Model-Based Ranking of Source Languages for Zero-Shot Cross-Lingual Transfer

Anonymous ACL submission

Abstract

We present NN-RANK, an algorithm for ranking source languages for cross-lingual transfer, which leverages hidden representations from multilingual models and unlabeled target-language data. We experiment with two pretrained multilingual models and two tasks: part-of-speech tagging (POS) and named entity recognition (NER). We consider 51 source languages and evaluate on 56 and 72 target languages for POS and NER, respectively. When using in-domain data, NN-RANK beats state-of-the-art baselines that leverage lexical and linguistic features, with average improvements of up to 35.56 NDCG for POS and 18.14 NDCG for NER. As prior approaches can fall back to language-level features if target language data is not available, we show that NN-RANK remains competitive using only the Bible, an out-of-domain corpus available for a large number of languages. Ablations on the amount of unlabeled target data show that, for subsets consisting of as few as 25 examples, NN-RANK produces high-quality rankings which achieve 92.8% of the NDCG achieved using all available target data for ranking.

1 Introduction

Cross-lingual transfer, where knowledge from a *source* language is used to improve performance for a *target* language, extends the coverage of languages supported by natural language processing tools. Pretrained multilingual models (Devlin et al., 2019; Conneau et al., 2020a) are effective in this framework: the model is first finetuned on labeled data in the source language, and then evaluated directly on target language data. Models show strong performance, even in a zero-shot setting where no labeled examples from the target language are used (Hu et al., 2020; Ruder et al., 2021).

A critical element for successful transfer is the choice of source language, and while many prior works use only a single high-resource source language such as English, this has been called into question (Turc et al., 2021). An alternative is to rank all available source datasets for a given target language, as in LangRank (Lin et al., 2019), which uses lexical features (such as word overlap) as well as linguistic features (such as word order, syntactic, and phylogenetic information). Pairs of languages with more similar features are predicted to be better transfer languages. Other highly related prior works, which focus on analyzing how, and for which language pairs, these multilingual models are able to achieve strong cross-lingual transfer also focus on these features (Wu and Dredze, 2019; Pires et al., 2019; K et al., 2020; Dufter and Schütze, 2020; de Vries et al., 2022; Rice et al., 2025).

While these *static* features – which we define as those independent of the model being evaluated – are intuitive, we hypothesize that they cannot sufficiently describe the relationships between the representations of the languages contained within a pretrained multilingual model, and, as such, do not provide an adequate signal for choosing source languages. For example, a model may be able to strongly transfer between two languages with dissimilar static features, e.g., inverted word order or no lexical overlap, given sufficient pretraining data. Conversely, two languages with high lexical overlap or similar language-level features may yield poor transfer performance due to weak representations from e.g., low amounts of data. Consider a theoretically "perfect" source language for a specific target language; if the model is unable to encode the source language, e.g., due to script or vocabulary restrictions, strong cross-lingual performance can never be realized.

We hypothesize that hidden representations extracted from the intermediate layers of the model implicitly capture the complex interactions between the pretraining languages, beyond the abil-

ities of static features, and therefore serve as a stronger predictor of cross-lingual performance. In this work, we present Nearest Neighbor Rank (NN-RANK), a model and data-based approach to ranking source languages. Our experiments on part-of-speech tagging (POS) and named entity recognition (NER), using two popular multilingual models, multilingual-BERT (mBERT; Devlin et al., 2019) and XLM-RoBERTa (XLM-R; Conneau et al., 2020a), show that this approach beats LangRank – a state-of-the-art approach which relies on static lexical and linguistic features – in all settings, when using the unlabeled development sets to create the ranking. We include 51 source languages for both tasks, 56 target languages for POS, and 72 target languages for NER. One drawback of NN-RANK is the requirement for unlabeled data, which may not be available for all target languages in the domain of interest. This challenge is not faced by LangRank, which can fall back to language-level features. As such, we also experiment with rankings that are generated using out-of-domain data taken from the Bible, a corpus covering over 1600 languages commonly used for data-scarce languages (McCarthy et al., 2020), and find that NN-RANK remains competitive with LangRank. Finally, we conduct ablations on the amount of data required for NN-RANK, and show that for target languages with sufficient representation quality after pretraining, NN-RANK produces viable rankings with as little as 25 examples.

2 Related Work

Multilingual Model Analysis Since their release, there has been growing interest in analyzing pretrained multilingual models, in order to better understand what factors lead to strong cross-lingual transfer performance (Philippy et al., 2023). Lexical overlap – the percentage overlap of subwords between two languages – is often considered, though its importance is unclear: Pires et al. (2019) find no correlation with downstream performance, while Wu and Dredze (2019) find a positive correlation. Linguistic similarity, calculated using typological databases (Dryer and Haspelmath, 2013; Skirgård et al., 2023), has also been considered. Transfer performance is often better for more similar languages, particularly languages with similar word order (Pires et al., 2019; K et al., 2020; Dufter and Schütze, 2020; Littell et al., 2017).

Architectural properties have been shown to correlate with downstream performance. Deshpande et al. (2022) show that embedding similarity is correlated with zero-shot performance. Similarly, Conneau et al. (2020b) show that shared parameters in the lower layers of the model are important for multilingual representations, and Muller et al. (2021) show that these lower layers focus on aligning representations across languages. In a similar vein, Dou and Neubig (2021) find that word alignment performance (Och and Ney, 2000), calculated using vector similarity metrics, is strongest when using representations from middle layers of the model. NN-RANK is motivated by these works, and calculates similarity metrics between representations taken from intermediate layers.

Ranking Source Languages LangRank (Lin et al., 2019) is a learned model which uses lexical and linguistic features (Littell et al., 2017) to rank source languages for four supported tasks, including part-of-speech tagging (POS) and entity linking (EL). Ranking models are trained using pairs of valid source and target datasets, with ranking signal from a trained cross-lingual model. The final model is a gradient boosted decision tree (Ke et al., 2017). In our experiments, we use the provided POS and EL ranking models for POS and NER experiments, respectively. Rice et al. (2025) extend the application of LangRank to pretrained multilingual models.

3 NN-Rank

3.1 Method

The inputs to NN-RANK are an unlabeled dataset in the target language, a pool of unlabeled source datasets to be ranked, and a model which can encode the input text. While we describe the approach using task-specific datasets, NN-RANK can use any unlabeled data which represents the source or target languages of interest (see §7). For each target subword, the general approach is to find the k nearest neighbor subwords from the source pool and tally the source datasets which yielded these neighboring subwords. An example can be found in Table 5. The ranking is calculated by simply sorting source datasets by their final count in descending order. We describe the steps in detail below:

1a. **Encoding the Target Dataset** Each example from the unlabeled target dataset is inputted into the model, and for each subword, the representation from layer ℓ is extracted. Assum-

ing a model hidden dimension of h_d, this step yields a matrix $T \in \mathbb{R}^{N_{tgt} \times h_d}$, where N_{tgt} is the total number of subword representations extracted from the target dataset, omitting special tokens. We consider all target *tokens*, not target *types*, as the representations will depend on the context.

1b. **Encoding each Source Dataset** Let $P = [s_1, \ldots, s_n]$ define the pool of n source datasets, with each yielding $N_{s_1}, \ldots, N_{s_n}$ subwords, respectively, after tokenization. Repeat Step 1a. for each source dataset available. This yields a pool of available source subwords, which is represented as a single concatenated matrix $S \in \mathbb{R}^{N_{src} \times h_d}$, where $N_{src} = \sum_{i=0}^{n} N_{s_i}$. Define a function $m : \mathbb{R}^{h_d} \rightarrow \{s_1, \ldots, s_n\}$, which maps every source hidden representation to the dataset which yielded it.

2. **Calculating Nearest Neighbors** Create a tally C which maps each source dataset to a count initialized at 0. Iterate over the rows of T, i.e., every target subword, and find the k nearest neighbors from the rows of S. For each of these top-k neighboring representations, use m to lookup the origin dataset of the source representation, and for each dataset, increment its tally in C by 1.

3. **Calculating the Ranking** To calculate the final ranking for a given target dataset, sort the source datasets in C by their tally in descending order.

3.2 Hyperparameters

For all experiments, we use either mBERT or XLM-R as the encoding model, and set $h_d = 768$ and $\ell = 8$. We describe the distributions of each N_{tgt} and N_{s_i} in Tables 24–27. Nearest neighbor calculations are performed using FAISS (Douze et al., 2025), a vector database that performs efficient retrieval using the inner product.We set k to 5 for our main results, chosen empirically using the development set performance presented in Tables 16–17.

3.3 Considerations

Importantly, because all initial counts are set to 0, NN-RANK cannot rank any source dataset whose tokens do not appear as a nearest-neighbor, as the final count would remain 0. We highlight the rate of candidate discovery in Figure 8. Furthermore, we expect NN-RANK to work best for target languages which the model can encode with high-quality representations. However, we note that this may be a minor issue: if a target language is not represented well enough to produce an adequate ranking, model performance for that language on a downstream task will likely be poor regardless of source language selection. Furthermore, this is a benefit in the reverse direction: NN-RANK is unlikely to give a high rank to source datasets that the model cannot represent well – regardless of features such as linguistic similarity. This is useful as source datasets with poor representations are not likely to lead to good downstream performance. We discuss these trade-offs in §5.

4 Experimental Setup

We present experiments comparing various methods for ranking source datasets. First, we finetune each task model on every source training dataset, yielding one finetuned model per training dataset. Each finetuned model is then evaluated zero-shot on every target dataset and the performance – accuracy for POS and F1 for NER – is recorded. This yields *num. source datasets* × *num. target datasets* total scores for each pretrained model. Finally, we use these performances to evaluate each ranking method using the metrics described below. A small worked example and additional details are in Appendix A.

Tasks We focus on two tasks which allow for large scale evaluation: POS and NER. For POS, we use Universal Dependencies (UD; Nivre et al., 2020). For NER, we use the WikiANN dataset (Pan et al., 2017; Rahimi et al., 2019).

Languages For both datasets, the amount of data available for each language varies greatly. Therefore, we consider different language splits[1] – *all*, *medium*, and *large* – based on a minimum threshold number of examples, which defines both our pool of possible source datasets as well as target datasets included in the evaluation. Because we have no maximum threshold, each split builds upon the prior: $large \subset medium \subset all$, which allows for comparison across splits. Splits mark increasing difficulty, with *all* being the hardest; ranking becomes more difficult as the pool becomes larger,

[1]Split names reflect the amount of data available *per-dataset*. Datasets in *large* have the most data, but the split itself contains the smallest number of languages.

Source Split	Task Model	Ranking Method	POS *test-all* Avg. Acc.@5	POS *test-all* NDCG@5	NER *test-all* Avg. F1.@5	NER *test-all* NDCG@5
train-large	mBERT	NN-RANK-mBERT	**74.59**	**62.91**	60.77	**47.94**
		NN-RANK-XLM-R	73.49	58.92	**60.85**	46.35
		LangRank	70.60	36.47	57.48	28.55
		N-LangRank-mBERT	69.01	32.05	56.02	21.19
		N-LangRank-XLM-R	68.70	33.75	55.07	20.17
	XLM-R	NN-RANK-mBERT	**78.38**	**60.81**	60.74	47.16
		NN-RANK-XLM-R	77.75	60.69	**61.55**	**49.09**
		LangRank	76.00	37.32	58.14	33.02
		N-LangRank-mBERT	73.90	29.96	56.26	20.41
		N-LangRank-XLM-R	74.24	32.75	55.71	20.78
train-med	mBERT	NN-RANK-mBERT	**74.69**	**55.46**	60.78	**47.20**
		NN-RANK-XLM-R	72.97	50.76	**60.96**	45.35
		LangRank	68.98	24.61	57.54	27.50
		N-LangRank-mBERT	54.21	13.98	55.73	18.14
		N-LangRank-XLM-R	57.73	17.97	54.93	18.76
	XLM-R	NN-RANK-mBERT	**78.02**	**53.65**	60.86	45.07
		NN-RANK-XLM-R	76.82	52.45	**61.75**	**48.52**
		LangRank	73.18	26.22	58.35	32.07
		N-LangRank-mBERT	55.87	14.39	56.24	18.73
		N-LangRank-XLM-R	60.80	17.71	55.55	18.50
train-all	mBERT	NN-RANK-mBERT	**73.41**	**44.51**	**59.61**	**44.07**
		NN-RANK-XLM-R	71.46	41.07	57.97	38.88
		LangRank	58.67	8.95	57.09	25.93
		N-LangRank-mBERT	55.39	12.62	53.88	17.17
		N-LangRank-XLM-R	57.32	13.43	54.69	17.71
	XLM-R	NN-RANK-mBERT	**75.67**	**42.04**	**58.68**	**41.36**
		NN-RANK-XLM-R	74.46	39.84	57.61	39.95
		LangRank	60.13	8.89	57.17	29.68
		N-LangRank-mBERT	57.40	13.04	54.23	17.36
		N-LangRank-XLM-R	59.87	13.31	55.18	17.69

Table 1: Main Results. *Task Model* denotes the model which was finetuned and evaluated. *Ranking Method* denotes how the rankings were produced. The model used for hidden representations, in the case of NN-RANK and the model used for the training signal, in the case of N-LangRank, are denoted.

and target languages in *all* are more likely to be poorly represented by the model. We only consider languages which are supported by the released LangRank models: source datasets are limited to those in the model index, and target datasets are limited to the languages supported by lang2vec (Littell et al., 2017).

The UD dataset often provides various treebanks for the same language. Any dataset which meets the minimum threshold is included in the pool of source languages to be ranked; therefore our experiments using UD data are not ranking source *languages* but source *datasets*. The same rule applies to the target languages: any treebank which meets the evaluation threshold is included. As NER data is extracted from Wikipedia, there is only one dataset per language.

For the UD training data, the minimum thresholds are 500, 7500, and 15000 examples for the *all*, *medium*, and *large* splits respectively. For UD evaluation data, the thresholds are 100, 750, and 2000. For NER the thresholds are set to 1000, 10000, and 15000 for training, and 100, 1000, and 10000 for evaluation. In the most restrictive setting (i.e., *train-large x test-large*), we have 20 unique source languages and 21 target languages for POS (corresponding to 25 source and 25 target datasets), as well as 37 source languages and 34 target languages for NER. In the least restrictive setting (i.e., *train-all x test-all*), we have 51 source languages and 56 target languages for POS (corresponding to 78 source datasets and 118 test datasets), as well as 51 source languages and 72 target languages for NER. Both tasks include languages from up to 13 different language families; however the majority are Indo-European. Detailed information on all

languages can be found in Tables 18–21.

Ranking Methods We consider five different ranking methods in our main experiments. The pretrained LangRank model released by Lin et al. (2019) is used as a baseline. For this model, lexical features are taken from the development set. As LangRank always produces a ranking of all available source languages, for each source language split, we skip any language in the ranking which is not valid for that case. We also skip any source language with the same ISO code. We also consider two LangRank-based models trained from scratch (N-LangRank), using relevance scores calculated from the development set performance of either mBERT or XLM-R. We follow the general experimental setup of Lin et al. (2019) and train a different ranking model for each target language. Training examples are created by considering all available pairs of train and development set datasets – dependent on the language split – and excluding any source or target dataset which shares the same ISO code as the target language. For N target datasets, this yields N different models, each of which is used at test time. Lexical features from the target development set are used for inference.

We also consider two NN-RANK rankings, depending on if hidden representations are taken from mBERT or XLM-R. Source representations are calculated using the training split for each target task, and target representations use the development set. We set a limit of 1000 total input lines, and the source language split determines which datasets are included in the source pool P.

Task Models We consider two task models: the `base` versions of mBERT and XLM-R, as they show strong zero-shot POS and NER performance. We omit large language models from this work, as they are often pretrained on a fewer number of languages and may not encode all source or target languages with sufficient quality.

Metrics We use Normalized Discounted Cumulative Gain (NDCG; Järvelin and Kekäläinen, 2002) for ranking evaluation. We follow Lin et al. (2019) and assign a relevance score of γ_{max} to the top predicted transfer dataset, $\gamma_{max} - 1$ to the second predicted, and continue until the top-γ_{max} source datasets have a relevance score greater than 0. The other source datasets are given a score of 0.

We additionally implement performance-based metrics, *Average Accuracy@p* and *Average F1@p* for POS and NER, respectively. For a given target dataset, we average the accuracy or F1 scores achieved by the task models finetuned on the predicted top-p source datasets. We then average these scores across all target datasets to get the final value for the evaluation split (e.g., *test-all*). γ_{max} is set to 10, and p is set to 5 (Lin et al., 2019; Rice et al., 2025) for all metrics.

5 Results

We present a summary of results in Table 1, where we calculate metrics using test set performances. Because all rankings are created with development set data, this setting ensures that we do not evaluate on any data used to generate the rankings. NN-RANK greatly outperforms LangRank in every setting in terms of NDCG, highlighting the strength of rankings generated from model hidden states. The increase in performance-based metrics shows that these ranking differences also have a practical impact on model performance. While there is some variance across task and split depending on the pretrained model, we find that mBERT is often a stronger choice for ranking than XLM-R, even when XLM-R is used as the task model. This may be due to the pretraining data of mBERT, which uses Wikipedia for all languages. The similar writing style and domain may lead to better implicit alignments between languages during pretraining.

More specific results are available in the appendix: full results detailing all language splits can be found in Tables 8–9. Results which use development set performances can be found in Tables 24–27.

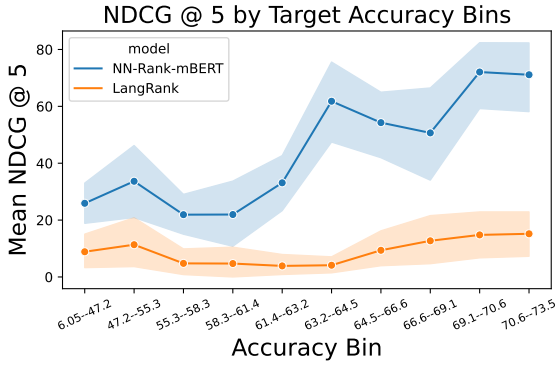

Figure 1: Mean Acc. of a target language over all source languages compared to NDCG for that target language. Each bin contains 10 languages, and the y-axis is the average NDCG for each bin. Shading represents the 95% confidence interval for NDCG scores.

5.1 Analysis

In this section, we analyze the main results and focus on the setting which uses the *all* split, and mBERT for both NN-RANK and evaluation.

5.1.1 Per-Language Performance

NN-RANK outperforms LangRank for almost every target dataset with the exception of 8 datasets covering Korean, Latin, Armenian and Turkish. When considering the true gold ranking, we find that a Finnish dataset is in the top-5 source datasets for 7 out of 8 of these datasets. This highlights the fact that NN-RANK is not only sensitive to the representation quality of the target dataset, but the source as well. In cases such as these, where the model cannot properly encode a high-performing source dataset, NN-RANK will fail to produce a strong ranking. This is supported by the fact that all 4 target languages are contained in the pretraining data of mBERT.

In Figure 1 we present the mean NDCG of target languages, ordered by their representation quality under mBERT. Here, we approximate quality by taking the mean POS accuracy achieved by mBERT on the target language, over all source languages. NN-RANK is sensitive to quality, where it predicts stronger rankings for languages that tend to be higher accuracy. LangRank, on the other hand, is invariant to this quality measure. This illustrates one fault of NN-RANK: it may struggle to find the best source datasets for poorly represented target datasets. However, for every bin, including the lowest accuracy bin, NN-RANK outperforms LangRank on average.

5.1.2 Quality of Highly Ranked Candidates

To measure the quality of highly ranked candidates for a given target dataset, we count the number of *poor* source candidates in the predicted top-5, which we consider analogous to false positives. Here, *poor candidates* are defined as those found in the bottom 15% of the gold ranking for a given target language. We present results in Figure 2, which shows that poor-quality source candidates are rarely ranked highly by NN-RANK, however the distribution is flat for LangRank.

We additionally consider the ranking distribution of source candidates with greater than 5% unknown tokens, presented in Figure 3. For all 5 datasets, NN-RANK consistently gives datasets with a high proportion of UNKs a lower rank than LangRank. This trend is most apparent for Coptic, a language where over 85% of tokens are UNK.

POS Tagging Results {train-all x test-all}			
Task Model	Ranking Method	Acc.@5	NDCG@5
mBERT	NN-RANK-mBERT	74.85	38.66
	NN-RANK-XLM-R	68.95	28.26
	LangRank	53.20	5.95
XLM-R	NN-RANK-mBERT	78.57	38.88
	NN-RANK-XLM-R	73.21	26.17
	LangRank	57.00	4.70
NER Results {train-all x test-all}			
Task Model	Ranking Method	F1@5	NDCG@5
mBERT	NN-RANK-mBERT	59.95	28.74
	NN-RANK-XLM-R	58.77	28.72
	LangRank	61.27	29.18
XLM-R	NN-RANK-mBERT	58.23	32.12
	NN-RANK-XLM-R	56.74	32.80
	LangRank	60.30	31.43

Table 2: Experiment 2 Results. Target language representations are taken from the Bible, while source representations are taken from the task-specific datasets.

6 Experiment 2: Ranking with no Target Task Dataset

While ranking approaches such as LangRank can fall back to language-level features, a drawback of NN-RANK is the requirement for unlabeled data in the target language. Relevant domain-specific data in the target language may be difficult to collect, particularly for data-scarce languages. For this experiment, we ask the following question: *How does* NN-RANK *perform if we assume that no in-domain data exists in the target language?* Here, we assume that the only data available for the target language is the Bible, a corpus which is currently available for around 1600 languages (McCarthy et al., 2020). While the Bible is often used for its coverage of languages, it has multiple drawbacks which include biased language, infrequently used vocabulary, and a limited domain – all of which may impact the performance of NN-RANK.

Experimental Setup These experiments largely follow the experimental setup of the main experiments. However, instead of using the unlabeled development set to extract model hidden representations, we use the associated Bible from the JHU Bible Corpus (McCarthy et al., 2020). The specific mapping from the target-task dataset to Bible can be found in Tables 22–23. As with the main experiments, representations for the source datasets are taken from the available training sets for each task. For fair comparison, we present results when using LangRank with lexical features from the Bible

as well. Performances however are not directly comparable to the main results, as not all train and evaluation languages have a corresponding Bible. For POS, there are 103 target datasets with Bibles, and 62 target datasets for NER.

Results We present a summary of results in Table 2, with full results in Tables 12–13. For POS, NN-RANK continues to outperform LangRank across both metrics, and using mBERT hidden representations offers the best ranking. For NER, ranking performances are much more mixed. When considering *Avg.F1*, LangRank is consistently stronger. However, for NDCG, LangRank is only stronger when mBERT is used. This indicates that NN-RANK may be sensitive to domain mismatch.

7 Experiment 3: General Rankings with No Task Datasets

For this experiment, we ask: *Can* NN-RANK *produce a general purpose ranking of source languages, without task or domain specific data in either the source and target languages?* This may be useful in cases where, for example, we wish to create an all-purpose ranking of source languages prior to having access to data in the domain of interest, such as in an online setting.

Experimental Setup We extend Experiment 2, and assume the only available data *for both source and target languages*, is from the Bible. We omit POS in this experiment, as this general ranking is at the language level, while the UD dataset has multiple train datasets for a single source language. Table 23 describes the Bible used for each language. For reference, we include LangRank results when using task-specific data in the target language. There are 46 source candidates in the *train-all* split, and again 62 target datasets.

Results Summary of results can be found in Table 3, and full results can be found in Table 14. For both task models, the *Avg. F1* is very close between the best NN-RANK ranking and LangRank, with a difference of 0.85 when evaluating with mBERT, and 0.74 for XLM-R. For both cases, using LangRank outperforms rankings generated using XLM-R, but rankings created using mBERT achieve the best performance. When considering NDCG scores however, NN-RANK – using either mBERT or XLM-R to generate representations – outperforms LangRank. Across both task models, the worst-performing NN-RANK ranking beats LangRank by 8.47 and 6.85 NDCG, respectively.

In summary, these results show that NN-RANK does not require in-domain data to create a strong ranking for the tasks in our experiments. They further indicate that NN-RANK works better when the source and target representations are calculated using data from more similar domains.

		NER *Test-all*	
Task Model	Ranking Method	F1.@5	NDCG@5
mBERT	NN-RANK-mBERT	61.23	36.07
	NN-RANKXLM-R	60.08	35.76
	LangRank	60.38	27.29
XLM-R	NN-RANK-mBERT	60.43	39.87
	NN-RANKXLM-R	58.70	37.67
	LangRank	59.69	30.79

Table 3: Experiment 3 Results. NN-RANK results take both source and target representations from the Bible. For reference, LangRank outputs using lexical features from the task-specific datasets are included.

8 Ablation Experiments

We conduct two ablation experiments, focused on (1) the layer at which model representations are taken and (2) the number of target subwords used for ranking. We use task-specific data for all ablation experiments.

8.1 Layer Ablation

For this ablation, we focus on the difference in ranking performance if we use hidden representations taken from Layer 8 and Layer 0 (embedding layer). A summary of results can be found in Table 4 with full results in Tables 11–10. In practically all cases, performance improves when using the intermediate layer, with large gains in NDCG. Using embedding representations only leads to better performance

POS Tagging Results {train-all x test-all}			
Task Model	Ranking Model	Δ Acc.@5	Δ NDCG@5
mBERT	mBERT	3.27	12.01
	XLM-R	1.17	7.57
XLM-R	mBERT	3.36	12.27
	XLM-R	1.55	7.68
NER Results {train-all x test-all}			
Eval Model	Ranking Model	Δ F1@5	Δ NDCG@5
mBERT	mBERT	2.07	10.53
	XLM-R	0.55	7.78
XLM-R	mBERT	1.24	5.61
	XLM-R	-0.26	4.02

Table 4: Layer Ablation Results. Positive scores indicate higher performance when using Layer 8.

when measuring *Avg. Acc* or *Avg. F1*, with the max difference across both tasks being less than -0.5. This result further highlights the weakness of static features; ranking quality improves as we move away from the embedding layer – the closest model-based feature to lexical overlap – and allow the model the encode and align the input sequences.

8.2 Target Data Ablations

Here, we are interested in understanding how the amount of target language data available influences ranking quality. We focus solely on NN-RANK performance using mBERT – as both the ranking and task model – for POS only. We discuss the limitations of these experiments in §9.

Experimental Setup For this experiment, we subsample the number of *target hidden representation* used, i.e., the rows of T, and consider sample sizes from between 10 and 2000. We take three different samples for each size, calculate a ranking from each, and consider the mean NDCG and *Avg. Acc* across each sample.

Results Full data ablation results can be found in Table 15. The *Avg. Acc* is surprisingly stable across all sample sizes, while the NDCG, on average, increases consistently with larger samples. This indicates that the best source datasets are found quickly, and the ranking quality improves with more target tokens. We present the distribution of performances in Figure 4, which compares the ranking performance using each subsample to the *main* ranking, which uses all available data (limited to 1000 input sequences); we also analyze the difference between each consecutive subsample in Figure 5. The former shows that even in the smallest setting, we recover over half of the source candidates predicted in the *main* top-5, when considering the median over all target languages. Similarly, when considering how the NDCG achieved using the subsamples compares to when we use all target data, the median using a sample size of 10 hidden representations is 85.6%. For subsamples of 25 and 50, the median is 92.8% and 97.4%, respectively.

To understand how a ranking can be created with so few target representations (see §3.3), we conduct a case study on French. The goal is to examine exactly how each sub-sampled target subword contributes to the ranking. Ideally, each target token will provide a large number of varied candidates to the total tally. We quantify the contribution of each target subword using two measures which describe the neighbor distribution. The first is target token *diversity*, defined as the average number of unique source datasets found in the top-5 nearest neighbors of a target token. The second is the *total number of unique source datasets* found across every instance of a target token in the sample. We provide additional details in Appendix D. We present results, calculating using the three samples of size 2000, in Figure 6 and a selected sample of neighbor results in Table 5. Here, we see that the majority of sampled tokens are diverse, with the largest bin of tokens yielding at least 3 different source datasets in their neighbors. Importantly, we also see that diversity is not correlated with frequency; while diverse tokens are more frequently found, frequency is not necessary for a token to be diverse. This applies to the number of unique source datasets as well: we see a broad range in the number of total unique neighbors across all token frequencies.

We hypothesize that the multilingual information encoded by each hidden representation is highly sensitive to the context. For example, as shown in Figure 7, we find that across all target languages, even punctuation tokens – the period and comma – yield a diverse set of neighbors and large number of source languages. As punctuation tokens do not carry any inherent multilingual meaning, differences in the neighboring tokens must come from changes in the context. The high average diversity and number of unique source datasets explains why a high-quality ranking can be created with only small number of target representations: each target subword in the sample contributes a large number of source candidates to the tally, and the contributed candidates change with context.

9 Conclusion

In this work, we present NN-RANK, a data-driven approach to ranking source languages for cross-lingual transfer, which leverages model hidden representations. The approach outperforms LangRank, and remains competitive when using out-of-domain data. Our results highlight a critical weakness of prior approaches to ranking, and general multilingual analysis: the use of *static* features, such as language-level or lexical features. These features fail to account for the *model itself*, and as such, cannot be used to sufficiently explain cross-lingual performance. We hope that our findings help motivate future work on better understanding the cross-lingual properties of multilingual models.

Limitations

Model Selection Decoder-only multilingual large language models (LLMs) such as BLOOM (Workshop et al., 2023) have also been proposed, and have shown strong cross-lingual performance. In this work, we choose to focus solely on encoder-only models. This choice was guided by the multilingual LLMs available at the time of experimentation, which are less multilingual than their encoder-only counterparts. For example, BLOOM only covers 46 languages and does not include particular high-resource languages that we expect may be helpful source languages like German. Furthermore, a method to rank source languages should ideally be lightweight and quick to run – for example, LangRank does not require any GPU resources, which makes it more applicable to a real-world setting.

We believe that using NN-RANK to evaluate and analyze multilingual LLMs is a promising direction for better understanding the dynamics of multilingual pretraining. However, due to the size of these models, the differences in pretraining procedures and objectives, as well as the various ways in which they can be used to achieve cross-lingual transfer, we believe that these experiments are better suited to dedicated future work.

Target Data Ablations For this experiment, we subsample the number of target hidden representations available independently across all available target tokens, i.e. sampling directly from T with no constraints. This is an unrealistic setting if we want to simulate the case where we only have, e.g., 10 sentences available in a dataset. Because representations are taken from Layer 8, the sampled vectors will change depending on the context. As such, 10 tokens sampled from the same input sequence will likely yield different results than 10 tokens each sampled from different contexts. For these reasons, results from the target data ablation cannot be extrapolated to cases where we have less input sequences than the sample size – especially for the smaller sample sizes.

Furthermore, we stress that the performance of this method relies on the quality of model representation for the target languages. While we show that the approach works with as little as 10 input sequences – considering the mean NDCG – this is qualified with the assumption that the representations are strong. A high-quality ranking using 10 input sequences should not be expected for, e.g., a low-resource language not contained in the pretraining data of the model.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging Cross-lingual Structure in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the Best of Cross-lingual Transfer: Evidence from POS Tagging with over 100 Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2022. When is BERT Multilingual? Isolating Crucial Ingredients for Cross-lingual Transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623, Seattle, United States. Association for Computational Linguistics.

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

Zi-Yi Dou and Graham Neubig. 2021. Word Alignment by Fine-tuning Embeddings on Parallel Corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss library. *Preprint*, arXiv:2401.08281.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (V2020.3)*. Zenodo.

Philipp Dufter and Hinrich Schütze. 2020. Identifying Elements Essential for BERT's Multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Abteen Ebrahimi and Katharina Kann. 2021. How to Adapt Your Pretrained Multilingual Model to 1600 Languages. 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 4555–4567, Online. Association for Computational Linguistics.

Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2022. Glottolog Database 4.7.

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. *Preprint*, arXiv:2003.11080.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. *Preprint*, arXiv:1912.07840.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing Transfer Languages for Cross-Lingual Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible Corpus: 1600+ Tongues for Typological Exploration. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First Align, then Predict: Understanding the Cross-Lingual Ability of Multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231, Online. Association for Computational Linguistics.

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

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics - ACL '00*, pages 440–447, Hong Kong. Association for Computational Linguistics.

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.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Towards a Common Understanding of Contributing Factors for Cross-Lingual Transfer in Multilingual Language Models: A Review. arXiv.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. 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.

Enora Rice, Ali Marashian, Hannah Haynie, Katharina von der Wense, and Alexis Palmer. 2025. Untangling the Influence of Typology, Data and Model Architecture on Ranking Transfer Languages for Cross-Lingual POS Tagging. *Preprint*, arXiv:2503.19979.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards More Challenging and Nuanced Multilingual Evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in*

Natural Language Processing, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, and 86 others. 2023. Grambank reveals the importance of genealogical constraints on linguistic diversity and highlights the impact of language loss. *Science Advances*, 9(16):eadg6175.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the Primacy of English in Zero-shot Cross-lingual Transfer. *Preprint*, arXiv:2106.16171.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, and 374 others. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *Preprint*, arXiv:2211.05100.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Appendix

A Additional Metric Details

We provide a small worked example for our experimental setup and how to calculate *Average F1*@2 (here $p = 2$ is used for simplicity) for XLM-R. Assume we have 3 source NER datasets, associated with English (en), Spanish (es), and French (fr). Assume we have 5 target NER datasets: Czech (cs), Igbo (ig), Irish (ga), Finnish (fi), and German (de). We first finetune XLM-R on each source dataset, yielding 3 finetuned models. Each model is then evaluated zero-shot on every target dataset, yielding 15 total pairs of *(source_dataset, target_dataset)* F1 scores.

For each target dataset, the selected ranking method generates an ordering of the 3 source datasets. Assume for German, the predicted ranking is `[English, French, Spanish]`. We then average the F1 scores for the English-finetuned model and French-finetuned model to get the average F1 for German. This process is repeated for the 4 other target datasets. To get the final *Average F1*@2, we average the resultant 5 averages.

NDCG As discussed in §3.3, NN-RANK may not always provide an ordering of all source datasets. As shown in the case study and target data ablations, in practice this is highly unlikely. However, in our evaluation, any unordered source dataset is given a rank of infinity, which yields a relevance score of 0, in order to maintain fair evaluation. This problem could be alleviated by initializing the tally with values that induce some default ordering; this represents a way in which linguistic features could be incorporated into NN-RANK.

B Model Training

We use established hyperparameters: a batch size of 32, learning rate of 2e-5, and train for 10 epochs (Ebrahimi and Kann, 2021) and assign labels to the last subword.

C Languages

We use Glottolog (Hammarström et al., 2022) to get language name and family information for each ISO code. To map between 2 letter and 3 letter ISO codes, we use the map provided by the LangRank implementation.

D Case Study: French and Wolof

For this analysis we use *token diversity*, defined as the number of unique source datasets found in the top-5 nearest neighbors of a given target token, averaged across every instance of the token. Therefore, the lower bound of target diversity is 0, and the upper bound is 5. We also measure the total number of unique source datasets discovered in the top-5 neighbors of a specific target token – summed across all instances. The lower bound of this value is 0, and the upper bound is 5 times the number of occurrences of the token (a token which appears once can maximally have 5 total unique neighbors, while a token that appears twice can have 10). Both metrics are required for a complete picture. A specific token may have low diversity (e.g., the nearest neighbors all come from the same source dataset), but a large number of unique source datasets (e.g., the source datasets of the neighbors change depending on the token context). Conversely, a token may have high diversity (e.g., all 5 neighbors come from a different source dataset), but a low number of unique source datasets (e.g., every instance of the token yields the same 5 source datasets). Counts are calculated across the three samples of size 2000.

French is a relatively high-resource language which is very closely related to English. In addition to the main case study, here we also include Wolof, a low-resource language not contained in the pretraining data of mBERT. Results can be found in Figure 6. Wolof tokens are less diverse, however the majority still yield on average two different source datasets. There is still no correlation between token frequency and diversity, and similar to French, Wolof target tokens still yield a large number of unique neighbor datasets.

E Figures

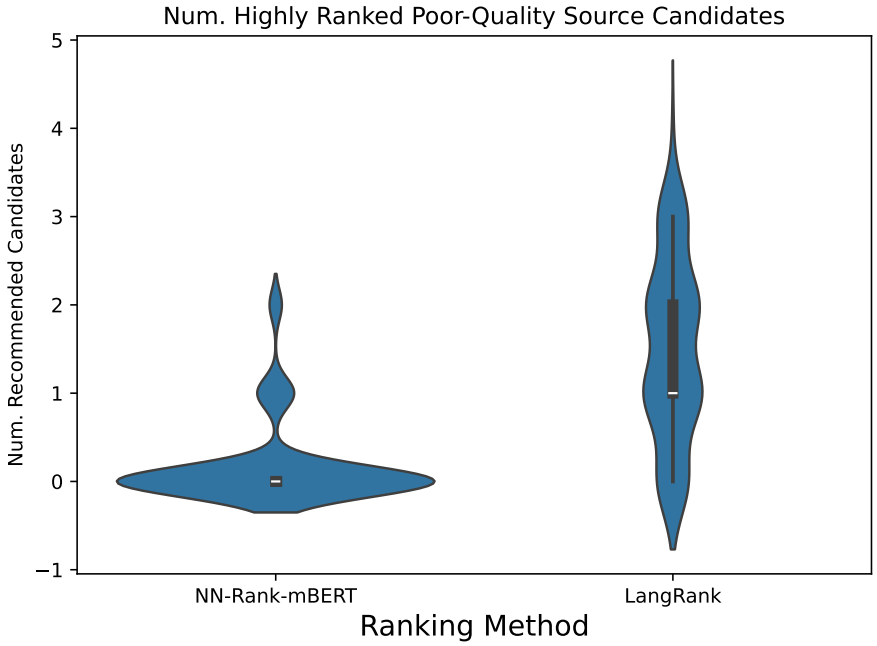

Figure 2: Number of Highly Ranked *Poor-Quality* Source Candidates. *Poor-quality* is defined as any source language in the bottom 15% of the gold ranking for each target language.

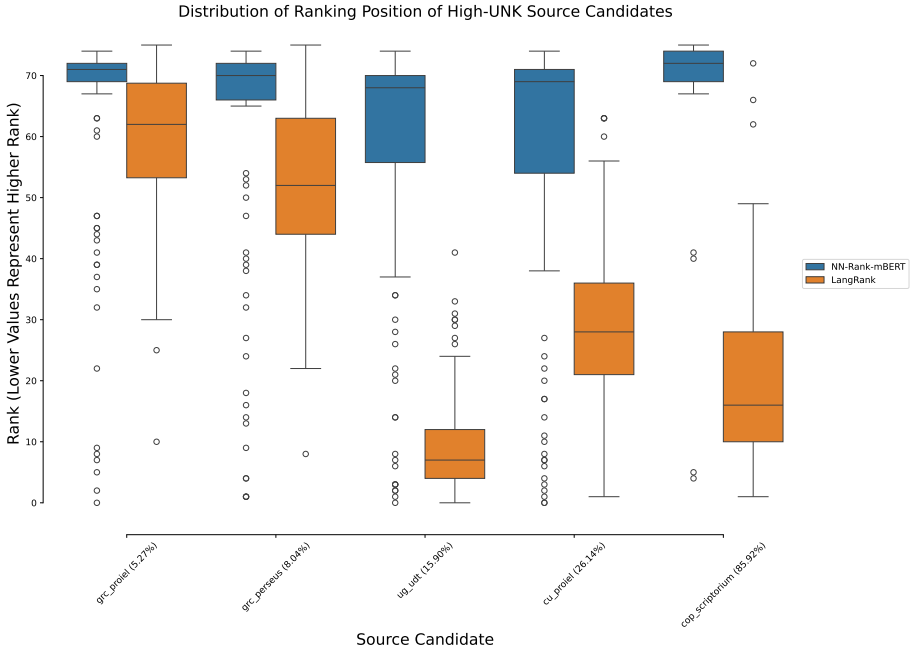

Figure 3: Distribution of Ranking Position for Source Candidates with High Unk Percentage. A rank position of 0 is used to mark the top-ranked candidate; in the figure, a lower value signifies that the ranking method gave the source candidate a higher rank. We consider the source candidates with greater than 5% UNK tokens, using the mBERT tokenizer.

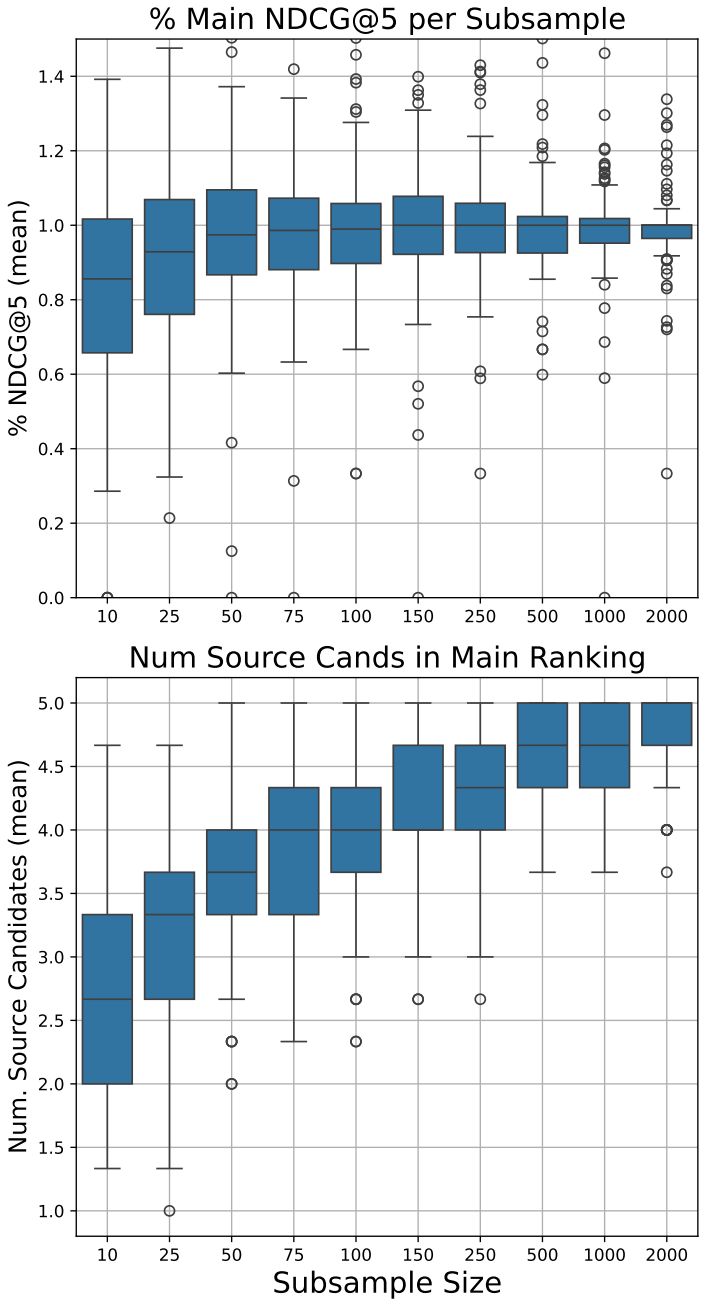

Figure 4: Data Ablation Results. Compares the performance from each subsample to the *Main* results, which use all available target data (limited to 1000 input sequences). Sample sizes represent the number of *target hidden representations*, i.e. *target tokens* used. The lower subplot considers the number of overlapping of source candidates in the top-5 predicted candidates.

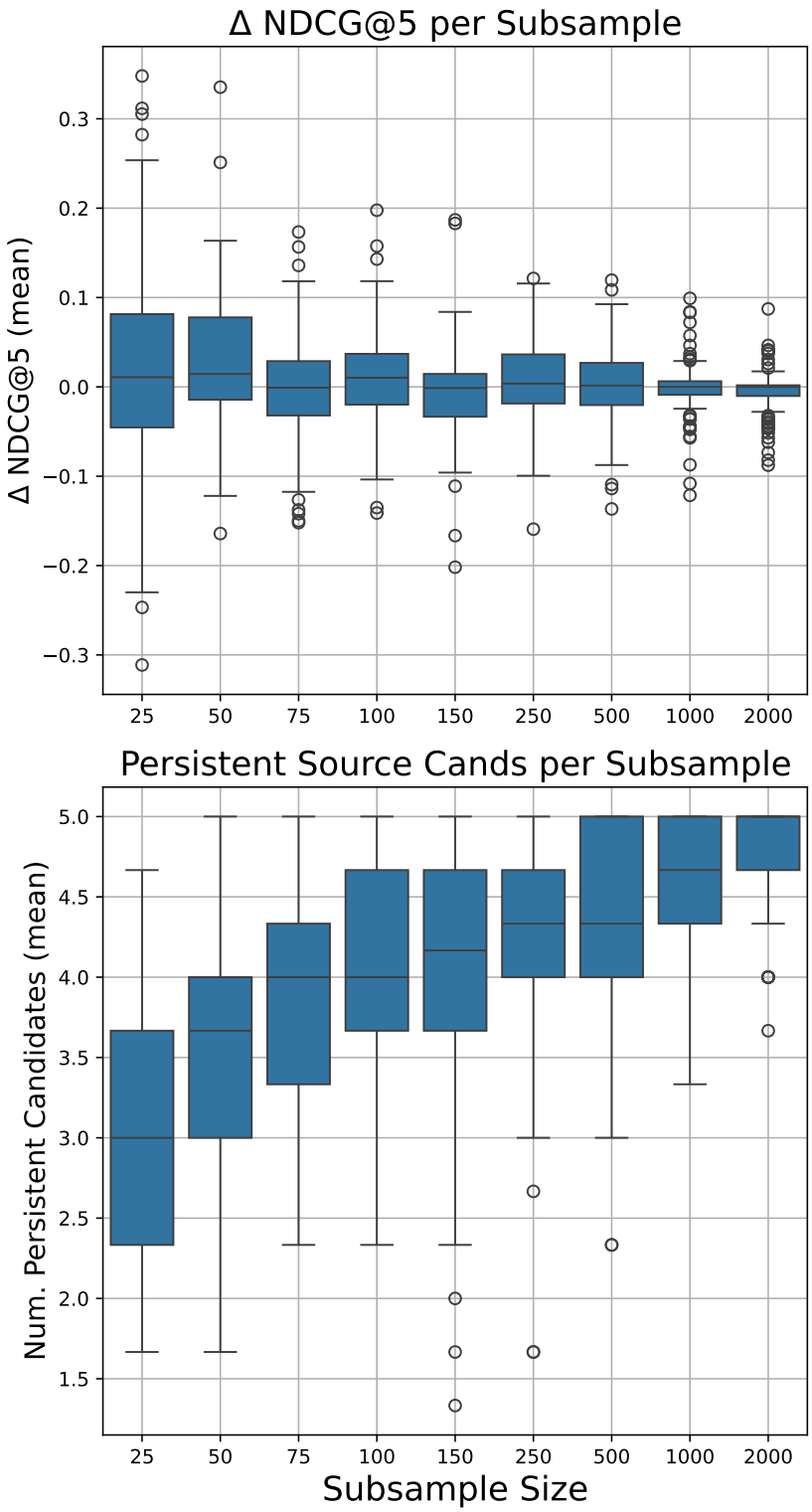

Figure 5: Data Ablation Results - Performance from one subsample to the next, omitting the first subsample of size 10. ΔNDCG refers to the change in NDCG per target language, from one subsample to the next. *Persistent Source Candidates* refers to the number of source candidates found in the top-5 predicted candidates of the subsample which were also in the top-5 predicted candidates of the previous subsample.

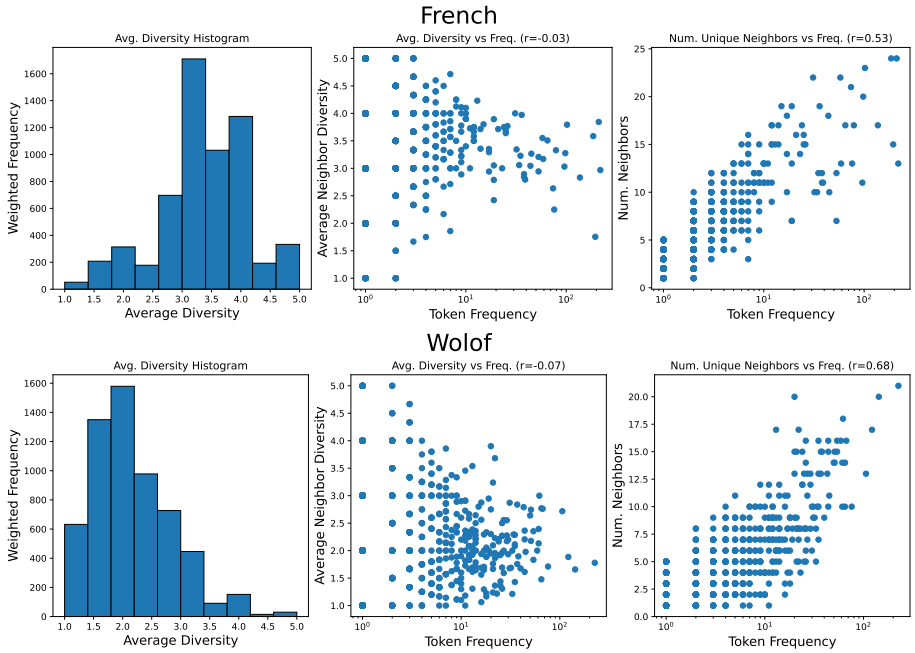

Figure 6: Case Study results for French. Token Frequency plotted using log scale. Pearson's r is used. We also include Wolof, as it is a low-resource and unseen language. Details provided in Appendix D.

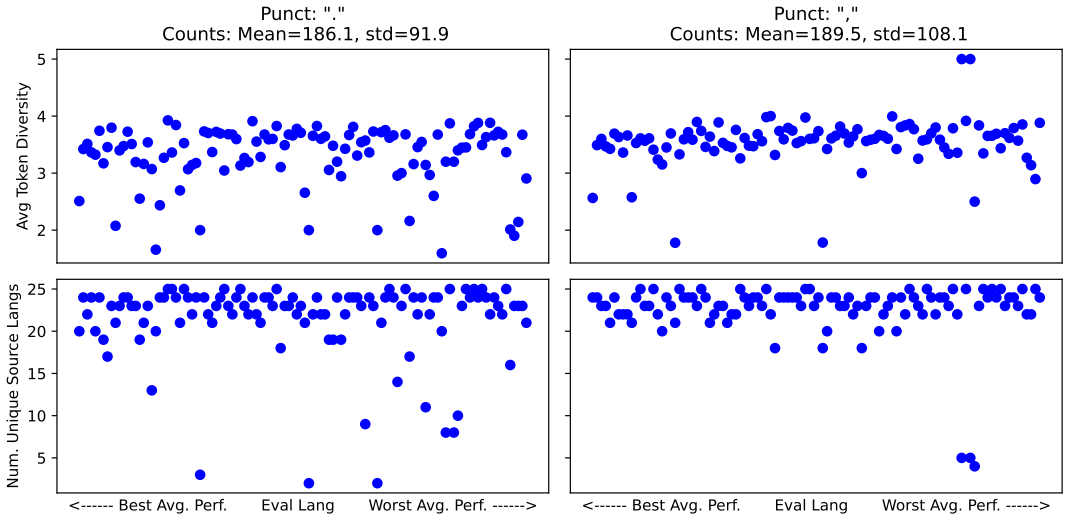

Figure 7: Diversity and total unique languages for "." and ",". The *x-axis* is a categorical ordering of each target language, ordered by average performance, which is calculated by averaging target language performance across all source datasets.

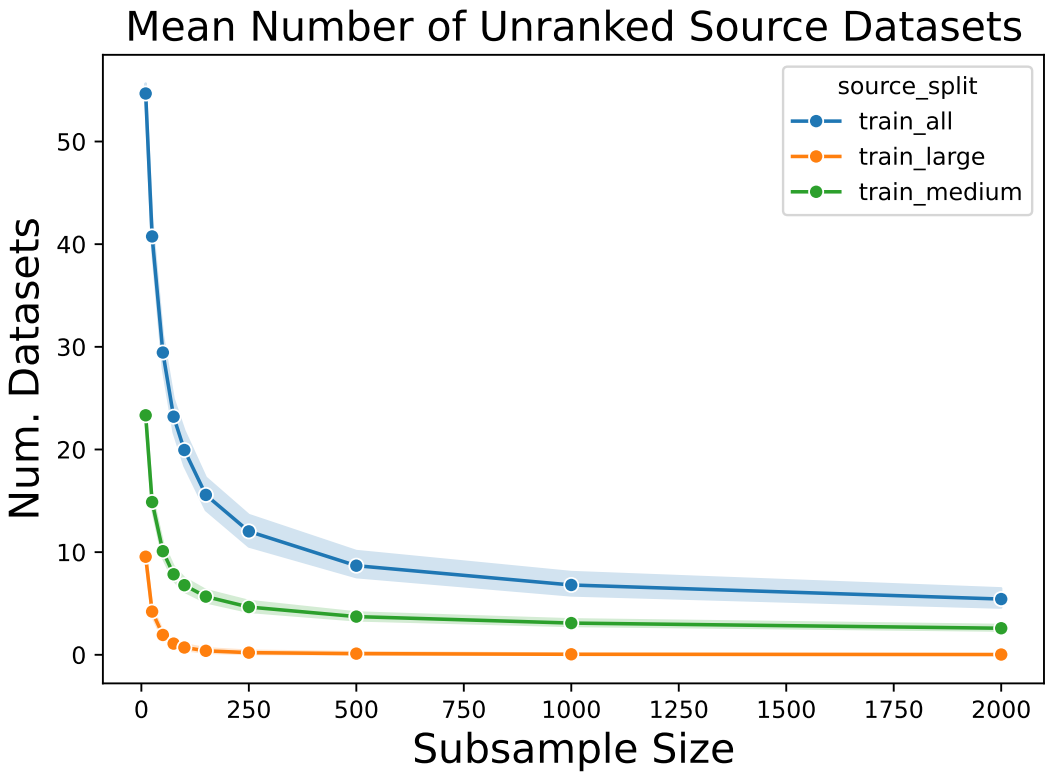

Figure 8: Number of unranked source datasets.

F Nearest Neighbor Examples

Target Dataset	Target Token	Top-5 Neighbors	Source File	Source Treebank
fr_gsd-ud-dev	['le']	['el']	ca_ancora-ud-train	UD_Catalan-AnCora
		['el']	es_ancora-ud-train	UD_Spanish-AnCora
		['il']	it_isdt-ud-train	UD_Italian-ISDT
		['il']	it_isdt-ud-train	UD_Italian-ISDT
		['il']	it_isdt-ud-train	UD_Italian-ISDT
	['.']	['.']	es_ancora-ud-train	UD_Spanish-AnCora
		['.']	es_gsd-ud-train	UD_Spanish-GSD
		['.']	it_isdt-ud-train	UD_Italian-ISDT
		['.']	ca_ancora-ud-train	UD_Catalan-AnCora
		['.']	it_isdt-ud-train	UD_Italian-ISDT
	['française']	['civil']	es_ancora-ud-train	UD_Spanish-AnCora
		[*Al Arabiya**]	ar_padt-ud-train	UD_Arabic-PADT
		['Wereldoorlog']	nl_lassysmall-ud-train	UD_Dutch-LassySmall
		['1918']	ar_padt-ud-train	UD_Arabic-PADT
		['Wereldoorlog']	nl_lassysmall-ud-train	UD_Dutch-LassySmall
	['reliant']	['liga']	pt_gsd-ud-train	UD_Portuguese-GSD
		['from']	en_ewt-ud-train	UD_English-EWT
		['entre']	es_gsd-ud-train	UD_Spanish-GSD
		['des']	ca_ancora-ud-train	UD_Catalan-AnCora
		['##ndo']	pt_gsd-ud-train	UD_Portuguese-GSD
	Maria	['Maria']	it_isdt-ud-train	UD_Italian-ISDT
		['Maria']	ro_nonstandard-ud-train	UD_Romanian-Nonstandard
		['Maria']	ro_nonstandard-ud-train	UD_Romanian-Nonstandard
		['Maria']	pt_gsd-ud-train	UD_Portuguese-GSD
		['Maria']	it_isdt-ud-train	UD_Italian-ISDT
	['.']	['.']	it_isdt-ud-train	UD_Italian-ISDT
		['.']	sl_ssj-ud-train	UD_Slovenian-SSJ
		['.']	it_isdt-ud-train	UD_Italian-ISDT
		['.']	ru_syntagrus-ud-train	UD_Russian-SynTagRus
		['.']	cs_cac-ud-train	UD_Czech-CAC
	['##ttes']	['##eras']	pt_gsd-ud-train	UD_Portuguese-GSD
		['##der']	no_nynorsk-ud-train	UD_Norwegian-Nynorsk
		['##s']	pt_gsd-ud-train	UD_Portuguese-GSD
		['##os']	pt_gsd-ud-train	UD_Portuguese-GSD
		['##ers']	nl_lassysmall-ud-train	UD_Dutch-LassySmall

Table 5: Example of nearest neighbors for French tokens. Arabic tokens are transliterated for the table and marked with an (*). Tokens and neighbors are calculated using mBERT representations. To calculate the ranking, we tally the number of occurences of each treebank or source dataset found in the right-most column, and sort them in decreasing order.

G Full Results Tables: Main Results

Source Split	Eval Model	Ranking Method	*test-large* Acc.@5	*test-large* NDCG@5	*test-medium* Acc.@5	*test-medium* NDCG@5	*test-all* Acc.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	**82.00**	**69.11**	**77.18**	**64.28**	**74.59**	**62.91**
		NN-Rank-XLM-R	80.43	65.99	76.17	60.92	73.49	58.92
		LangRank	77.54	40.90	73.80	40.52	70.60	36.47
		N-LangRank-mBERT	77.23	38.00	71.66	31.93	69.01	32.05
		N-LangRank-XLM-R	75.92	36.01	72.05	35.20	68.70	33.75
	XLM-R	NN-Rank-mBERT	**84.09**	65.47	**81.10**	62.70	**78.38**	**60.81**
		NN-Rank-XLM-R	83.11	**67.89**	80.51	**62.80**	77.75	60.69
		LangRank	82.07	42.55	79.22	41.65	76.00	37.32
		N-LangRank-mBERT	80.73	36.97	76.75	31.84	73.90	29.96
		N-LangRank-XLM-R	80.03	30.97	77.48	33.78	74.24	32.75
train-medium	mBERT	NN-Rank-mBERT	**82.56**	**63.45**	77.44	**57.76**	**74.69**	**55.46**
		NN-Rank-XLM-R	80.68	61.55	**75.72**	53.98	72.97	50.76
		LangRank	75.69	29.20	72.32	28.16	68.98	24.61
		N-LangRank-mBERT	62.98	20.49	56.02	14.41	54.21	13.98
		N-LangRank-XLM-R	65.95	21.72	61.38	19.33	57.73	17.97
	XLM-R	NN-Rank-mBERT	**84.42**	61.43	**80.86**	55.86	**78.02**	**53.65**
		NN-Rank-XLM-R	82.94	**64.00**	79.59	**56.81**	76.82	52.45
		LangRank	78.25	30.26	76.42	29.55	73.18	26.22
		N-LangRank-mBERT	63.92	19.27	58.02	15.29	55.87	14.39
		N-LangRank-XLM-R	68.46	20.94	64.46	19.13	60.80	17.71
train-all	mBERT	NN-Rank-mBERT	**81.46**	**54.51**	**75.92**	**47.12**	**73.41**	**44.51**
		NN-Rank-XLM-R	79.94	51.79	74.19	43.68	71.46	41.07
		LangRank	63.12	10.84	61.06	10.19	58.67	8.95
		N-LangRank-mBERT	65.29	19.92	57.64	13.38	55.39	12.62
		N-LangRank-XLM-R	66.81	15.87	60.90	14.17	57.32	13.43
	XLM-R	NN-Rank-mBERT	**82.63**	**51.44**	**78.07**	**45.20**	**75.67**	**42.04**
		NN-Rank-XLM-R	81.22	49.86	77.20	43.53	74.46	39.84
		LangRank	63.49	10.47	62.79	10.76	60.13	8.89
		N-LangRank-mBERT	66.08	16.40	59.88	14.42	57.40	13.04
		N-LangRank-XLM-R	68.83	15.45	63.37	13.99	59.87	13.31

Table 6: Main POS results – Full table highlighting ranking performances for all language split combinations.

Source Split	Evaluation Model	Ranking Method	*test-large* F1.@5	*test-large* NDCG@5	*test-medium* F1.@5	*test-medium* NDCG@5	*test-all* F1.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	69.41	53.98	67.17	50.95	60.77	47.94
		NN-Rank-XLM-R	69.29	51.88	67.10	49.15	60.85	46.35
		LangRank	65.82	29.76	64.01	30.74	57.48	28.55
		N-LangRank-mBERT	64.51	22.87	62.21	22.34	56.02	21.19
		N-LangRank-XLM-R	63.92	22.94	61.81	22.41	55.07	20.17
	XLM-R	NN-Rank-mBERT	68.29	61.21	66.35	53.69	60.74	47.16
		NN-Rank-XLM-R	68.31	60.09	66.43	53.80	61.55	49.09
		LangRank	64.03	39.33	62.72	36.95	58.14	33.02
		N-LangRank-mBERT	62.53	24.23	60.53	21.34	56.26	20.41
		N-LangRank-XLM-R	62.42	25.83	60.70	23.18	55.71	20.78
train-medium	mBERT	NN-Rank-mBERT	69.56	53.20	67.12	49.65	60.78	47.20
		NN-Rank-XLM-R	69.37	52.12	67.21	48.95	60.96	45.35
		LangRank	66.01	27.56	64.14	28.59	57.54	27.50
		N-LangRank-mBERT	63.78	20.37	61.80	20.19	55.73	18.14
		N-LangRank-XLM-R	63.98	22.18	61.43	21.52	54.93	18.76
	XLM-R	NN-Rank-mBERT	68.20	57.56	66.28	50.41	60.86	45.07
		NN-Rank-XLM-R	68.19	58.37	66.50	52.68	61.75	48.52
		LangRank	64.06	36.99	62.84	34.39	58.35	32.07
		N-LangRank-mBERT	61.98	21.47	60.39	19.52	56.24	18.73
		N-LangRank-XLM-R	62.43	25.41	60.41	21.91	55.55	18.50
train-all	mBERT	NN-Rank-mBERT	69.05	51.97	66.36	47.59	59.61	44.07
		NN-Rank-XLM-R	68.03	47.98	64.93	44.21	57.97	38.88
		LangRank	65.46	25.84	63.31	26.94	57.09	25.93
		N-LangRank-mBERT	61.95	19.38	59.90	19.32	53.88	17.17
		N-LangRank-XLM-R	64.08	21.74	61.27	20.38	54.69	17.71
	XLM-R	NN-Rank-mBERT	67.33	56.04	64.87	48.08	58.68	41.36
		NN-Rank-XLM-R	66.11	53.36	63.19	46.79	57.61	39.95
		LangRank	62.95	35.44	61.32	32.32	57.17	29.68
		N-LangRank-mBERT	59.95	20.10	58.27	18.25	54.23	17.36
		N-LangRank-XLM-R	62.55	25.19	60.08	21.08	55.18	17.69

Table 7: Main NER results – Full table highlighting ranking performances for all language split combinations.

H Full Results Tables: Standard Deviations

Source Split	Eval Model	Ranking Method	*test-large* Acc.@5	*test-large* NDCG@5	*test-medium* Acc.@5	*test-medium* NDCG@5	*test-all* Acc.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	9.56	24.05	15.81	23.16	17.71	22.87
		NN-Rank-XLM-R	11.05	29.39	16.18	26.86	18.09	26.48
		LangRank	11.40	26.21	16.22	24.49	17.41	23.75
		N-LangRank-mBERT	8.76	20.18	14.99	19.78	16.45	18.40
		N-LangRank-XLM-R	8.64	18.14	14.81	19.56	16.21	19.26
	XLM-R	NN-Rank-mBERT	9.03	27.39	12.54	24.00	15.61	23.93
		NN-Rank-XLM-R	9.57	27.35	12.55	26.19	15.40	25.74
		LangRank	8.70	27.16	13.23	23.23	15.70	23.23
		N-LangRank-mBERT	8.38	21.11	12.54	19.54	14.99	19.07
		N-LangRank-XLM-R	6.82	17.65	11.73	19.79	14.25	19.20
train-medium	mBERT	NN-Rank-mBERT	10.23	30.07	15.48	27.63	17.56	27.60
		NN-Rank-XLM-R	11.40	34.22	16.04	32.07	17.90	31.91
		LangRank	10.64	26.63	15.52	24.39	16.78	22.64
		N-LangRank-mBERT	14.38	18.03	14.73	17.85	14.95	17.75
		N-LangRank-XLM-R	10.46	17.35	13.98	18.05	15.22	17.65
	XLM-R	NN-Rank-mBERT	9.90	28.18	12.48	27.83	15.73	28.29
		NN-Rank-XLM-R	10.20	30.64	12.65	30.23	15.53	30.95
		LangRank	9.74	26.05	12.83	22.62	15.42	21.65
		N-LangRank-mBERT	15.34	18.31	14.50	17.93	14.89	17.68
		N-LangRank-XLM-R	11.38	16.08	13.13	17.92	14.79	17.73
train-all	mBERT	NN-Rank-mBERT	10.83	24.27	16.15	28.00	17.82	27.76
		NN-Rank-XLM-R	11.79	22.83	16.80	27.46	18.45	28.55
		LangRank	10.45	12.65	12.98	12.21	13.36	11.79
		N-LangRank-mBERT	14.16	19.25	15.70	17.65	15.85	17.79
		N-LangRank-XLM-R	10.61	16.81	14.10	17.65	15.17	16.56
	XLM-R	NN-Rank-mBERT	9.71	21.15	13.85	27.84	16.05	27.62
		NN-Rank-XLM-R	10.55	21.90	13.70	26.61	16.23	27.97
		LangRank	9.54	12.20	10.98	12.48	12.16	11.79
		N-LangRank-mBERT	14.92	17.69	14.84	17.32	15.35	17.03
		N-LangRank-XLM-R	11.69	15.57	13.27	17.26	14.66	16.61

Table 8: Main POS results – Full table showing standard deviations across *all target languages* for all language split combinations.

Source Split	Eval Model	Ranking Method	*test-large*		*test-medium*		*test-all*	
			Acc.@5	NDCG@5	Acc.@5	NDCG@5	Acc.@5	NDCG@5
train-large	mBERT	NN-Rank-mBERT	20.19	20.50	19.30	20.62	21.81	19.60
		NN-Rank-XLM-R	20.07	20.53	19.47	19.90	21.76	20.42
		LangRank	21.81	24.59	20.42	24.04	22.53	23.01
		N-LangRank-mBERT	20.78	17.34	19.55	17.39	21.58	16.47
		N-LangRank-XLM-R	21.39	20.00	20.27	18.87	22.25	17.16
	XLM-R	NN-Rank-mBERT	17.94	20.28	16.35	23.81	17.68	25.69
		NN-Rank-XLM-R	18.13	18.57	16.82	22.24	17.02	22.29
		LangRank	19.10	24.92	17.37	24.07	17.42	23.69
		N-LangRank-mBERT	18.57	17.55	16.97	17.68	16.86	17.38
		N-LangRank-XLM-R	18.58	19.99	16.73	19.09	17.04	18.06
train-medium	mBERT	NN-Rank-mBERT	20.19	19.64	19.31	20.59	21.70	19.71
		NN-Rank-XLM-R	20.07	19.84	19.56	19.93	21.77	21.56
		LangRank	22.03	26.26	20.68	24.48	22.77	23.60
		N-LangRank-mBERT	20.60	17.34	19.53	17.31	21.48	16.14
		N-LangRank-XLM-R	21.46	19.83	20.28	18.91	22.09	17.34
	XLM-R	NN-Rank-mBERT	17.89	20.82	16.13	23.67	17.28	24.52
		NN-Rank-XLM-R	18.15	18.92	16.59	21.68	16.75	21.60
		LangRank	19.55	26.39	17.76	25.93	17.58	24.91
		N-LangRank-mBERT	18.39	17.32	16.96	17.56	16.73	16.80
		N-LangRank-XLM-R	18.61	20.19	16.69	19.91	16.84	18.80
train-all	mBERT	NN-Rank-mBERT	20.29	20.64	19.58	21.57	21.93	20.72
		NN-Rank-XLM-R	19.99	21.84	19.89	22.07	22.23	22.72
		LangRank	21.92	25.77	20.82	24.38	22.67	23.03
		N-LangRank-mBERT	20.93	17.59	19.90	17.56	21.46	16.33
		N-LangRank-XLM-R	21.52	19.77	20.34	18.46	22.09	17.06
	XLM-R	NN-Rank-mBERT	18.19	21.28	16.70	23.98	18.30	25.43
		NN-Rank-XLM-R	18.31	22.79	17.28	23.68	17.86	24.30
		LangRank	19.16	25.88	17.71	25.39	17.37	24.51
		N-LangRank-mBERT	18.89	17.41	17.65	17.61	17.25	16.66
		N-LangRank-XLM-R	18.68	20.05	16.88	19.81	16.94	18.61

Table 9: Main NER results – Full table showing standard deviations across *all target languages* for all language split combinations.

I Full Results Tables: Layer Ablations

Source Split	Task Model	Ranking Method	*test-large* Acc.@5	*test-large* NDCG@5	*test-medium* Acc.@5	*test-medium* NDCG@5	*test-all* Acc.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	1.38	9.29	1.36	7.92	1.58	6.87
		NN-Rank XLM-R	0.06	9.47	0.13	4.50	0.40	3.63
	XLM-R	NN-Rank-mBERT	0.70	6.02	0.67	4.74	0.92	3.20
		NN-Rank XLM-R	-0.47	9.71	-0.26	4.74	-0.01	3.96
train-medium	mBERT	NN-Rank-mBERT	1.98	6.95	2.06	9.01	2.08	9.24
		NN-Rank XLM-R	0.94	6.43	0.74	4.27	0.99	3.62
	XLM-R	NN-Rank-mBERT	1.64	8.18	1.47	6.26	1.59	7.90
		NN-Rank XLM-R	0.47	9.69	0.38	4.39	0.75	3.90
train-all	mBERT	NN-Rank-mBERT	1.90	10.83	3.16	11.85	3.27	12.01
		NN-Rank XLM-R	-0.01	6.12	1.19	7.56	1.17	7.57
	XLM-R	NN-Rank-mBERT	2.42	12.16	2.95	11.28	3.36	12.27
		NN-Rank XLM-R	0.51	9.08	1.56	8.24	1.55	7.68

Table 10: Layer Ablation: POS results. Difference when using Layer 8 - Layer 0 (positive is better)

Source Split	Task Model	Ranking Method	*test-large* F1.@5	*test-large* NDCG@5	*test-medium* F1.@5	*test-medium* NDCG@5	*test-all* F1.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	0.24	9.59	0.70	9.32	0.65	7.65
		NN-Rank XLM-R	0.91	11.98	1.06	11.86	1.41	11.60
	XLM-R	NN-Rank-mBERT	0.38	8.32	0.84	7.67	0.21	5.21
		NN-Rank XLM-R	0.96	9.94	0.94	9.18	1.21	9.24
train-medium	mBERT	NN-Rank-mBERT	2.70	15.86	3.19	13.84	0.51	10.21
		NN-Rank XLM-R	1.47	13.71	1.87	13.18	2.08	12.87
	XLM-R	NN-Rank-mBERT	2.29	9.68	2.92	9.42	0.58	4.16
		NN-Rank XLM-R	1.58	13.39	1.54	11.44	1.41	10.74
train-all	mBERT	NN-Rank-mBERT	3.73	13.33	3.03	10.93	2.07	10.53
		NN-Rank XLM-R	0.60	8.36	-0.18	6.22	0.55	7.78
	XLM-R	NN-Rank-mBERT	3.59	11.45	3.14	7.63	1.24	5.61
		NN-Rank XLM-R	0.93	5.89	-0.22	4.03	-0.26	4.02

Table 11: Layer Ablation: NER tagging results. Difference when using Layer 8 - Layer 0 (positive is better).

J Full Results using Bible Data

			test-large		*test-medium*		*test-all*	
Source Split	Task Model	Ranking Method	Acc.@5	NDCG@5	Acc.@5	NDCG@5	Acc.@5	NDCG@5
train-large	mBERT	NN-Rank-mBERT	81.23	54.82	78.15	56.98	75.77	58.62
		NN-Rank-XLM-R	76.40	25.30	73.84	29.05	70.97	31.53
		LangRank	78.67	46.35	76.22	47.35	72.86	44.42
	XLM-R	NN-Rank-mBERT	84.05	55.17	82.61	57.05	79.87	58.45
		NN-Rank-XLM-R	80.22	25.12	78.93	26.11	76.07	29.44
		LangRank	82.95	46.95	81.77	49.31	78.45	47.26
train-medium	mBERT	NN-Rank-mBERT	80.80	45.79	77.37	46.73	74.70	43.73
		NN-Rank-XLM-R	74.75	18.90	72.48	21.09	70.13	23.00
		LangRank	77.29	36.27	74.48	35.97	71.59	33.61
	XLM-R	NN-Rank-mBERT	83.31	47.00	81.32	47.11	78.49	44.33
		NN-Rank-XLM-R	78.18	19.66	76.97	19.69	74.52	21.57
		LangRank	79.96	35.74	78.68	36.73	75.53	35.34
train-all	mBERT	NN-Rank-mBERT	81.90	48.03	77.12	40.94	74.85	38.66
		NN-Rank-XLM-R	75.18	35.97	71.45	30.43	68.95	28.26
		LangRank	55.72	4.92	55.68	5.82	53.20	5.95
	XLM-R	NN-Rank-mBERT	84.34	46.67	80.83	40.42	78.57	38.88
		NN-Rank-XLM-R	78.61	33.61	75.89	28.76	73.21	26.17
		LangRank	59.49	4.13	59.74	4.89	57.00	4.70

Table 12: Experiment 2: POS Results when No Target Task Data Available – Full Results. Here, the source datasets are from UD, and the target data is taken from the Bible.

Source Split	Task Model	Ranking Method	*test-large* F1.@5	*test-large* NDCG@5	*test-medium* F1.@5	*test-medium* NDCG@5	*test-all* F1.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	71.41	41.26	69.24	40.76	61.92	36.88
		NN-Rank-XLM-R	71.06	40.01	68.77	39.07	61.49	37.25
		LangRank	71.63	38.12	69.01	35.68	62.05	35.23
	XLM-R	NN-Rank-mBERT	69.46	48.19	67.16	42.61	61.06	39.93
		NN-Rank-XLM-R	69.20	50.67	67.03	44.32	60.93	42.31
		LangRank	69.78	45.87	67.44	41.35	61.87	38.26
train-medium	mBERT	NN-Rank-mBERT	70.92	36.06	68.64	35.97	61.75	33.40
		NN-Rank-XLM-R	71.04	37.57	68.55	37.76	61.65	35.95
		LangRank	70.99	31.94	68.51	30.21	61.89	31.31
	XLM-R	NN-Rank-mBERT	68.78	41.70	66.71	37.81	60.88	36.67
		NN-Rank-XLM-R	69.12	45.50	66.83	40.90	61.00	41.13
		LangRank	68.89	39.18	66.81	35.01	61.62	33.19
train-all	mBERT	NN-Rank-mBERT	69.61	31.94	67.06	31.87	59.95	28.74
		NN-Rank-XLM-R	68.99	33.38	66.43	32.64	58.77	28.72
		LangRank	70.02	29.21	67.86	28.77	61.27	29.18
	XLM-R	NN-Rank-mBERT	66.18	36.64	63.92	33.70	58.23	32.12
		NN-Rank-XLM-R	65.75	40.35	63.16	35.74	56.74	32.80
		LangRank	67.00	36.26	65.41	33.59	60.30	31.43

Table 13: Experiment 2: NER Results when No Target Task Data Available – Full Results. Here, the source datasets are from the Rahimi splits, and the target data is taken from the Bible.

Source Split	Task Model	Ranking Method	*test-large* F1.@5	*test-large* NDCG@5	*test-medium* F1.@5	*test-medium* NDCG@5	*test-all* F1.@5	*test-all* NDCG@5
train-large	mBERT	NN-Rank-mBERT	71.84	43.55	69.47	41.95	62.34	39.15
		NN-Rank XLM-R	71.33	45.26	69.17	43.27	61.98	40.27
		LangRank (Rahimi)	69.87	32.03	67.54	30.76	60.55	30.11
	XLM-R	NN-Rank-mBERT	69.68	51.44	67.54	46.57	61.68	43.98
		NN-Rank XLM-R	69.18	50.47	67.36	46.62	61.18	42.92
		LangRank (Rahimi)	67.54	40.92	65.70	36.85	60.39	34.37
train-medium	mBERT	NN-Rank-mBERT	71.56	42.78	69.11	40.37	61.98	36.86
		NN-Rank XLM-R	71.18	44.73	68.93	42.46	61.79	39.32
		LangRank (Rahimi)	70.06	29.94	67.73	28.97	60.69	28.99
	XLM-R	NN-Rank-mBERT	69.36	49.20	67.36	42.94	61.63	41.37
		NN-Rank XLM-R	69.10	49.63	67.30	44.84	61.08	41.64
		LangRank (Rahimi)	67.59	37.93	65.78	33.74	60.63	32.44
train-all	mBERT	NN-Rank-mBERT	71.62	43.06	68.85	40.57	61.23	36.07
		NN-Rank XLM-R	70.14	42.24	67.80	40.61	60.08	35.76
		LangRank (Rahimi)	69.37	27.86	67.09	27.53	60.38	27.29
	XLM-R	NN-Rank-mBERT	68.96	48.80	66.55	42.67	60.43	39.87
		NN-Rank XLM-R	67.54	46.69	65.48	42.56	58.70	37.67
		LangRank (Rahimi)	66.27	36.24	64.55	32.35	59.69	30.79

Table 14: Experiment 3 - General Rankings with No Task Dataset - NER Results Full. NN method uses the Bible as the source dataset and target dataset. LangRank takes lexical features from the Rahimi split (same as main results, slightly different source/target pools due to bible availability).

K Data Ablation Results

Source Split	Task Model	Ranking Model	Subsample Size	Avg. Acc@5		Avg. NDCG@5	
				Mean	STD	Mean	STD
large	mBERT	mBERT	10	73.56	0.0999	57.67	1.0486
			25	74.02	0.2591	59.48	0.5208
			50	74.48	0.2190	62.06	0.0999
			75	74.35	0.2800	61.95	0.7042
			100	74.44	0.0677	62.90	0.5349
			150	74.43	0.0522	62.22	0.5954
			250	74.46	0.0612	62.99	0.6453
			500	74.48	0.0679	63.26	0.3153
			1000	74.55	0.0324	63.16	0.3059
			2000	75.13	0.0314	63.05	0.1436
		XLM-R	10	72.64	0.2656	52.46	0.8792
			25	72.93	0.1687	55.80	0.7510
			50	72.89	0.1090	55.47	0.3483
			75	73.09	0.1661	57.04	0.8348
			100	73.33	0.1113	57.75	0.1754
			150	73.18	0.2106	57.39	0.9737
			250	73.32	0.1253	58.57	0.5841
			500	73.37	0.0942	58.30	0.2289
			1000	73.35	0.1191	58.46	0.1992
			2000	73.83	0.0841	58.77	0.3297
	XLM-R	mBERT	10	77.81	0.0988	55.75	0.4481
			25	78.01	0.1861	57.08	0.7400
			50	78.30	0.2015	60.14	0.5176
			75	78.23	0.2273	60.17	0.5350
			100	78.27	0.0507	60.74	0.8992
			150	78.36	0.0364	60.39	0.4339
			250	78.28	0.0775	60.53	0.7546
			500	78.31	0.0616	60.94	0.4437
			1000	78.38	0.0234	61.19	0.1991
			2000	79.01	0.0151	61.05	0.2984
		XLM-R	10	77.20	0.1125	53.67	1.4657
			25	77.42	0.1231	56.56	0.6499
			50	77.30	0.1159	56.25	0.7295
			75	77.56	0.1931	58.83	0.7268
			100	77.62	0.1107	59.39	0.4887
			150	77.55	0.1795	59.35	1.1807
			250	77.62	0.1000	60.19	0.3460
			500	77.70	0.1177	60.59	0.4153
			1000	77.66	0.1100	60.44	0.3318
			2000	78.17	0.0506	60.65	0.2407
medium	mBERT	mBERT	10	73.68	0.3690	50.08	1.9981
			25	74.22	0.3696	53.05	1.5703
			50	74.53	0.2225	55.32	0.4839
			75	74.53	0.2037	54.80	0.4711
			100	74.60	0.0304	55.91	0.5138
			150	74.61	0.1233	55.53	0.2667
			250	74.60	0.0907	55.79	1.1553
			500	74.73	0.0621	55.91	0.2530
			1000	74.75	0.0310	55.96	0.1029
			2000	75.32	0.1101	55.82	0.3875
		XLM-R	10	72.10	0.2370	44.67	1.2159
			25	72.69	0.2353	48.87	1.4164
			50	72.86	0.1843	49.16	0.2496
			75	72.74	0.1238	49.88	0.8218
			100	73.03	0.1312	50.18	0.3240
			150	73.05	0.3915	50.50	0.9673
			250	72.99	0.2487	50.03	0.3608
			500	72.98	0.0635	50.70	0.0975
			1000	72.93	0.0945	50.38	0.3225

Continued on next page

Source Split	Task Model	Ranking Model	Subsample Size	Avg. Acc@5		Avg. NDCG@5	
				Mean	STD	Mean	STD
medium	XLM-R	mBERT	2000	73.41	0.1078	50.64	0.1503
			10	77.40	0.2719	48.40	1.8551
			25	77.65	0.3435	51.10	1.7390
			50	77.89	0.1766	53.58	0.5910
			75	77.89	0.1505	53.20	0.8143
			100	77.95	0.0438	54.17	0.2357
			150	78.02	0.1706	54.10	0.5164
			250	77.92	0.0311	54.06	0.5838
			500	78.06	0.0560	54.17	0.2143
			1000	78.07	0.0402	53.92	0.2182
			2000	78.70	0.0901	54.08	0.0361
		XLM-R	10	76.10	0.3626	45.51	1.8007
			25	76.59	0.2389	48.73	1.6025
			50	76.68	0.1508	49.21	0.6561
			75	76.63	0.1566	50.16	0.5225
			100	76.88	0.1513	50.96	0.2359
			150	76.85	0.3256	51.24	0.9151
			250	76.86	0.2388	51.49	0.6931
			500	76.83	0.0592	51.89	0.3107
			1000	76.71	0.1380	51.26	0.5007
			2000	77.26	0.0542	51.80	0.4134
all	mBERT	mBERT	10	72.58	0.1915	39.17	1.4141
			25	72.78	0.5296	41.88	1.6804
			50	73.14	0.1272	43.94	0.3211
			75	73.31	0.1674	43.95	0.5767
			100	73.28	0.0378	43.88	0.7011
			150	73.32	0.0831	44.61	0.6676
			250	73.42	0.3133	44.29	0.6229
			500	73.30	0.2252	44.22	0.3289
			1000	73.35	0.0902	44.23	0.3403
			2000	73.88	0.0583	44.16	0.3167
		XLM-R	10	70.86	0.2847	35.36	1.6877
			25	71.14	0.5082	38.69	1.3061
			50	71.64	0.2958	41.12	0.6685
			75	71.50	0.2919	41.23	0.8403
			100	71.66	0.1017	42.16	0.8231
			150	71.59	0.0429	40.91	0.4555
			250	71.53	0.1310	41.40	0.3779
			500	71.43	0.2147	40.86	0.3017
			1000	71.43	0.2595	40.76	0.0447
			2000	71.84	0.0482	40.66	0.2772
	XLM-R	mBERT	10	74.98	0.2255	35.60	1.2733
			25	75.02	0.6532	39.23	1.5762
			50	75.33	0.0895	40.94	0.4181
			75	75.60	0.1585	41.25	0.5508
			100	75.30	0.1640	40.92	0.4279
			150	75.62	0.1676	41.48	0.9218
			250	75.50	0.2844	42.03	0.7664
			500	75.52	0.3166	41.30	0.2536
			1000	75.55	0.1301	41.68	0.2655
			2000	76.15	0.1159	41.58	0.1400
		XLM-R	10	73.90	0.3098	33.60	0.5184
			25	74.13	0.6088	36.48	1.2740
			50	74.66	0.4388	38.75	1.2252
			75	74.46	0.1867	39.36	0.5701
			100	74.64	0.0280	40.01	1.3144
			150	74.59	0.0899	39.33	0.3540
			250	74.55	0.0918	39.93	0.6842
			500	74.39	0.2216	39.55	0.3978
			1000	74.44	0.2580	39.21	0.1768
			2000	74.83	0.0892	39.23	0.4762

Table 15: All POS Data Ablation Results

L Development Set Results

Avg. of Accuracy @ 5			NN-Rank k			
Task Model	Source Split	Evaluation Split	5	10	20	25
mBERT	train-all	dev-all	72.54	72.49	72.52	72.42
		dev-large	78.78	78.81	78.85	78.85
		dev-medium	75.63	75.57	75.50	75.42
	train-large	dev-all	74.08	73.91	73.78	73.75
		dev-large	78.80	78.92	78.70	78.68
		dev-medium	77.11	76.92	76.75	76.73
	train-medium	dev-all	73.90	73.85	73.73	73.62
		dev-large	79.85	79.96	79.89	79.80
		dev-medium	77.05	76.96	76.80	76.68
XLM-R	train-all	dev-all	75.12	75.08	75.09	74.92
		dev-large	80.19	80.23	80.33	80.33
		dev-medium	78.19	78.15	78.12	78.01
	train-large	dev-all	78.06	77.93	77.83	77.80
		dev-large	81.56	81.69	81.55	81.53
		dev-medium	81.22	81.07	80.94	80.91
	train-medium	dev-all	77.45	77.36	77.18	77.11
		dev-large	82.11	82.21	82.15	82.07
		dev-medium	80.68	80.57	80.43	80.31
Avg. of NDCG @ 5			NN-Rank k			
Evaluation Model	Source Split	Evaluation Split	5	10	20	25
mBERT	train-all	dev-all	43.47	43.02	42.88	42.69
		dev-large	52.66	52.21	52.24	52.08
		dev-medium	46.30	45.89	45.73	45.49
	train-large	dev-all	61.43	60.90	60.12	59.75
		dev-large	69.46	69.15	68.36	67.88
		dev-medium	64.35	63.41	62.41	62.19
	train-medium	dev-all	53.81	53.67	53.39	52.94
		dev-large	63.60	63.57	64.04	63.94
		dev-medium	57.41	57.30	57.10	56.46
XLM-R	train-all	dev-all	41.42	41.16	41.19	40.91
		dev-large	50.19	49.73	49.84	49.68
		dev-medium	44.63	44.49	44.59	44.35
	train-large	dev-all	60.95	60.21	59.38	59.06
		dev-large	67.62	67.16	66.56	66.05
		dev-medium	64.20	63.24	62.24	61.87
	train-medium	dev-all	53.50	53.12	52.67	52.41
		dev-large	63.68	63.66	63.57	63.49
		dev-medium	57.29	56.91	56.58	56.19

Table 16: POS NN-RANK Development Set Results. Values are averaged over the two ranking models.

AVERAGE of F1@5			NN-Rank *k*			
Task Model	Source Split	Evaluation Split	5	10	20	25
mBERT	train-all	dev-all	58.41	58.09	57.72	57.64
		dev-large	68.41	68.19	68.06	67.98
		dev-medium	65.51	65.13	64.91	64.86
	train-large	dev-all	60.33	60.20	60.10	60.07
		dev-large	69.22	69.11	69.01	68.99
		dev-medium	67.07	66.98	66.87	66.83
	train-medium	dev-all	60.43	60.26	60.25	60.23
		dev-large	69.34	69.18	69.10	69.05
		dev-medium	67.08	66.89	66.81	66.77
XLM-R	train-all	dev-all	58.00	57.61	57.13	57.06
		dev-large	66.64	66.40	66.20	66.18
		dev-medium	63.96	63.52	63.26	63.24
	train-large	dev-all	60.84	60.61	60.42	60.37
		dev-large	68.21	68.14	68.01	67.97
		dev-medium	66.28	66.19	66.03	65.98
	train-medium	dev-all	61.09	60.90	60.84	60.82
		dev-large	68.12	67.94	67.92	67.88
		dev-medium	66.31	66.04	65.98	65.94
Avg. of NDCG@5			NN-Rank *k*			
Task Model	Source Split	Evaluation Split	5	10	20	25
mBERT	train-all	dev-all	40.91	40.57	39.89	39.37
		dev-large	50.46	50.19	49.93	49.41
		dev-medium	45.76	45.34	44.80	44.35
	train-large	dev-all	46.09	45.49	45.10	44.92
		dev-large	53.33	52.58	52.93	52.90
		dev-medium	50.48	49.68	49.31	48.99
	train-medium	dev-all	45.39	44.85	45.06	44.81
		dev-large	52.78	51.96	52.18	51.98
		dev-medium	49.19	48.45	48.35	48.09
XLM-R	train-all	dev-all	41.36	40.90	40.14	40.15
		dev-large	54.75	54.68	53.88	53.90
		dev-medium	47.43	46.97	46.28	46.35
	train-large	dev-all	47.35	46.48	46.26	45.90
		dev-large	60.04	59.25	59.05	58.70
		dev-medium	53.54	52.98	52.85	52.50
	train-medium	dev-all	46.08	45.90	45.79	45.73
		dev-large	57.79	57.49	57.58	57.65
		dev-medium	51.41	50.97	50.77	50.77

Table 17: NER NN-RANK Development Set Results. Values are averaged over the two ranking models.

M Language Tables

UD_ISO	ISO-3	Language Family	Treebank	Language Splits
af	afr	Indo-European	UD_Afrikaans-AfriBooms	train_all
hy	hye	Indo-European	UD_Armenian-ArmTDP	train_all
eu	eus	-	UD_Basque-BDT	train_all
zh	zho	Sino-Tibetan	UD_Chinese-GSD	train_all
cop	cop	Afro-Asiatic	UD_Coptic-Scriptorium	train_all
cs	ces	Indo-European	UD_Czech-CLTT	train_all
da	dan	Indo-European	UD_Danish-DDT	train_all
en	eng	Indo-European	UD_English-LinES	train_all
en	eng	Indo-European	UD_English-ParTUT	train_all
fr	fra	Indo-European	UD_French-ParTUT	train_all
fr	fra	Indo-European	UD_French-Sequoia	train_all
gl	glg	Indo-European	UD_Galician-CTG	train_all
gl	glg	Indo-European	UD_Galician-TreeGal	train_all
got	got	Indo-European	UD_Gothic-PROIEL	train_all
el	ell	Indo-European	UD_Greek-GDT	train_all
he	heb	Afro-Asiatic	UD_Hebrew-HTB	train_all
hu	hun	Uralic	UD_Hungarian-Szeged	train_all
ga	gle	Indo-European	UD_Irish-IDT	train_all
it	ita	Indo-European	UD_Italian-ParTUT	train_all
it	ita	Indo-European	UD_Italian-PoSTWITA	train_all
ko	kor	Koreanic	UD_Korean-GSD	train_all
la	lat	Indo-European	UD_Latin-Perseus	train_all
sme	sme	Uralic	UD_North_Sami-Giella	train_all
fa	fas	Indo-European	UD_Persian-Seraji	train_all
pl	pol	Indo-European	UD_Polish-LFG	train_all
ru	rus	Indo-European	UD_Russian-GSD	train_all
sr	srp	Indo-European	UD_Serbian-SET	train_all
sk	slk	Indo-European	UD_Slovak-SNK	train_all
sl	slv	Indo-European	UD_Slovenian-SST	train_all
sv	swe	Indo-European	UD_Swedish-LinES	train_all
sv	swe	Indo-European	UD_Swedish-Talbanken	train_all
ta	tam	Dravidian	UD_Tamil-TTB	train_all
tr	tur	Turkic	UD_Turkish-IMST	train_all
ur	urd	Indo-European	UD_Urdu-UDTB	train_all
ug	uig	Turkic	UD_Uyghur-UDT	train_all
vi	vie	Austroasiatic	UD_Vietnamese-VTB	train_all
grc	grc	Indo-European	UD_Ancient_Greek-Perseus	train_all, train_medium
grc	grc	Indo-European	UD_Ancient_Greek-PROIEL	train_all, train_medium
ar	ara	Afroasiatic	UD_Arabic-NYUAD	train_all, train_medium
bg	bul	Indo-European	UD_Bulgarian-BTB	train_all, train_medium
hr	hrv	Indo-European	UD_Croatian-SET	train_all, train_medium
cs	ces	Indo-European	UD_Czech-FicTree	train_all, train_medium
nl	nld	Indo-European	UD_Dutch-Alpino	train_all, train_medium
fi	fin	Uralic	UD_Finnish-FTB	train_all, train_medium
id	ind	Austronesian	UD_Indonesian-GSD	train_all, train_medium
ja	jpn	Japonic	UD_Japanese-BCCWJ	train_all, train_medium
ja	jpn	Japonic	UD_Japanese-GSD	train_all, train_medium
ko	kor	Koreanic	UD_Korean-Kaist	train_all, train_medium
la	lat	Indo-European	UD_Latin-PROIEL	train_all, train_medium
cu	chu	Indo-European	UD_Old_Church_Slavonic-PROIEL	train_all, train_medium
pt	por	Indo-European	UD_Portuguese-Bosque	train_all, train_medium
ro	ron	Indo-European	UD_Romanian-RRT	train_all, train_medium
uk	ukr	Indo-European	UD_Ukrainian-IU	train_all, train_medium
ar	ara	Afroasiatic	UD_Arabic-PADT	train_all, train_medium, train_large
be	bel	Indo-European	UD_Belarusian-HSE	train_all, train_medium, train_large
ca	cat	Indo-European	UD_Catalan-AnCora	train_all, train_medium, train_large
cs	ces	Indo-European	UD_Czech-CAC	train_all, train_medium, train_large
cs	ces	Indo-European	UD_Czech-PDT	train_all, train_medium, train_large
nl	nld	Indo-European	UD_Dutch-LassySmall	train_all, train_medium, train_large
en	eng	Indo-European	UD_English-EWT	train_all, train_medium, train_large
en	eng	Indo-European	UD_English-GUM	train_all, train_medium, train_large
et	est	Uralic	UD_Estonian-EDT	train_all, train_medium, train_large
fi	fin	Uralic	UD_Finnish-TDT	train_all, train_medium, train_large
fr	fra	Indo-European	UD_French-GSD	train_all, train_medium, train_large

Continued on next page

UD_ISO	ISO-3	Language Family	Treebank	Language Splits
de	deu	Indo-European	UD_German-GSD	train_all, train_medium, train_large
hi	hin	Indo-European	UD_Hindi-HDTB	train_all, train_medium, train_large
it	ita	Indo-European	UD_Italian-ISDT	train_all, train_medium, train_large
la	lat	Indo-European	UD_Latin-ITTB	train_all, train_medium, train_large
lv	lav	Indo-European	UD_Latvian-LVTB	train_all, train_medium, train_large
no	nor	Indo-European	UD_Norwegian-Bokmaal	train_all, train_medium, train_large
no	nor	Indo-European	UD_Norwegian-Nynorsk	train_all, train_medium, train_large
pt	por	Indo-European	UD_Portuguese-GSD	train_all, train_medium, train_large
ro	ron	Indo-European	UD_Romanian-Nonstandard	train_all, train_medium, train_large
ru	rus	Indo-European	UD_Russian-SynTagRus	train_all, train_medium, train_large
ru	rus	Indo-European	UD_Russian-Taiga	train_all, train_medium, train_large
sl	slv	Indo-European	UD_Slovenian-SSJ	train_all, train_medium, train_large
es	spa	Indo-European	UD_Spanish-AnCora	train_all, train_medium, train_large
es	spa	Indo-European	UD_Spanish-GSD	train_all, train_medium, train_large

Table 18: POS Train Languages and Datasets. There are 78 datasets in the *train-all*, 42 in train-medium, and 25 in train-large. The number of unique languages in train-all is 51, 28 in train-medium, and 20 in train-large. In train-all, there are languages from 11 language families and one language isolate, however this distribution is heavily biased towards Indo-European languages.

UD_ISO	ISO 639-3	Language Family	Treebank	Language Splits
hy	hye	Indo-European	UD_Armenian-ArmTDP	test_all
zh	zho	Tino-Sibetan	UD_Chinese-GSD	test_all
zh	zho	Tino-Sibetan	UD_Chinese-GSDSimp	test_all
da	dan	Indo-European	UD_Danish-DDT	test_all
en	eng	Indo-European	UD_English-Atis	test_all
en	eng	Indo-European	UD_English-ESLSpok	test_all
en	eng	Indo-European	UD_English-GUMReddit	test_all
en	eng	Indo-European	UD_English-ParTUT	test_all
fo	fao	Indo-European	UD_Faroese-FarPaHC	test_all
fr	fra	Indo-European	UD_French-GSD	test_all
fr	fra	Indo-European	UD_French-ParisStories	test_all
fr	fra	Indo-European	UD_French-ParTUT	test_all
fr	fra	Indo-European	UD_French-Rhapsodie	test_all
fr	fra	Indo-European	UD_French-Sequoia	test_all
he	heb	Afro-Asiatic	UD_Hebrew-HTB	test_all
he	heb	Afro-Asiatic	UD_Hebrew-IAHLTknesset	test_all
he	heb	Afro-Asiatic	UD_Hebrew-IAHLTwiki	test_all
is	isl	Indo-European	UD_Icelandic-Modern	test_all
ga	gle	Indo-European	UD_Irish-IDT	test_all
it	ita	Indo-European	UD_Italian-ISDT	test_all
it	ita	Indo-European	UD_Italian-MarkIT	test_all
it	ita	Indo-European	UD_Italian-Old	test_all
it	ita	Indo-European	UD_Italian-ParTUT	test_all
it	ita	Indo-European	UD_Italian-TWITTIRO	test_all
ko	kor	Koreanic	UD_Korean-GSD	test_all
ko	kor	Koreanic	UD_Korean-KSL	test_all
lt	lit	Indo-European	UD_Lithuanian-HSE	test_all
mt	mlt	Afro-Asiatic	UD_Maltese-MUDT	test_all
gd	gla	Indo-European	UD_Scottish_Gaelic-ARCOSG	test_all
sl	slv	Indo-European	UD_Slovenian-SST	test_all
ta	tam	Dravidian	UD_Tamil-TTB	test_all
tr	tur	Turkic	UD_Turkish-Atis	test_all
tr	tur	Turkic	UD_Turkish-FrameNet	test_all
vi	vie	Austroasiatic	UD_Vietnamese-VTB	test_all
wo	wol	Atlantic-Congo	UD_Wolof-WTB	test_all
af	afr	Indo-European	UD_Afrikaans-AfriBooms	test_all, test_medium
hy	hye	Indo-European	UD_Armenian-BSUT	test_all, test_medium
bg	bul	Indo-European	UD_Bulgarian-BTB	test_all, test_medium
hr	hrv	Indo-European	UD_Croatian-SET	test_all, test_medium
cs	ces	Indo-European	UD_Czech-CAC	test_all, test_medium
cs	ces	Indo-European	UD_Czech-CLTT	test_all, test_medium
cs	ces	Indo-European	UD_Czech-FicTree	test_all, test_medium

Continued on next page

UD_ISO	ISO 639-3	Language Family	Treebank	Language Splits
nl	nld	Indo-European	UD_Dutch-Alpino	test_all, test_medium
en	eng	Indo-European	UD_English-LinES	test_all, test_medium
et	est	Uralic	UD_Estonian-EWT	test_all, test_medium
fi	fin	Uralic	UD_Finnish-FTB	test_all, test_medium
fi	fin	Uralic	UD_Finnish-TDT	test_all, test_medium
ka	kat	Kartvelian	UD_Georgian-GLC	test_all, test_medium
de	deu	Indo-European	UD_German-GSD	test_all, test_medium
el	ell	Indo-European	UD_Greek-GDT	test_all, test_medium
hu	hun	Uralic	UD_Hungarian-Szeged	test_all, test_medium
is	isl	Indo-European	UD_Icelandic-GC	test_all, test_medium
id	ind	Austronesian	UD_Indonesian-GSD	test_all, test_medium
ga	gle	Indo-European	UD_Irish-TwittIrish	test_all, test_medium
it	ita	Indo-European	UD_Italian-PoSTWITA	test_all, test_medium
it	ita	Indo-European	UD_Italian-VIT	test_all, test_medium
ja	jpn	Japonic	UD_Japanese-GSD	test_all, test_medium
ja	jpn	Japonic	UD_Japanese-GSDLUW	test_all, test_medium
ko	kor	Koreanic	UD_Korean-Kaist	test_all, test_medium
la	lat	Indo-European	UD_Latin-LLCT	test_all, test_medium
la	lat	Indo-European	UD_Latin-PROIEL	test_all, test_medium
la	lat	Indo-European	UD_Latin-UDante	test_all, test_medium
lt	lit	Indo-European	UD_Lithuanian-ALKSNIS	test_all, test_medium
cu	chu	Indo-European	UD_Old_Church_Slavonic-PROIEL	test_all, test_medium
fa	fas	Indo-European	UD_Persian-PerDT	test_all, test_medium
fa	fas	Indo-European	UD_Persian-Seraji	test_all, test_medium
pl	pol	Indo-European	UD_Polish-LFG	test_all, test_medium
pt	por	Indo-European	UD_Portuguese-Bosque	test_all, test_medium
pt	por	Indo-European	UD_Portuguese-GSD	test_all, test_medium
pt	por	Indo-European	UD_Portuguese-PetroGold	test_all, test_medium
ro	ron	Indo-European	UD_Romanian-Nonstandard	test_all, test_medium
ro	ron	Indo-European	UD_Romanian-RRT	test_all, test_medium
ro	ron	Indo-European	UD_Romanian-SiMoNERo	test_all, test_medium
ru	rus	Indo-European	UD_Russian-GSD	test_all, test_medium
ru	rus	Indo-European	UD_Russian-Poetry	test_all, test_medium
ru	rus	Indo-European	UD_Russian-Taiga	test_all, test_medium
sa	san	Indo-European	UD_Sanskrit-Vedic	test_all, test_medium
sr	srp	Indo-European	UD_Serbian-SET	test_all, test_medium
sk	slk	Indo-European	UD_Slovak-SNK	test_all, test_medium
es	spa	Indo-European	UD_Spanish-GSD	test_all, test_medium
sv	swe	Indo-European	UD_Swedish-LinES	test_all, test_medium
tr	tur	Turkic	UD_Turkish-BOUN	test_all, test_medium
tr	tur	Turkic	UD_Turkish-IMST	test_all, test_medium
tr	tur	Turkic	UD_Turkish-Kenet	test_all, test_medium
tr	tur	Turkic	UD_Turkish-Penn	test_all, test_medium
tr	tur	Turkic	UD_Turkish-Tourism	test_all, test_medium
uk	ukr	Indo-European	UD_Ukrainian-IU	test_all, test_medium
uk	ukr	Indo-European	UD_Ukrainian-ParlaMint	test_all, test_medium
ur	urd	Indo-European	UD_Urdu-UDTB	test_all, test_medium
ug	uig	Turkic	UD_Uyghur-UDT	test_all, test_medium
cy	cym	Indo-European	UD_Welsh-CCG	test_all, test_medium
ar	ara	Afro-Asiatic	UD_Arabic-PADT	test_all, test_medium, test_large
eu	eus	-	UD_Basque-BDT	test_all, test_medium, test_large
be	bel	Indo-European	UD_Belarusian-HSE	test_all, test_medium, test_large
ca	cat	Indo-European	UD_Catalan-AnCora	test_all, test_medium, test_large
cs	ces	Indo-European	UD_Czech-PDT	test_all, test_medium, test_large
nl	nld	Indo-European	UD_Dutch-LassySmall	test_all, test_medium, test_large
en	eng	Indo-European	UD_English-EWT	test_all, test_medium, test_large
en	eng	Indo-European	UD_English-GUM	test_all, test_medium, test_large
et	est	Uralic	UD_Estonian-EDT	test_all, test_medium, test_large
gl	glg	Indo-European	UD_Galician-CTG	test_all, test_medium, test_large
de	deu	Indo-European	UD_German-HDT	test_all, test_medium, test_large
hi	hin	Indo-European	UD_Hindi-HDTB	test_all, test_medium, test_large
is	isl	Indo-European	UD_Icelandic-IcePaHC	test_all, test_medium, test_large
la	lat	Indo-European	UD_Latin-ITTB	test_all, test_medium, test_large
lv	lav	Indo-European	UD_Latvian-LVTB	test_all, test_medium, test_large
no	nor	Indo-European	UD_Norwegian-Bokmaal	test_all, test_medium, test_large
no	nor	Indo-European	UD_Norwegian-Nynorsk	test_all, test_medium, test_large
pl	pol	Indo-European	UD_Polish-PDB	test_all, test_medium, test_large

Continued on next page

UD_ISO	ISO 639-3	Language Family	Treebank	Language Splits
pt	por	Indo-European	UD_Portuguese-CINTIL	test_all, test_medium, test_large
pt	por	Indo-European	UD_Portuguese-Porttinari	test_all, test_medium, test_large
ru	rus	Indo-European	UD_Russian-SynTagRus	test_all, test_medium, test_large
sl	slv	Indo-European	UD_Slovenian-SSJ	test_all, test_medium, test_large
es	spa	Indo-European	UD_Spanish-AnCora	test_all, test_medium, test_large
sv	swe	Indo-European	UD_Swedish-Talbanken	test_all, test_medium, test_large

Table 19: POS Test Languages and Datasets. There are 118 total dataset in test-all, 83 in test-medium, and 25 in test-large. There are 56 unique languages in test-all, 46 in test-medium, and 21 in test-large. Languages cover 12 language families and 1 language isolate.

UD_ISO	ISO-3	Language Family	treebank	Language Split
ckb	ckb	Indo-European	Central Kurdish	train_all
la	lat	Indo-European	Latin	train_all
br	bre	Indo-European	Breton	train_all
hi	hin	Indo-European	Hindi	train_all
ga	gle	Indo-European	Irish	train_all
af	afr	Indo-European	Afrikaans	train_all
tt	tat	Turkic	Tatar	train_all
cy	cym	Indo-European	Welsh	train_all,train_medium
eu	eus	-	Basque	train_all,train_medium
lv	lav	Indo-European	Latvian	train_all,train_medium
tl	tgl	Austronesian	Tagalog	train_all,train_medium
mk	mkd	Indo-European	Macedonian	train_all,train_medium
bn	ben	Indo-European	Bengali	train_all,train_medium
lt	lit	Indo-European	Lithuanian	train_all,train_medium
it	ita	Indo-European	Italian	train_all,train_medium,train_large
sr	srp	Indo-European	Serbian Standard	train_all,train_medium,train_large
sl	slv	Indo-European	Slovenian	train_all,train_medium,train_large
ko	kor	Koreanic	Korean	train_all,train_medium,train_large
eo	epo	Artificial Language	Esperanto	train_all,train_medium,train_large
pt	por	Indo-European	Portuguese	train_all,train_medium,train_large
ta	tam	Dravidian	Tamil	train_all,train_medium,train_large
es	spa	Indo-European	Spanish	train_all,train_medium,train_large
et	est	Uralic	Estonian	train_all,train_medium,train_large
ja	jpn	Japonic	Japanese	train_all,train_medium,train_large
fi	fin	Uralic	Finnish	train_all,train_medium,train_large
fr	fra	Indo-European	French	train_all,train_medium,train_large
be	bel	Indo-European	Belarusian	train_all,train_medium,train_large
nl	nld	Indo-European	Dutch	train_all,train_medium,train_large
uk	ukr	Indo-European	Ukrainian	train_all,train_medium,train_large
ur	urd	Indo-European	Urdu	train_all,train_medium,train_large
de	deu	Indo-European	German	train_all,train_medium,train_large
id	ind	Austronesian	Standard Indonesian	train_all,train_medium,train_large
el	ell	Indo-European	Modern Greek	train_all,train_medium,train_large
ru	rus	Indo-European	Russian	train_all,train_medium,train_large
pl	pol	Indo-European	Polish	train_all,train_medium,train_large
da	dan	Indo-European	Danish	train_all,train_medium,train_large
bg	bul	Indo-European	Bulgarian	train_all,train_medium,train_large
vi	vie	Austroasiatic	Vietnamese	train_all,train_medium,train_large
sv	swe	Indo-European	Swedish	train_all,train_medium,train_large
hu	hun	Uralic	Hungarian	train_all,train_medium,train_large
zh	zho	Sino-Tibetan	Chinese	train_all,train_medium,train_large
hy	hye	Indo-European	Eastern Armenian	train_all,train_medium,train_large
th	tha	Tai-Kadai	Thai	train_all,train_medium,train_large
nn	nno	Indo-European	Norwegian Nynorsk	train_all,train_medium,train_large
ro	ron	Indo-European	Romanian	train_all,train_medium,train_large
ca	cat	Indo-European	Catalan	train_all,train_medium,train_large
tr	tur	Turkic	Turkish	train_all,train_medium,train_large
sk	slk	Indo-European	Slovak	train_all,train_medium,train_large
cs	ces	Indo-European	Czech	train_all,train_medium,train_large
hr	hrv	Indo-European	Croatian Standard	train_all,train_medium,train_large

Continued on next page

UD_ISO	ISO-3	Language Family	treebank	Language Split
ms	msa	Austronesian	Malay	train_all,train_medium,train_large

Table 20: NER Train Languages and Datasets. There are 51 datasets in train-all, 44 in train-medium, and 37 in train-large. As the WikiANN dataset only has one dataset per language, these counts represent unique languages as well. There are languages from 10 different languages families, 1 artificial language, and 1 language isolate.

UD_ISO	ISO-3	Language Family	treebank	Language Split
am	amh	Afro-Asiatic	Amharic	test_all
my	mya	Sino-Tibetan	Burmese	test_all
ceb	ceb	Austronesian	Cebuano	test_all
km	khm	Austroasiatic	Central Khmer	test_all
ce	che	Nakh-Daghestanian	Chechen	test_all
crh	crh	Turkic	Crimean Tatar	test_all
ne	nep	Indo-European	Eastern Pahari	test_all
fo	fao	Indo-European	Faroese	test_all
ig	ibo	Atlantic-Congo	Igbo	test_all
ilo	ilo	Austronesian	Iloko	test_all
jv	jav	Austronesian	Javanese	test_all
rw	kin	Atlantic-Congo	Kinyarwanda	test_all
mg	mlg	Austronesian	Malagasy	test_all
mi	mri	Austronesian	Maori	test_all
pdc	pdc	Indo-European	Pennsylvania German	test_all
gd	gla	Indo-European	Scottish Gaelic	test_all
sd	snd	Indo-European	Sindhi	test_all
so	som	Afro-Asiatic	Somali	test_all
tg	tgk	Indo-European	Tajik	test_all
ug	uig	Turkic	Uighur	test_all
yo	yor	Atlantic-Congo	Yoruba	test_all
af	afr	Indo-European	Afrikaans	test_all,test_medium
be	bel	Indo-European	Belarusian	test_all,test_medium
bn	ben	Indo-European	Bengali	test_all,test_medium
br	bre	Indo-European	Breton	test_all,test_medium
ckb	ckb	Indo-European	Central Kurdish	test_all,test_medium
hy	hye	Indo-European	Eastern Armenian	test_all,test_medium
hi	hin	Indo-European	Hindi	test_all,test_medium
ga	gle	Indo-European	Irish	test_all,test_medium
la	lat	Indo-European	Latin	test_all,test_medium
mk	mkd	Indo-European	Macedonian	test_all,test_medium
ms	msa	Austronesian	Malay	test_all,test_medium
nn	nno	Indo-European	Norwegian Nynorsk	test_all,test_medium
tl	tgl	Austronesian	Tagalog	test_all,test_medium
ta	tam	Dravidian	Tamil	test_all,test_medium
tt	tat	Turkic	Tatar	test_all,test_medium
ur	urd	Indo-European	Urdu	test_all,test_medium
cy	cym	Indo-European	Welsh	test_all,test_medium
eu	eus	-	Basque	test_all,test_medium,test_large
bg	bul	Indo-European	Bulgarian	test_all,test_medium,test_large
ca	cat	Indo-European	Catalan	test_all,test_medium,test_large
zh	zho	Sino-Tibetan	Chinese	test_all,test_medium,test_large
hr	hrv	Indo-European	Croatian Standard	test_all,test_medium,test_large
cs	ces	Indo-European	Czech	test_all,test_medium,test_large
da	dan	Indo-European	Danish	test_all,test_medium,test_large
nl	nld	Indo-European	Dutch	test_all,test_medium,test_large
eo	epo	Artificial Language	Esperanto	test_all,test_medium,test_large
et	est	Uralic	Estonian	test_all,test_medium,test_large
fi	fin	Uralic	Finnish	test_all,test_medium,test_large
fr	fra	Indo-European	French	test_all,test_medium,test_large
de	deu	Indo-European	German	test_all,test_medium,test_large
hu	hun	Uralic	Hungarian	test_all,test_medium,test_large
it	ita	Indo-European	Italian	test_all,test_medium,test_large
ja	jpn	Japonic	Japanese	test_all,test_medium,test_large
ko	kor	Koreanic	Korean	test_all,test_medium,test_large
lv	lav	Indo-European	Latvian	test_all,test_medium,test_large

Continued on next page

UD_ISO	ISO-3	Language Family	treebank	Language Splits
lt	lit	Indo-European	Lithuanian	test_all,test_medium,test_large
el	ell	Indo-European	Modern Greek	test_all,test_medium,test_large
pl	pol	Indo-European	Polish	test_all,test_medium,test_large
pt	por	Indo-European	Portuguese	test_all,test_medium,test_large
ro	ron	Indo-European	Romanian	test_all,test_medium,test_large
ru	rus	Indo-European	Russian	test_all,test_medium,test_large
sr	srp	Indo-European	Serbian Standard	test_all,test_medium,test_large
sk	slk	Indo-European	Slovak	test_all,test_medium,test_large
sl	slv	Indo-European	Slovenian	test_all,test_medium,test_large
es	spa	Indo-European	Spanish	test_all,test_medium,test_large
id	ind	Austronesian	Standard Indonesian	test_all,test_medium,test_large
sv	swe	Indo-European	Swedish	test_all,test_medium,test_large
th	tha	Tai-Kadai	Thai	test_all,test_medium,test_large
tr	tur	Turkic	Turkish	test_all,test_medium,test_large
uk	ukr	Indo-European	Ukrainian	test_all,test_medium,test_large
vi	vie	Austroasiatic	Vietnamese	test_all,test_medium,test_large

Table 21: NER Test Languages and Datasets. In test-all, there are 72 dataset, 51 in test-medium, and 34 in test-large. Languages come from 13 language families, in addition to one language isolate and one artificial language.

Treebank	UD File	Bible ISO	Bible File
UD_Lithuanian-ALKSNIS	lt_alksnis-ud-dev	lit	lit-x-bible-lit-v1.txt
UD_English-LinES	en_lines-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Portuguese-PetroGold	pt_petrogold-ud-dev	por	por-x-bible-almeidaatualizada-v1.txt
UD_Czech-FicTree	cs_fictree-ud-dev	ces	ces-x-bible-preklad-v1.txt
UD_Portuguese-CINTIL	pt_cintil-ud-dev	por	por-x-bible-almeidaatualizada-v1.txt
UD_Czech-CLTT	cs_cltt-ud-dev	ces	ces-x-bible-preklad-v1.txt
UD_Romanian-Nonstandard	ro_nonstandard-ud-dev	ron	ron-x-bible-cornilescu-v1.txt
UD_English-ParTUT	en_partut-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Maltese-MUDT	mt_mudt-ud-dev	mlt	mlt-x-bible-mlt-v1.txt
UD_Polish-PDB	pl_pdb-ud-dev	pol	pol-x-bible-gdansk-v1.txt
UD_Icelandic-Modern	is_modern-ud-dev	isl	isl-x-bible-isl-v1.txt
UD_Dutch-Alpino	nl_alpino-ud-dev	nld	nld-x-bible-2004-v1.txt
UD_English-GUM	en_gum-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Turkish-Kenet	tr_kenet-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_Italian-Old	it_old-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_Russian-Taiga	ru_taiga-ud-dev	rus	rus-x-bible-synodal-v1.txt
UD_Ukrainian-IU	uk_iu-ud-dev	ukr	ukr-x-bible-2007-v1.txt
UD_Hindi-HDTB	hi_hdtb-ud-dev	hin	hin-HNDSKV.txt
UD_Wolof-WTB	wo_wtb-ud-dev	wol	wol-x-bible-wol-v1.txt
UD_Korean-GSD	ko_gsd-ud-dev	kor	kor-x-bible-latinscript-v1.txt
UD_Estonian-EDT	et_edt-ud-dev	est	est-x-bible-portions-v1.txt
UD_Persian-PerDT	fa_perdt-ud-dev	fas	fas-x-bible-1995-v1.txt
UD_French-GSD	fr_gsd-ud-dev	fra	fra-FRNPDC.txt
UD_Latin-ITTB	la_ittb-ud-dev	lat	lat-LTNNVV.txt
UD_Vietnamese-VTB	vi_vtb-ud-dev	vie	vie-VIEVOV.txt
UD_Latvian-LVTB	lv_lvtb-ud-dev	lav	lav-x-bible-ljd-youversion-v1.txt
UD_Finnish-FTB	fi_ftb-ud-dev	fin	fin-x-bible-1766-v1.txt
UD_Icelandic-IcePaHC	is_icepahc-ud-dev	isl	isl-x-bible-isl-v1.txt
UD_Latin-PROIEL	la_proiel-ud-dev	lat	lat-LTNNVV.txt
UD_Romanian-RRT	ro_rrt-ud-dev	ron	ron-x-bible-cornilescu-v1.txt
UD_Czech-CAC	cs_cac-ud-dev	ces	ces-x-bible-preklad-v1.txt
UD_English-ESLSpok	en_eslspok-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Russian-SynTagRus	ru_syntagrus-ud-dev	rus	rus-x-bible-synodal-v1.txt
UD_Italian-ParTUT	it_partut-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_Turkish-IMST	tr_imst-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_Swedish-LinES	sv_lines-ud-dev	swe	swe-SWESFV.txt
UD_Russian-GSD	ru_gsd-ud-dev	rus	rus-x-bible-synodal-v1.txt
UD_Icelandic-GC	is_gc-ud-dev	isl	isl-x-bible-isl-v1.txt
UD_Persian-Seraji	fa_seraji-ud-dev	fas	fas-x-bible-1995-v1.txt
UD_Latin-UDante	la_udante-ud-dev	lat	lat-LTNNVV.txt
UD_Greek-GDT	el_gdt-ud-dev	ell	ell-x-bible-hellenic1-v1.txt
UD_Norwegian-Bokmaal	no_bokmaal-ud-dev	nor	nor-x-bible-student-v1.txt

Continued on next page

Treebank	UD File	Bible ISO	Bible File
UD_Turkish-FrameNet	tr_framenet-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_Swedish-Talbanken	sv_talbanken-ud-dev	swe	swe-SWESFV.txt
UD_Danish-DDT	da_ddt-ud-dev	dan	dan-x-bible-1931-v1.txt
UD_Italian-ISDT	it_isdt-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_Slovak-SNK	sk_snk-ud-dev	slk	slk-x-bible-standard-v1.txt
UD_Latin-LLCT	la_llct-ud-dev	lat	lat-LTNNVV.txt
UD_English-EWT	en_ewt-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Welsh-CCG	cy_ccg-ud-dev	cym	cym-x-bible-revised2004-v1.txt
UD_Portuguese-DANTEStocks	pt_dantestocks-ud-dev	por	por-x-bible-almeidaatualizada-v1.txt
UD_Hebrew-IAHLTknesset	he_iahltknesset-ud-dev	heb	heb-x-bible-2009-v1.txt
UD_Portuguese-Porttinari	pt_porttinari-ud-dev	por	por-x-bible-almeidaatualizada-v1.txt
UD_Hungarian-Szeged	hu_szeged-ud-dev	hun	hun-x-bible-revised-v1.txt
UD_Russian-Poetry	ru_poetry-ud-dev	rus	rus-x-bible-synodal-v1.txt
UD_Catalan-AnCora	ca_ancora-ud-dev	cat	cat-x-bible-cat-v1.txt
UD_French-ParTUT	fr_partut-ud-dev	fra	fra-FRNPDC.txt
UD_Italian-VIT	it_vit-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_German-GSD	de_gsd-ud-dev	deu	deu-x-bible-freebible-v1.txt
UD_Armenian-BSUT	hy_bsut-ud-dev	hye	hye-x-bible-eastern-v1.txt
UD_Lithuanian-HSE	lt_hse-ud-dev	lit	lit-x-bible-lit-v1.txt
UD_English-GUMReddit	en_gumreddit-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Italian-PoSTWITA	it_postwita-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_Korean-KSL	ko_ksl-ud-dev	kor	kor-x-bible-latinscript-v1.txt
UD_Spanish-AnCora	es_ancora-ud-dev	spa	spa-SPNBDA.txt
UD_Portuguese-GSD	pt_gsd-ud-dev	por	por-x-bible-almeidaatualizada-v1.txt
UD_Portuguese-Bosque	pt_bosque-ud-dev	por	por-x-bible-almeidaatualizada-v1.txt
UD_Polish-LFG	pl_lfg-ud-dev	pol	pol-x-bible-gdansk-v1.txt
UD_Czech-PDT	cs_pdt-ud-dev	ces	ces-x-bible-preklad-v1.txt
UD_Turkish-Atis	tr_atis-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_Finnish-TDT	fi_tdt-ud-dev	fin	fin-x-bible-1766-v1.txt
UD_Italian-MarkIT	it_markit-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_Romanian-SiMoNERo	ro_simonero-ud-dev	ron	ron-x-bible-cornilescu-v1.txt
UD_German-HDT	de_hdt-ud-dev	deu	deu-x-bible-freebible-v1.txt
UD_Hebrew-IAHLTwiki	he_iahltwiki-ud-dev	heb	heb-x-bible-2009-v1.txt
UD_French-Sequoia	fr_sequoia-ud-dev	fra	fra-FRNPDC.txt
UD_Estonian-EWT	et_ewt-ud-dev	est	est-x-bible-portions-v1.txt
UD_Uyghur-UDT	ug_udt-ud-dev	uig	uig-x-bible-uig-v1.txt
UD_Italian-TWITTIRO	it_twittiro-ud-dev	ita	ita-x-bible-2009-v1.txt
UD_Slovenian-SSJ	sl_ssj-ud-dev	slv	slv-x-bible-slv-v1.txt
UD_English-Atis	en_atis-ud-dev	eng	eng-x-bible-books-v1.txt
UD_Armenian-ArmTDP	hy_armtdp-ud-dev	hye	hye-x-bible-eastern-v1.txt
UD_Korean-Kaist	ko_kaist-ud-dev	kor	kor-x-bible-latinscript-v1.txt
UD_Serbian-SET	sr_set-ud-dev	srp	srp-x-bible-srp-v1.txt
UD_Slovenian-SST	sl_sst-ud-dev	slv	slv-x-bible-slv-v1.txt
UD_Hebrew-HTB	he_htb-ud-dev	heb	heb-x-bible-2009-v1.txt
UD_Old_Church_Slavonic-PROIEL	cu_proiel-ud-dev	chu	chu-x-bible-chu-v1.txt
UD_Urdu-UDTB	ur_udtb-ud-dev	urd	urd-x-bible-revised2010-v1.txt
UD_Norwegian-Nynorsk	no_nynorsk-ud-dev	nor	nor-x-bible-student-v1.txt
UD_Turkish-BOUN	tr_boun-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_Bulgarian-BTB	bg_btb-ud-dev	bul	bul-x-bible-veren-v1.txt
UD_Indonesian-GSD	id_gsd-ud-dev	ind	ind-x-bible-suciinjil-v1.txt
UD_Dutch-LassySmall	nl_lassysmall-ud-dev	nld	nld-x-bible-2004-v1.txt
UD_Turkish-Penn	tr_penn-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_Georgian-GLC	ka_glc-ud-dev	kat	kat-x-bible-kat-v1.txt
UD_Ukrainian-ParlaMint	uk_parlamint-ud-dev	ukr	ukr-x-bible-2007-v1.txt
UD_Afrikaans-AfriBooms	af_afribooms-ud-dev	afr	afr-x-bible-1953-v1.txt
UD_Spanish-GSD	es_gsd-ud-dev	spa	spa-SPNBDA.txt
UD_Basque-BDT	eu_bdt-ud-dev	eus	eus-x-bible-Hautin1571-v1.txt
UD_French-ParisStories	fr_parisstories-ud-dev	fra	fra-FRNPDC.txt
UD_French-Rhapsodie	fr_rhapsodie-ud-dev	fra	fra-FRNPDC.txt
UD_Tamil-TTB	ta_ttb-ud-dev	tam	tam-x-bible-tam-v1.txt
UD_Croatian-SET	hr_set-ud-dev	hrv	hrv-x-bible-hrv-v1.txt
UD_Turkish-Tourism	tr_tourism-ud-dev	tur	tur-x-bible-tur-v1.txt
UD_English-LinES	en_lines-ud-train	eng	eng-x-bible-books-v1.txt
UD_Czech-FicTree	cs_fictree-ud-train	ces	ces-x-bible-preklad-v1.txt
UD_Czech-CLTT	cs_cltt-ud-train	ces	ces-x-bible-preklad-v1.txt
UD_Romanian-Nonstandard	ro_nonstandard-ud-train	ron	ron-x-bible-cornilescu-v1.txt
UD_English-ParTUT	en_partut-ud-train	eng	eng-x-bible-books-v1.txt

Continued on next page

Treebank	UD File	Bible ISO	Bible File
UD_Dutch-Alpino	nl_alpino-ud-train	nld	nld-x-bible-2004-v1.txt
UD_English-GUM	en_gum-ud-train	eng	eng-x-bible-books-v1.txt
UD_Russian-Taiga	ru_taiga-ud-train	rus	rus-x-bible-synodal-v1.txt
UD_Ukrainian-IU	uk_iu-ud-train	ukr	ukr-x-bible-2007-v1.txt
UD_Hindi-HDTB	hi_hdtb-ud-train	hin	hin-HNDSKV.txt
UD_Korean-GSD	ko_gsd-ud-train	kor	kor-x-bible-latinscript-v1.txt
UD_Estonian-EDT	et_edt-ud-train	est	est-x-bible-portions-v1.txt
UD_French-GSD	fr_gsd-ud-train	fra	fra-FRNPDC.txt
UD_Latin-ITTB	la_ittb-ud-train	lat	lat-LTNNVV.txt
UD_Vietnamese-VTB	vi_vtb-ud-train	vie	vie-VIEVOV.txt
UD_Latvian-LVTB	lv_lvtb-ud-train	lav	lav-x-bible-ljd-youversion-v1.txt
UD_Finnish-FTB	fi_ftb-ud-train	fin	fin-x-bible-1766-v1.txt
UD_Latin-PROIEL	la_proiel-ud-train	lat	lat-LTNNVV.txt
UD_Romanian-RRT	ro_rrt-ud-train	ron	ron-x-bible-cornilescu-v1.txt
UD_Czech-CAC	cs_cac-ud-train	ces	ces-x-bible-preklad-v1.txt
UD_Russian-SynTagRus	ru_syntagrus-ud-train	rus	rus-x-bible-synodal-v1.txt
UD_Italian-ParTUT	it_partut-ud-train	ita	ita-x-bible-2009-v1.txt
UD_Turkish-IMST	tr_imst-ud-train	tur	tur-x-bible-tur-v1.txt
UD_Swedish-LinES	sv_lines-ud-train	swe	swe-SWESFV.txt
UD_Russian-GSD	ru_gsd-ud-train	rus	rus-x-bible-synodal-v1.txt
UD_Persian-Seraji	fa_seraji-ud-train	fas	fas-x-bible-1995-v1.txt
UD_Greek-GDT	el_gdt-ud-train	ell	ell-x-bible-hellenic1-v1.txt
UD_Norwegian-Bokmaal	no_bokmaal-ud-train	nor	nor-x-bible-student-v1.txt
UD_Swedish-Talbanken	sv_talbanken-ud-train	swe	swe-SWESFV.txt
UD_Danish-DDT	da_ddt-ud-train	dan	dan-x-bible-1931-v1.txt
UD_Italian-ISDT	it_isdt-ud-train	ita	ita-x-bible-2009-v1.txt
UD_Slovak-SNK	sk_snk-ud-train	slk	slk-x-bible-standard-v1.txt
UD_English-EWT	en_ewt-ud-train	eng	eng-x-bible-books-v1.txt
UD_Hungarian-Szeged	hu_szeged-ud-train	hun	hun-x-bible-revised-v1.txt
UD_Catalan-AnCora	ca_ancora-ud-train	cat	cat-x-bible-cat-v1.txt
UD_French-ParTUT	fr_partut-ud-train	fra	fra-FRNPDC.txt
UD_German-GSD	de_gsd-ud-train	deu	deu-x-bible-freebible-v1.txt
UD_Italian-PoSTWITA	it_postwita-ud-train	ita	ita-x-bible-2009-v1.txt
UD_Spanish-AnCora	es_ancora-ud-train	spa	spa-SPNBDA.txt
UD_Portuguese-GSD	pt_gsd-ud-train	por	por-x-bible-almeidaatualizada-v1.txt
UD_Portuguese-Bosque	pt_bosque-ud-train	por	por-x-bible-almeidaatualizada-v1.txt
UD_Polish-LFG	pl_lfg-ud-train	pol	pol-x-bible-gdansk-v1.txt
UD_Latin-Perseus	la_perseus-ud-train	lat	lat-LTNNVV.txt
UD_Czech-PDT	cs_pdt-ud-train	ces	ces-x-bible-preklad-v1.txt
UD_Finnish-TDT	fi_tdt-ud-train	fin	fin-x-bible-1766-v1.txt
UD_French-Sequoia	fr_sequoia-ud-train	fra	fra-FRNPDC.txt
UD_Uyghur-UDT	ug_udt-ud-train	uig	uig-x-bible-uig-v1.txt
UD_Slovenian-SSJ	sl_ssj-ud-train	slv	slv-x-bible-slv-v1.txt
UD_Armenian-ArmTDP	hy_armtdp-ud-train	hye	hye-x-bible-eastern-v1.txt
UD_Korean-Kaist	ko_kaist-ud-train	kor	kor-x-bible-latinscript-v1.txt
UD_Serbian-SET	sr_set-ud-train	srp	srp-x-bible-srp-v1.txt
UD_Slovenian-SST	sl_sst-ud-train	slv	slv-x-bible-slv-v1.txt
UD_Hebrew-HTB	he_htb-ud-train	heb	heb-x-bible-2009-v1.txt
UD_Old_Church_Slavonic-PROIEL	cu_proiel-ud-train	chu	chu-x-bible-chu-v1.txt
UD_Urdu-UDTB	ur_udtb-ud-train	urd	urd-x-bible-revised2010-v1.txt
UD_Norwegian-Nynorsk	no_nynorsk-ud-train	nor	nor-x-bible-student-v1.txt
UD_Bulgarian-BTB	bg_btb-ud-train	bul	bul-x-bible-veren-v1.txt
UD_Indonesian-GSD	id_gsd-ud-train	ind	ind-x-bible-suciinjil-v1.txt
UD_Dutch-LassySmall	nl_lassysmall-ud-train	nld	nld-x-bible-2004-v1.txt
UD_Afrikaans-AfriBooms	af_afribooms-ud-train	afr	afr-x-bible-1953-v1.txt
UD_Spanish-GSD	es_gsd-ud-train	spa	spa-SPNBDA.txt
UD_Basque-BDT	eu_bdt-ud-train	eus	eus-x-bible-Hautin1571-v1.txt
UD_Tamil-TTB	ta_ttb-ud-train	tam	tam-x-bible-tam-v1.txt
UD_Croatian-SET	hr_set-ud-train	hrv	hrv-x-bible-hrv-v1.txt

Table 22: Mapping from UD datasets to Bible datasets.

Rahimi ISO	Bible ISO	Bible File
it	ita	ita-x-bible-riveduta-v1.txt
sr	srp	srp-x-bible-srp-v1.txt
sl	slv	slv-x-bible-slv-v1.txt
so	som	som-SOMSIM.txt
ko	kor	kor-x-bible-kor-v1.txt
crh	crh	crh-CRHIBT.txt
cy	cym	cym-x-bible-colloquial2013-v1.txt
eo	epo	epo-x-bible-epo-v1.txt
pt	por	por-PORARC.txt
ta	tam	tam-TCVWTC.txt
es	spa	spa-SPNWTC.txt
la	lat	lat-x-bible-vulgataclementina-v1.txt
ceb	ceb	ceb-x-bible-popular-v1.txt
et	est	est-x-bible-portions-v1.txt
yo	yor	yor-x-bible-yor-v1.txt
br	bre	bre-x-bible-bre-v1.txt
fi	fin	fin-x-bible-1766-v1.txt
eu	eus	eus-x-bible-batua-v1.txt
hi	hin	hin-HNDSKV.txt
fr	fra	fra-x-bible-kingjames-v1.txt
ug	uig	uig-UI1UMK.txt
lv	lav	lav-x-bible-1997-v1.txt
ilo	ilo	ilo-x-bible-ilo-v1.txt
ce	che	che-CHEIBT.txt
tl	tgl	tgl-TGLPBS.txt
nl	nld	nld-x-bible-2007-v1.txt
rw	kin	kin-x-bible-bird-youversion-v1.txt
mg	mlg	mlg-MLGRCV.txt
uk	ukr	ukr-x-bible-2009-v1.txt
mk	mkd	mkd-x-bible-2004-v1.txt
ur	urd	urd-x-bible-devanagari-v1.txt
de	deu	deu-x-bible-greber-v1.txt
id	ind	ind-INZNTV.txt
el	ell	ell-x-bible-hellenic1-v1.txt
am	amh	amh-x-bible-amh-v1.txt
ru	rus	rus-x-bible-kulakov-v1.txt
af	afr	afr-x-bible-boodskap-v1.txt
pl	pol	pol-x-bible-gdansk-v1.txt
da	dan	dan-x-bible-1931-v1.txt
bg	bul	bul-x-bible-veren-v1.txt
my	mya	mya-x-bible-common-v1.txt
vi	vie	vie-x-bible-bd2011-youversion-v1.txt
tt	tat	tat-TTRIBT.txt
tg	tgk	tgk-TGKIBT.txt
sv	swe	swe-SWESFV.txt
hu	hun	hun-x-bible-revised-v1.txt
hy	hye	hye-x-bible-eastern-v1.txt
th	tha	tha-THATSV.txt
ig	ibo	ibo-x-bible-ibo-v1.txt
jv	jav	jav-x-bible-jav-v1.txt
nn	nno	nno-x-bible-2011-v1.txt
bn	ben	ben-x-bible-common-v1.txt
mi	mri	mri-x-bible-mri-v1.txt
lt	lit	lit-x-bible-1999-v1.txt
ro	ron	ron-RONBSR.txt
ca	cat	cat-x-bible-cat-v1.txt
tr	tur	tur-TRKBST.txt
sk	slk	slk-x-bible-standard-v1.txt
cs	ces	ces-x-bible-novakarlica-v1.txt
hr	hrv	hrv-x-bible-hrv-v1.txt
km	khm	khm-x-bible-2011-v1.txt
ms	msa	msa-x-bible-1996-v1.txt

Table 23: Mapping from (Rahimi et al., 2019) datasets to Bible datasets.

Source	Target Tokens		Num. Examples
	mBERT	XLM-R	
af_afribooms-ud-dev	8521	7515	398
ar_nyuad-ud-dev	40000	40000	1000
ar_padt-ud-dev	24390	18741	1000
be_hse-ud-dev	26740	20806	1000
bg_btb-ud-dev	25337	19559	1000
ca_ancora-ud-dev	20738	19755	1000
cs_cac-ud-dev	22820	18853	856
cs_cltt-ud-dev	24100	19053	917
cs_fictree-ud-dev	27022	23007	1000
cs_pdt-ud-dev	21609	17678	1000
cu_proiel-ud-dev	22228	39918	1000
cy_ccg-ud-dev	16210	13478	564
da_ddt-ud-dev	15924	14106	711
de_gsd-ud-dev	17983	16517	944
de_hdt-ud-dev	18014	18195	1000
el_gdt-ud-dev	24439	16737	799
en_atis-ud-dev	8875	9388	478
en_eslspok-ud-dev	2420	2697	122
en_ewt-ud-dev	17333	16780	1000
en_gum-ud-dev	15137	15423	1000
en_gumreddit-ud-dev	1755	1555	71
en_lines-ud-dev	18652	18990	1000
en_partut-ud-dev	3181	3351	183
es_ancora-ud-dev	19084	18510	1000
es_gsd-ud-dev	18506	18786	1000
et_edt-ud-dev	25452	21031	1000
et_ewt-ud-dev	18747	15140	699
eu_bdt-ud-dev	23246	20001	1000
fa_perdt-ud-dev	27040	22247	1000
fa_seraji-ud-dev	24608	20241	962
fi_ftb-ud-dev	23579	20469	1000
fi_tdt-ud-dev	23211	18989	1000
fo_farpahc-ud-dev	15190	13542	521
fr_gsd-ud-dev	19507	20455	1000
fr_parisstories-ud-dev	12645	12226	521
fr_partut-ud-dev	2408	2401	124
fr_rhapsodie-ud-dev	16030	15763	684
fr_sequoia-ud-dev	13779	13514	679
ga_idt-ud-dev	18391	15299	643
ga_twittirish-ud-dev	31893	26959	1000
gd_arcosg-ud-dev	20167	17136	669
gl_ctg-ud-dev	18628	17014	1000
he_htb-ud-dev	16549	13905	522
he_iahltknesset-ud-dev	8838	7217	317
he_iahltwiki-ud-dev	13857	11748	514
hi_hdtb-ud-dev	30255	21135	1000
hr_set-ud-dev	22897	19128	1000
hu_szeged-ud-dev	24410	18650	1000
hy_armtdp-ud-dev	13562	9144	456
hy_bsut-ud-dev	26762	16933	977
id_gsd-ud-dev	17783	16168	978
is_gc-ud-dev	21888	16661	786
is_icepahc-ud-dev	29366	23409	1000
is_modern-ud-dev	15990	12184	565
it_isdt-ud-dev	15137	14639	782
it_markit-ud-dev	12895	12992	705
it_old-ud-dev	16504	16190	686
it_partut-ud-dev	3812	3665	200
it_postwita-ud-dev	21299	19585	854
it_twittiro-ud-dev	4871	4577	202
it_vit-ud-dev	18397	17271	1000
ja_bccwj-ud-dev	80653	6395	1000
ja_bccwjluw-ud-dev	80586	6443	1000
ja_gsd-ud-dev	16004	12800	538

Continued on next page

Source	Target Tokens		Num. Examples
	mBERT	XLM-R	
ja_gsdluw-ud-dev	16004	12800	538
ka_glc-ud-dev	31320	18913	895
ko_gsd-ud-dev	27817	24689	561
ko_kaist-ud-dev	48999	42660	1000
ko_ksl-ud-dev	13806	11485	288
la_ittb-ud-dev	22941	19384	1000
la_llct-ud-dev	24171	20836	1000
la_proiel-ud-dev	23033	19301	1000
la_udante-ud-dev	20738	18207	887
lt_alksnis-ud-dev	25320	19442	985
lt_hse-ud-dev	2205	1822	108
lv_lvtb-ud-dev	24574	19819	1000
mt_mudt-ud-dev	22456	22470	715
nl_alpino-ud-dev	16672	16126	825
nl_lassysmall-ud-dev	20204	19797	1000
no_bokmaal-ud-dev	20175	17823	1000
no_nynorsk-ud-dev	21417	20205	1000
pl_lfg-ud-dev	24459	20574	921
pl_pdb-ud-dev	25627	21685	1000
pt_bosque-ud-dev	20164	19222	1000
pt_cintil-ud-dev	19545	20356	1000
pt_dantestocks-ud-dev	20825	18335	673
pt_gsd-ud-dev	19702	18554	1000
pt_petrogold-ud-dev	19890	18738	1000
pt_porttinari-ud-dev	20162	19148	1000
ro_nonstandard-ud-dev	30543	26920	1000
ro_rrt-ud-dev	23287	20163	1000
ro_simonero-ud-dev	23378	19727	1000
ru_gsd-ud-dev	21848	19077	932
ru_poetry-ud-dev	19372	16107	869
ru_syntagrus-ud-dev	22194	19571	1000
ru_taiga-ud-dev	19948	17288	755
sa_vedic-ud-dev	35281	32561	1000
sk_snk-ud-dev	25869	21288	985
sl_ssj-ud-dev	23134	19058	1000
sl_sst-ud-dev	15180	15014	615
sr_set-ud-dev	20676	17723	896
sv_lines-ud-dev	22032	19407	1000
sv_talbanken-ud-dev	16240	13716	794
ta_ttb-ud-dev	3597	2188	120
tr_atis-ud-dev	10765	8658	481
tr_boun-ud-dev	25020	19336	984
tr_framenet-ud-dev	2998	2616	115
tr_imst-ud-dev	21716	16889	863
tr_kenet-ud-dev	25741	22390	1000
tr_penn-ud-dev	14326	11950	594
tr_tourism-ud-dev	38899	28671	801
ug_udt-ud-dev	25272	22254	863
uk_iu-ud-dev	23320	18934	1000
uk_parlamint-ud-dev	20090	15537	997
ur_udtb-ud-dev	25949	19464	847
vi_vtb-ud-dev	22978	22816	1000
wo_wtb-ud-dev	17495	17275	553
zh_gsd-ud-dev	19099	14906	248
zh_gsdsimp-ud-dev	19099	14553	248

Table 24: POS Target Token Counts

Source	Source Tokens		Num. Examples
	mBERT	XLM-R	
af_afribooms-ud-train	22032	19344	1000

Continued on next page

Source	Source Tokens		Num. Examples
	mBERT	XLM-R	
ar_nyuad-ud-train	40000	40000	1000
ar_padt-ud-train	24926	19355	1000
be_hse-ud-train	30807	23987	1000
bg_btb-ud-train	27479	21340	1000
ca_ancora-ud-train	20705	19615	1000
cop_scriptorium-ud-train	12065	22713	1000
cs_cac-ud-train	25907	20818	1000
cs_cltt-ud-train	26619	21440	1000
cs_fictree-ud-train	26956	22541	1000
cs_pdt-ud-train	21379	17393	1000
cu_proiel-ud-train	31190	39693	1000
da_ddt-ud-train	22707	20025	1000
de_gsd-ud-train	19558	17229	1000
el_gdt-ud-train	31139	20363	1000
en_ewt-ud-train	17188	17915	1000
en_gum-ud-train	15677	16395	1000
en_lines-ud-train	18533	18951	1000
en_partut-ud-train	16296	16640	1000
es_ancora-ud-train	18108	17223	1000
es_gsd-ud-train	18481	18543	1000
et_edt-ud-train	25274	20247	1000
eu_bdt-ud-train	23293	20023	1000
fa_seraji-ud-train	26436	22096	1000
fi_ftb-ud-train	23958	21113	1000
fi_tdt-ud-train	24186	19118	1000
fr_gsd-ud-train	19488	20265	1000
fr_partut-ud-train	18435	18196	1000
fr_sequoia-ud-train	20856	20610	1000
ga_idt-ud-train	28360	23489	1000
gl_ctg-ud-train	18809	17081	1000
gl_treegal-ud-train	19181	17820	991
got_proiel-ud-train	29054	27840	1000
grc_perseus-ud-train	39994	44806	1000
grc_proiel-ud-train	41031	41922	1000
he_htb-ud-train	32107	26899	1000
hi_hdtb-ud-train	30541	22278	1000
hr_set-ud-train	23100	19243	1000
hu_szeged-ud-train	24914	20141	1000
hy_armtdp-ud-train	30744	21104	1000
id_gsd-ud-train	18133	16515	1000
it_isdt-ud-train	20357	19516	1000
it_partut-ud-train	17912	16512	1000
it_postwita-ud-train	24636	22727	1000
ja_bccwj-ud-train	80347	6827	1000
ja_gsd-ud-train	29394	23727	1000
ko_gsd-ud-train	49639	44018	1000
ko_kaist-ud-train	48678	43974	1000
la_ittb-ud-train	22431	19385	1000
la_perseus-ud-train	24084	21151	1000
la_proiel-ud-train	23178	19169	1000
lv_lvtb-ud-train	26548	21311	1000
nl_alpino-ud-train	20144	19048	1000
nl_lassysmall-ud-train	19747	19830	1000
no_bokmaal-ud-train	19767	17679	1000
no_nynorsk-ud-train	21361	19908	1000
pl_lfg-ud-train	26914	22629	1000
pt_bosque-ud-train	20416	19109	1000
pt_gsd-ud-train	19664	18483	1000
ro_nonstandard-ud-train	32138	28198	1000
ro_rrt-ud-train	25965	22750	1000
ru_gsd-ud-train	23059	20180	1000
ru_syntagrus-ud-train	25673	21795	1000
ru_taiga-ud-train	21552	17830	1000
sk_snk-ud-train	28149	23430	1000
sl_ssj-ud-train	23635	19730	1000

Continued on next page

Source	Source Tokens		Num. Examples
	mBERT	XLM-R	
sl_sst-ud-train	24226	24141	1000
sme_giella-ud-train	31075	30260	1000
sr_set-ud-train	23153	19811	1000
sv_lines-ud-train	21788	19220	1000
sv_talbanken-ud-train	19128	16001	1000
ta_ttb-ud-train	18592	11484	609
tr_imst-ud-train	25381	19909	1000
ug_udt-ud-train	29758	25737	1000
uk_iu-ud-train	25311	21342	1000
ur_udtb-ud-train	30512	22656	1000
vi_vtb-ud-train	22842	22841	1000
zh_gsd-ud-train	76547	60503	1000

Table 25: POS Source Token Counts

Target Lang	Target Tokens		Num. Lines
	mBERT	XLM-R	
af_dev	17736	18092	1000
am_dev	669	1296	100
be_dev	16748	14468	1000
bg_dev	15589	14525	1000
bn_dev	13388	10384	1000
br_dev	12928	13464	1000
ca_dev	9460	9566	1000
ce_dev	2137	2838	100
ceb_dev	1106	1372	100
ckb_dev	8342	16355	1000
crh_dev	1376	1284	100
cs_dev	14561	14109	1000
cy_dev	15155	14544	1000
da_dev	13352	13423	1000
de_dev	14628	16019	1000
el_dev	23649	18368	1000
eo_dev	12316	11864	1000
es_dev	10755	10384	1000
et_dev	16242	15177	1000
eu_dev	15957	17482	1000
fi_dev	16814	16361	1000
fo_dev	1832	1760	100
fr_dev	10486	11192	1000
ga_dev	15008	14162	1000
gd_dev	1634	1537	100
hi_dev	14286	10476	1000
hr_dev	14429	13888	1000
hu_dev	16757	15377	1000
hy_dev	21359	15959	1000
id_dev	10086	9982	1000
ig_dev	1124	1165	100
ilo_dev	808	834	100
it_dev	12340	12520	1000
ja_dev	29334	50743	1000
jv_dev	1001	994	100
km_dev	553	1897	100
ko_dev	17710	17560	1000
la_dev	10895	12183	1000
lt_dev	14285	13411	1000
lv_dev	15584	13843	1000
mg_dev	1621	1758	100
mi_dev	2988	2981	100
mk_dev	18411	17093	1000
ms_dev	9490	9162	1000

Continued on next page

Target Lang	Target Tokens		Num. Lines
	mBERT	XLM-R	
my_dev	4976	3125	100
ne_dev	1949	1382	100
nl_dev	12195	12649	1000
nn_dev	15423	16451	1000
pdc_dev	1570	1616	100
pl_dev	14440	14606	1000
pt_dev	10296	10399	1000
ro_dev	12034	12160	1000
ru_dev	13738	13835	1000
rw_dev	1387	1326	100
sd_dev	3302	2834	100
sk_dev	15142	14131	1000
sl_dev	13155	12487	1000
so_dev	1800	1394	100
sr_dev	15092	12946	1000
sv_dev	13972	15943	1000
ta_dev	22159	16942	1000
tg_dev	1681	1918	100
th_dev	60413	72584	1000
tl_dev	7327	7352	1000
tr_dev	15011	13392	1000
tt_dev	17031	19500	1000
ug_dev	2663	2360	100
uk_dev	17042	16073	1000
ur_dev	14189	11129	1000
vi_dev	8754	8755	1000
yo_dev	1354	1463	100
zh_dev	22833	37362	1000

Table 26: NER Target Token Counts

Source	Source Tokens		Num. Examples
	mBERT	XLM-R	
af_train	18515	18890	1000
be_train	16393	14160	1000
bg_train	15437	14366	1000
bn_train	13055	10154	1000
br_train	14047	14708	1000
ca_train	9613	9694	1000
ckb_train	8503	16505	1000
cs_train	15216	14844	1000
cy_train	15764	15279	1000
da_train	12868	13042	1000
de_train	14609	15870	1000
el_train	22705	17864	1000
eo_train	12347	11731	1000
es_train	10965	10681	1000
et_train	16654	15899	1000
eu_train	16099	17887	1000
fi_train	16416	16082	1000
fr_train	10894	11579	1000
ga_train	15228	14307	1000
hi_train	14020	10345	1000
hr_train	14915	14417	1000
hu_train	17808	16664	1000
hy_train	20793	15550	1000
id_train	9783	9839	1000
it_train	11989	12270	1000
ja_train	31087	52334	1000
ko_train	18293	18308	1000
la_train	10372	11720	1000

Continued on next page

Source	Source Tokens		Num. Examples
	mBERT	XLM-R	
lt_train	14992	14137	1000
lv_train	15948	14318	1000
mk_train	18375	17090	1000
ms_train	9696	9343	1000
nl_train	13036	13436	1000
nn_train	15713	16631	1000
pl_train	14120	14366	1000
pt_train	10089	9931	1000
ro_train	12168	12339	1000
ru_train	13698	13703	1000
sk_train	14820	13884	1000
sl_train	12743	11799	1000
sr_train	15033	12886	1000
sv_train	13362	15089	1000
ta_train	22669	17414	1000
th_train	59299	70855	1000
tl_train	7363	7399	1000
tr_train	14478	12894	1000
tt_train	16811	19726	1000
uk_train	17615	16453	1000
ur_train	14422	11286	1000
vi_train	8981	8983	1000
zh_train	21121	34428	1000

Table 27: NER Source Token Counts