From Transfer to Typology: Using Cross-Lingual NLP to Reveal Linguistic Structure

Anonymous ACL submission

Abstract

We investigate whether linguistic properties can be inferred from cross-lingual transfer performance. Using PoS tagging and dependency parsing as case studies, we show that transfer scores encode typological information that can be recovered through regression and feature-imputation methods. Our results demonstrate that transfer performance not only reflects linguistic distance but can also reveal missing or inconsistent entries in typological databases, offering a novel perspective on multilingual representation learning.

1 Introduction

Recent advances in multilingual models have made it possible to train a model for a target language T using only annotated data from a different source language S, as first shown in the seminal work of Pires et al. (2019). A growing body of research has since explored how to predict cross-lingual transfer performance in advance, drawing either on linguistic distances—based on syntactic, lexical or phonological information from large typological databases such as `URIEL` or `PHOIBLE` (e.g. Lin et al., 2019; Eronen et al., 2023) or on internal, model-based similarity measures (e.g. Yun et al., 2023).

In this work, we position ourselves within the first line of research but take the inverse perspective: rather than predicting transfer performance from linguistic features, we ask whether linguistic properties themselves can be recovered from observed transfer performance. This inversion is both theoretically revealing, showing that transferability indeed reflects linguistic structure, and practically useful, as it could help identify inconsistencies or missing values in typological databases and even suggest new entries where empirical transfer data strongly imply a given property, questions whose significance has been emphasised by several studies such as Bjerva and Augenstein (2021).

Building on this intuition, we use PoS tagging and dependency parsing as case studies to test whether multilingual transfer performance implicitly encodes typological information. This paper makes two main contributions: (i) we show that cross-lingual transfer performance can be used to predict key linguistic features of the target language, extending earlier observations within a more rigorous experimental framework that enforces a strict separation between training and test languages and explicitly avoids data leakage; and (ii) we demonstrate that such predictions uncover a latent structure of language similarity, reflecting typological relationships as captured by multilingual models.

2 The Promise and Pitfalls of Zero-Shot Syntax Transfer

This work builds upon a long line of research investigating the zero-shot capabilities of multilingual language models. To ground our analysis, we first evaluate cross-lingual transfer performance for two syntactic tasks: PoS tagging and dependency parsing. These experiments provide both an overview of transfer variability across languages and a foundation for the predictive and inversion models introduced in later sections.

Data and Model We rely on models developed within the `UDPipe` project (Straka, 2018).[1] The Universal Dependencies (UD) project provides annotated treebanks with predefined training, development, and test splits for a wide range of languages.[2] Based on these resources, the `UDPipe` project releases a collection of 147 pre-trained models covering 78 languages (some languages are represented by multiple treebanks).

To estimate cross-lingual transferability, we use the PoS taggers and dependency parsers from the

[1] `UDPipe` provides trainable pipelines for tokenisation, tagging, lemmatisation, and dependency parsing.

[2] We use version 2.15 of the UD project.

UDPipe model collection. Each model is trained by the UDPipe project on a single UD source treebank S, using the corresponding UD training data. We evaluate these models on all 296 UD target treebanks T, spanning 168 languages, which results in 43,512 transfer scores per task.[3]

Results Fig. 1 shows the distribution of scores when training on a source treebank S and testing on a target treebank T. We distinguish three conditions: *in-domain*, where $S = T$; *out-of-domain*, where they belong to the same language and *cross-lingual*, where source and target differ.

These distributions reveal substantial variation in performance across source corpora. In the best cases (what we term *oracle performance*) transfer systems approach out-of-domain accuracy, with median gaps between the best-performing system (always trained on data from the same language and domain) and the oracle performance of 17.5 points for PoS tagging and 21.4 for dependency parsing.[4] Although these performances do not reach in-domain levels, they are comparable to out-of-domain results and demonstrate that it is possible to develop reasonably accurate systems, suitable, for instance, for pre-annotating data, even in the absence of annotated resources. More importantly, they show that such models are capable of internalising and abstracting syntactic knowledge shared across multiple languages.

Conversely, a significant proportion of transfers underperform markedly. For instance, 44.9 % of PoS tagging systems perform worse than a baseline that correctly predicts punctuation and returns the most frequent label (NOUN) for all other tokens, underscoring the importance of determining which source language will transfer best. This motivates the next section, where we formalise this prediction problem and demonstrate that it can be modelled effectively using linguistic and data-derived features.

3 Estimating the Quality of Cross-Lingual Transfer

To identify the source language yielding the best performance for a given target, we train a regressor to predict test performance for each (source, target) pair, based on models trained on the source and evaluated on the target.

[3]Dependency parsers use gold PoS tags to control for error propagation. All systems are evaluated using the F_1 score on UPOS and UAS, computed with the official CoNLL shared task script.

[4]Full distributions are reported in Appendix A.

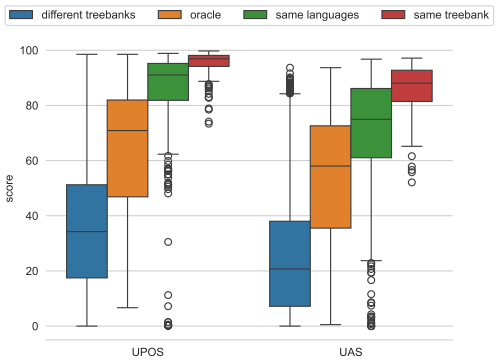

Figure 1: Distribution of scores obtained by training a PoS tagger and a dependency parser on a treebank of language S and testing on a treebank of language B

Features Following standard practice, we employ several feature types to describe language pairs:

- **Linguistic features** (968 features, 1,452 when including combinations): binary indicators capturing structural properties of source and target languages (e.g., word order such as SOV, or whether adjectives precede or follow nouns). These are extracted from the lang2vec (Littell et al., 2017) and Grambank (Skirgård et al., 2023) databases. In some experiments, we also include combination features indicating whether a property is shared between source and target languages.
- **Similarity features** (7 features): real-valued measures of inter-language similarity, computed as Gower similarity (Gower, 1971) over linguistic feature vectors. We distinguish 7 similarity types reflecting phonological, grammatical, and geographical characteristics.
- **Data- and model-related features** (10 features): descriptors of datasets and model behaviour, including the number of available models for that language, number of target languages, average target baseline performance, average source and target dataset sizes, ...

As already noted by Littell et al. (2017) or Bjerva and Augenstein (2021), the linguistic databases contain substantial missing values (up to 53.7 % in our experiments).

Model Given that numerous missing values, mixed feature types, and complex interdependencies all favour a non-linear model such as gradient boosting, we cast the problem as regression and use a single type of model: the Histogram-based Gradient Boosting Classifier (Ke et al., 2017), an efficient

implementation of gradient-boosted decision trees. It discretises continuous features into histograms before splitting and iteratively builds small trees that focus on correcting previous errors. Crucially, it handles missing values natively by learning dedicated "missing" branches during split selection, optimally routing incomplete observations without imputation. This property makes it particularly robust to sparse and heterogeneous feature spaces.

As a baseline, we also trained a linear regression model for comparison. All models were implemented using the corresponding scikit-learn classes (Pedregosa et al., 2011). Model hyperparameters were selected through cross-validation, and the full set of tested values is detailed in Appendix B.

Dataset Constructing a dataset that accurately evaluates a regressor's generalisation ability is non-trivial, as it requires avoiding data leakage, a concern often overlooked in prior work. Earlier studies did not explicitly ensure that test languages were excluded from the set of source languages used during training, guaranteeing only distinct target languages and leaving open the possibility of overlap between training and test sets.

To mitigate this issue, we perform the train–validation–test split at the language level, ensuring that no language appears in more than one partition. We adopt an 80–10–10 split ratio and select test languages greedily to approximate this proportion while preserving the overall distribution of language families. Owing to these typological and balancing constraints, not all available data can be included, and corpus sizes may vary slightly across splits.

A second challenge arises when constructing the datasets: as noted above, multiple treebanks are available for each language, which makes it difficult to determine a single score to associate with each language pair. In our experiments, we therefore chose to use the average of the scores obtained by training on any treebank of the source language and testing on any treebank of the target language. This averaging avoids bias introduced by selecting a particular treebank. In our experiments, we restricted our analysis to cross-lingual transfer scores, excluding results from same-language treebanks.

Metrics We report two types of evaluation metrics. (i) Regression metrics, the coefficient of determination R^2 and the MAE, are used to assess how well predicted scores approximate the observed ones. Unlike MAE, R^2 provides a scale-independent indicator of overall fit: A value closer to 1 indicates a better prediction, irrespective of the absolute magnitude of the predicted score; (ii) Ranking metrics, the mean reciprocal rank (MRR)[5] and precision at five `P@5`, are used to evaluate how effectively the regressor's predictions identify the best source language, where the correct item corresponds to the true optimal source.

Experimental Results The results obtained with the two models under consideration are presented in Table 1 and discussed in greater detail in Appendix C, which also includes `P@1` and `MSE`scores and a breakdown of the contribution of the different feature sets. These results show that the HGBR regressor can predict, with remarkably high accuracy, the performance achieved in a cross-lingual transfer experiment. In contrast, the linear model performs worse than chance, as reflected by a negative R^2. Nevertheless, in both cases the predicted scores are sufficient to identify the most relevant source languages: The MRR obtained with HGBR is 0.64, meaning that, on average, the true best source language is ranked among the top few candidates predicted by the model. Such information substantially reduces the number of models that need to be trained and evaluated in a cross-lingual transfer scenario. These findings corroborate earlier observations (Lin et al., 2019; Philippy et al., 2023), however, in a more robust experimental setting that enforces a strict separation between training and test languages.[6]

4 Feature Imputation from Performance Metrics

The results presented in the previous section show that the performance of a PoS tagger and a dependency parser can be predicted from features describing the language pair. A natural follow-up question is whether the reverse is also possible: can feature values themselves be inferred from observed performance metrics? This corresponds to a feature imputation problem, in which missing or latent attributes are inferred from observed data. Feature imputation has been widely studied in machine learning, for instance in the context of missing values in high-dimensional data (see, e.g., (Little and Rubin, 2002; van Buuren, 2018)).

[5]The MRR measures the average of the inverse rank assigned to the correct item; higher values indicate that the true best source is ranked closer to the top.

[6]A systematic comparison of results obtained under two distinct train/test splitting strategies is provided in Appendix D.

features	PoS tagging				Dependency Parsing			
	R^2	MAE	MRR	P@5	R^2	MAE	MRR	P@5
HGBR								
datasets	0.64 ± 0.01	8.9 ± 0.44	0.29 ± 0.06	0.33 ± 0.06	0.53 ± 0.02	10.6 ± 0.50	0.25 ± 0.05	0.30 ± 0.05
linguistic	0.22 ± 0.07	13.5 ± 1.07	0.49 ± 0.06	0.51 ± 0.06	0.18 ± 0.06	15.2 ± 1.10	0.45 ± 0.06	0.48 ± 0.06
similarities	0.18 ± 0.03	14.5 ± 0.57	0.52 ± 0.08	0.54 ± 0.09	0.15 ± 0.03	15.8 ± 0.60	0.48 ± 0.08	0.50 ± 0.09
all	0.75 ± 0.02	7.6 ± 0.46	0.64 ± 0.04	0.74 ± 0.05	0.63 ± 0.02	9.8 ± 0.48	0.60 ± 0.04	0.68 ± 0.05
Linear								
datasets	-2.33 ± 0.38	33.3 ± 0.79	0.25 ± 0.03	0.34 ± 0.04	-2.17 ± 0.40	35.8 ± 0.85	0.23 ± 0.03	0.31 ± 0.04
linguistic	-2.69 ± 0.43	32.1 ± 0.86	0.45 ± 0.06	0.46 ± 0.07	-2.50 ± 0.45	34.5 ± 0.90	0.41 ± 0.06	0.43 ± 0.07
similarities	-2.48 ± 0.27	31.9 ± 0.68	0.52 ± 0.08	0.54 ± 0.10	-2.66 ± 0.30	34.2 ± 0.70	0.48 ± 0.08	0.50 ± 0.10
all	-2.39 ± 0.63	31.9 ± 1.96	0.52 ± 0.09	0.59 ± 0.10	-2.73 ± 0.65	34.0 ± 2.00	0.47 ± 0.09	0.54 ± 0.10

Table 1: Perf. on the test set for our two models (95% CI are computed by bootraping (Efron and Tibshirani, 1993)).

5 Experimental Method

We propose an ad hoc exhaustive-search approach that explicitly exploits the predictive model introduced in the previous section. Considering a small subset of the n binary linguistic features of interest (i.e., those directly defined in linguistic databases and from which similarity-based features are derived), we enumerate all 2^n possible feature combinations. For each combination, we estimate the expected score using our performance predictor and select the configuration whose predicted score is closest to the observed one. Although this method has exponential complexity and is limited to small feature subsets, it places model performance at the core of the imputation process, unlike standard approaches.

For evaluation, we randomly select a set of non-missing features in the test data, mask them, and attempt to recover them using each imputation method. Performance is measured as the proportion of correctly recovered feature values. As baselines, we consider two common strategies: (i) replacing each missing feature with its most frequent value, and (ii) a k-nearest neighbours imputation method, which assigns feature values by averaging over the k closest data points based on non-missing features.

Results The accuracy of the different imputation methods[7] is reported in Table 2. Although all methods achieve comparable performance levels, the overall results remain modest and are insufficient to replace human linguistic expertise. Importantly, however, all approaches perform substantially better than random chance, indicating that they capture non-trivial patterns in typological information. Given both the observed performance and the computational complexity of ourmethod, its primary contribution should be understood as theoretical rather than practical: it serves to highlight the extent and nature of the linguistic information implicitly encoded by the models of § 3, rather than as a directly applicable tool for typological inference.

[7]In our experiments, we focused on our best-performing model predicting PoS cross-lingual transfer scores.

	# predicted values		
	1	3	5
Our Model			
prediction-based	76.6 ± 0.38	76.9 ± 0.59	76.3 ± 0.91
Baselines			
most frequent	66.6 ± 0.82	66.1 ± 0.23	67.1 ± 0.46
k-nn	68.3 ± 0.14	68.4 ± 0.68	68.2 ± 0.75

Table 2: Accuracy of different imputation methods.

6 Conclusion

In this paper, we explored how cross-lingual transfer performance can be used not only to predict but also to reconstruct linguistic structure. Using PoS tagging and dependency parsing as case studies, we showed that transfer results encode typological information that can be recovered through regression and feature-imputation techniques.

Our findings suggest that multilingual models capture dimensions of linguistic similarity that align with typological databases, while also exposing inconsistencies and gaps. This opens promising directions for data-driven typology, where empirical transfer data may help refine and expand existing linguistic resources. Future work will extend this framework to other tasks and model architectures, and investigate how typological regularities emerge during multilingual training.

7 Limitations

Our findings rely on the data available in the Universal Dependencies (UD) project. Despite the large number of languages it covers, UD is not fully representative of the world's linguistic diversity, and some language families are considerably more represented than others.

We also rely on typological databases that contain numerous errors, inconsistencies, and missing entries. As a result, our estimates of the ability to predict missing linguistic values may be affected by inaccuracies in the underlying data. Only a careful manual verification by expert linguists could fully assess the reliability of our imputation results. To the best of our knowledge, no proprietary or sensitive data were used.

We note, however, that the coverage of languages in UD and typological databases is uneven: some language families are well represented, while others, particularly low-resource and endangered languages, remain largely absent. Consequently, our analyses and conclusions should not be interpreted as representative of the full diversity of the world's languages.

We used `ChatGPT` to review the fluency and accuracy of our English.

8 Ethical Considerations

This work does not involve the training of large neural models. All experiments were conducted using pre-existing models from the `UDPipe` project and simple, non-neural predictive models such as gradient boosting and linear regression. To the best of our knowledge, the computational and environmental cost of training these models is negligible.

All datasets used in this study are publicly and freely available. Linguistic data were obtained from open typological databases, and all annotated corpora originate from the Universal Dependencies (UD) project. No proprietary or sensitive data were used.

We note, however, that the coverage of languages in UD and typological databases is uneven: some language families are well represented, while others—particularly low-resource and endangered languages—remain largely absent. Consequently, our analyses and conclusions should not be interpreted as representative of the full diversity of the world's languages.

References

Johannes Bjerva and Isabelle Augenstein. 2021. Does typological blinding impede cross-lingual sharing? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 480–486, Online. Association for Computational Linguistics.

Bradley Efron and Robert Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, London.

Juuso Eronen, Michal Ptaszynski, and Fumito Masui. 2023. Zero-shot cross-lingual transfer language selection using linguistic similarity. *Information Processing & Management*, 60(3):103250.

J. C. Gower. 1971. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA. 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.

Roderick J.A. Little and Donald B. Rubin. 2002. *Statistical Analysis with Missing Data*, 2nd edition. Wiley, Hoboken, NJ.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5877–5891,

Toronto, Canada. Association for Computational Linguistics.

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.

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).

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Stef van Buuren. 2018. *Flexible Imputation of Missing Data*, 2nd edition. CRC Press, Boca Raton, FL.

Taejun Yun, Jinhyeon Kim, Deokyeong Kang, Seonghoon Lim, Jihoon Kim, and Taeuk Kim. 2023. X-SNS: Cross-lingual transfer prediction through sub-network similarity. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13131–13144, Singapore. Association for Computational Linguistics.

A Distribution of scores

Figure 2 and Figure 3 show the distribution, across all UD treebanks, of the difference between oracle scores (i.e., the best performance achievable on a given treebank through cross-lingual transfer) and the performance obtained by an in-domain system. Extreme values—though rare—are mainly due to the presence of highly divergent languages (notably sign languages), which we did not exclude, unlike several previous studies.

B Hyper-Parameters

The Figure 4 defines the hyperparameter grid explored during model selection for the linear regressor and the HGBR regressor).

C Detailed Evaluation Results

Table 3 details the performance (for PoS prediction) obtained on the different datasets by our two models. In addition to the main evaluation scores, we

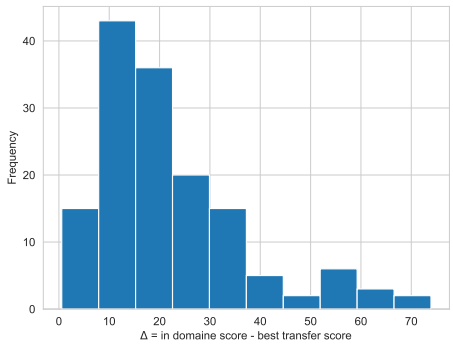

Figure 2: Distribution of the difference between the in-domain PoS tagging score and the best score obtained in cross-lingual transfer.

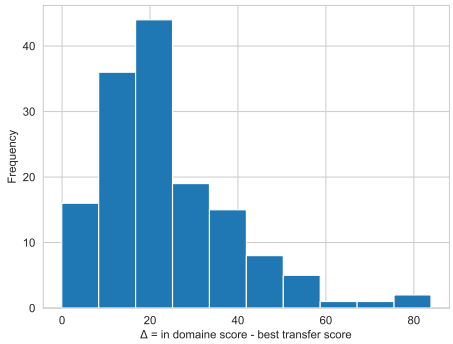

Figure 3: Distribution of the difference between the in-domain UAS and the best UAS obtained in cross-lingual transfer.

```
linear regressor:
{
    "reg__alpha": [0.03, 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, 100.0],
    "reg__fit_intercept": [True, False],
}

hgbr regressor:
{
    "reg__learning_rate": [0.2, 0.1, 0.05, 0.02],
    "reg__max_iter": [200, 400, 800],
    "reg__max_leaf_nodes": [15, 31, 63, 127],
    "reg__min_samples_leaf": [5, 10, 20, 40],
    "reg__l2_regularization": [0.0, 0.05, 0.1, 0.5],
    "reg__max_bins": [128, 255],
    "reg__early_stopping": [True],
    "reg__validation_fraction": [0.1],
}
```

Figure 4: Hyper-Parameters grid.

also include a metric assessing the quality of the predictions (MSE) and another measuring the regressor's ability to identify the best source language (P@1).

D Impact of the strict split strategy

We report in Table 4 a systematic comparison of the two ways of splitting our data into training and test sets for different feature configurations. The data are split either by ensuring that the target languages in the test set never appear in the training data, whether as source or target languages (strict mode), or by ensuring that they do not appear as targets only (but may appear as sources), as is commonly done (e.g. (Lin et al., 2019)). As expected, the strict split leads to a slight but consistent decrease in performance, likely because it forces us to discard part of the training data.

features	MRR	P@1	P@5	R^2	MAE	MSE
HGBR						
datasets	0.29 ± 0.06	0.20 ± 0.06	0.33 ± 0.06	0.64 ± 0.01	8.92 ± 0.44	11.79 ± 0.41
datasets+linguistic	0.59 ± 0.05	0.52 ± 0.05	0.65 ± 0.05	0.74 ± 0.02	7.61 ± 0.5	9.91 ± 0.57
linguistic w/o match	0.12 ± 0.03	0.04 ± 0.02	0.14 ± 0.05	0.18 ± 0.07	13.88± 1.03	17.74 ± 1.15
linguistic	0.49 ± 0.06	0.45 ± 0.06	0.51 ± 0.06	0.22 ± 0.07	13.58 ± 1.07	17.24 ± 1.22
similarities	0.52 ± 0.08	0.48 ± 0.09	0.54 ± 0.09	0.18 ± 0.03	14.51± 0.57	17.64 ± 0.64
similarities+datasets	0.63 ± 0.05	0.54 ± 0.05	0.71 ± 0.05	0.68 ± 0.01	8.43 ± 0.29	10.98 ± 0.36
similarities+datasets+linguistic w/o match	0.65 ± 0.04	0.58 ± 0.05	0.69 ± 0.06	0.75 ± 0.02	7.58 ± 0.48	9.8 ± 0.51
similarities+datasets+linguistic	0.64 ± 0.04	0.56 ± 0.04	0.74 ± 0.05	0.75 ± 0.02	7.61 ± 0.46	9.86 ± 0.5
similarities+linguistic	0.55 ± 0.05	0.49 ± 0.04	0.6 ± 0.08	0.27 ± 0.05	13.27 ± 0.72	16.71 ± 0.91
Linear						
datasets	0.25 ± 0.03	0.14 ± 0.02	0.34 ± 0.04	-2.33 ± 0.38	33.34 ± 0.79	35.46 ± 0.55
datasets+linguistic	0.51 ± 0.10	0.45 ± 0.12	0.53 ± 0.10	-2.37 ± 0.58	31.99 ± 1.77	35.61 ± 1.92
linguistic w/o match	0.07 ± 0.02	0.02 ± 0.02	0.06 ± 0.03	-2.72 ± 0.48	32.13 ± 1.20	37.53 ± 1.59
linguistic	0.45 ± 0.06	0.41 ± 0.06	0.46 ± 0.07	-2.69 ± 0.43	32.16 ± 0.86	37.37 ± 1.10
similarities	0.52 ± 0.08	0.48 ± 0.08	0.54 ± 0.10	-2.48 ± 0.27	31.94 ± 0.68	36.35 ± 0.67
similarities+datasets	0.60 ± 0.04	0.50 ± 0.04	0.69 ± 0.05	-2.32 ± 0.43	33.42 ± 0.92	35.40 ± 0.73
all	0.52 ± 0.09	0.45 ± 0.10	0.59 ± 0.10	-2.39 ± 0.63	31.91 ± 1.96	35.65 ± 2.30
similarities+linguistic	0.48 ± 0.08	0.44 ± 0.08	0.49 ± 0.09	-2.65 ± 0.38	32.19 ± 0.78	37.22 ± 1.04
similarities+datasets+linguistic (w/o match)	0.46 ± 0.10	0.35 ± 0.12	0.58 ± 0.10	-2.4 ± 0.72	32.11 ± 2.04	35.69 ± 2.50

Table 3: Performance on the test set for our two regression models. The 95% CI are computed following bootstraping (Efron and Tibshirani, 1993)

features	strict split	MRR	R^2	MAE
datasets	✗	0.33 ± 0.06	0.68 ± 0.02	8.22 ± 0.39
	✓	0.29 ± 0.06	0.64 ± 0.01	8.92 ± 0.44
datasets+linguistic	✗	0.59 ± 0.05	0.78 ± 0.02	7.04 ± 0.33
	✓	0.59 ± 0.05	0.74 ± 0.02	7.61 ± 0.5
linguistic	✗	0.5 ± 0.05	0.31 ± 0.06	12.53 ± 0.58
	✓	0.49 ± 0.06	0.22 ± 0.07	13.58 ± 1.07
linguistic w/o match	✗	0.17 ± 0.02	0.3 ± 0.07	12.54 ± 0.85
	✓	0.12 ± 0.03	0.18 ± 0.07	13.88 ± 1.03
similarities	✗	0.52 ± 0.08	0.25 ± 0.03	13.85 ± 0.45
	✓	0.52 ± 0.08	0.18 ± 0.03	14.51 ± 0.57
similarities+datasets	✗	0.64 ± 0.05	0.73 ± 0.02	7.81 ± 0.24
	✓	0.63 ± 0.05	0.68 ± 0.01	8.43 ± 0.29
similarities+datasets+linguistic	✗	0.63 ± 0.06	0.78 ± 0.02	6.99 ± 0.31
	✓	0.64 ± 0.04	0.75 ± 0.02	7.61 ± 0.46
similarities+linguistic	✗	0.53 ± 0.06	0.37 ± 0.05	12.13 ± 0.48
	✓	0.55 ± 0.05	0.27 ± 0.05	13.27 ± 0.72

Table 4: Comparison of performance under different train/test splitting strategies for PoS performance prediction (see text for details).