperforms all denoising methods. The same denoising method evaluated in their work, `cmlm-iid`, is the best performer after autoencoding. Then we evaluate in our setting, which constrains the supervised data to a smaller training set, while maintaining the same UD dataset for self-supervision. Here, we attain the same relative result but the gap in performance between autoencoding and the best denoising setup is much larger (4.56 absolute accuracy), due to the smaller supervised dataset. When we sample the unlabeled UD data with a minimum word-length constraint (ud-wl), we see an increase in accuracy across the board, though autoencoding still performs best by a large margin. Here, however, the suffix strategy is the best denoising method. Additionally constraining data to only verbs, adjectives and nouns (ud-vnadj) actually reduces performance when compared to the ud-wl dataset. This is likely because it reduces the diversity of the dataset due to those constraints (See: Table 1, N-grams).

Sampling Words without Replacement Table 1 shows that all datasets so far have several fewer unique types than the total number of samples. This is because we followed Purushothama et al. (2024) by sampling words from UD with replacement, causing many duplicate words. When we resample the datasets without replacement, effectively increasing the size of the unlabeled datasets to be truly 5000 samples, there is an increase in accuracy across the board, with a substantial increase for all denoising strategies (up to 9.13 absolute accuracy). For these datasets, denoising clearly outperforms autoencoding.

Objective Figure 3a presents a comparison of the two self-supervised objectives averaged over all languages and datasets on the left. For all strategies, CMLM outperforms T5 on average. Despite this, when combined with the best strategy (`iid`), the difference of these averages is rather small.

Strategy Figure 3b presents a comparison of the three self-supervised strategies averaged over all languages and datasets. The `iid` strategy performs best on average for both objectives, followed closely by the `suffix` strategy.

Segmentation Finally, we present results for the five segmentation languages in Table 4. When using the CMLM objective, the segmentation variant—which samples entire morphemes for masking rather than individual characters—always outperforms the corresponding setup without segmentation. However, the opposite is true for T5 – in these cases the segmentation variant *underperforms*. Figure 2 also shows that deleting tends to outperform masking for the T5 objective at the segment level. The best performing system in this experiment on average is `cmlm-seg-iid`. However, the same setup without segmentation performs only 0.58 absolute accuracy lower on average.

6.1 Language-Specific Results

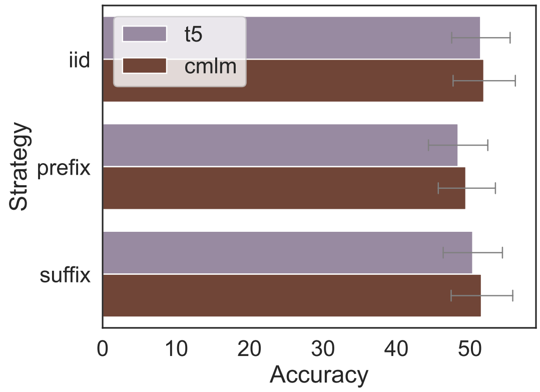
We perform a small language-specific analysis focused on the self-supervised strategy, which we expect to be most relevant to language typology. We focus on the ud-vnadj-NR dataset, which we had expected to contain the most morphologically rich data of our full 19-language datasets. In Figure 4 we present the best performing variation for each strategy on this dataset. For almost all languages, the `iid` masking strategy performs best. There are 4 languages for which `AE` outperforms every denoising strategy. In three of these four languages either we sampled far fewer than 5k self-supervised words (amh, san), or the character vocabulary is extremely large (jap, amh). Both of these scenarios support the hypothesis that autoencoding is a strong objective under extreme data sparsity. It is unclear which factor explains the autoencoding performance in Armenian (hye). Despite the fact that our dataset contains no languages with typical prefix inflection, the `prefix` strategy also performs best in a few cases. Upon further analysis, we noticed that in the development sets, particles like pronouns are sometimes inserted before an inflected form as in the Italian example:

$$\textit{sdebitarsi} \rightarrow \textit{ti sdebitavi}. \tag{7}$$

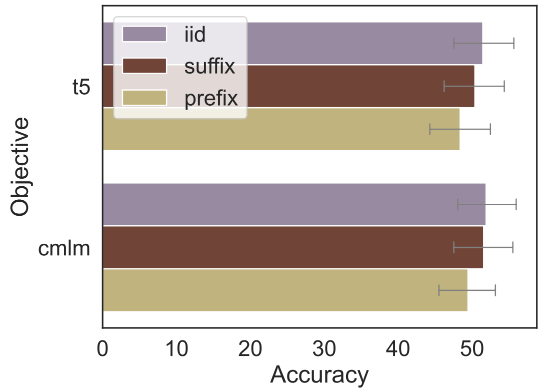
When we quantify this phenomenon, we find that in 1000 development set examples, it happens most frequently in fin (284) and ita (141), followed by hye (99) and deu (71). The German (deu) instances are caused by separable verbs, and Armenian (hye) additionally exhibits this phenomenon pervasively at the end of the string. We interpret this as evidence that skewing the mask distribution does teach the model an inductive bias towards certain typological phenomenon. The fact that many languages that exhibit a strong bias towards suffixes perform best with the `iid` strategy, however, shows that masking uniformly is typically stronger overall.

6.2 Discussion

The main results presented in this work show a clear trend between autoencoding—which performs strongly when there is little external unlabeled data—and denoising, which performs strongly as we train on a larger set of unlabeled words. This is apparent in the trend in the number of unique types per dataset in Table 1 vis-à-vis the accuracy in Table 3. We hypothesize that this is because autoencoding learns a strong inductive bias towards *copying* – a common operation in morphological inflection. On the other hand, denoising encourages models to generate particular character sequences that were masked in the input. It is thus prone to overfitting to such sequences in cases of extreme data sparsity. Denoising also encourages the model to learn to generate character sequences that are potentially morphologically rich



(a) Comparison of objective performance



(b) Comparison of strategy performance

Figure 3: Average accuracy over all ud datasets when masking.

	eng	hun	ita	rus	spa	mean
AE	<u>90.70</u>	52.00	<u>53.70</u>	68.40	55.20	64.00
t5-pref	88.70	58.60	46.90	69.80	63.30	65.46
t5-suff	89.40	66.60	50.20	67.70	57.90	66.36
t5-iid	90.10	**68.80**	47.90	70.40	**68.00**	69.04
cmlm-pref	88.70	56.90	42.10	69.50	59.90	63.42
cmlm-suff	90.50	66.70	51.90	68.50	61.00	67.72
cmlm-iid	90.60	67.30	52.60	<u>73.10</u>	66.30	<u>69.98</u>
t5-seg-pref	86.60	58.00	46.70	67.60	60.70	63.92
t5-seg-suff	89.50	58.40	45.30	71.60	56.10	64.18
t5-seg-iid	89.10	63.70	48.70	71.60	62.50	67.12
cmlm-seg-pref	89.70	57.60	**58.20**	68.60	64.20	67.66
cmlm-seg-suff	**91.00**	67.00	47.90	**73.20**	64.10	68.64
cmlm-seg-iid	90.50	<u>68.30</u>	53.20	**73.20**	<u>67.60</u>	**70.56**

Table 4: Accuracy for the languages with segmentation data. -seg- indicates the masking objective sampled morphemes, rather than characters. The best model per language is **bolded**, and the second best is <u>underlined</u>

and unavailable in the source form in an inflection (e.g., an inflectional suffix). A morphological inflection model must learn to do this competently, and with sufficiently diverse data, we hypothesize that denoising objectives encourage this capability. We are not the first to hypothesize that autoencoding benefits inflection via copy bias. Liu and Hulden (2022) proposed generating synthetic words or lemmas and training inflection models to copy them. Yang et al. (2022) confirmed that this strategy strongly increases inflection performance.

However, if we consider `cmlm-iid` with masking as a baseline denoising method, all of our attempts to change the objective with a different inductive bias can be considered negative results. Deleting, merging span masks, and skewing the positions of the mask distribution do not lead to better results overall. One hypothesis here is simply that morphology is complicated in most languages: even if suffix concatenation, for instance, is pervasive, it does not account for all of the inflectional transformations and is thus too strong of an inductive bias. Instead, we believe that `iid` sampling with masks is the best heuristic because it leads to the highest variance in training data. That is, given a small unlabeled dataset, we sample each word many times during training. A uniform sampling of masks will lead to the highest variance in word positions that are masked for the same word, and thus the most variance in our dataset. This is likely a stronger learning method than any particular inductive bias.

In subsection 6.1 we presented evidence that `prefix` masking is beneficial to inflection datasets with pervasive prefix inflection. This suggests that the auxiliary self-supervised task *does* modify model behavior in typologically interpretable ways. However, the relatively weak `suffix` results when compared to `iid` suggests that inductive bias is less beneficial than simply learning the most diverse token distribution, even in a low-resource setting.

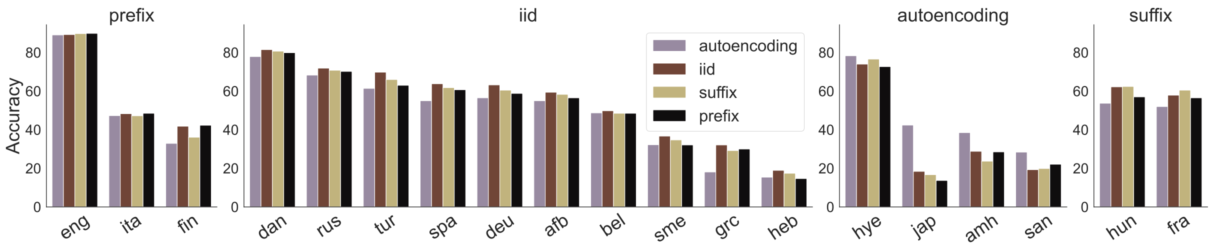


Figure 4: Language specific results for the ud-vnadj-NR dataset. Bars represent the max test accuracy of each strategy. Languages are grouped by which strategy maximized performance on this dataset.

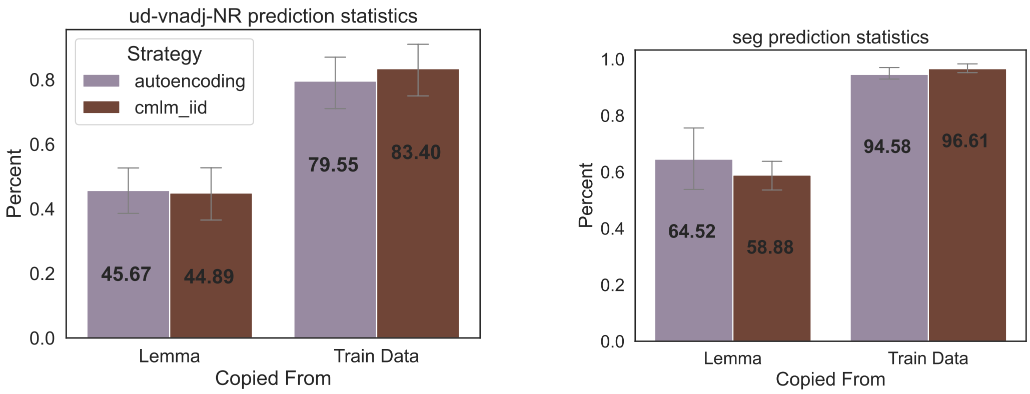


Figure 5: The percent of trigams in predicted inflections that were copied from the lemma or from the training outputs. This compares models trained with autoencoding to `cmlm-iid` with masking on two different datasets.

Segment Masking Masking morphemes rather than characters is more closely aligned to the downstream inflection task for concatenative inflections. In our experiments, we tend to see a slight positive effect from incorporating morpheme information to this end. This indicates that smarter masking *can* increase performance, though on average sampling characters uniformly is a very strong alternative. We observe two other results that are unique to the seg dataset: (i) morpheme masking *underperforms* character masking with the T5 objective (ii) deleting sometimes outperforms masking. The first result may be due to the fact that T5 on the seg dataset is most likely to have a large mismatch between the number of symbols on the source and target side. That is, an entire segment comprising many characters is replaced by a single mask symbol. For instance, if we sample the stem in the following example: <*X*>*s* + [TASK] → *recovers*. Instead, the CMLM objective would result in seven individual mask tokens that align with the stem. The fact that deletion still performs strongly for the T5 objective with segments (See: Figure 2), however, may warrant further investigation into the impact of masks. See Table 11 in the appendix for full results on the seg dataset with deletion.

7 Additional Analysis: Copying

In order to explore our hypothesis that autoencoding teaches the model to copy, whereas denoising teaches the model to generate from the training distribution, we quantify the extent to which models copy n-grams from the lemma, versus generate n-grams from the training data. To this end, we analyze predictions on the development set for the ud-vnadj-NR dataset used in the language-specific analysis. We compare the autoencoding model, and the `cmlm-iid` mask model. For each prediction, we compute the percent of character trigrams that appear in the trigrams of the input lemma in order to quantify copying. Then, we build the set of all character trigrams in the training data output forms, and compute the percent of predicted character trigrams that are attested in the training trigrams in order to quantify generating from the training distribution. In Figure 5 we compare the average percent of the two models over all languages for those predictions where the models disagree. This represents 7286 total disagreements for ud-vnadj-NR and 2511 for the seg dataset. In both datasets, autoencoding tends to copy more from the lemma and denoising tends to produce more trigrams from the training data. We interpret this as support for our hypothesis. In the segmentation dataset, the difference in copy percent is much more substantial, and the training data percent difference is much smaller. The latter effect is likely because this morphologically rich dataset contains most n-grams that a model is likely to generate. The former effect illustrates that many AE mistakes probably come from over-copying.

8 Conclusion

In this work, we compared several self-supervised tasks performed on unlabeled data as auxiliary training objectives for morphological inflection. We found that for small sets of unlabeled words, autoencoding is a strong objective due to its inductive bias towards copying form the source string. However, tasks based on MLM perform better as the unlabeled dataset grows because it encourages the model to generate new character sequences. Other denoising objectives with more inductive bias have predictable effects, but are not necessarily better than the standard MLM with uniform masking. We believe this is because uniform masking produces the most diverse dataset. Finally, we find that providing a morpheme segment oracle leads to the best results, establishing a promising direction if segments can be approximated without supervision.

Limitations

First, this study is limited with respect to the morphological typology of the languages. Because we use UD data, we filter out the SIGMORPHON 2023 shared task languages that tend to inflect with prefixes, for instance.

We also explore a limited set of simple self-supervised heuristics. Other methods may be more effective for inflection. Our segmentation dataset contains strictly concatenative languages, and it is unclear what this means for non-concatenative morphology, which tends to be understudied in the computational morphology community.

Finally, we claim that better inflection models can be useful for language documentation efforts, but we test all methods on a standard benchmark by *simulating* a low-resource scenario. Languages with data needs like those that we discuss may yield different results.

Ethics Statement

This work is partly inspired by applications of language technology to language documentation and educational tools. We note that many of these applications traditionally apply to small indigenous language communities, who may not actually desire or need such NLP technology. We thus emphasize that the methods proposed here are intended to explore the development of NLP tools under severe data constraints, but do not test the feasibility of using them in a real applied setting.

Acknowledgments

We would like to thank Abteen Ebrahimi, Stéphane Aroca-Ouellette, Ananya Ganesh, and Kyle Gorman for feedback on this work. We would also like to thank Kyle Gorman for help with the alignment algorithm for turning canonical segmentation into surface segmentation.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Kyle Gorman. 2016. Pynini: A python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

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.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Peter Makarov and Simon Clematide. 2018. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882, Brussels, Belgium. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

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

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Abhishek Purushothama, Adam Wiemerslage, and Katharina Von Der Wense. 2024. Getting the most out of your training data: Exploring unsupervised tasks for morphological inflection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18955–18970, Miami, Florida, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Adam Wiemerslage, Kyle Gorman, and Katharina von der Wense. 2024. Quantifying the hyperparameter sensitivity of neural networks for character-level sequence-to-sequence tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 674–689, St. Julian's, Malta. Association for Computational Linguistics.

Adam Wiemerslage, Changbing Yang, Garrett Nicolai, Miikka Silfverberg, and Katharina Kann. 2023. An investigation of noise in morphological inflection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3351–3365.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537.

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.

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.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Salih Furkan Akkurt, Gabrielė Aleksandravičiūtė, Ika Alfina, Avner Algom, Khalid Alnajjar, Chiara Alzetta, Erik Andersen, Lene Antonsen, Tatsuya Aoyama, Katya Aplonova, Angelina Aquino, Carolina Aragon, Glyd Aranes, Maria Jesus Aranzabe, Bilge Nas Arıcan, Ḣórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Katla Ásgeirsdóttir, Deniz Baran Aslan, Cengiz Asmazoğlu, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Mariana Avelãs, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Shabnam Behzad, Kepa Bengoetxea, İbrahim Benli, Yifat Ben Moshe, Gözde Berk, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, António Branco, Kristina Brokaitė, Aljoscha Burchardt, Marisa Campos, Marie Candito, Bernard Caron, Gauthier Caron, Catarina Carvalheiro, Rita Carvalho, Lauren Cassidy, Maria Clara Castro, Sérgio Castro, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Liyanage Chamila, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Daniela Corbetta, Francisco Costa, Marine Courtin, Mihaela Cristescu, Ingerid Løyning Dale, Philemon Daniel, Elizabeth Davidson, Leonel Figueiredo de Alencar, Mathieu Dehouck, Martina de Laurentiis, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Adrian Doyle, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Christian Ebert, Hanne Eckhoff, Masaki Eguchi, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Farah Essaidi, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Federica Favero, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Amal Fethi, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Federica Gamba, Marcos Garcia, Moa Gärdenfors, Fabrício Ferraz Gerardi, Kim Gerdes, Luke Gessler, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Takahiro Harada, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Marivel Huerta Mendez, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Artan Islamaj, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Katharine Jiang, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Ritván Karahóǧa, Andre Kåsen, Tolga Kayadelen, Sarveswaran Kengatharaiyer, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Alexey Koshevoy, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Adrian Kuqi, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Kris Kyle, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Lauren Levine, Cheuk Ying Li, Josie Li, Keying Li, Yixuan Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Yi-Ju Jessica Lin, Krister Lindén, Yang Janet Liu, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Ilya Makarchuk, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Stella Markantonatou, Héctor Martínez Alonso, Lorena Martín Rodríguez, André Martins, Cláudia Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Aaron Miller, Karina Mischenkova, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lng Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Hulda Óladóttir, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Noam Ordan, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Teresa Paccosi, Alessio Palmero Aprosio, Anastasia Panova, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Giulia Pedonese, Angelika Peljak-Łapińska, Siyao Peng, Siyao Logan Peng, Rita Pereira, Sílvia Pereira, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Andrea Peverelli, Jason Phelan, Jussi Piitulainen, Yuval Pinter, Clara Pinto, Tommi A Pirinen, Emily Pitler, Magdalena Plamada, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Robert Pugh, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andreia Querido, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Joana Ramos, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Arij Riabi, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Ivan Roksandic, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Ben Rozonoyer, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Marta Sartor, Mitsuya Sasaki, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Maria Shvedova, Janine Siewert, Einar Freyr Sigurðsson, João Silva, Aline Silveira, Natalia Silveira, Sara Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Haukur Barri Símonarson, Kiril Simov, Dmitri Sitchinava, Ted Sither, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Per Erik Solberg, Barbara Sonnenhauser, Shafi Sourov, Rachele Sprugnoli, Vivian Stamou, Steinh͡ór Steingrímsson, Antonio Stella, Abishek Stephen, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Daniel Swanson, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Mirko Tavoni, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Sara Tonelli, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sveinbjörn H͡órðarson, Vilhjálmur H͡orsteinsson, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Elena Vagnoni, Sowmya Vajjala, Socrates Vak, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Uliana Vedenina, Giulia Venturi, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Shira Wigderson, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Yilun Zhu, Anna Zhuravleva, and Rayan Ziane. 2023. Universal dependencies 2.12. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Appendix

A UD Data

When we sample from UD data, we use exactly the corpora from Purushothama et al. (2024) except for two cases. For both English and Turkish, there is not sufficient data in the existing corpora to sample 5k unique words from, so we additionally add the English-EWT corpus and the Turkish-BOUN, respectively. The English-EWT contains web data, and so we ignore tokens with special sequences "@" and "www". This process is imperfect though, so we still introduce noise into our data via this corpus.

B Surface Segmentation Data

Canonically segmented data may insert or delete characters into or from a surface word form. We thus present a simple method for attaining a surface segmentation given a surface form s and its canonical segmentation c. First, we build a Pynini `EditTransducer` (Gorman, 2016)—which is a weighted finite state transducer aligning strings—over the language's character vocabulary Σ. Then, we attain the shortest path from s to c: the alignment that produces their minimal Levenshtein distance. Finally, we segment s (i) at any insertion that maps to a segmentation boundary in c or (ii) after any character that maps to a segmentation boundary in c.

For example, given the pair $\langle$*chugged, chug-ed*$\rangle$, we attain the alignment

$$\begin{array}{ccccccc} c & h & u & g & g & e & d \\ c & h & u & g & \text{-} & e & d \end{array}$$

Then we match on rule (ii) and segment s into *chugg - ed*.

C Hyperparameters

Our architecture hyperparameters match Wu et al. (2021): We use a transformer encoder and decoder, each with 4 layers, embedding size 256, hidden size 1024, and 4 attention heads. We optimize with cross-entropy loss with label smoothing of 0.1. We train with Adam optimization (Kingma, 2014) with a learning rate of 1e-3, and a beta2 of .98. We train up to 800 epochs with a batch size of 400, and evaluate every 16 epochs. For the baseline experiments, we lower the batch size to 100 and train for up to 10k epochs with a patience of 400. We train with the Warmup Inverse Square Root schedule with 4k warmup steps. When masking unlabeled data, we always sample 25% of tokens for masking. All experiments are trained on NVIDIA GPUs and implemented with yoyodyne[1] (Wiemerslage et al., 2024).

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

	t5-prefix	cmlm-prefix	t5-suffix	cmlm-suffix	t5-iid	cmlm-iid	AE
afb	**75.80**	<u>74.50</u>	71.90	73.70	74.40	73.30	72.60
amh	59.90	62.80	61.80	62.00	60.60	<u>63.80</u>	**69.20**
bel	<u>64.00</u>	63.30	62.40	62.40	62.40	**64.90**	62.20
dan	81.70	82.20	82.90	**83.90**	<u>83.50</u>	82.70	82.30
deu	74.10	<u>77.60</u>	73.80	77.50	76.10	**79.00**	76.90
eng	88.10	**90.50**	89.00	89.50	89.40	88.50	<u>90.10</u>
fin	<u>73.90</u>	72.50	**74.60**	73.20	**74.60**	70.70	<u>73.90</u>
fra	75.80	74.10	**76.40**	<u>76.10</u>	73.70	74.70	75.00
grc	52.50	49.60	50.80	47.90	<u>53.40</u>	**55.30**	49.20
heb	76.94	77.04	<u>77.84</u>	77.24	77.64	**79.66**	76.84
hun	77.70	3.40	77.90	79.50	**80.50**	77.60	<u>79.80</u>
hye	91.30	<u>92.40</u>	91.60	92.20	**93.30**	92.30	**93.30**
ita	**92.40**	<u>92.10</u>	91.20	90.80	92.00	91.00	91.30
jap	37.80	39.10	41.50	<u>44.20</u>	37.80	<u>44.20</u>	**53.60**
rus	79.50	80.10	80.20	<u>82.20</u>	**82.40**	<u>82.20</u>	81.30
san	56.40	56.10	54.80	56.80	<u>59.10</u>	57.10	**61.10**
sme	61.00	61.30	61.10	59.60	<u>66.20</u>	64.50	**69.20**
spa	91.20	91.60	90.30	<u>92.40</u>	90.60	92.30	**93.10**
tur	84.60	<u>88.00</u>	85.10	86.10	85.70	84.90	**89.90**
Mean	73.40	69.91	73.43	74.07	74.39	<u>74.67</u>	**75.83**

Table 5: All language results for the ud-1k dataset from (Purushothama et al., 2024).

	t5-prefix	cmlm-prefix	t5-suffix	cmlm-suffix	t5-iid	cmlm-iid	AE
afb	54.30	53.30	54.30	53.10	**56.50**	52.90	<u>54.40</u>
amh	26.00	<u>29.30</u>	26.40	<u>29.30</u>	26.00	28.40	**37.10**
bel	47.00	**50.60**	45.20	<u>48.60</u>	45.20	<u>48.60</u>	44.90
dan	79.00	**80.10**	79.80	79.60	<u>79.90</u>	79.60	79.40
deu	51.40	55.60	54.40	57.90	<u>58.50</u>	**60.50**	53.90
eng	28.90	<u>51.60</u>	50.90	46.60	35.40	48.70	**75.40**
fin	**38.70**	28.40	31.60	33.30	<u>38.50</u>	30.00	28.70
fra	40.40	<u>47.50</u>	46.30	45.30	46.80	45.10	**49.70**
grc	24.00	19.50	22.50	26.00	**30.70**	<u>29.20</u>	21.60
heb	14.60	**16.01**	13.29	12.49	<u>15.11</u>	14.60	14.30
hun	49.00	55.30	**58.70**	56.60	51.80	<u>58.40</u>	56.90
hye	60.90	65.10	**73.10**	71.40	68.90	61.10	<u>72.40</u>
ita	27.10	37.00	34.20	40.10	33.30	<u>41.40</u>	**48.60**
jap	0.80	1.70	2.30	<u>3.70</u>	2.00	2.00	**21.40**
rus	66.80	66.00	63.10	66.20	66.10	**72.00**	<u>67.40</u>
san	20.10	23.50	26.50	25.50	23.20	<u>27.40</u>	**33.80**
sme	29.30	**33.20**	28.70	28.80	<u>32.10</u>	28.20	31.90
spa	53.30	51.10	49.10	54.80	54.40	**57.50**	<u>56.90</u>
tur	16.20	29.80	25.00	<u>33.20</u>	15.40	29.80	**53.40**
Mean	38.31	41.82	41.34	42.76	41.04	<u>42.92</u>	**47.48**

Table 6: All language results for the ud-200 dataset.

	t5-prefix	cmlm-prefix	t5-suffix	cmlm-suffix	t5-iid	cmlm-iid	AE
afb	54.90	54.20	56.20	55.70	**56.70**	55.60	54.60
amh	22.40	26.30	26.30	24.30	27.40	25.00	**37.80**
bel	48.80	**50.20**	43.10	48.70	47.80	48.20	49.30
dan	80.20	79.40	77.30	78.80	80.60	**80.70**	79.20
deu	54.10	57.90	52.90	58.10	57.90	**62.70**	52.90
eng	23.30	44.80	40.30	45.80	29.60	36.80	**72.70**
fin	36.80	28.70	33.70	34.00	**43.50**	38.10	32.20
fra	43.80	43.80	48.40	49.60	**52.60**	51.50	52.50
grc	23.30	20.70	21.60	26.10	26.30	**28.70**	16.90
heb	15.41	**16.72**	15.41	13.09	15.71	13.60	15.41
hun	53.20	55.80	59.80	60.10	**62.70**	58.30	53.70
hye	69.00	66.20	72.20	73.40	69.10	63.40	**73.90**
ita	38.80	43.10	42.10	43.70	44.00	45.30	**49.70**
jap	6.90	9.30	11.10	9.20	5.40	7.90	**30.80**
rus	64.30	**70.50**	65.00	68.80	67.40	70.00	67.90
san	14.00	21.00	15.90	17.70	18.90	20.30	**30.50**
sme	31.00	31.60	30.60	28.30	32.60	33.50	**33.70**
spa	52.20	45.60	55.10	56.30	**59.70**	55.40	52.20
tur	18.20	22.00	28.30	32.80	16.40	20.30	**52.20**
Mean	39.51	41.46	41.86	43.39	42.86	42.91	**47.80**

Table 7: All language results for the ud-vnadj dataset

	t5-prefix	cmlm-prefix	t5-suffix	cmlm-suffix	t5-iid	cmlm-iid	AE
afb	54.40	53.50	**57.10**	55.50	56.30	54.10	55.20
amh	25.90	34.30	32.70	33.60	32.10	34.00	**40.70**
bel	44.70	**49.50**	42.30	47.90	47.50	45.40	48.10
dan	77.00	**79.70**	77.80	79.00	78.90	78.60	79.20
deu	52.30	56.20	56.40	56.80	60.10	**63.40**	60.10
eng	40.50	58.80	49.60	52.50	37.10	50.60	**80.00**
fin	**38.40**	28.50	32.60	30.30	38.10	31.60	34.40
fra	40.30	45.20	51.40	53.50	**56.60**	54.10	53.70
grc	25.90	25.30	25.60	**30.30**	28.60	29.10	21.90
heb	15.61	15.31	14.00	13.90	16.62	13.60	**18.53**
hun	50.20	54.90	53.90	54.50	**59.30**	54.10	53.80
hye	61.30	64.80	70.60	**77.00**	72.20	70.20	74.70
ita	32.40	45.10	41.60	45.00	38.90	43.80	**46.60**
jap	3.80	5.50	9.30	9.00	8.20	7.80	**29.00**
rus	66.00	69.00	65.70	66.80	**69.40**	66.80	68.10
san	17.30	26.50	21.70	25.80	18.00	26.50	**33.30**
sme	28.90	31.00	27.80	**32.10**	26.60	30.80	31.00
spa	45.80	47.30	51.90	54.50	54.10	55.00	**57.10**
tur	18.50	29.70	36.30	36.70	19.30	26.20	**56.10**
Mean	38.91	43.16	43.07	44.98	43.05	43.98	**49.55**

Table 8: All language results for the ud-wl dataset

	t5-prefix	cmlm-prefix	t5-suffix	cmlm-suffix	t5-iid	cmlm-iid	AE
afb	56.50	54.50	<u>58.30</u>	56.80	**59.40**	55.70	55.00
amh	27.90	28.50	23.70	23.30	27.90	<u>28.80</u>	**38.50**
bel	47.60	48.50	48.40	48.50	<u>49.60</u>	**49.80**	48.70
dan	79.10	79.90	76.70	<u>80.70</u>	**81.50**	80.40	77.90
deu	58.50	58.80	60.50	58.90	**63.20**	<u>62.50</u>	56.50
eng	87.20	**89.90**	87.30	<u>89.80</u>	89.30	88.80	89.10
fin	**42.30**	35.70	36.10	35.30	<u>41.80</u>	38.80	32.90
fra	56.50	43.10	**60.50**	51.30	<u>57.90</u>	53.80	52.00
grc	30.00	26.00	29.20	28.00	**32.10**	<u>31.30</u>	18.10
heb	14.70	13.70	<u>17.42</u>	16.21	**18.93**	12.99	15.41
hun	55.60	57.00	55.30	**62.40**	61.70	<u>62.20</u>	53.70
hye	72.70	69.60	74.40	<u>76.60</u>	74.00	73.40	**78.30**
ita	39.50	**48.50**	46.60	47.20	46.70	<u>48.30</u>	47.30
jap	13.70	13.40	16.70	13.50	<u>18.40</u>	17.10	**42.40**
rus	65.90	70.20	70.30	<u>70.80</u>	69.20	**71.90**	68.30
san	13.10	<u>22.10</u>	19.90	17.10	18.60	19.30	**28.40**
sme	32.10	31.80	29.90	<u>34.70</u>	34.20	**36.70**	32.20
spa	60.70	55.60	59.30	61.80	**63.80**	<u>62.80</u>	55.00
tur	52.20	63.00	64.20	<u>66.00</u>	**69.80**	64.10	61.40
Mean	47.67	47.88	49.20	49.42	**51.48**	<u>50.46</u>	50.06

Table 9: All language results for the ud-vnadj-NR dataset.

	t5-prefix	cmlm-prefix	t5-suffix	cmlm-suffix	t5-iid	cmlm-iid	AE
afb	55.80	52.70	**56.60**	56.00	<u>56.40</u>	54.60	55.70
amh	28.30	<u>36.10</u>	33.00	34.20	31.90	35.80	**40.40**
bel	49.20	49.10	47.10	49.30	<u>49.40</u>	**51.10**	48.60
dan	**81.40**	80.00	81.00	**81.40**	<u>81.30</u>	81.20	79.60
deu	56.80	<u>63.10</u>	56.40	61.30	61.70	**64.60**	57.70
eng	87.70	87.80	86.60	88.10	88.30	**89.50**	<u>89.40</u>
fin	<u>39.30</u>	32.10	34.70	33.40	**41.90**	36.70	33.70
fra	55.70	55.20	51.50	53.90	<u>56.20</u>	**59.60**	51.40
grc	28.00	28.20	28.30	<u>31.70</u>	30.60	**32.10**	22.60
heb	<u>18.43</u>	15.01	17.22	16.31	**20.95**	17.62	17.42
hun	56.70	52.70	53.90	<u>59.80</u>	**63.10**	58.60	53.50
hye	76.10	71.90	76.10	<u>77.30</u>	**78.30**	73.40	75.30
ita	36.30	43.90	46.10	<u>49.00</u>	46.30	**50.00**	48.80
jap	11.20	11.20	14.60	13.30	<u>16.80</u>	<u>16.80</u>	**34.70**
rus	65.20	68.90	68.70	67.40	**70.80**	<u>70.20</u>	69.70
san	21.80	<u>27.80</u>	24.60	25.10	20.40	25.70	**32.30**
sme	31.80	**36.80**	33.90	<u>35.70</u>	32.90	**36.80**	33.50
spa	57.40	52.20	58.70	<u>62.60</u>	59.50	**64.30**	55.30
tur	55.70	61.70	<u>65.60</u>	63.80	**67.10**	63.30	59.70
Mean	48.04	48.76	49.19	50.51	<u>51.26</u>	**51.68**	50.49

Table 10: All language results for the ud-wl-NR dataset.

	eng	hun	ita	rus	spa	Mean
`t5-prefix`	80.10	46.60	13.90	56.50	35.30	46.48
`cmlm-prefix`	88.70	56.20	34.90	63.50	47.80	58.22
`t5-suffix`	88.50	64.50	44.00	60.50	49.80	61.46
`cmlm-suffix`	87.30	62.40	45.40	61.50	57.20	62.76
`t5-iid`	75.90	51.10	27.40	58.20	41.20	50.76
`cmlm-iid`	81.10	58.40	32.90	59.20	46.50	55.62
`t5-seg-prefix`	85.90	55.20	44.90	66.30	58.60	62.18
`t5-seg-suffix`	<u>89.60</u>	61.60	<u>51.90</u>	69.20	59.30	66.32
`t5-seg-iid`	89.20	<u>66.00</u>	48.60	<u>71.60</u>	<u>61.60</u>	**67.40**
`cmlm-seg-prefix`	85.00	56.90	**58.20**	65.40	58.40	64.78
`cmlm-seg-suffix`	**90.00**	65.60	44.20	**73.20**	59.40	66.48
cmlm-seg-iid	89.00	**66.10**	47.80	69.60	**61.70**	<u>66.84</u>

Table 11: All language results for the segmentation dataset when deleting rather than masking.