Corpus Frequencies in Morphological Inflection: Do They Matter?

Tomáš Sourada*, Jana Straková

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Prague, Czech Republic

Abstract

The traditional approach to morphological inflection (the task of modifying a base word (lemma) to express grammatical categories) has been, for decades, to consider lexical entries of lemma-tag-form triples uniformly, lacking any information about their frequency distribution. However, in production deployment, one might expect the user inputs to reflect a real-world distribution of frequencies in natural texts. With future deployment in mind, we explore the incorporation of corpus frequency information into the task of morphological inflection along three key dimensions during system development: (i) for train-dev-test split, we combine a lemma-disjoint approach, which evaluates the model's generalization capabilities, with a frequency-weighted strategy to better reflect the realistic distribution of items across different frequency bands in training and test sets; (ii) for evaluation, we complement the standard *type accuracy* (often referred to simply as *accuracy*), which treats all items equally regardless of frequency, with *token accuracy*, which assigns greater weight to frequent words and better approximates performance on running text; (iii) for training data sampling, we introduce a method novel in the context of inflection, *frequency-aware* training, which explicitly incorporates word frequency into the sampling process. We show that frequency-aware training outperforms uniform sampling in 26 out of 43 languages.

Keywords

Morphological inflection, Frequency-weighting, Frequency-weighted split, Frequency-weighted sampling, Frequency-weighted evaluation

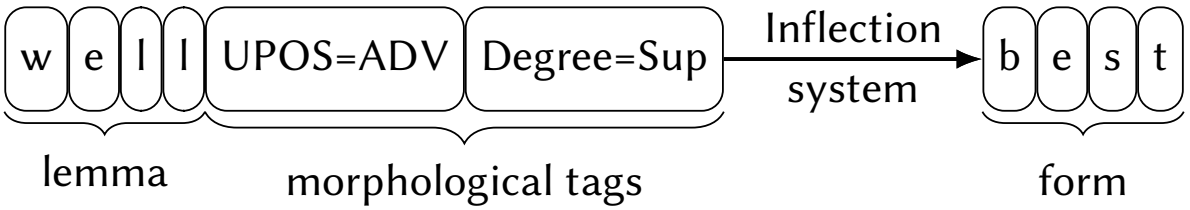


Figure 1: The morphological inflection task: an input-output example, inflection of English lemma "well" to superlative, inflected form "best" (figure adopted from Sourada and Straková [1]).

1. Introduction

Morphological inflection, the task of producing the inflected form from a base word form (lemma) to express grammatical categories (see Figure 1), has become a highly active research area in recent years, largely driven by the SIGMORPHON shared tasks held from 2016 to 2023 [2, 3, 4, 5, 6, 7, 8]. The traditional approach to inflection has been to treat the lexical entries of lemma-tag-form triples uniformly, without any information about their frequency distribution. This is natural when a morphological lexicon (e.g., UniMorph (UM) [9]) is used, which is a prevalent approach.

However, in production deployment of a morphological inflection system, we may expect the user input to reflect a real-world distribution of frequencies: users will probably enter generally frequent

ITAT'25: Information Technologies – Applications and Theory, September 26–30, 2025, Telgárt, Slovakia
*Corresponding author.
sourada@ufal.mff.cuni.cz (T. Sourada); strakova@ufal.mff.cuni.cz (J. Straková)
0009-0003-6792-825X (T. Sourada); 0000-0003-0075-2408 (J. Straková)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

words more commonly than rare words. This fact has been ignored for decades until quite recently [10, 1].

With a focus on future deployment, we address this gap by exploring the incorporation of corpus frequency information into the task of morphological inflection along three key axes: train-dev-test split, data sampling during training, and evaluation, experimenting with Universal Dependencies (UD) [11] corpora in 43 languages.

Our contributions are as follows:

1. We experiment with a new train-dev-test split technique that satisfies both recent methodological standards (being lemma-disjoint [12, 8]) and practical requirements for real-world deployment by ensuring a realistic distribution of items with varying corpus frequencies through frequency-weighted sampling [10, 1].
2. Throughout the experiments, we complement the standard type accuracy evaluation with token accuracy [13], a previously proposed but never used metric that better reflects the deployment conditions by assigning greater importance to frequent words.
3. Furthermore, motivated by the focus on frequent words in evaluation, we conduct pioneering experiments with *frequency-aware* training; directly incorporating the importance (corpus occurrence counts) of training data items (lemma-tag-form triples) during training by frequency-weighted sampling. Our results indicate that this approach is promising for the majority of the languages.

We make all our work open-source by publicly releasing our code at GitHub: https://github.com/tomsouri/corpus-frequencies-in-inflection.

2. Related Work

In the field of morphological inflection, there has been a lot of recent work, mostly pushed forward by the SIGMORPHON shared tasks [2, 3, 4, 5, 6, 7, 8]. The shared tasks leveraged the UniMorph database [14, 15, 9], an ever-growing morphological lexicon. The data used in the shared tasks, as well as in other relevant works, are used in the form of a lexicon of unique lemma-tag-form triplets, without any notion of their frequency: the train, dev and test sets are simple lists of the triplets.

2.1. Train-Dev-Test Data Split

Until 2021, common practice was to split the lexicon of lemma-tag-form triplets into train, dev, and test uniformly randomly, without any additional control over the lemma overlap or feature overlap between train and test. Liu and Hulden [16] showed that lemma overlap artificially inflates the model's performance. Goldman et al. [12] investigated this problem more and suggested a lemma-disjoint split technique, which was then adopted by the next installation of the SIGMORPHON shared task [8].

Finally, Kodner et al. [10] have focused on different split practices and thoroughly evaluated and discussed their properties. In addition to the issues with overlap control, he also revealed drawbacks of standard *uniform sampling*. Uniform sampling over a set of triplets lemma-tag-form means that we sample them uniformly randomly into train, dev, and test, regardless of their corpus occurrence frequencies.[1] According to Kodner et al. [10], this leads to an unnatural train-test split, with a bias towards low-frequency and thus more regular and reliable training items. To mitigate this problem, they suggest a new, *frequency-weighted* split strategy: the types (lemma-tag-form triples) are partitioned at random weighted by their frequency in a corpus. The train set is sampled first, then the dev+test is sampled and then uniformly separated. This sampling biases the train set towards items with high corpus frequency, and dev and test towards low-frequency items, leading to a more realistic train-test distribution. Sourada and Straková [1] have built on this and proposed a new split technique

[1]This is the only approach one may use if working with a morphological lexicon that only contains the lemma-tag-form triplets and lacks the information about their frequency in natural text.

that combines the frequency-weighted split with the lemma-disjoint approach. We employ this new technique, as the realistic train-test frequency distribution fits better our needs, and lemma-disjoint split allows evaluating generalization abilities.

2.2. Evaluation

When evaluating morphological inflection, exact-match accuracy is used, as the ratio of correctly predicted forms (also called *type accuracy* or simply *accuracy*, see Equation (1)), on a lexical test set, not considering the corpus frequencies of the items.

$$\text{type acc} = \frac{1}{\text{total items}} \cdot \sum_{\substack{\text{correct} \\ \text{items}}} 1 \tag{1}$$

Nicolai et al. [13] proposed to use the so called *token accuracy* (see Equation (2)), where the items are weighted by their corpus frequency.

$$\text{token acc} = \frac{1}{\text{total occurrences}} \sum_{\substack{\text{correct} \\ \text{items}}} \text{occurrences(item)} \tag{2}$$

They argue that although type accuracy is easy to compute from the morphological lexicons, it "*may over-represent rare forms, which are often regularly inflected and thus simpler to produce*" [13, Section 7]. However, they conclude that computing token accuracy is infeasible because it would require a corpus of running text annotated for morphological inflection, and they themselves do not evaluate using that metric, but only its approximation. To our knowledge, no further work on morphological inflection used the token accuracy for the evaluation.

To meet the standards, we report traditional type accuracy and also include the proposed token accuracy, which better reflects real-world inflection system usage.

2.3. Frequency-Aware Training

To the best of our knowledge, we are not aware of previous relevant work in morphological inflection that incorporates corpus frequencies in (supervised) machine learning context, especially a neural-network-based one.

3. Methodology

With the focus on future deployment, we decide to perform the frequency-weighted train-dev-test split (for a realistic train-test distribution), evaluate with token accuracy, which corresponds to evaluating on a running text and therefore reflects a real-world usage, and experiment with frequency-aware training.

3.1. Data Selection

To be able to experiment with the techniques using corpus frequency information, we need to first have that information. The commonly used data source, UniMorph (UM), is a lexicon and completely lacks information on frequency.

We consider several possible approaches to obtain morphological data (lemma-tag-form triples) along with their corpus frequencies:

1. Use a manually, morphologically annotated corpus of running text, such as Universal Dependencies (UD) [17] directly as the data source. This approach has the advantage that it is simple and does not require any matching or tag conversion. Furthermore, if UD is used as the annotated corpus, the morphological tags are already in the CoNLL-U format, which is suitable for connecting the system into pipelines with other systems that work with UD (for example, UDPipe [18]). However, using UD as the only source of morphologically annotated data could lead to

lower coverage of lemmas/forms, compared to using UniMorph. This approach was used by Sourada and Straková [1], and a variation of it (specifically intersecting an annotated corpus with UniMorph and thus enriching UniMorph with corpus frequencies) was used by Kodner et al. [10].

2. Use a raw corpus of running text and align it with UM, while dropping UM entries with corpus count = 0, that is, not covered by the corpus. This would lead to slightly higher coverage of lemmas/forms should a large corpus be used. However, for a proper alignment, we need to resolve ambiguities of two types: a form equal by string equality, corresponding to two different morphological categories of the same lemma (e.g., form "*hradu*" as both dative/locative of lemma "*hrad*" (*castle* in Czech)), or even corresponding to two different lemmas (e.g., form "*ženu*" as form of lemma "*žena*" (*woman* in Czech) or as form of lemma "*hnát*" (*to hurry* in Czech)). For resolving such ambiguities, we consider several possible approaches:
   a) divide counts evenly between ambiguous forms (used by Kodner et al. [10]), which is problematic, because in reality, the counts are uneven;
   b) use a tag-based back-off, either choosing the more frequent tag in general, no matter the lemma, or dividing the counts proportionally to the overall distribution of the tags, which is still not realistic, but it may work as a good approximation;
   c) use a tagger (a morphological analyzer, such as UDPipe [18]) to automatically annotate the raw corpus, which is problematic because of high computational demands to run the tagger on large running texts, and also brings a potential leak of test data, as UDPipe was trained on a part of UD from which some lemmas are in our test set.

   Nevertheless, none of these techniques to resolve the ambiguities removes the problem of still having a high number of UM items with zero corpus occurrences and thus throwing away a lot of data.
3. Use option 2) and do not drop UM items with 0 count: Since we work with corpus counts as numbers, we could assign some constant (less than 1) as the corpus count to all theoretically possible forms (those present in UM) that are not found in the corpus of running text.

After carefully considering the advantages and drawbacks of each method, we select the first method, using the annotated corpus of running text (UD) directly as the data source, as it is the most straightforward approach, while keeping in mind that it leads to lower coverage of lemmas/forms, and we leave the other approaches for future work.

From Universal Dependencies, we select corpora for 43 languages,[2] specifically, Indo-European languages written in the Latin script and languages spoken in Europe. We lexicalize them by extracting unique lemma-tag-form triples along with their occurrence counts for training and evaluation. For hyperparameter tuning, we select a subset of five development languages: Czech, English, Spanish, Breton, and Basque.

3.2. Train-Dev-Test Split

The canonical split of UD is not suitable for the task of morphological inflection, because a naive extraction of lemma-tag-form triples from the canonical UD split naturally produces a train-test overlap, both on the item level (lemma-tag-form triples), and on the lemma level. Therefore, we need to resplit the data. We aim for potential future deployment, therefore, we need a frequency-weighted split, as advocated by Kodner et al. [10], for a realistic train-test frequency distribution. However, their proposed split is not lemma-disjoint and thus breaks recent methodological requirements [12, 8] by obscuring an evaluation of the model's generalization abilities. To meet both requirements, we employ a new split technique proposed by Sourada and Straková [1], which combines the lemma-disjoint approach with the frequency-weighted approach. It consists of the following steps:

1. obtain the total occurrence count for each lemma in the corpus,

[2] See Table 4 for the selected languages and corpora.

hyperparam	value	hyperparam	value	hyperparam	value
attn drop	0.1	L2-norm	0.01	batch size	512
drop	0.15	attn heads	4	learning rate	0.001
activation drop	0.35	layers	3	max epochs	960
layer drop	0.2	layer dim	256	checkpoint sel	on
grad-clip-norm	1.0	ffnn dim	64	decay	cosine

Table 1
Hyperparameters as tuned on the five development languages and used for all experiments. The model is trained with Adam. The checkpoint is selected based on dev performance (token accuracy) for each language separately.

2. sample lemmas first into the train set, randomly, weighted by the occurrence counts (until we obtain a desired amount of data in the train set),
3. sample lemmas from the rest uniformly randomly (no weighted sampling) to the dev set (until a desired amount of data in the dev set is achieved),
4. put the rest of lemmas into the test set.

We use a train:dev:test ratio 8:1:1 in terms of total occurrence counts.

3.3. Evaluation

For evaluating the inflection systems, we use the standard *type accuracy* (also called simply *accuracy*, see Equation (1)), working at the level of a lexicon, and complement it with *token accuracy* (see Equation (2)) [13], which gives more weight to frequent words.

In its natural interpretation, type accuracy corresponds to evaluation on a lexicon of unique lemma-tag-form triples (ignoring frequencies), while token accuracy corresponds to evaluation over a running text annotated with lemmas and tags (naturally reflecting frequencies). Therefore, token accuracy better reflects real-world usage of an inflection system. Clearly, token accuracy is better suited for distinguishing between two models when one is superior in predicting more frequent words — a situation that type accuracy cannot detect.

3.4. Model

For model training, we use a state-of-the-art, small-capacity, encoder-decoder, sequence-to-sequence Transformer architecture, trained from scratch on the inflection data, with the lemma-tag pair as input and the inflected form as output (see Figure 1), as standard in the field. We tune the model's hyperparameters on the five development languages (see Table 1 for the final values). The tuned model (without frequency-aware training) outperforms both SIGMORPHON neural baselines [19] in all five development languages.

3.5. Frequency-Aware Training

As we give more weight to frequent words in evaluation with token accuracy, one might ask whether it could be beneficial to give more weight to such words also in training. To support this, we introduce sample weights and use weighted random sampling to sample training data items into batches.

If we wanted to mimic training on a running corpus, the direct approach would be to use raw frequency counts of the data items as sample weights. Another specific case is setting all sample weights to 1, which would lead to uniform sampling, which is a standard approach in morphological inflection. For a continuous scale between these two extremes, we introduce a *corpus-frequency temperature*, denoted as τ, a real-valued hyperparameter, usually $\tau \in [0, 1]$. For training data T, denote by $c_T(A)$ the corpus occurrence count for a data item (lemma-tag-form triple) A. Let M_T be the sum of $c_T(B)$ on all lemma-tag-form triples B (the total number of lemma-tag-form triples in training data before aggregation, which can be viewed as the original corpus length in tokens) and let V_T be the number of

unique lemma-tag-form triples in training data. The sample weight for weighted random sampling is computed from the raw corpus count and the corpus-frequency temperature value τ:

$$w_T(A) = (c_T(A))^\tau \tag{3}$$

The probability that the training data item A will be sampled in an individual sampling step is given by the sample weight with normalization:

$$p(A) = \frac{(c_T(A))^\tau}{\sum_B (c_T(B))^\tau} \tag{4}$$

The ratio of the probabilities of two data items A and B being sampled in a single sampling step is:

$$\frac{p(A)}{p(B)} = \left(\frac{c_T(A)}{c_T(B)}\right)^\tau \tag{5}$$

3.5.1. Specific Values of Corpus-Frequency Temperature and Their Interpretation

With $\tau = 1$, sample weights are equal to the raw corpus frequencies, and the probability that the training data item A will be sampled in an individual sampling step is then

$$p_{\tau=1}(A) = \frac{c_T(A)}{\sum_B c_T(B)} = \frac{c_T(A)}{M_T} \tag{6}$$

The value $\tau = 0$ corresponds to the uniform sample weights, and the probability that any specific data item will be sampled in an individual sampling step is then

$$p_{\tau=0}(A) = \frac{1}{V_T} \tag{7}$$

With $\tau = 1/2$, sample weights are equal to the square root of raw corpus frequencies. Then item A with raw occurrence count 400 would be sampled with $\sqrt{400} = 20$ times higher probability than item B with raw occurrence count 1 (with sample weight $\sqrt{1} = 1$).

Corpus-frequency temperature values from the interval $[0, 1]$ correspond to the continuous scale between the two extremes (uniform sample weights ignoring the corpus frequencies with $\tau = 0$, and sample weights equal to the raw corpus frequencies $\tau = 1$). Nevertheless, it is technically possible to set it also to values outside the range, and it also has a natural meaning.

The value $\tau = -1$ leads to sample weights being the inverse of the raw corpus counts. Then the ratio of the probabilities of two data items A, B being sampled in a single sampling step is

$$\frac{p_{\tau=-1}(A)}{p_{\tau=-1}(B)} = \frac{c_T(B)}{c_T(A)} \tag{8}$$

That is, less frequent data items are sampled more frequently during training. The value $\tau = 2$ leads to extremely emphasizing the frequent data items. In the example of $c_T(A) = 400$, $c_T(B) = 1$, with $\tau = 2$, A would be sampled 160,000 times more frequently than B during training.

4. Experiments and Results

To measure the effect of frequency-aware training, we experiment with the tuned Transformer model (as described in Section 3.4), trained for 960 epochs with checkpoint selection based on token accuracy on the dev set, with different values of corpus-frequency temperature. Our hypothesis is that giving more importance to frequent words during training could improve the performance of the system when evaluating with token accuracy, which itself gives more importance to frequent words in evaluation.

We experiment with different values of corpus-frequency temperature (τ) from the range $[0, 1]$. For experimental purposes, we also try τ values 1.1 and 2, which lead to over-emphasizing the frequent items during training, and negative values of τ, which go directly against our objective (giving more importance to frequent words), as they lead to focusing more on rare words during training.

4.1. Effect of Corpus-Frequency Temperature Value τ on Token Accuracy

In Table 2, we report the performance comparison in terms of token accuracy on the dev set with different corpus-frequency temperature values for each language separately.

We observe that $\tau = 2.0$ leads to a complete failure,[3] probably because of the extreme emphasis on the frequent words. When we focus on $\tau \in [-1, 1]$, we can observe a clear tendency in some languages (Basque, Old French, Turkish): the higher the temperature value, the better the performance. In a larger set of languages we can see more or less consistent improvement when increasing τ from -1 to 0.5 (or another value close 0.5), followed by a drop, again more or less consistent: Belarusian, Bulgarian, Croatian, Dutch, German, Gothic, Irish, Italian, Latin, Polish, Pomak, Portuguese, Russian, Spanish, and Welsh. In other languages, frequency-aware training does not seem to have a major effect, as the best performance is achieved with temperature around 0 (uniform sampling): Afrikaans, Catalan, Czech, Danish, Estonian, Finnish, French, Greek, Hungarian, Icelandic, Latvian, Manx, Norwegian, Romanian, Scottish Gaelic, Slovenian and Swedish.

Quite interestingly, there are also languages in which negative values of corpus-frequency temperature (which give more weight to rare forms during training) lead to the best performance, such as English, Galician, and Slovak. In other languages, such as Breton, Lithuanian, Low Saxon, or Ukrainian, the performance at different temperature levels is quite noisy and does not provide any insight.

On average in all languages (rightmost column in the bottom part of Table 2), there is a continuous improvement when the temperature increases to 0.5, and then a continuous decline, indicating that if we shall select a single temperature value to be used in all languages, it would be $\tau = 0.5$ (sample weights equal to the square root of the raw corpus occurrence counts).

To sum up, for almost half of the languages the performance corresponds to our hypothesis that giving more importance to frequent words during training should increase the performance of the system in terms of token accuracy: either steadily with the best results achieved with $\tau = 1$, weights being the raw corpus occurrence counts (Basque, Old French, Turkish), or rather up to the square root of occurrence counts ($\tau = 0.5$). On the other hand, there are languages where frequency-aware training does not seem to have a major effect, and even some with opposite results (English, Galician, Slovak).

4.2. Effect of Corpus-Frequency Temperature Value τ on Type Accuracy

In Table 3, we report a similar comparison, now evaluated with type accuracy (which disregards the corpus occurrence counts of the evaluation items, treating all items uniformly). Although the results appear to be slightly more noisy than with token accuracy, we observe similar trends in most of the languages: the continuous improvement in Basque and Turkish when increasing the temperature value to 1.0, and the languages with a peak close to $\tau = 0.5$ (Belarusian, Dutch, German, Irish, Italian, Latin, Low Saxon, Portuguese, Scottish Gaelic, and Welsh). This is quite surprising: why should we benefit from giving more importance to frequent words in training if we disregard the frequencies in evaluation? We discuss it further in Section 5.

4.3. Token vs. Type Accuracy

Another interesting observation is that although token accuracy and type accuracy differ in the absolute value (compare English with $\tau = 0.0$ with 96.22% in token accuracy and 93.37% in type accuracy), they seem to be similar in terms of model ranking, at least when comparing systems trained with the same hyperparameters with the only difference in temperature value (and it is a trend we observed also during the development experiments with hyperparameter tuning).

To further investigate whether using type accuracy as the main objective during the experiments could actually affect the final performance in token accuracy (our primary objective for deployment of the system), we conduct the following experiment: we first train a system, selecting the checkpoint based on token accuracy, and evaluate using token accuracy. Then, we train another system, selecting

[3] With the only exception of Old French, where $\tau = 2.0$ leads to the best performance.

Table 2

Dev results, reported **token accuracy** (%) for different corpus-frequency temperature values (τ). Macro average over all languages is reported in the rightmost column in the bottom part of the table. A separate color scale was applied to each language, shading from dark red (lowest) through white (median) to dark green (highest).

τ	Afrikaans	Basque	Belarusian	Breton	Bulgarian	Catalan	Croatian	Czech	Danish	Dutch	English
-1.0	85.61	87.51	88.14	70.40	92.67	95.60	92.65	97.67	90.18	78.36	96.34
-0.8	85.54	87.89	88.34	70.13	92.67	95.81	92.82	97.72	90.39	78.97	96.38
-0.6	85.65	88.20	88.71	70.10	92.72	95.94	92.90	97.81	90.44	79.34	96.19
-0.5	85.72	88.47	88.81	70.67	92.81	96.19	93.06	97.78	90.32	79.65	96.39
-0.4	86.08	88.59	89.01	70.97	92.92	96.41	93.06	97.81	90.53	79.94	96.38
-0.3	85.89	88.86	89.09	70.63	92.97	96.41	93.40	97.83	90.35	80.11	96.33
-0.2	86.30	89.02	89.34	70.20	93.02	96.48	93.52	97.81	90.69	80.52	96.27
-0.1	86.34	89.40	89.58	70.83	93.02	96.60	93.56	97.88	90.45	80.85	96.13
0.0	86.12	89.50	89.51	70.27	93.11	96.73	93.72	97.88	90.58	80.96	96.22
0.1	86.63	89.70	89.71	71.00	93.10	96.82	93.74	97.94	90.36	81.05	96.21
0.2	86.66	89.91	89.78	70.43	93.15	96.84	93.72	97.89	90.49	81.15	96.22
0.3	86.37	90.01	89.84	70.23	93.12	96.75	93.86	97.87	90.47	81.35	96.25
0.4	86.59	89.93	89.87	69.90	93.18	96.77	93.76	97.83	90.33	81.36	96.17
0.5	86.34	89.89	90.01	70.70	93.10	96.79	93.78	97.79	90.38	81.54	96.25
0.6	86.47	90.15	89.92	69.87	93.06	96.73	93.77	97.75	90.15	81.36	96.07
0.8	85.74	90.17	89.92	70.27	92.95	96.49	93.68	97.71	89.81	81.00	95.71
1.0	83.87	90.27	89.61	69.00	92.68	96.01	93.74	97.46	88.94	79.66	95.39
1.1	82.53	90.15	89.19	68.53	92.11	95.42	93.34	97.10	88.90	78.23	95.11
2.0	41.00	25.13	10.36	17.77	4.76	7.29	9.40	7.13	20.45	17.36	49.67

τ	Estonian	Finnish	French	Galician	German	Gothic	Greek	Hungarian	Icelandic	Irish	Italian
-1.0	89.71	91.73	96.18	95.40	89.50	72.31	83.63	93.88	74.23	85.32	94.37
-0.8	89.87	91.75	96.43	95.45	89.61	72.46	84.08	93.96	73.92	85.62	94.77
-0.6	89.99	91.82	96.43	95.92	89.75	72.73	84.07	94.03	74.13	85.85	95.07
-0.5	90.09	92.00	96.44	95.79	89.84	72.59	84.31	93.90	74.55	85.91	95.29
-0.4	90.24	91.96	96.72	95.78	89.81	73.54	84.31	93.93	74.75	85.88	95.40
-0.3	90.21	92.09	96.58	95.87	89.82	73.80	84.35	93.85	75.11	86.46	95.53
-0.2	90.39	91.95	96.72	95.82	89.85	74.21	84.49	93.84	75.22	86.47	95.72
-0.1	90.34	92.06	96.78	95.77	89.92	74.17	84.57	93.96	75.28	86.48	95.88
0.0	90.31	92.21	96.78	95.71	89.96	74.35	84.61	93.87	75.51	86.67	95.96
0.1	90.71	92.19	96.89	95.61	90.02	74.69	84.49	94.05	74.59	86.67	95.80
0.2	90.49	92.15	96.90	95.44	90.02	74.77	84.38	94.04	75.19	86.81	95.97
0.3	90.61	92.12	96.86	95.75	90.10	74.35	84.61	94.00	75.11	86.91	96.09
0.4	90.55	92.12	96.77	95.79	90.04	74.52	84.59	93.71	74.76	87.09	95.95
0.5	90.43	92.18	96.87	95.77	90.07	74.23	84.45	93.92	74.81	87.03	95.97
0.6	90.32	92.08	96.82	95.66	90.16	74.97	84.00	93.70	74.94	86.87	96.04
0.8	90.12	92.07	96.74	95.62	90.00	74.16	83.99	93.58	74.68	86.94	95.89
1.0	89.57	91.80	96.26	95.32	89.76	72.06	83.40	93.27	72.99	86.47	95.53
1.1	88.63	91.34	96.04	94.87	89.46	72.28	82.82	92.64	73.20	85.81	95.18
2.0	10.61	9.04	10.41	26.41	18.43	24.17	27.10	10.26	29.73	22.19	9.78

τ	Latin	Latvian	Lithuanian	Low Saxon	Manx	Norwegian	Old French	Polish	Pomak	Portuguese	Romanian
-1.0	76.17	96.82	93.90	55.51	68.71	93.71	5.07	93.70	48.10	96.07	93.47
-0.8	77.92	96.82	93.98	56.14	69.01	93.84	4.86	93.84	48.96	96.12	93.88
-0.6	78.65	96.88	93.98	56.39	69.50	93.94	5.34	93.88	50.22	96.26	93.64
-0.5	79.42	96.84	93.92	56.71	69.53	94.00	7.46	93.86	49.44	96.26	93.70
-0.4	79.74	96.86	93.89	56.86	69.37	94.16	4.80	94.01	50.08	96.33	93.89
-0.3	80.90	96.94	93.84	56.89	70.21	94.17	5.45	93.97	50.86	96.36	94.00
-0.2	80.49	96.91	94.08	56.64	70.38	94.16	6.01	94.01	51.35	96.40	93.95
-0.1	80.86	96.86	94.02	57.07	70.45	94.22	7.77	93.96	51.19	96.50	93.95
0.0	81.38	96.87	93.94	57.01	70.56	94.07	5.43	93.99	51.20	96.51	94.04
0.1	81.81	96.94	93.99	56.95	70.11	94.18	11.72	93.96	51.71	96.63	94.06
0.2	81.97	96.85	93.86	57.40	69.82	94.08	13.47	93.90	51.93	96.55	93.88
0.3	81.93	96.88	94.03	57.77	69.42	94.12	14.95	94.00	51.74	96.69	93.85
0.4	82.59	96.89	94.09	57.63	69.38	94.16	17.76	94.03	52.60	96.67	93.78
0.5	82.17	96.84	93.99	57.25	69.77	94.00	19.58	94.03	52.56	96.61	93.97
0.6	82.18	96.81	93.89	57.86	68.87	93.86	20.68	94.01	52.80	96.52	93.87
0.8	81.27	96.67	93.84	57.58	67.10	93.39	21.25	93.88	52.33	96.44	93.57
1.0	80.17	96.43	93.38	56.82	62.97	92.71	21.93	93.59	52.65	96.14	93.10
1.1	79.77	96.32	93.23	57.30	61.44	91.90	22.31	93.49	51.76	96.11	92.66
2.0	32.28	5.12	13.83	17.65	7.99	21.04	22.78	1.51	13.81	3.40	15.95

τ	Russian	Sanskrit	Scottish Gaelic	Slovak	Slovenian	Spanish	Swedish	Turkish	Ukrainian	Welsh	**macro avg**
-1.0	93.53	76.45	63.29	94.73	95.56	96.65	88.75	83.91	93.28	88.23	**84.58**
-0.8	93.74	77.26	63.53	94.80	95.64	96.96	88.79	83.93	93.40	88.38	**84.80**
-0.6	93.91	77.09	64.47	94.90	95.62	97.23	89.15	84.30	93.26	88.60	**85.00**
-0.5	94.09	77.76	64.30	94.79	95.71	97.27	89.08	84.41	93.35	88.75	**85.14**
-0.4	94.13	77.68	64.74	94.91	95.80	97.43	89.23	84.59	93.40	88.71	**85.22**
-0.3	94.27	77.85	65.66	94.91	95.72	97.39	89.17	84.64	93.26	88.64	**85.36**
-0.2	94.26	78.30	65.55	94.95	95.76	97.53	89.54	84.75	93.36	88.89	**85.47**
-0.1	94.43	78.44	66.05	94.86	95.95	97.52	89.35	84.73	93.29	88.82	**85.58**
0.0	94.56	78.94	66.43	94.83	95.91	97.60	89.35	84.82	93.38	88.69	**85.58**
0.1	94.55	78.88	66.43	94.87	95.76	97.63	89.19	84.89	93.30	89.01	**85.78**
0.2	94.63	79.28	67.15	94.84	95.78	97.61	89.26	85.14	93.39	88.85	**85.86**
0.3	94.62	78.90	67.07	94.74	95.80	97.69	89.45	85.09	93.37	89.04	**85.90**
0.4	94.60	79.12	66.89	94.71	95.83	97.63	89.18	85.08	93.40	89.12	**85.97**
0.5	94.64	79.31	66.94	94.75	95.75	97.64	89.12	85.23	93.18	89.06	**86.02**
0.6	94.60	79.27	66.73	94.80	95.72	97.43	89.14	85.23	93.34	88.73	**85.98**
0.8	94.45	79.43	65.92	94.68	95.65	97.27	89.03	85.37	93.27	89.00	**85.78**
1.0	94.08	78.81	64.47	94.46	95.41	96.87	88.08	85.56	93.32	88.28	**85.17**
1.1	93.70	78.64	63.17	94.22	95.27	96.57	87.77	85.53	92.99	87.63	**84.76**
2.0	1.33	58.25	26.72	2.21	3.57	7.28	31.39	12.51	1.91	45.22	**17.54**

Table 3

Dev results, reported **type accuracy** (%) for different corpus-frequency temperature values (τ). Macro average over all languages is reported in the rightmost column in the bottom part of the table. A separate color scale was applied to each language, shading from dark red (lowest) through white (median) to dark green (highest).

τ	Afrikaans	Basque	Belarusian	Breton	Bulgarian	Catalan	Croatian	Czech	Danish	Dutch	English
-1.0	82.82	86.37	87.47	64.41	91.13	93.48	91.95	97.35	89.44	75.71	93.44
-0.8	82.85	86.66	87.65	64.27	91.08	93.56	91.87	97.40	89.75	76.09	93.47
-0.6	82.86	86.81	87.92	64.22	91.17	93.68	92.04	97.45	89.70	76.39	93.25
-0.5	83.01	87.22	87.96	64.61	91.13	93.79	92.14	97.44	89.52	76.63	93.65
-0.4	83.78	87.13	88.11	64.61	91.31	93.82	92.13	97.45	89.84	76.65	93.56
-0.3	83.49	87.47	88.15	64.70	91.27	93.78	92.26	97.51	89.68	76.74	93.54
-0.2	83.82	87.40	88.34	64.27	91.49	93.72	92.32	97.42	89.91	77.13	93.55
-0.1	83.70	87.74	88.51	65.18	91.46	93.93	92.19	97.46	89.69	77.18	93.19
0.0	83.41	87.63	88.41	64.70	91.55	93.91	92.45	97.48	89.94	77.35	93.37
0.1	83.93	87.85	88.67	65.61	91.50	94.02	92.35	97.58	89.69	77.47	93.31
0.2	84.07	88.09	88.67	65.13	91.51	94.03	92.40	97.47	89.78	77.63	93.21
0.3	83.94	88.07	88.66	64.70	91.52	93.90	92.55	97.47	89.76	77.85	93.47
0.4	83.94	88.01	88.80	64.42	91.55	93.96	92.45	97.50	89.47	77.79	93.40
0.5	83.96	88.03	88.93	65.09	91.51	93.85	92.42	97.42	89.62	77.90	93.28
0.6	84.05	88.35	88.79	64.13	91.41	93.70	92.33	97.40	89.43	77.78	93.14
0.8	82.89	88.25	88.79	64.61	91.32	93.41	92.21	97.29	88.95	77.23	92.41
1.0	81.29	88.46	88.46	63.27	90.90	92.64	92.30	97.02	87.94	75.61	92.01
1.1	80.00	88.25	88.02	62.64	90.29	91.69	91.77	96.52	87.70	73.84	91.54
2.0	32.44	19.47	7.11	13.03	3.64	4.87	6.93	3.34	14.53	14.02	39.88

τ	Estonian	Finnish	French	Galician	German	Gothic	Greek	Hungarian	Icelandic	Irish	Italian
-1.0	88.51	90.53	95.15	94.26	88.99	69.56	81.26	93.18	70.90	82.64	93.75
-0.8	88.67	90.57	95.32	94.20	89.07	69.72	81.64	93.29	71.41	83.00	93.70
-0.6	88.92	90.58	95.24	94.59	89.29	70.16	81.59	93.39	71.35	83.17	94.15
-0.5	88.87	90.78	95.34	94.51	89.30	70.25	81.73	93.18	71.58	82.97	94.26
-0.4	89.07	90.73	95.42	94.58	89.29	70.65	81.85	93.28	71.79	82.85	94.36
-0.3	88.97	90.86	95.22	94.64	89.28	71.10	81.88	93.17	72.18	83.41	94.66
-0.2	89.18	90.75	95.37	94.57	89.32	71.62	81.90	93.17	72.44	83.25	94.69
-0.1	89.17	90.86	95.36	94.49	89.36	71.06	81.97	93.30	72.38	83.42	95.00
0.0	88.99	90.98	95.39	94.47	89.42	71.92	82.14	93.28	72.66	83.36	95.03
0.1	89.45	90.96	95.52	94.32	89.51	71.99	81.98	93.35	72.13	83.53	94.84
0.2	89.30	90.89	95.47	94.20	89.54	71.82	81.84	93.36	72.29	83.73	95.01
0.3	89.37	90.87	95.44	94.47	89.57	71.62	81.99	93.34	72.10	83.53	95.02
0.4	89.29	90.89	95.36	94.62	89.53	71.98	82.05	93.05	71.99	83.97	94.97
0.5	89.18	90.95	95.44	94.49	89.59	71.55	81.91	93.19	72.03	84.08	95.03
0.6	88.98	90.86	95.40	94.35	89.59	72.00	81.64	93.05	72.08	83.63	95.01
0.8	88.76	90.77	95.23	94.26	89.44	71.31	81.35	92.83	71.63	83.73	94.84
1.0	88.01	90.50	94.61	93.97	89.32	68.57	80.65	92.45	70.05	83.21	94.28
1.1	86.84	90.00	94.34	93.67	89.02	68.70	80.13	91.81	69.31	82.51	93.75
2.0	6.89	6.34	7.23	21.93	16.34	17.02	20.86	8.65	22.20	16.57	6.74

τ	Latin	Latvian	Lithuanian	Low Saxon	Manx	Norwegian	Old French	Polish	Pomak	Portuguese	Romanian
-1.0	72.72	96.11	92.64	52.35	63.68	92.52	0.32	93.68	43.95	95.30	92.79
-0.8	73.71	96.13	92.66	53.00	64.04	92.69	0.24	93.79	44.70	95.23	93.12
-0.6	74.58	96.21	92.66	53.16	64.41	92.80	0.34	93.74	45.15	95.43	92.98
-0.5	74.58	96.13	92.61	53.37	64.59	92.84	0.29	93.76	44.73	95.50	93.03
-0.4	74.63	96.18	92.62	53.36	64.27	92.90	0.31	93.86	45.83	95.36	93.11
-0.3	75.55	96.24	92.52	53.55	64.85	93.01	0.29	93.81	45.79	95.56	93.21
-0.2	75.31	96.20	92.77	53.26	65.17	93.03	0.36	93.89	46.52	95.47	93.16
-0.1	76.20	96.21	92.69	53.38	65.50	93.09	0.35	93.83	45.80	95.56	93.06
0.0	75.80	96.15	92.60	53.31	65.47	92.99	0.28	93.83	46.49	95.44	93.13
0.1	76.54	96.25	92.66	53.40	64.85	93.00	0.41	93.84	46.75	95.60	93.08
0.2	76.34	96.19	92.59	53.77	64.59	93.09	0.42	93.76	46.78	95.50	93.05
0.3	76.64	96.23	92.68	54.03	63.65	92.99	0.48	93.82	46.64	95.67	93.08
0.4	77.14	96.17	92.75	53.89	64.68	92.97	0.53	93.89	47.15	95.63	92.95
0.5	76.87	96.16	92.66	53.80	64.42	92.81	0.59	93.88	46.97	95.54	93.01
0.6	76.67	96.13	92.54	53.94	63.74	92.86	0.52	93.85	47.36	95.42	93.02
0.8	75.67	95.95	92.46	54.04	60.67	92.24	0.59	93.71	46.61	95.29	92.69
1.0	74.09	95.67	91.87	53.21	56.78	91.35	0.51	93.36	47.38	95.00	92.19
1.1	73.09	95.53	91.72	53.38	54.68	90.32	0.54	93.20	46.36	94.80	91.69
2.0	16.21	3.27	9.01	15.64	4.59	14.32	0.53	0.91	10.72	2.22	12.22

τ	Russian	Sanskrit	Scottish Gaelic	Slovak	Slovenian	Spanish	Swedish	Turkish	Ukrainian	Welsh	**macro avg**
-1.0	92.91	75.15	56.64	94.49	95.62	94.90	87.86	80.63	92.87	86.03	**82.69**
-0.8	92.93	75.67	57.46	94.52	95.72	94.98	88.00	80.77	92.95	86.12	**82.88**
-0.6	93.14	75.87	58.09	94.63	95.70	95.18	88.22	81.18	92.90	86.43	**83.04**
-0.5	93.14	76.14	58.00	94.58	95.73	95.23	88.30	81.21	92.91	86.62	**83.10**
-0.4	93.14	75.98	58.78	94.64	95.80	95.25	88.23	81.29	92.94	86.56	**83.19**
-0.3	93.22	75.99	59.04	94.71	95.75	95.24	88.27	81.43	92.85	86.48	**83.29**
-0.2	93.22	76.73	58.99	94.66	95.75	95.33	88.63	81.40	92.93	86.64	**83.37**
-0.1	93.31	76.41	59.71	94.63	95.92	95.32	88.46	81.50	92.90	86.67	**83.42**
0.0	93.30	76.82	59.88	94.60	95.87	95.37	88.65	81.55	92.97	86.57	**83.46**
0.1	93.33	76.93	59.97	94.60	95.71	95.36	88.16	81.58	92.96	86.87	**83.52**
0.2	93.39	77.05	60.15	94.59	95.82	95.33	88.37	81.75	93.00	86.66	**83.53**
0.3	93.39	76.75	60.37	94.50	95.78	95.44	88.47	81.80	92.92	86.59	**83.51**
0.4	93.33	76.97	60.09	94.49	95.86	95.41	88.49	81.77	93.01	86.94	**83.56**
0.5	93.37	76.95	60.01	94.46	95.75	95.34	88.23	81.88	92.76	86.82	**83.53**
0.6	93.36	77.19	59.55	94.56	95.75	95.11	88.15	81.97	92.90	86.48	**83.45**
0.8	93.18	77.17	59.25	94.42	95.55	94.84	88.12	81.88	92.85	86.54	**83.14**
1.0	92.70	76.79	56.67	94.18	95.26	94.39	86.83	82.14	92.86	85.40	**82.42**
1.1	92.14	76.18	55.18	94.02	95.14	93.70	86.61	82.18	92.57	84.55	**81.86**
2.0	0.59	54.15	18.23	1.25	1.95	4.69	24.85	8.85	1.39	37.44	**12.86**

the checkpoint based on type accuracy, and again evaluate using token accuracy. If a significant drop in performance is observed, it would suggest that relying on type accuracy for design decisions during experimentation might lead to a worse final performance in token accuracy.

For different maximum numbers of training epochs (60, 120, 240, 480, and 960), we run five instances[4] of the proposed experiment on the five development languages. We find that performance drops are negligible: in Basque, the largest absolute drop is 0.7%; in Spanish and English, it is 0.3%; in Czech, 0.2%; and in Breton, 0.9%.

The fact that the differences are minor may be attributed to the nature of our splits — data with high corpus-frequency counts tend to appear in the training set, resulting in a limited variance in the development set. The difference between token accuracy and type accuracy should be explored further, as we have not shown a substantial difference between them in terms of the relative ordering of models under our split and evaluation conditions.

4.4. Test Comparison

For the evaluation on the test set, we use four different systems: a naive COPY baseline, which copies the input lemma to output, a model without frequency-aware training (with uniform sampling to batches, ignoring the corpus frequencies, denoted by $\tau = 0.0$), a model with frequency-aware training (weighted random sampling into batches during training) with $\tau = 0.5$ (the best value in average over all languages, weights being the square root of the raw corpus occurrence count), and finally, τ_{BEST}, a model that for each language separately selects the best temperature value based on the performance on the dev set (token accuracy) and uses frequency-aware training with the selected temperature value.

In terms of token accuracy (see Table 4, left part), frequency-aware training with τ_{BEST} outperforms uniform sampling in 26 languages and achieves the overall best result in 23 languages out of 43 languages. It is outperformed by $\tau = 0.5$ in 12 languages and by uniform sampling in additional 5 languages. That is relatively disappointing, and it probably shows overfitting to the dev set (by hyperparameter tuning, checkpoint selection and temperature value selection), because otherwise we would expect the τ_{BEST} model to outperform the rest of models. Compared to uniform sampling, $\tau = 0.5$ brings improvement in 23 languages and outperforms all other models in 13 languages. In two languages (Low Saxon and Old French), none of our models outperformed the COPY baseline.

In terms of type accuracy (see Table 4, right part), the results are more straightforward: τ_{BEST} improves over uniform sampling in 41 out of 43 languages, achieving the overall best result in 35 languages. It is outperformed by $\tau = 0.5$ in only 6 languages. As in token accuracy, it is outperformed by COPY baseline in two languages. It is surprising that when choosing the temperature value based on token accuracy performance on the dev set, on the test set it helps more the type accuracy than the token accuracy. It would be beneficial to explore this more in future work and to seek an explanation.

Regarding the comparison of token accuracy and type accuracy, note that at least in the test results we can observe some differences (τ_{BEST} system clearly better than the rest according to type accuracy, yet not so clearly according to token accuracy).

5. Discussion

We found that, in at least some languages, frequency-aware training provides a clear benefit for both token accuracy and type accuracy.

If the data split was not lemma-disjoint, the reason for its positive effect on token accuracy would be fairly straightforward: Suppose a frequent lemma A appears in the training set. Under frequency-aware training, it receives a higher weight, allowing the model to learn it more effectively. If the same lemma A also occurs in the development or test set, the model will (likely) inflect it correctly. Since lemma A is frequent, its correct inflection contributes disproportionately to the overall token accuracy, due to the higher evaluation weight it receives.

[4]Differing by the random seed used for initialization.

Table 4
Test results, **token accuracy** (left) and **type accuracy** (right). The best system for each language (selected separately for token accuracy and type accuracy) is marked with **bold**. COPY copies the input lemma to output. $\tau = 0.0$ is uniform sampling. $\tau = 0.5$ is frequency-aware training with weights equal to the square root of the raw occurrence counts. τ_{BEST} is system, where the specific τ value is selected separately for each language, based on token accuracy on dev set.

lang ↓	corpus ↓	token accuracy (%)				type accuracy (%)			
system →		COPY	$\tau = 0.0$	$\tau = 0.5$	τ_{BEST}	COPY	$\tau = 0.0$	$\tau = 0.5$	τ_{BEST}
Afrikaans	AfriBooms	72.19	85.89	**86.41**	86.31	69.18	82.61	82.93	**83.82**
Basque	BDT	45.23	88.62	89.02	**89.59**	39.74	85.71	85.79	**87.57**
Belarusian	HSE	43.44	88.64	**88.90**	**88.90**	40.88	87.07	87.38	**88.14**
Breton	KEB	61.54	67.63	67.40	**69.06**	57.66	63.56	63.55	**65.04**
Bulgarian	BTB	45.08	**92.64**	92.40	92.41	37.90	90.61	90.81	**91.06**
Catalan	AnCora	73.01	96.06	**96.12**	96.11	64.73	93.18	93.35	**93.49**
Croatian	SET	40.50	93.05	93.11	**93.25**	36.10	91.64	91.80	**92.29**
Czech	PDT	45.96	97.41	97.48	**97.52**	37.53	97.38	97.43	**97.50**
Danish	DDT	59.24	**90.54**	90.31	90.49	55.61	89.84	89.53	**89.87**
Dutch	Alpino	61.60	80.71	**81.76**	**81.76**	55.48	76.40	76.36	**78.13**
English	EWT	82.10	**96.11**	95.84	96.03	76.67	93.77	93.70	**93.91**
Estonian	EDT	30.33	90.52	90.70	**90.71**	23.24	88.74	88.97	**89.53**
Finnish	TDT	30.27	**91.83**	**91.83**	**91.83**	23.88	90.47	90.30	**90.69**
French	GSD	74.27	96.71	**96.79**	96.74	72.55	95.32	95.21	**95.44**
Galician	TreeGal	63.39	92.61	92.21	**92.75**	59.67	93.86	**94.01**	**94.01**
German	GSD	75.35	89.61	**89.92**	**89.92**	75.22	88.77	88.97	**89.34**
Gothic	PROIEL	30.62	75.89	77.26	**77.72**	17.50	69.71	70.63	**72.45**
Greek	GDT	40.62	**84.70**	84.52	**84.70**	34.45	81.63	81.41	**82.04**
Hungarian	Szeged	56.17	**93.50**	93.37	93.44	51.34	92.52	**92.84**	92.64
Icelandic	Modern	41.48	**72.36**	72.24	**72.36**	35.71	68.34	69.10	**70.02**
Irish	IDT	55.69	86.60	**87.40**	**87.40**	51.93	83.40	83.78	**84.17**
Italian	ISDT	68.49	95.74	95.86	**96.04**	62.95	93.82	94.04	**95.12**
Latin	ITTB	28.87	81.29	**83.08**	80.99	14.58	74.25	74.74	**77.39**
Latvian	LVTB	33.59	**96.81**	96.73	96.79	27.81	96.32	96.41	**96.42**
Lithuanian	ALKSNIS	31.20	93.93	93.92	**93.96**	27.79	92.51	**92.77**	92.71
Low Saxon	LSDC	**59.03**	55.03	56.49	56.32	**54.42**	50.84	51.23	51.55
Manx	Cadhan	66.20	71.47	**71.69**	71.47	65.15	63.97	65.09	**66.41**
Norwegian	Bokmaal	58.60	**93.40**	93.34	93.07	53.62	92.68	92.80	**93.00**
Old French	PROFITEROLE	**46.75**	0.71	3.70	10.80	**19.82**	0.10	0.11	0.31
Polish	PDB	30.53	93.93	94.03	**94.17**	28.42	93.69	**93.88**	93.86
Pomak	Philotis	27.75	**51.17**	50.63	50.54	20.98	42.19	43.69	**45.27**
Portuguese	Bosque	72.11	**96.93**	96.91	96.91	67.66	95.83	96.05	**96.19**
Romanian	RRT	44.17	92.25	**92.49**	92.25	42.00	**92.46**	92.38	92.31
Russian	SynTagRus	33.01	94.64	**94.87**	**94.87**	26.35	92.76	92.94	**93.28**
Sanskrit	Vedic	12.51	78.39	78.98	**79.25**	9.43	74.65	74.88	**76.07**
Scottish Gaelic	ARCOSG	66.16	65.80	**66.49**	66.20	59.71	59.00	59.41	**61.39**
Slovak	SNK	35.75	**94.57**	94.56	94.33	31.83	93.98	**94.17**	94.05
Slovenian	SSJ	38.18	95.65	95.60	**95.71**	33.31	95.17	95.22	**95.39**
Spanish	AnCora	70.42	**97.31**	97.30	97.22	61.38	94.86	94.97	**95.17**
Swedish	Talbanken	52.61	90.73	90.14	**90.91**	48.63	89.44	89.55	**90.03**
Turkish	BOUN	51.91	84.46	84.66	**85.16**	42.80	80.70	81.10	**81.97**
Ukrainian	IU	35.41	**92.60**	92.49	92.49	32.11	92.51	**92.55**	92.51
Welsh	CCG	65.16	88.15	88.15	**88.27**	60.11	84.39	84.41	**84.95**
macro avg		50.15	85.04	85.28	**85.51**	44.37	82.57	82.80	**83.41**

Nevertheless, the split is lemma-disjoint, which raises a question: Why does frequency-aware training help? Furthermore, why should it help even in terms of type accuracy? Why does it help only in some languages, and why we even achieve an opposite effect some other languages?

One possible explanation for why frequency-aware training is beneficial, even with the lemma-disjoint split, is that frequent words in the dev/test sets tend to inflect in a similar way to frequent words in the training set. If this is the case, assigning greater weight to frequent words during training should help the model inflect frequent dev/test words more accurately, thereby increasing token accuracy. A more straightforward explanation is that frequency-aware training helps due to a derivational leak. For instance, a frequent lemma in the training set, such as the Czech verb "*jít*" ("*to go*"), may have derived lemmas in the dev/test sets, e.g., "*přijít*" ("*to come*"), "*projít*" ("*to go through*"). These derived forms are also frequent and thus receive a higher weight during evaluation. Since derived verbs in Czech inflect in very similar ways, giving more weight to "*jít*" during training improves performance on "*přijít*" and "*projít*" during evaluation.

As for why frequency-aware training might improve also type accuracy, the gain may come from the fact that frequent lemmas occur in corpus not only with a higher occurrence count (reflected in token accuracy) but also with a greater variety of distinct forms. Since type accuracy treats each lemma-tag-form triple equally, this increased form diversity could contribute to the improvement.

5.1. Future Work

Regarding why frequency-aware training helps in some languages but has the opposite effect in others, its success appears to depend on factors that are not yet well understood. Investigating these in future work would be worthwhile. Possible explanations include linguistic properties such as morphological richness and regularity, or corpus-related factors such as data size and variability.

In addition, other ways of obtaining corpus frequencies for morphological data could be explored, as described in Section 3.1, especially using large raw data to extract the frequencies with some tag-based back-off for disambiguation, not dropping the UniMorph items with zero actual corpus occurrences.

Kodner et al. [10] argues that frequency-weighted split better reflects real-world frequency distribution of train-test, and so we use it. However, it would be beneficial to explore more the split compared to uniform split, to experimentally answer whether it would hurt the real-world performance, if we developed a system on a uniform train-dev-test split.

Also, it would be worth experimenting with the canonical UD train-dev-test split (the original split of UD, not lemma-disjoint).

In terms of metrics (token accuracy and type accuracy), it would be worth further exploring whether there is a difference in terms of what metric we use during the development of a model, in contrast to the metric used in the final evaluation (that is, tuning hyperparameters for two systems, each according to one of the metrics, and then evaluate both systems according to token accuracy and see if the performance differs). Furthermore, it would be particularly interesting to measure the differences between them on a uniform (not frequency-weighted) split of UD. Moreover, enriching a standard benchmark like SIGMORPHON with corpus frequency information and re-evaluating the shared task using our token accuracy metric could potentially yield valuable insights.

6. Conclusion

We explored the incorporation of corpus frequencies in the task of morphological inflection along three key dimensions: train-dev-test split of data (frequency-weighted lemma-disjoint split), training data sampling (frequency-aware training), and evaluation. We designed and consistently applied a previously suggested but never before used evaluation metric that assigns greater importance to frequent word forms (token accuracy). We demonstrated that in our setting (lemma-disjoint, frequency-weighted split), the difference between the original metric (type accuracy or just accuracy) and token accuracy is minimal with respect to the relative ranking of different systems. Our pioneering experiments with frequency-aware training showed promising results, improving over uniform sampling in 26 out of

43 languages. The overall best weighting scheme seems to be with corpus frequency temperature 0.5 (each sample is given a weight corresponding to the square root of its corpus frequency).

All three dimensions (frequency-weighted split, frequency-aware training, and token accuracy) are worth further investigation.

Acknowledgments

This research was supported by the Johannes Amos Comenius Programme (P JAC) project No. CZ.02.01.01/00/22_008/0004605, Natural and anthropogenic georisks.

Computational resources for this work were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

The work described herein uses resources hosted by the LINDAT/CLARIAH-CZ Research Infrastructure (projects LM2018101 and LM2023062, supported by the Ministry of Education, Youth and Sports of the Czech Republic).

We thank the anonymous reviewers for their valuable comments.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and Writefull to check grammar and spelling. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] T. Sourada, J. Straková, Flexing in 73 languages: A single small model for multilingual inflection, in: K. Ekštein, M. Konopík, O. Pražák, F. Pártl (Eds.), 28th International Conference on Text, Speech and Dialogue, Springer, Cham, Switzerland, 2025.

[2] R. Cotterell, C. Kirov, J. Sylak-Glassman, D. Yarowsky, J. Eisner, M. Hulden, The SIGMORPHON 2016 shared Task—Morphological reinflection, in: Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 10–22. URL: https://aclanthology.org/W16-2002. doi:10.18653/v1/W16-2002.

[3] R. Cotterell, C. Kirov, J. Sylak-Glassman, G. Walther, E. Vylomova, P. Xia, M. Faruqui, S. Kübler, D. Yarowsky, J. Eisner, M. Hulden, CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages, in: Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection, Association for Computational Linguistics, Vancouver, 2017, pp. 1–30. URL: https://aclanthology.org/K17-2001. doi:10.18653/v1/K17-2001.

[4] R. Cotterell, C. Kirov, J. Sylak-Glassman, G. Walther, E. Vylomova, A. D. McCarthy, K. Kann, S. J. Mielke, G. Nicolai, M. Silfverberg, D. Yarowsky, J. Eisner, M. Hulden, The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection, in: Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, Association for Computational Linguistics, Brussels, 2018, pp. 1–27. URL: https://aclanthology.org/K18-3001. doi:10.18653/v1/K18-3001.

[5] E. Vylomova, J. White, E. Salesky, S. J. Mielke, S. Wu, E. M. Ponti, R. H. Maudslay, R. Zmigrod, J. Valvoda, S. Toldova, F. Tyers, E. Klyachko, I. Yegorov, N. Krizhanovsky, P. Czarnowska, I. Nikkarinen, A. Krizhanovsky, T. Pimentel, L. Torroba Hennigen, ..., M. Hulden, SIGMORPHON 2020 shared task 0: Typologically diverse morphological inflection, in: Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Association for Computational Linguistics, Online, 2020, pp. 1–39. URL: https://aclanthology.org/2020.sigmorphon-1.1. doi:10.18653/v1/2020.sigmorphon-1.1.

[6] T. Pimentel, M. Ryskina, S. J. Mielke, S. Wu, E. Chodroff, B. Leonard, G. Nicolai, Y. Ghanggo Ate, S. Khalifa, N. Habash, C. El-Khaissi, O. Goldman, M. Gasser, W. Lane, M. Coler, A. Oncevay, J. R. Montoya Samame, G. C. Silva Villegas, A. Ek, ..., E. Vylomova, SIGMORPHON 2021 shared task on morphological reinflection: Generalization across languages, in: Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Association for Computational Linguistics, Online, 2021, pp. 229–259. URL: https://aclanthology.org/2021.sigmorphon-1.25. doi:10.18653/v1/2021.sigmorphon-1.25.

[7] J. Kodner, S. Khalifa, K. Batsuren, H. Dolatian, R. Cotterell, F. Akkus, A. Anastasopoulos, T. Andrushko, A. Arora, N. Atanalov, G. Bella, E. Budianskaya, Y. Ghanggo Ate, O. Goldman, D. Guriel, S. Guriel, S. Guriel-Agiashvili, W. Kieraś, A. Krizhanovsky, ..., E. Vylomova, 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, Association for Computational Linguistics, Seattle, Washington, 2022, pp. 176–203. URL: https://aclanthology.org/2022.sigmorphon-1.19. doi:10.18653/v1/2022.sigmorphon-1.19.

[8] O. Goldman, K. Batsuren, S. Khalifa, A. Arora, G. Nicolai, R. Tsarfaty, E. Vylomova, 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, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 117–125. URL: https://aclanthology.org/2023.sigmorphon-1.13. doi:10.18653/v1/2023.sigmorphon-1.13.

[9] K. Batsuren, O. Goldman, S. Khalifa, N. Habash, W. Kieraś, G. Bella, B. Leonard, G. Nicolai, K. Gorman, Y. G. Ate, M. Ryskina, S. Mielke, E. Budianskaya, C. El-Khaissi, T. Pimentel, M. Gasser, W. A. Lane, M. Raj, M. Coler, ..., E. Vylomova, UniMorph 4.0: Universal Morphology, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 840–855. URL: https://aclanthology.org/2022.lrec-1.89/.

[10] J. Kodner, S. Payne, S. Khalifa, Z. Liu, Morphological inflection: A reality check, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 6082–6101. URL: https://aclanthology.org/2023.acl-long.335/. doi:10.18653/v1/2023.acl-long.335.

[11] D. Zeman, J. Nivre, M. Abrams, E. Ackermann, N. Aepli, H. Aghaei, Ž. Agić, A. Ahmadi, L. Ahrenberg, C. K. Ajede, S. F. Akkurt, G. Aleksandravičiūtė, I. Alfina, A. Algom, K. Alnajjar, C. Alzetta, E. Andersen, L. Antonsen, T. Aoyama, ..., R. Ziane, Universal dependencies 2.14, 2024. URL: http://hdl.handle.net/11234/1-5502, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[12] O. Goldman, D. Guriel, R. Tsarfaty, (un)solving morphological inflection: Lemma overlap artificially inflates models' performance, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 864–870. URL: https://aclanthology.org/2022.acl-short.96/. doi:10.18653/v1/2022.acl-short.96.

[13] G. Nicolai, D. Lewis, A. D. McCarthy, A. Mueller, W. Wu, D. Yarowsky, Fine-grained morphosyntactic analysis and generation tools for more than one thousand languages, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 3963–3972. URL: https://aclanthology.org/2020.lrec-1.488/.

[14] C. Kirov, R. Cotterell, J. Sylak-Glassman, G. Walther, E. Vylomova, P. Xia, M. Faruqui, S. J. Mielke, A. McCarthy, S. Kübler, D. Yarowsky, J. Eisner, M. Hulden, UniMorph 2.0: Universal Morphology, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh

International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://aclanthology.org/L18-1293/.

[15] A. D. McCarthy, C. Kirov, M. Grella, A. Nidhi, P. Xia, K. Gorman, E. Vylomova, S. J. Mielke, G. Nicolai, M. Silfverberg, T. Arkhangelskiy, N. Krizhanovsky, A. Krizhanovsky, E. Klyachko, A. Sorokin, J. Mansfield, V. Ernštreits, Y. Pinter, C. L. Jacobs, R. Cotterell, M. Hulden, D. Yarowsky, UniMorph 3.0: Universal Morphology, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 3922–3931. URL: https://aclanthology.org/2020.lrec-1.483/.

[16] L. Liu, M. Hulden, Can a transformer pass the wug test? tuning copying bias in neural morphological inflection models, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 739–749. URL: https://aclanthology.org/2022.acl-short.84/. doi:10.18653/v1/2022.acl-short.84.

[17] J. Nivre, M.-C. de Marneffe, F. Ginter, J. Hajič, C. D. Manning, S. Pyysalo, S. Schuster, F. Tyers, D. Zeman, Universal Dependencies v2: An evergrowing multilingual treebank collection, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 4034–4043. URL: https://aclanthology.org/2020.lrec-1.497/.

[18] M. Straka, J. Straková, Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe, in: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 88–99. URL: http://www.aclweb.org/anthology/K/K17/K17-3009.pdf.

[19] S. Wu, R. Cotterell, M. Hulden, Applying the transformer to character-level transduction, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Online, 2021, pp. 1901–1907. URL: https://aclanthology.org/2021.eacl-main.163/. doi:10.18653/v1/2021.eacl-main.163.