

On the Interplay between Positional Encodings, Morphological Complexity, and Word Order Flexibility

Kushal Tatariya* and Wessel Poelman* and Miryam de Lhoneux

L^AG_OM-NLP, Department of Computer Science, KU Leuven

{kushaljayesh.tatariya, wessel.poelman, miryam.delhoneux}@kuleuven.be

Abstract

Language model architectures are predominantly first created for English and subsequently applied to other languages. It is an open question whether this architectural bias leads to degraded performance for languages that are structurally different from English. We examine one specific architectural choice: positional encodings, through the lens of the *trade-off hypothesis*: the supposed interplay between morphological complexity and word order flexibility. This hypothesis posits a trade-off between the two: a more morphologically complex language can have a more flexible word order, and vice-versa. Positional encodings are a direct target to investigate the implications of this hypothesis in relation to language modelling. We pretrain monolingual model variants with absolute, relative and no positional encodings for seven typologically diverse languages and evaluate them on four downstream tasks. Contrary to previous findings, we do not observe a clear interaction between position encodings and morphological complexity or word order flexibility, as measured by various proxies. Our results show that the choice of tasks, languages, and metrics are essential for drawing stable conclusions.

1 Introduction

A longstanding theory in linguistics posits that human languages are roughly equally "complex", but that there exists a trade-off between the complexity of syntax and morphology (e.g., Hockett, 1958; Sinnemäki, 2014; Miestamo, 2017; Bentz et al., 2022; Nijs et al., 2025). The *trade-off hypothesis* states that languages exhibiting higher morphological complexity tend to use less complex syntax, and vice-versa. The hypothesis does not get a lot of attention in NLP, even though it is a useful

*Equal contribution. Order determined by a coin toss.

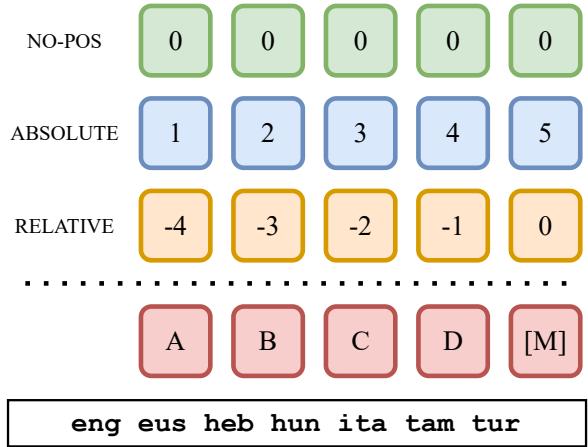


Figure 1 – We investigate the relationship between positional encodings and morphology by training three model variants per language: **NO-POS**, **ABSOLUTE**, and **RELATIVE**. We use seven typologically diverse languages, varying in morphological complexity and word order flexibility.

lens through which we can investigate *architectural choices* of language models for different languages. In NLP, models are typically developed for English first, and then applied to other languages without significant architectural changes (Søgaard, 2022). Subword-based transformers (Vaswani et al., 2017) are the de-facto standard architecture, with the implicit assumption that it is the best architecture for language modelling, regardless of the language. However, some architectures might work better for some languages than for others. We look at a specific architectural choice: *positional encodings*, and their interplay with morphological complexity and word order flexibility.

Positional encodings in transformers signal to the model where a given token occurs in a sequence (Figure 1). Numerous methods to encode positional information have been proposed (e.g., Liu et al., 2020; He et al., 2020; Dufter et al., 2022). However, these strategies have generally been optimized for English. This opens the question of whether these

strategies work just as well for other languages with structures distinct from English. The trade-off hypothesis gives us two areas to look for potential answers to this question: morphological complexity and word order flexibility. Intuitively, the more flexible the word order of a language, the less useful positional encodings might be, and vice-versa. Similarly, a model for a morphologically complex language could depend more on positional encodings if longer words are segmented into more tokens, and vice-versa.

We study the impact of three different position encoding strategies in monolingual masked language modelling across seven typologically diverse languages, specifically: no positional encodings (**NO-POS**), relative encodings (**RELATIVE**), and absolute encodings (**ABSOLUTE**). We aim to understand whether the importance of position encodings decreases with an increase in word order flexibility and how this relates to morphological complexity. As a starting point we take the work of [Ghosh et al. \(2024\)](#), who study the effect of *removing* positional encodings from monolingual language models trained *with* them. Their results show that performance decreases from removing these encodings differ significantly between languages and that this is negatively correlated with a rough proxy of morphological complexity, namely the *type-token ratio* (*TTR*) of parallel texts. We expand upon this line of work in three ways:

- We systematically select languages with diverse typological profiles in terms of morphological complexity and word order strategies. Additionally, we make sure all languages have coverage in the downstream tasks we use and analyses we carry out.
- We train monolingual models from scratch¹ with the three mentioned positional encoding methods, we keep all other variables constant.
- We analyse the impact of position encodings in the context of the trade-off hypothesis, using fine-grained proxies of morphological complexity and word order flexibility.

Our findings show that even for languages with relatively flexible word order, position encodings are essential for processing linguistic structures.

¹This is important since the *removal* of positional information from a trained model is notably different than the interplay between architectural choices and inherent language differences (*cf.* [Ghosh et al., 2024](#)).

Contrary to the findings of [Ghosh et al. \(2024\)](#), we do not see much impact of word order flexibility or morphological complexity on the utility of position encodings. Rather, we observe the impact of position encodings to be more task specific. We also find that *relative* position encodings are generally more suitable than *absolute* encodings for most applications.

2 Background

2.1 Position Encodings

A transformer requires an explicit signal regarding the ordering of its input. Strategies for this include adding position encodings ([Vaswani et al., 2017](#); [Devlin et al., 2019](#); [Takase and Okazaki, 2019](#)), adding modifications to the attention matrices ([Shaw et al., 2018](#); [Huang et al., 2020](#)), or using an RNN to process the input ([Neishi and Yoshinaga, 2019](#)). [Dufter et al. \(2022\)](#) identify two reference points that are commonly used to encode the position of a unit in a sentence: the *absolute* position and the position of a unit *relative* to other units in a sentence. Most transformers employ one or both ([He et al., 2020](#); [Ke et al., 2021](#)) as the basis for position encoding strategies. For both of these reference points, [Dufter et al. \(2022\)](#) note two methods of injecting position information into the model that are commonly associated with each. The first is "Adding Position Encodings" where position information is added to the input before it is fed to the model, and commonly uses the *absolute* reference point. The second is "Modifying the Attention Matrix" which includes approaches that directly alter the attention matrix. This approach is more commonly associated with *relative* position encodings. In this work, we focus on these two approaches, and compare them to a model that uses no position encodings at all.

2.2 Position Encodings and Linguistic Structures

[Wang and Chen \(2020\)](#) show that different architectures utilise position encodings differently for various NLP tasks. [Wang et al. \(2020\)](#) conduct a theoretical comparison of absolute, relative and no position encodings in BERT ([Devlin et al., 2019](#)) and show that a BERT model without position encodings is essentially a bag-of-words representation. They also show that absolute position encodings can be uninformative since any random truncations of the input can arbitrarily shift them

with an offset, while relative position encodings can be beneficial for longer sequences.

Understanding the interplay of position information and the learning of linguistic structures is also done using data-level perturbations, where [Sinha et al. \(2021\)](#) randomly shuffle words and train masked language models to negate the effect of position encodings. They show shuffled language models still show a remarkable understanding of syntax and semantics, which indicates masked language models may not rely on word order information as much as we think they do. [Abdou et al. \(2022\)](#) extend this work to show that certain NLU tasks are sensitive to word order information. They also show results differ depending on *when* data is shuffled: before or after tokenization.

While these findings are linguistically motivated, they only focus on English. When making architectural decisions for languages that are typologically distinct from English, these conclusions may not hold. This is why we present an investigation into the interaction of inherent language differences and position encodings.

2.3 Position Encodings and Multilingual Language Modelling

[Ji et al. \(2023\)](#) study the interplay of syntax and position encodings in a multilingual setting. They find distances of typological feature vectors of languages to be correlated with their distance in the position space (referring to the vectors representing position information). They leverage linguistic typology as a prior to facilitate positive transfer from high resource to low resource languages in dependency parsing. A direct investigation of the impact of various types of positional encodings on cross-lingual performance is done by [Ravishankar and Søgaard \(2021\)](#). They find absolute position encodings work well in a multilingual language model to facilitate cross-lingual transfer. However, their experiments use synthetically perturbed English data, not "real" multilingual data.

In a monolingual setting, recent work by [Ghosh et al. \(2024\)](#) investigates the importance of position encodings across languages with different levels of morphological complexity. They use monolingually pre-trained BERT-style models across 22 languages, and set positional encodings to 0 before finetuning the models on various NLI tasks. This approach does not account for the fact that the models have learned position information in the pretraining stage that may confound results. Us-

ing the *type-token ratio (TTR)* of parallel texts as a proxy for morphological complexity, they find a negative correlation between morphological complexity and the relative drop in downstream performance between models with and without positional encodings. We find the use of TTR as a proxy for morphological complexity problematic, as TTR is not an inverse metric for word order flexibility. We elaborate on this in §3.1.

3 Methodology

3.1 Language Selection and Metrics

Word order flexibility can be considered as a gradient (e.g., [Futrell et al., 2015](#); [Naranjo and Becker, 2018](#); [Levshina et al., 2023](#); [Baylor et al., 2024](#)), instead of coarse groupings found in typological databases ([Dryer, 2013](#)). In an effort to create a language sample that is representative of typological features across this gradient, we sample languages based on the typological sampling framework by [Ploeger et al. \(2025\)](#) using order-related features from Grambank ([Skirgård et al., 2023](#)).² These coarse features allow us to create a representative language sample that is consistent across datasets. With the languages covered by [Futrell et al. \(2015\)](#) as a starting point, we consider languages with enough pretraining data and that are covered in potential downstream tasks as our sampling frame.³ We then select a sample from our frame that covers (or saturates) the order-related typological features of interest. This results in a sample of seven languages listed in Table 1.

Morphological Complexity. For a proxy of morphological complexity, we use the *Accessor Variety (AV)* and *Accessor Efficiency (η)* metrics introduced by [Poelman et al. \(2025\)](#). AV measures how often types from the vocabulary of a tokenizer co-occur, either on the left or right in a corpus. η measures the Shannon efficiency of this distribution. We follow [Poelman et al.](#) and use the *right accessor* (successor) for both. Intuitively, AV and η capture some properties of morphological complexity when using a subword tokenizer: a language that uses many affixes that can occur in different places results in more *choice* or *ambiguity* from one token to another; a proxy for complexity. Similar to the *moving average TTR* (MATTR, [Covington and McFall, 2010](#)), we calculate AV in a sliding widow

²Details are listed in §A.1.

³The set from which we are going to draw a sample.

Language	Dominant Order	Word Order			Morphological Complexity		
		HDE	ROE	SO-ROE	AV	η	MATTR
Basque	SOV	0.50	0.21	0.72	30.22	0.38	0.68
English	SVO	0.16	0.03	0.20	25.20	0.34	0.53
Hebrew	SVO	0.38	0.13	0.25	27.48	0.32	0.69
Hungarian	NDO	0.40	0.12	0.83	42.57	0.37	0.63
Italian	SVO	0.26	0.06	0.27	24.63	0.33	0.58
Tamil	SOV	0.07	0.09	0.94	41.50	0.45	0.69
Turkish	SOV	0.22	0.13	0.31	33.32	0.36	0.65

Table 1 – Language selection. See §A.1 for details regarding the creation of this language sample. The *head direction entropy* (*HDE*), *relation order entropy* (*ROE*), and *subject object ROE* (*SO-ROE*) are taken from [Futrell et al. \(2015\)](#). These are estimated on 1000 sentences from UD corpora. The *dominant order* from [Dryer \(2013\)](#) is added to show there is more granularity need at this stage than coarse groupings. The measure for morphological complexity are *accessor variety* (*AV*), the Shannon efficiency of *AV* (η), and the *moving average TTR* (*MATTR*). These are all averages of sliding windows of 1000 tokens. All are calculated using monolingual tokenizers trained on 250k lines samples from the same dataset the models are trained on.

of tokens to combat the effect of text length.

[Ghosh et al. \(2024\)](#) approximate morphological complexity by calculating TTR on FLORES-200 using words or characters. This creates a mismatch between (1) the units a model sees (subword tokens) and what the proxy is calculated on (words or characters), and (2) the data that is used for model training and what data the proxy is calculated on. We remove these confounds by maintaining consistency across units and corpora. For the sake of completeness and comparison with [Ghosh et al.](#), we include *MATTR* using our tokenizers.

Word Order Flexibility. For word order flexibility, we use the UD-based metrics introduced by [Futrell et al. \(2015\)](#). *Head direction entropy* (*HDE*) measures the conditional entropy of whether a head is to the right or left of a dependent. It is a proxy for consistency in head direction, giving us some indication of word order flexibility once conditioned on all relation types and part of speech labels. *Relation order entropy* (*ROE*) is the conditional entropy of the order of words in a local sub-tree. This measure is sensitive to the size of a corpus. To alleviate this, it can be restricted to specific relations of interest, such as the position of the subject and object in the main clause. This is referred to as the *subject object ROE* (*SO-ROE*).

We want to emphasize a point made by [Kiparsky \(1997\)](#) and reinforced by [Futrell et al. \(2015\)](#): languages with flexible word order (*SO-ROE* higher than 0.625, for instance Tamil, Basque and Hungar-

ian, see Table 1) tend to use case marking. Thus, a more flexible word order implies the presence of case marking. However, inversely, the presence of case marking does not necessarily imply a flexible word order. Turkish, for example, has low *SO-ROE* and *HDE*, indicating a more rigid word order, but it does use case markings and is considered in the NLP landscape a more "morphologically complex" language. Put differently, a metric for morphological complexity is not directly an "inverse" metric for word order flexibility (see Table 1). Therefore, using measures like *TTR* or *MATTR*, as a rough proxy for word order flexibility (as is done in [Ghosh et al. \(2024\)](#)) can be problematic, and the two must not be conflated.

With *HDE* and *SO-ROE*, unfortunately, we also introduce the confound we alluded to earlier: the units of this measure are not the same as the units the model sees. Still, it gives a more realistic and gradient view of word order flexibility than shuffling English ([Ravishankar and Søgaard, 2021](#); [Abdou et al., 2022](#)), using typological groupings ([Dryer, 2013](#)), or using a proxy for morphological complexity as an inverse metric for word order flexibility ([Ghosh et al., 2024](#)).

3.2 Models and Tokenization

We use the training setup from [Tatariya et al. \(2024b\)](#). For each of our seven languages, we train three monolingual RoBERTa-base models ([Liu et al., 2019](#)), one for each of the positional encoding strategies, resulting in 21 models total.

Experimental setting	Ghosh et al. 2024	Ours
Languages studied	6-22, depending on the task	7 sampled for diversity and consistency across tasks
Tasks	POS, UD, XNLI, NER, PAWS-X	UD, NER, SIB-200, MultiBLiMP
Model comparison	Removing vs keeping positional encodings	Models with three strategies trained from scratch
Morphological complexity	TTR	AV, η , MATTR
Word order flexibility	TTR	HDE, SO-ROE
Metric unit	Words or characters	Subwords from tokenizer
Corpus used for metrics	2k lines from FLORES-200	250k lines from tokenizer and model training data

Table 2 – Comparison of experimental conditions.

We choose not to remove positional encodings from a model after pretraining (Ghosh et al., 2024) since we are interested in inherent model architecture choices and their relation to inherent language characteristics. We keep hyperparameters and other settings as consistent as possible. Our tokenizers are monolingual BPE (Gage, 1994; Sennrich et al., 2016) tokenizers.⁴

3.3 Data and Evaluation

For training our tokenizers and pretraining our monolingual models, we sample one hundred million lines per language from FineWeb (English only, Penedo et al., 2024) and FineWeb-2 (the other languages, Penedo et al., 2025). We sample the same amount of data to remove the confound of data disparity between languages. Similarly, we choose *lines* instead of *tokens*, as the latter is confounded by language characteristics (Poelman et al., 2025) and circular: we need a tokenizer to count tokens to create a sample, but we need a sample to train our tokenizers. Our corpus-based metrics are also calculated on 250k lines randomly sampled from this dataset.

We use the following downstream tasks:

- Dependency Parsing using Universal Dependencies (UD v2.15, Nivre et al., 2020; de Marneffe et al., 2021; Zeman et al., 2024) represents a challenging token- and sentence-level task.
- Named-Entity Recognition (NER) using WikiAnn (Pan et al., 2017; Rahimi et al., 2019), which represents an in-between: named entities can be multi-word expressions or single tokens.
- SIB-200 (Adelani et al., 2024) is a simple, sentence-level text classification task.

⁴Details are listed in §A.

- MultiBLiMP 1.0 (Jumelet et al., 2025) is a minimal pair, sentence level task where ungrammatical sentences should have higher (worse) perplexity values than grammatical ones. The dataset specifically targets *agreement*, which represents an interesting setting for both word order and morphological features. Since we use encoder-models, we use pseudo-perplexity (Salazar et al., 2020; Kauf and Ivanova, 2023), as implemented in the `minicons` library (Misra, 2022).

We selected these tasks to represent both token-level and sentence-level tasks, as well as to cover different difficulty levels.⁵ In contrast with Ghosh et al. (2024), who used a subset of XTREME (Hu et al., 2020) covering different languages per task (see Ploeger et al. (2024)), in our setup all languages are available for all tasks, making comparisons easier and conclusions more consistent. Table 2 summarizes the differences between our and Ghosh et al.’s study.

4 Results

In this section, we discuss *general* results of the monolingual masked language modelling performance with `ABSOLUTE`, `RELATIVE` and `NO-POS`, followed by their respective downstream results. In §5 we go into more detail regarding individual language differences.

4.1 Pretraining

The loss curves for the FineWeb-2 validation set of each language for each position encoding type are shown in Figure 2. With the same hyperparameters, none of the `NO-POS` models achieve a loss below 6.⁶ All languages also have better losses with `RELATIVE` than with `ABSOLUTE`. Additionally, the

⁵We experimented with Belebele (Bandarkar et al., 2024), a challenging reading comprehension task, but found that none of the models scored above the majority choice.

⁶It could be argued that the hyperparameters we used are tailored to the models *with* positional encodings. While we

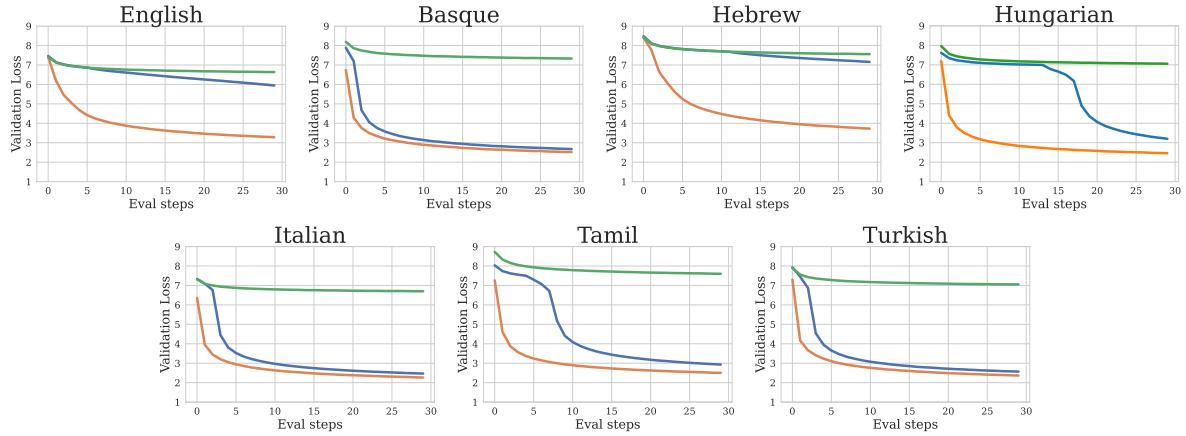


Figure 2 – Loss curves on validation set for pretraining for **ABSOLUTE**, **RELATIVE**, and **NO-POS**.

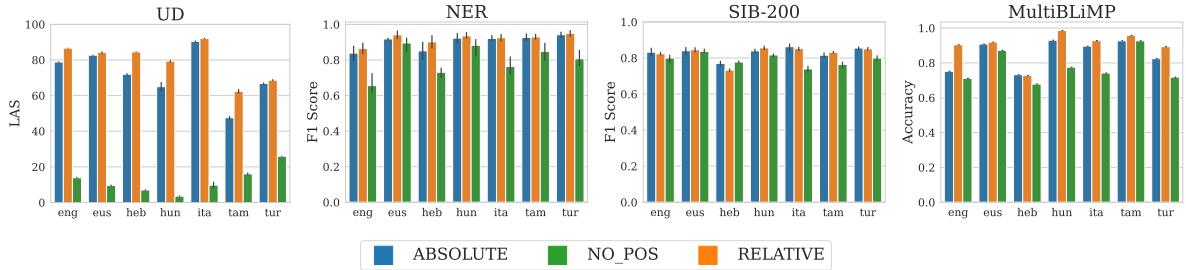


Figure 3 – Results per task, language, and positional encoding type. Scores are averaged over 5 runs. Full results are in §B.2.

models achieve a lower loss more quickly with **RELATIVE** compared to **ABSOLUTE**. Better losses with **RELATIVE** encodings indicate that Huang et al. (2020)’s findings are applicable to languages beyond English. Still, the learning dynamics are quite different between the languages; English and Hebrew do not converge with **ABSOLUTE**, for example, and their losses for **RELATIVE** are higher than the others. Similarly, Hungarian and Tamil take noticeably longer to start converging with **ABSOLUTE** than with **RELATIVE**.

4.2 Finetuning

Results for the finetuned models on downstream tasks are in Figure 3. We see that the importance of position encodings differs more per task than language. This echoes the findings from Wang and Chen (2020) and Wang et al. (2020). The difference between **RELATIVE** and **ABSOLUTE** for most tasks is, surprisingly, less pronounced than the difference of the loss in Figure 2 (e.g., for English and Hebrew). This means the models are able to “catch up” during finetuning, even if they did not

try to keep the comparison as fair as possible, we did not fully explore the effect of hyperparameters. We discuss this more in the limitations (§6).

fully converge during pretraining.

Dependency parsing requires syntactic knowledge and therefore needs access to positional information. We can see that the **NO-POS** models are drastically worse than **ABSOLUTE** and **RELATIVE**. Thus, position encodings are necessary for the model to learn syntax, regardless of the morphological complexity or word order flexibility of the language. Particularly, **RELATIVE** fares the best among the three strategies, reflective of the lower losses seen in Figure 2.

We see similar trends for NER, where some languages have noticeably lower performance with **NO-POS** than others, while **RELATIVE** position encodings generally have the strongest performance. However, NER does not require a lot of syntactic or morphological understanding, apart from identifying the multi-word named entities, or those that may be inflected in some languages.

With SIB-200 we observe a different pattern. The gap between the three variants is generally not as pronounced as with the other tasks. SIB-200 is a classification task that does not require much understanding of syntax, making position encodings not essential.

On MultiBLiMP, languages like Basque and Tamil show roughly the same performance across the three model variants. For English, Hungarian, and Turkish, **RELATIVE** strongly outperforms **ABSOLUTE** and **NO-POS**. The MultiBLiMP results somewhat mirror the UD results, albeit with much higher scores for the **NO-POS** model variant (it is ultimately a sentence-level task). Since both datasets are highly dependent on both word order and morphology, they should be ideal candidates to show the results found by Ghosh et al. We will discuss this in the next section.

5 Analysis

5.1 Position Encodings and Morphological Complexity

We attempt to verify the findings of Ghosh et al. (2024), who find a significant *negative* correlation between TTR (as a proxy for morphological complexity) and the relative decrease in performance from removing positional encodings per language. The authors consistently find this across tasks (even with different languages per task). As mentioned, instead of removing positional encodings of an already trained model, we train tokenizers and models from scratch (Table 2). Figure 4 shows the relation between downstream performance and our proxies for morphological complexity.

We fail to see the same patterns as Ghosh et al.: there is no clear trend from language to language within the tasks. A likely explanation for this is that the models used by Ghosh et al., despite having position embedding set to zero post-hoc, still had the benefits of being pre-trained with position encodings. Pretraining from scratch without position encodings, as illustrated in Figure 2, results in a model that is not able to converge.

Moreover, Ghosh et al.’s correlations could be attributed to morphologically complex languages potentially being harder to model (e.g., Mielke et al., 2019; Park et al., 2021). This is shown in the relation between UD results and AV and MATTR in Figure 4. The performance with position encodings degrades with an increase in AV and MATTR. The performance without position encodings remains steady, albeit perhaps decreasing slightly with morphological complexity. This results in a smaller gap between **NO-POS** and models trained with position encodings for languages with higher MATTR and AV. Ghosh et al. attribute this to the effect of position encodings (this is likely where the signifi-

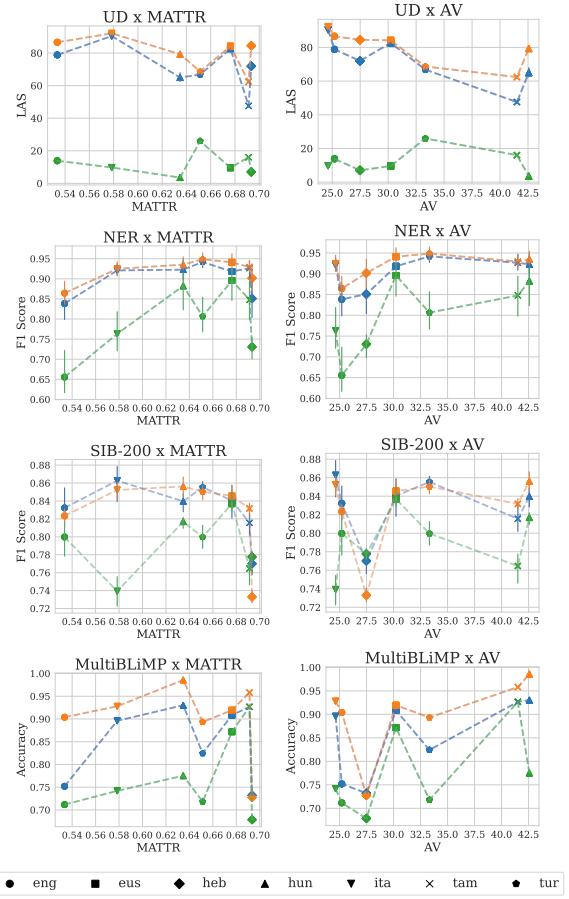


Figure 4 – Relation between proxies for morphological complexity and downstream performance. The line shows the groupings of positional encoding type.

cance in their statistical testing comes from), but it is potentially a result of morphologically complex languages performing worse.

Additionally, these findings are highly dependent on the languages studied and metrics used. Tamil and Basque, for example have similar MATTR of 0.69, but substantially different AVs: 41.50 and 30.22, respectively. The lack of a clear trend for *either* metric shows the impact of position encodings on model performance is not dependent upon the morphological complexity of the language.

Similarly, we observe comparable AV for both Hungarian and Tamil (42.57 and 41.50, respectively), but the entropic efficiency of their AV distribution (η) differs substantially: 0.368 and 0.445. This means that while the number of valid types to choose from is roughly the same from token to token, *how equally valid* those choices are is different (i.e., increased complexity; §3.1). Figure 5 shows the relation between task performance and efficiency, and again we fail to see a clear trend.

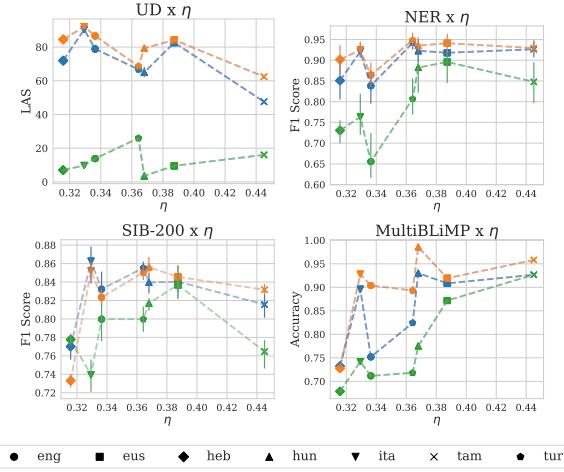


Figure 5 – Relation between entropic efficiency of the accessor variety and downstream performance.

5.2 Position Encodings and Word Order Flexibility

Ghosh et al. (2024) draw the conclusions that (1) languages with a higher TTR (more morphologically complex) show a smaller performance drop when removing positional encodings compared to those with a lower TTR; that (2) this is caused by morphologically complex languages having a more flexible word order; and therefore (3) languages with a more rigid word order benefit more from positional encodings.

As mentioned in §3 and shown in Table 1, metrics or proxies for morphological complexity and word order flexibility measure noticeably different phenomena. Therefore, drawing conclusions about word order flexibility from TTR alone is problematic. To validate their findings, we use gradient word order metrics (§3) and study their relation with downstream performance, shown in Figure 6.

We again fail to see the relation found by Ghosh et al.. Our results are strongly dependent on the task, but also on the choice of languages and metrics. Tamil, for example, has the highest SO-ROE in our sample, but the *lowest* HDE. This means that, while it has a very flexible word order for subjects and objects, this is *not* true for general word order flexibility.

As noted in §3.1, there is a mismatch in the granularity of the units used to calculate the metrics (words) and the units seen by a model (subword). Any variations in *word*-order may not be entirely reflected in the *subword*-order of a sentence. For example, a very long word may be able to shift positions in a sentence, but its subwords will have

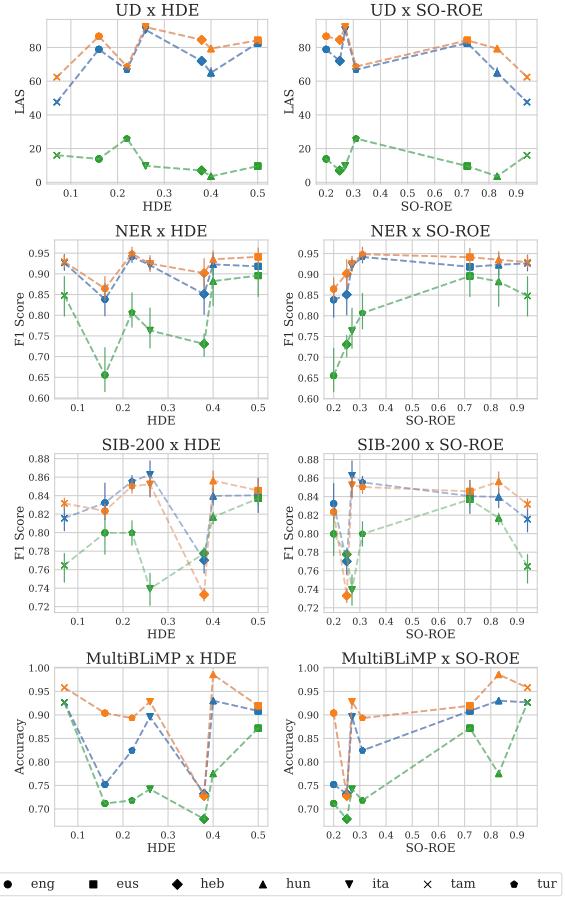


Figure 6 – Relation between proxies for word order flexibility and downstream performance. The connecting line is just to show groupings of positional encoding type, not a regression line.

more fixed positions. Thus, perhaps there is a need to adapt these proxies to tokens as well. We leave this investigation to future work.

5.3 Position Encodings and Task Performance

As noted from the results in Figure 3, the impact of position encodings is task specific. This is also observed by Ghosh et al., who find more variability in performance for syntactic tasks than semantic tasks. Token-level tasks like UD require more position information than sentence-level tasks like SIB-200. A clear example is seen with Basque. It has a high HDE and SO-ROE, indicating a language with very flexible word order. One would expect that position information would not matter much in learning its structure. However, with UD we observe a 40 point drop in LAS with **NO-POS**, as compared to **RELATIVE**, indicating that even with flexible word order, dependency structures are learnt better with position encodings. The variability decreases with NER, since the task does not require much syn-

tactic understanding, but some understanding of token positions in spans is needed. MultiBLiMP has more variability than NER between the position types, even though it is a sentence level task, since understanding of some syntactic structures is essential for predicting the grammaticality of a sentence. SIB-200 has the least variability in scores between **ABSOLUTE**, **RELATIVE**, and **NO-POS**, not being dependent on the learning of syntax.

Without position encodings, as Wang et al. (2020) note, a transformer model is basically a bag of words model. It does not have enough information to infer syntax, therefore **NO-POS** is still able to compete with **ABSOLUTE** and **RELATIVE** in tasks where the order of tokens is not essential. Wang et al. found absolute embeddings to outperform relative embeddings for sentence-level classification tasks. We do not observe this consistently across languages. Notably, with Tamil, Hungarian and Basque, **RELATIVE** performs better. Thus, in this aspect, conclusions made from English do not carry over to other languages. Perhaps a useful investigation for future work would be to find optimal task-specific position encoding strategies for different languages. Wang et al. also find relative encodings to be better for span prediction tasks, which is reflected in the NER results across languages.

5.4 Position Encodings and Syntax

Results from token-level tasks indicate that position encodings are essential to learning syntax, regardless of typological characteristics. We further verify this by probing the models on Bigram Shift, a syntactic task from the SentEval (Conneau et al., 2018) framework. We use UD datasets and leverage their annotations to create this probing task for each language (Vijaykumar and Tatariya, 2025). Using the setup from Tatariya et al. (2024a), we train logistic regression classifiers on the outputs from each layer of the models to understand whether syntactic information is can be recovered from the hidden representations. Figure 7 shows the results. The **NO-POS** models do not score higher than the majority baseline of 50% (Bigram Shift is a binary task) for most languages, clearly indicating that the model does not learn much syntactic information through the layers. Notably, both **ABSOLUTE** and **RELATIVE** for Hebrew, and **ABSOLUTE** for English also plateau at 50%, likely due to the lack of convergence as seen in Figure 2.

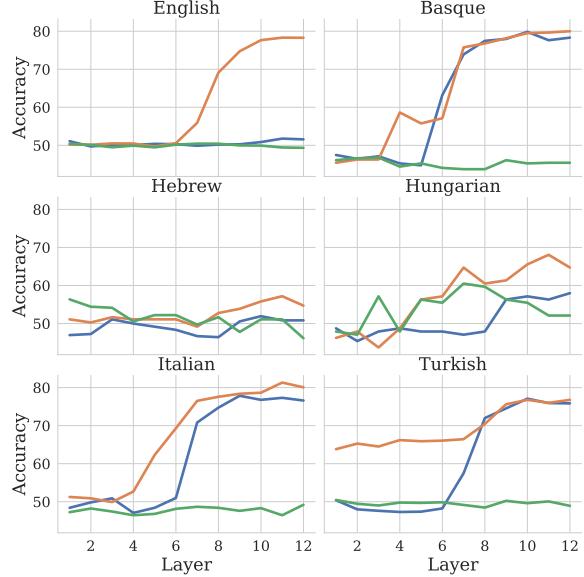


Figure 7 – Probing results for **ABSOLUTE**, **RELATIVE**, and **NO-POS** models. The Tamil dataset generated from UD annotations resulted in only 70 examples, so it is excluded from the plot.

6 Conclusion

We investigate the relation between positional encodings, morphological complexity, and word order flexibility. We systematically select typologically diverse languages in terms of morphological complexity and word order flexibility. For each of our seven languages, we pretrain three monolingual models, one with **ABSOLUTE** positional encodings, one with **RELATIVE**, and one without (**NO-POS**). We use multiple gradient proxies for both morphological complexity and word order flexibility and study their relation with the performance of the different models on various token- and sentence-level downstream tasks. Our models, tokenizers, and metrics are all based on the same dataset.

We find (1) the impact of position encodings is task specific, (2) relative position encoding gives the most consistent performance across languages and tasks, (3) position encodings are important to learn syntax, regardless of word order flexibility or morphological complexity, and (4) morphological complexity and word order flexibility do not have a direct interaction with position encodings, contradicting previous research.

For future work, we suggest developing a corpus-based proxy for word order flexibility that uses the same units that the model uses (i.e., tokenizer-based), thus removing potential confounds of domain shift and differing units.

Limitations

Language Coverage. We systematically select languages based on typological features in order to get a representative sample that covers the language characteristics we are interested in. Still, it is certainly possible we missed languages simply because they do not have enough data or are not included in the datasets we use.

Metrics. While we use gradient proxies for morphological complexity and word order flexibility instead of groupings, they are still *proxies*. No "perfect" measure for these phenomena exists.

Hyperparameters. The `NO-POS` models might be unfairly disadvantaged since we use hyperparameters that were arguably optimized for models with positional encodings. An exhaustive hyperparameter search was outside the scope and computational budget of this paper. Our main aim was to keep settings and metrics as comparable as possible.

Models. We use encoder-only models to study language phenomena, whereas the field is currently mainly using decoder-only models. We chose encoder-only models to directly compare findings from Ghosh et al. (2024), as well as to leverage the large body of research that investigates positional encodings in these models (see §2). Future work could look into architecture choices and language phenomena for decoder-only models.

Acknowledgements

We thank Coleman Haley for pointing us to Futrell et al. (2015). We thank the anonymous reviewers for their comments and suggestions. WP is funded by a KU Leuven Bijzonder Onderzoeksfonds C1 project with reference C14/23/096. The computational resources and services used were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government - department EWI.

References

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. **Word Order Does Matter and Shuffled Language Models Know It**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919. Association for Computational Linguistics.

David Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba Alabi, Yanke Mao, Haonan Gao, and En-Shiun Lee. 2024. **SIB-200: A Simple, Inclusive, and Big Evaluation Dataset for Topic Classification in 200+ Languages and Dialects**. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245. Association for Computational Linguistics.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. **The Belebele Benchmark: A Parallel Reading Comprehension Dataset in 122 Language Variants**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775. Association for Computational Linguistics.

Emi Baylor, Esther Ploeger, and Johannes Bjerva. 2024. **Multilingual Gradient Word-Order Typology from Universal Dependencies**. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 42–49. Association for Computational Linguistics.

Christian Bentz, Ximena Gutierrez-Vasques, Olga Sozanova, and Tanja Samardžić. 2022. **Complexity trade-offs and equi-complexity in natural languages: A meta-analysis**. *Linguistics Vanguard*, 9(s1):9–25.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. **What you can cram into a single \$&#!#* vector: Probing sentence embeddings for linguistic properties**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Michael A. Covington and Joe D. McFall. 2010. **Cutting the Gordian knot: The moving-average type-token ratio (MATTR)**. *Journal of Quantitative Linguistics*, 17(2):94–100.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. **Universal Dependencies**. *Computational Linguistics*, 47(2):255–308.

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. Association for Computational Linguistics.

Matthew S. Dryer. 2013. **Determining Dominant Word Order**. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*, (V2020.4). Zenodo.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. **Position Information in Transformers: An Overview**. *Computational Linguistics*, 48(3):733–763.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. **Quantifying Word Order Freedom in Dependency Corpora**. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100. Uppsala University, Uppsala, Sweden.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Poulami Ghosh, Shikhar Vashishth, Raj Dabre, and Pushpak Bhattacharyya. 2024. **A Morphology-Based Investigation of Positional Encodings**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21035–21045. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. **DeBERTa: Decoding-Enhanced Bert With Disentangled Attention**. In *International Conference on Learning Representations*.

C. F. Hockett. 1958. **A Course in Modern Linguistics**. *Language Learning*, 8(3-4):73–75.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. **XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization**. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML’20*, pages 4411–4421. JMLR.org.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. **Improve Transformer Models with Better Relative Position Embeddings**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3327–3335. Association for Computational Linguistics.

Tao Ji, Yuanbin Wu, and Xiaoling Wang. 2023. **Typology Guided Multilingual Position Representations: Case on Dependency Parsing**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13524–13541. Association for Computational Linguistics.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2025. **MultiBLiMP 1.0: A Massively Multilingual Benchmark of Linguistic Minimal Pairs**. *Transactions of the Association for Computational Linguistics*.

Carina Kauf and Anna Ivanova. 2023. **A Better Way to Do Masked Language Model Scoring**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 925–935. Association for Computational Linguistics.

Guolin Ke, Di He, and Tie-Yan Liu. 2021. **Rethinking positional encoding in language pre-training**. In *International Conference on Learning Representations*.

Paul Kiparsky. 1997. The rise of positional licensing. In *Parameters of Morphosyntactic Change*. Cambridge University Press.

Artur Kulmizev and Joakim Nivre. 2023. **Investigating UD treebanks via dataset difficulty measures**. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1076–1089, Dubrovnik, Croatia. Association for Computational Linguistics.

Natalia Levshina, Savithry Namboodiripad, Marc Allasonnière-Tang, Mathew Kramer, Luigi Talamo, Annemarie Verkerk, Sasha Wilmoth, Gabriela Garrido Rodriguez, Timothy Michael Gupton, Evan Kidd, Zoey Liu, Chiara Naccarato, Rachel Nordlinger, Anastasia Panova, and Natalia Stoyanova. 2023. **Why we need a gradient approach to word order**. *Linguistics*, 61(4):825–883.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Chou-Jui Hsieh. 2020. **Learning to Encode Position for Transformer with Continuous Dynamical Model**. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6327–6335. PMLR.

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**. *Preprint*, arXiv:1907.11692.

Sabrina J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. **What Kind of Language Is Hard to Language-Model?** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4975–4989. Association for Computational Linguistics.

Matti Miestamo. 2017. Linguistic diversity and complexity. *Lingue e Linguaggio*, XVI(2/2017):227–253.

Kanishka Misra. 2022. **Minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models**. *Preprint*, arXiv:2203.13112.

Matías Guzmán Naranjo and Laura Becker. 2018. Quantitative word order typology with UD. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, volume 155, pages 91–104. Linköping University Electronic Press.

Masato Neishi and Naoki Yoshinaga. 2019. **On the relation between position information and sentence length in neural machine translation**. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338, Hong Kong, China. Association for Computational Linguistics.

Julie Nijs, Freek Van de Velde, and Hubert Cuyckens. 2025. **Is Word Order Responsive to Morphology? Disentangling Cause and Effect in Morphosyntactic Change in Five Western European Languages.** *Entropy*, 27(1):53.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginster, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, Daniel Zeman, and 1 others. 2020. **Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection.** In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043. European Language Resources Association.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. **Cross-lingual name tagging and linking for 282 languages.** In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Hayley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. **Morphology Matters: A Multilingual Language Modeling Analysis.** *Transactions of the Association for Computational Linguistics*, 9:261–276.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Alal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. **The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale.** In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. **FineWeb2: One Pipeline to Scale Them All – Adapting Pre-Training Data Processing to Every Language.** *Preprint*, arXiv:2506.20920.

Esther Ploeger, Wessel Poelman, Miryam de Lhoneux, and Johannes Bjerva. 2024. **What is "Typological Diversity" in NLP?** In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5681–5700. Association for Computational Linguistics.

Esther Ploeger, Wessel Poelman, Andreas Holck Høeg-Petersen, Anders Schlichtkrull, Miryam de Lhoneux, and Johannes Bjerva. 2025. **A Principled Framework for Evaluating on Typologically Diverse Languages.** *Computational Linguistics*, pages 1–36.

Wessel Poelman, Thomas Bauwens, and Miryam de Lhoneux. 2025. **Confounding Factors in Relating Model Performance to Morphology.** In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7273–7298. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. **Massively multilingual transfer for NER.** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Vinit Ravishankar and Anders Søgaard. 2021. **The Impact of Positional Encodings on Multilingual Compression.** In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 763–777. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. **How good is your tokenizer? on the monolingual performance of multilingual language models.** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. **Masked Language Model Scoring.** In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. **Neural Machine Translation of Rare Words with Subword Units.** In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. **Self-Attention with Relative Position Representations.** In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. **Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little.** In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913. Association for Computational Linguistics.

Kaius Sinnemäki. 2014. **Global optimization and complexity trade-offs.** *Poznan Studies in Contemporary Linguistics*, 50(2):179–195.

Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Llatarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, and 1 others. 2023. **Grambank reveals the importance of genealogical constraints on linguistic**

diversity and highlights the impact of language loss. *Science Advances*, 9(16):eadg6175.

Anders Søgaard. 2022. **Should We Ban English NLP for a Year?** In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5260. Association for Computational Linguistics.

Sho Takase and Naoaki Okazaki. 2019. **Positional encoding to control output sequence length.** 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 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.

Kushal Tatariya, Vladimir Araujo, Thomas Bauwens, and Miryam de Lhoneux. 2024a. **Pixology: Probing the linguistic and visual capabilities of pixel-based language models.** In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3307–3320, Miami, Florida, USA. Association for Computational Linguistics.

Kushal Tatariya, Artur Kulmizev, Wessel Poelman, Esther Ploeger, Marcel Bollmann, Johannes Bjerva, Jiaming Luo, Heather Lent, and Miryam de Lhoneux. 2024b. **How Good is Your Wikipedia? Auditing Data Quality for Low-resource and Multilingual NLP.** *Preprint*, arXiv:2411.05527.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need.** In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Rohin Vijaykumar and Kushal Tatariya. 2025. Leveraging Universal Dependencies Annotations to Create Multilingual Probing Datasets.

Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. 2020. **On Position Embeddings in BERT.** In *International Conference on Learning Representations*.

Yu-An Wang and Yun-Nung Chen. 2020. **What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrette Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, and 1 others. 2020. **Transformers: State-of-the-Art Natural Language Processing.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. 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, Arofat Akhundjanova, Furkan Akkurt, Gabrielè Aleksandravičiūtė, Ika Alfina, Avner Algom, and 1 others. 2024. **Universal Dependencies 2.15.** *Preprint*, hdl:11234/1-5787.

A Experimental Setup

A.1 Language Sampling

We follow the procedure in Ploeger et al. (2025) to create a representative language sample of the features we are interested in. This allows us to make an informed choice about which and how many languages we use given our research question and computational budget. Our sampling starts with languages that are the intersection of (1) having enough data available in FineWeb-2 (Penedo et al., 2025), (2) being available in our downstream tasks of interest, (3) being available in Grambank (Skirgård et al., 2023), and (4) being available in the word order flexibility metrics of Futrell et al. (2015). We select the following word order related features from Grambank:

- GB024: What is the order of numeral and noun in the NP?
- GB025: What is the order of adnominal demonstrative and noun?
- GB065: What is the pragmatically unmarked order of adnominal possessor noun and possessed noun?
- GB130: What is the pragmatically unmarked order of S and V in intransitive clauses?
- GB131: Is a pragmatically unmarked constituent order verb-initial for transitive clauses?
- GB132: Is a pragmatically unmarked constituent order verb-medial for transitive clauses?
- GB133: Is a pragmatically unmarked constituent order verb-final for transitive clauses?
- GB134: Is the order of constituents the same in main and subordinate clauses?
- GB136: Is the order of core argument (i.e., S/A/P) constituents fixed?
- GB193: What is the order of adnominal property word and noun?
- GB203: What is the order of the adnominal collective universal quantifier ("all") and the noun?
- GB260: Can polar interrogation be indicated by a special word order?

These features create a feature vector per language, which is used to calculate pairwise distances. On these distances, we apply the sampling methods introduced by Ploeger et al. (2024, 2025). Figure 8 shows the results of the sampling procedures. We settle on seven languages that "saturate" (or cover) the features with the MaxSum objective: Basque, English, Hebrew, Hungarian, Italian, Tamil, and Turkish.

A.2 Model Training

All pretraining and finetuning hyperparameters are in Tables 3 and 4. We use `texieve`⁷ for pretraining and finetuning on UD, NER, and SIB-200. For the tokenizer we used a vocabulary size of 50k. All the training was run on one NVIDIA H100 GPU. Pretraining took approximately 16 hours for each `ABSOLUTE` and `NO-POS` model, and approximately 45 hours for each `RELATIVE` model, bringing it to a total of 530 hours of training for 21 models. Each finetuning run for UD took approximately 30 minutes (except Tamil, with a small dataset, which took 3 minutes); each SIB-200 run took 6 minutes; each NER run took 25 minutes. The calculation of the metrics took about 8 hours per language.

A.3 Datasets

Table 5 lists the details of the train/validation/test split sizes for each finetuning dataset. For WikiAnn we used the version of the dataset available on HuggingFace (Wolf et al., 2020) under `unimelb-nlp/wikiann`. For languages having more than one UD datasets, we select the ones with the highest quality as reported in Kulmizev and Nivre (2023).

⁷github.com/akulmizev/texieve

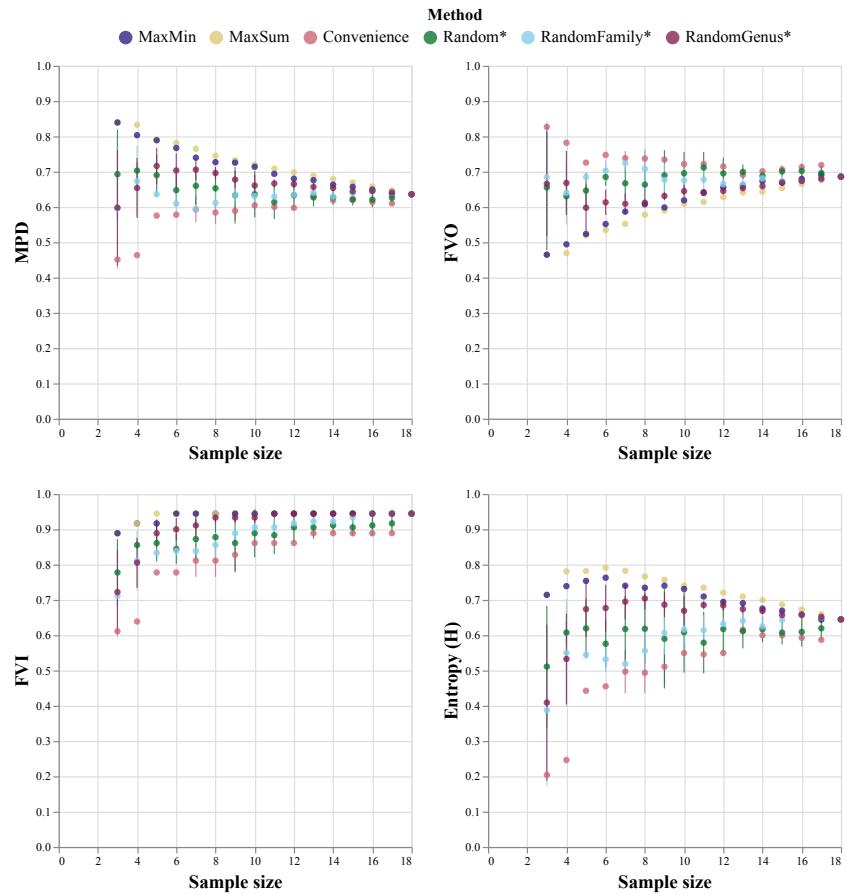


Figure 8 – Language sampling metrics for the Grambank features of interest. The features are "saturated" (when looking at FVI and Entropy) with about 6 or 7 languages with the MaxSum objective. \uparrow = higher is better, \downarrow = lower is better. The metrics are: **MPD** (\uparrow) = Mean Pairwise Distance, **FVI** (\uparrow) = Feature Value Inclusion, **FVO** (\downarrow) = Feature Value Overlap, **Entropy** (H ; \uparrow) = Shannon Entropy of feature vector.

Pretraining	
Model Type	roberta
Batch Size	128
Max Length	512
Train Steps	150k
Eval Steps	5k
Learning Rate	2×10^{-4}
Padding Strategy	longest
Mask Probability	0.12
Weight Decay	0.05
Gradient Accumulation Steps	4
Warmup Steps	1500

Table 3 – Pretraining hyperparameters.

Finetuning	
UD	
Max Length	512
Epochs	20
Batch Size	16
Learning Rate	5×10^{-5}
Padding Strategy	max_length
Warmup Steps	500
NER	
Max Length	512
Epochs	5
Batch Size	32
Learning Rate	5×10^{-5}
Padding Strategy	max_length
Warmup Steps	500
SIB-200	
Max Length	512
Epochs	20
Batch Size	32
Learning Rate	5×10^{-5}
Padding Strategy	max_length
Warmup Steps	500

Table 4 – Finetuning hyperparameters.

Dataset Splits			
	Train	Validation	Test
UD_Basque-BDT	5396	1798	1799
UD_English-EWT	12544	2001	2077
UD_Hebrew-HTB	5168	484	491
UD_Hungarian-Szeged	910	441	449
UD_Italian-IST	13121	564	482
UD_Tamil-TTB	400	80	120
UD_Turkish-Kenet	15398	1646	1643
wikiann-eu	10k	10k	10k
wikiann-en	20k	10k	10k
wikiann-he	20k	10k	10k
wikiann-hu	20k	10k	10k
wikiann-it	20k	10k	10k
wikiann-ta	15k	1k	1k
wikiann-tr	20k	10k	10k
SIB-200	701	99	99
MultiBLiMP-eus	-	-	278
MultiBLiMP-eng	-	-	770
MultiBLiMP-heb	-	-	2.3k
MultiBLiMP-hun	-	-	843
MultiBLiMP-ita	-	-	3k
MultiBLiMP-tam	-	-	382
MultiBLiMP-tur	-	-	1.7k

Table 5 – Dataset statistics. MultiBLiMP has a single split named "train" on Huggingface, but it is used as a test split.

B Additional Results

B.1 UD and Fertility

To better understand UD results, we look at [Rust et al. \(2021\)](#) who found a strong negative correlation between tokenizer fertility and performance on UD. We calculate fertility for all our tokenizers and plot their relation with LAS in Figure 9. We can observe a slight downward trend for [ABSOLUTE](#) and [RELATIVE](#) scores with increase in fertility. This trend is notably absent with [NO-POS](#) models.

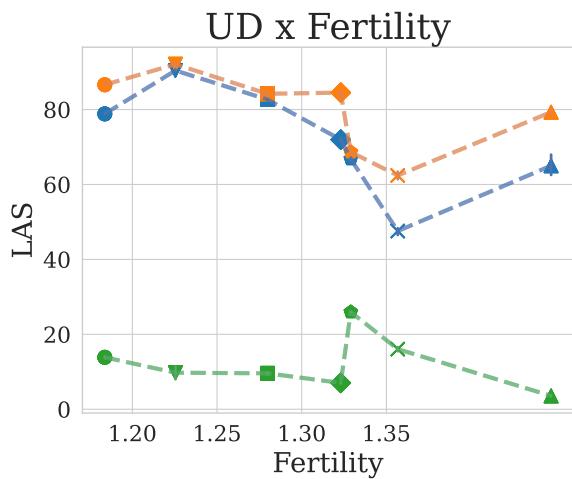


Figure 9 – Relation between tokenizer fertility and LAS for UD. The connecting line shows the groupings of positional encoding type. Fertility is calculated on the test split of FineWeb-2 (and a similar-sized sample of FineWeb for English that was also used for validation of the models).

Future work can look at the relation between positional encodings and tokenization. There are many interesting areas to explore in that direction, but they are outside of the scope of our current study.

B.2 Full Results

Language	Pos Type	UD	WikiAnn	SIB-200	MultiBLiMP
English	NO-POS	13.87 \pm 0.32	0.66 \pm 0.07	0.80 \pm 0.02	0.71
	ABSOLUTE	78.84 \pm 0.18	0.84 \pm 0.05	0.83 \pm 0.02	0.75
	RELATIVE	86.61 \pm 0.07	0.86 \pm 0.03	0.82 \pm 0.01	0.90
Basque	NO-POS	9.59 \pm 0.33	0.90 \pm 0.05	0.84 \pm 0.01	0.87
	ABSOLUTE	82.67 \pm 0.15	0.92 \pm 0.01	0.84 \pm 0.02	0.91
	RELATIVE	84.20 \pm 0.71	0.94 \pm 0.02	0.85 \pm 0.01	0.92
Hebrew	NO-POS	7.03 \pm 0.18	0.73 \pm 0.03	0.78 \pm 0.01	0.68
	ABSOLUTE	71.97 \pm 0.48	0.85 \pm 0.06	0.77 \pm 0.01	0.73
	RELATIVE	84.52 \pm 0.06	0.90 \pm 0.04	0.73 \pm 0.01	0.73
Hungarian	NO-POS	3.57 \pm 0.18	0.73 \pm 0.03	0.82 \pm 0.01	0.78
	ABSOLUTE	65.00 \pm 3.34	0.85 \pm 0.06	0.84 \pm 0.01	0.93
	RELATIVE	79.31 \pm 0.67	0.90 \pm 0.04	0.86 \pm 0.01	0.99
Italian	NO-POS	9.75 \pm 2.27	0.76 \pm 0.06	0.74 \pm 0.02	0.74
	ABSOLUTE	90.42 \pm 0.36	0.92 \pm 0.02	0.86 \pm 0.02	0.90
	RELATIVE	92.09 \pm 0.15	0.92 \pm 0.02	0.85 \pm 0.01	0.93
Tamil	NO-POS	16.05 \pm 0.71	0.85 \pm 0.06	0.76 \pm 0.02	0.93
	ABSOLUTE	47.57 \pm 1.16	0.93 \pm 0.02	0.82 \pm 0.02	0.93
	RELATIVE	62.40 \pm 1.32	0.93 \pm 0.02	0.83 \pm 0.01	0.96
Turkish	NO-POS	25.95 \pm 0.10	0.81 \pm 0.05	0.80 \pm 0.02	0.72
	ABSOLUTE	66.80 \pm 0.60	0.94 \pm 0.02	0.86 \pm 0.01	0.82
	RELATIVE	68.65 \pm 0.40	0.95 \pm 0.02	0.85 \pm 0.01	0.89

Table 6 – Results for UD (LAS), WikiAnn NER (F_1), SIB-200 (F_1), and MultiBLiMP (accuracy). UD, NER, and SIB-200 scores are an average of 5 runs with different random seeds. Results are the same as in Figure 3.