Predicting Performance for Natural Language Processing Tasks

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, Graham Neubig
Language Technologies Institute, Carnegie Mellon University
{mengzhox,aanastas,yiming,gneubig}@cs.cmu.edu
ruochenx@gmail.com

Abstract

Given the complexity of combinations of tasks, languages, and domains in natural language processing (NLP) research, it is computationally prohibitive to exhaustively test newly proposed models on each possible experimental setting. In this work, we attempt to explore the possibility of gaining plausible judgments of how well an NLP model can perform under an experimental setting, *without actually training or testing the model*. To do so, we build regression models to predict the evaluation score of an NLP experiment given the experimental settings as input. Experimenting on 9 different NLP tasks, we find that our predictors can produce meaningful predictions over unseen languages and different modeling architectures, outperforming reasonable baselines as well as human experts. Going further, we outline how our predictor can be used to find a small subset of representative experiments that should be run in order to obtain plausible predictions for all other experimental settings.[1]

1 Introduction

Natural language processing (NLP) is an extraordinarily vast field, with a wide variety of models being applied to a multitude of tasks across a plenitude of domains and languages. In order to measure progress in all these scenarios, it is necessary to compare performance on test datasets representing each scenario. However, the cross-product of tasks, languages, and domains creates an explosion of potential application scenarios, and it is infeasible to collect high-quality test sets for each. In addition, even for tasks where we do have a wide variety of test data, e.g. for well-resourced tasks such as machine translation (MT), it is still computationally prohibitive as well as not environmentally friendly (Strubell et al., 2019) to build and test on systems for all languages or domains we are interested in. Because of this, the common practice is to test new methods on a small number of languages or domains, often semi-arbitrarily chosen based on previous work or the experimenters' intuition.

As a result, this practice impedes the NLP community from gaining a comprehensive understanding of newly-proposed models. Table 1 illustrates this fact with an example from bilingual lexicon induction, a task that aims to find word translation pairs from cross-lingual word embeddings. As vividly displayed in Table 1, almost all the works report evaluation results on a different subset of language pairs. Evaluating only on a small subset raises concerns about making inferences when comparing the merits of these methods: there is no guarantee that performance on English–Spanish (EN–ES, the only common evaluation dataset) is representative of the expected performance of the models over all other language pairs (Anastasopoulos and Neubig, 2020). Such phenomena lead us to consider if it is possible to make a decently accurate estimation for the performance over an untested language pair without actually running the NLP model to bypass the computation restriction.

Toward that end, through drawing on the idea of characterizing an experiment from Lin et al. (2019), we propose a framework, which we call **NLPERF**, to provide an exploratory solution. We build regression models, to *predict the performance on a particular experimental setting* given past experimental records of the same task, with each record consisting of a characterization of its training dataset and a performance score of the corresponding metric. Concretely, in §2, we start with a partly populated table (such as the one from

[1]Code, data and logs are publicly available at https://github.com/xiamengzhou/NLPerf.

BLI Method	Evaluation Set											
	DE–EN	EN–DE	ES–EN	EN–ES	FR–EN	EN–FR	IT–EN	EN–IT	EN–PT	EN–RU	ES–DE	PT–RU
Zhang et al. (2017)	?	✓	✓	✓	?	?	✓	?	?	?	?	?
Chen and Cardie (2018)	✓	✓	✓	✓	✓	✓	✓	✓	✓	?	✓	?
Yang et al. (2019)	✓	✓	✓	✓	✓	✓	✓	?	?	?	?	?
Heyman et al. (2019)	?	✓	?	✓	?	✓	?	✓	?	?	?	?
Huang et al. (2019)	?	?	✓	✓	✓	✓	?	?	?	?	?	?
Artetxe et al. (2019)	✓	✓	✓	✓	✓	✓	?	?	?	✓	?	?

Table 1: An illustration of the comparability issues across methods and multiple evaluation datasets from the Bilingual Lexicon Induction task. Our prediction model can reasonably fill in the blanks, as illustrated in Section 4.

Table 1) and attempt to infer the missing values with the predictor. We begin by introducing the process of characterizing an NLP experiment for each task in §3. We evaluate the effectiveness and robustness of NLPERF by comparing to multiple baselines, human experts, and by perturbing a single feature to simulate a grid search over that feature (§4). Evaluations on multiple tasks show that NLPERF is able to outperform all baselines. Notably, on a machine translation (MT) task, the predictions made by the predictor turn out to be more accurate than human experts.

An effective predictor can be very useful for multiple applications associated with practical scenarios. In §5, we show how it is possible to adopt the predictor as a scoring function to find a small subset of experiments that are most *representative* of a bigger set of experiments. We argue that this will allow researchers to make informed decisions on what datasets to use for training and evaluation, in the case where they cannot experiment on all experimental settings. Last, in §6, we show that we can adequately predict the performance of new models even with a minimal number of experimental records.

2 Problem Formulation

In this section we formalize the problem of predicting performance on supervised NLP tasks. Given an NLP model of architecture $\mathcal{M}$ trained over dataset(s) $\mathcal{D}$ of a specific task involving language(s) $\mathcal{L}$ with a training procedure (optimization algorithms, learning rate scheduling etc.) $\mathcal{P}$, we can test the model on a test dataset $\mathcal{D}'$ and get a score $\mathcal{S}$ of a specific evaluation metric. The resulting score will surely vary depending on all the above mentioned factors, and we denote this relation as g:

$$\mathcal{S}_{\mathcal{M},\mathcal{P},\mathcal{L},\mathcal{D},\mathcal{D}'} = g(\mathcal{M},\mathcal{P},\mathcal{L},\mathcal{D},\mathcal{D}'). \quad (1)$$

In the ideal scenario, for each test dataset $\mathcal{D}'$ of a specific task, one could enumerate all different settings and find the one that leads to the best performance. As mentioned in Section §1, however, such a brute-force method is computationally infeasible. Thus, we turn to modeling the process and formulating our problem as a regression task by using a parametric function f_θ to approximate the true function g as follows:

$$\hat{\mathcal{S}}_{\mathcal{M},\mathcal{P},\mathcal{L},\mathcal{D},\mathcal{D}'} = f_\theta([\Phi_\mathcal{M};\Phi_\mathcal{P};\Phi_\mathcal{L};\Phi_\mathcal{D};\Phi_{\mathcal{D}'}])$$

where Φ_* denotes a set of features for each influencing factor.

For the purpose of this study, we mainly focus on dataset and language features $\Phi_\mathcal{L}$ and $\Phi_\mathcal{D}$, as this already results in a significant search space, and gathering extensive experimental results with fine-grained tuning over model and training hyperparameters is both expensive and relatively complicated. In the cases where we handle multiple models, we only use a single categorical model feature to denote the combination of model architecture and training procedure, denoted as $\Phi_\mathcal{C}$. We still use the term *model* to refer to this combination in the rest of the paper. We also omit the test set features, under the assumption that the data distributions for training and testing data are the same (a fairly reasonable assumption if we ignore possible domain shift). Therefore, for all experiments below, our final prediction function is the following:

$$\hat{\mathcal{S}}_{\mathcal{C},\mathcal{L},\mathcal{D}} = f_\theta([\Phi_\mathcal{C};\Phi_\mathcal{L};\Phi_\mathcal{D}])$$

In the next section we describe concrete instantiations of this function for several NLP tasks.

3 NLP Task Instantiations

To build a predictor for NLP task performance, we must 1) select a task, 2) describe its featurization, and 3) train a predictor. We describe details of these three steps in this section.

Task	Dataset Citation	Source Langs	Target Langs	Transfer Langs	# Models	# EXs	Task Metric
Wiki-MT	Schwenk et al. (2019)	39	39	–	single	995	BLEU
TED-MT	Qi et al. (2018)	54	1	–	single	54	BLEU
TSF-MT	Qi et al. (2018)	54	1	54	single	2862	BLEU
TSF-PARSING	Nivre et al. (2018)	–	30	30	single	870	Accuracy
TSF-POS	Nivre et al. (2018)	–	26	60	single	1531	Accuracy
TSF-EL	Rijhwani et al. (2019)	–	9	54	single	477	Accuracy
BLI	Lample et al. (2018)	44	44	–	3	88×3	Accuracy
MA	McCarthy et al. (2019)	–	66	–	6	107×6	F1
UD	Zeman et al. (2018a)	–	53	–	25	72×25	F1

Table 2: Statistics of the datasets we use for training predictors. # EXs denote the total number of experiment instances; Task Metric reflects how the models are evaluated.

Tasks We test on tasks including bilingual lexicon induction (BLI); machine translation trained on aligned Wikipedia data (Wiki-MT), on TED talks (TED-MT), and with cross-lingual transfer for translation into English (TSF-MT); cross-lingual dependency parsing (TSF-Parsing); cross-lingual POS tagging (TSF-POS); cross-lingual entity linking (TSF-EL); morphological analysis (MA) and universal dependency parsing (UD). Basic statistics on the datasets are outlined in Table 2.

For Wiki-MT tasks, we collect experimental records directly from the paper describing the corresponding datasets (Schwenk et al., 2019). For TED-MT and all the transfer tasks, we use the results of Lin et al. (2019). For BLI, we conduct experiments using published results from three papers, namely Artetxe et al. (2016), Artetxe et al. (2017) and Xu et al. (2018). For MA, we use the results of the SIGMORPHON 2019 shared task 2 (McCarthy et al., 2019). Last, the UD results are taken from the CoNLL 2018 Shared Task on universal dependency parsing (Zeman et al., 2018b).

Featurization For language features, we utilize six distance features from the URIEL Typological Database (Littell et al., 2017), namely geographic, genetic, inventory, syntactic, phonological, and featural distance.

The complete set of dataset features includes the following:

1. *Dataset Size:* The number of data entries used for training.
2. *Word/Subword Vocabulary Size:* The number of word/subword types.
3. *Average Sentence Length:* The average length of sentences from all experimental.
4. *Word/Subword Overlap:*
$$\frac{|T_1 \cap T_2|}{|T_1| + |T_2|}$$
where T_1 and T_2 denote vocabularies of any two corpora.
5. *Type-Token Ratio (TTR):* The ratio between the number of types and number of tokens (Richards, 1987) of one corpus.
6. *Type-Token Ratio Distance:*
$$\left(1 - \frac{\mathrm{TTR}_1}{\mathrm{TTR}_2}\right)^2$$
where TTR_1 and TTR_2 denote TTR of any two corpora.
7. *Single Tag Type:* Number of single tag types.
8. *Fused Tag Type:* Number of fused tag types.
9. *Average Tag Length Per Word:* Average number of single tags for each word.
10. *Dependency Arcs Matching WALS Features:* the proportion of dependency parsing arcs matching the following WALS features, computed over the training set: subject/object/oblique before/after verb and adjective/numeral before/after noun.

For transfer tasks, we use the same set of dataset features $\Phi_{\mathcal{D}}$ as Lin et al. (2019), including features 1–6 on the source and the transfer language side. We also include language distance features between source and transfer language, as well as between source and target language. For MT tasks, we use features 1–6 and language distance features, but only between the source and target language. For MA, we use features 1, 2, 5 and morphological tag related features 7–9. For UD, we

use features 1, 2, 5, and 10. For BLI, we use language distance features and URIEL syntactic features for the source and the target language.

Predictor Our prediction model is based on gradient boosting trees (Friedman, 2001), implemented with XGBoost (Chen and Guestrin, 2016). This method is widely known as an effective means for solving problems including ranking, classification and regression. We also experimented with Gaussian processes (Williams and Rasmussen, 1996), but settled on gradient boosted trees because performance was similar and Xgboost's implementation is very efficient through the use of parallelism. We use squared error as the objective function for the regression and adopted a fixed learning rate 0.1. To allow the model to fully fit the data we set the maximum tree depth to be 10 and the number of trees to be 100, and use the default regularization terms to prevent the model from overfitting.

4 Can We Predict NLP Performance?

In this section we investigate the effectiveness of NLPERF across different tasks on various metrics. Following Lin et al. (2019), we conduct k-fold cross validation for evaluation. To be specific, we randomly partition the experimental records of $\langle \mathcal{L}, \mathcal{D}, \mathcal{C}, \mathcal{S} \rangle$ tuples into k folds, and use $k-1$ folds to train a prediction model and evaluate on the remaining fold. Note that this scenario is similar to "filling in the blanks" in Table 1, where we have some experimental records that we can train the model on, and predict the remaining ones.

For evaluation, we calculate the average root mean square error (RMSE) between the predicted scores and the true scores.

Baselines We compare against a simple mean value baseline, as well as against language-wise mean value and model-wise mean value baselines. The simple mean value baseline outputs an average of scores s from the training folds for all test entries in the left-out fold (i) as follows:

$$\hat{s}^{(i)}_{\text{mean}} = \frac{1}{|\mathcal{S} \setminus \mathcal{S}^{(i)}|} \sum_{s \in \mathcal{S} \setminus \mathcal{S}^{(i)}} s; i \in 1 \dots k \quad (2)$$

Note that for tasks involving multiple models, we calculate the RMSE score separately on each model and use the mean RMSE of all models as the final RMSE score.

The language-wise baselines make more informed predictions, taking into account only training instances with the same transfer, source, or target language (depending on the task setting). For example, the source-language mean value baseline $\hat{s}^{(i,j)}_{\text{s-lang}}$ for j^{th} test instance in fold i outputs an average of the scores s of the training instances that share the *same* source language features s-lang, as shown in Equation 3:

$$\hat{s}^{(i,j)}_{\text{s-lang}} = \frac{\sum_{s,\phi} \delta(\phi_{\mathcal{L},\text{src}} = \text{s-lang}) \cdot s}{\sum_{s,\phi} \delta(\phi_{\mathcal{L},\text{src}} = \text{s-lang})} \quad (3)$$
$$\forall (s, \phi) \in (|\mathcal{S} \setminus \mathcal{S}^{(i)}|, |\Phi \setminus \Phi^{(i)}|)$$

where δ is the indicator function. Similarly, we define the target- and the transfer-language mean value baselines.

In a similar manner, we also compare against a model-wise mean value baseline for tasks that include experimental records from multiple models. Now, the prediction for the j^{th} test instance in the left-out fold i is an average of the scores on the same dataset (as characterized by the language $\phi_{\mathcal{L}}$ and dataset $\phi_{\mathcal{D}}$ features) from all other models:

$$\hat{s}^{(i,j)}_{\text{model}} = \frac{\sum_{s,\phi} \delta(\phi_{\mathcal{L}} = \text{lang}, \phi_{\mathcal{D}} = \text{data}) \cdot s}{\sum_{s,\phi} \delta(\phi_{\mathcal{L}} = \text{lang}, \phi_{\mathcal{D}} = \text{data})} \quad (4)$$
$$\forall (s, \phi) \in (|\mathcal{S} \setminus \mathcal{S}^{(i)}|, |\Phi \setminus \Phi^{(i)}|)$$

where $\text{lang} = \Phi^{(i,j)}_{\mathcal{L}}$ and $\text{data} = \Phi^{(i,j)}_{\mathcal{D}}$ respectively denote the language and dataset features of the test instance.

Main Results For multi-model tasks, we can do either **S**ingle **M**odel prediction (SM), restricting training and testing of the predictor within a single model, or **M**ulti-**M**odel (MM) prediction using a categorical model feature. The RMSE scores of NLPERF along with the baselines are shown in Table 3. For all tasks, our single model predictor is able to more accurately estimate the evaluation score of unseen experiments compared to the single model baselines, confirming our hypothesis that the there exists a correlation that can be captured between experimental settings and the downstream performance of NLP systems. The language-wise baselines are much stronger than the simple mean value baseline but still perform worse than our single model predictor. Similarly, the model-wise baseline significantly outperforms the mean value baseline because results from other models reveal much information about the dataset.

Model	Task								
	Wiki-MT	TED-MT	TSF-MT	TSF-PARSING	TSF-POS	TSF-EL	BLI	MA	UD
Mean	6.40	12.65	10.77	17.58	29.10	18.65	20.10	9.47	17.69
Transfer Lang-wise	–	–	10.96	15.68	29.98	20.55	–	–	–
Source Lang-wise	5.69	12.65	2.24	–	–	–	20.13	–	–
Target Lang-wise	5.12	12.65	10.78	12.05	8.92	8.61	20.00	9.47	–
NLPERF (SM)	2.50	6.18	1.43	6.24	7.37	7.82	12.63	6.48	12.06
Model-wise	–	–	–	–	–	–	8.77	5.22	4.96
NLPERF (MM)	–	–	–	–	–	–	6.87	3.18	3.54

Table 3: RMSE scores of three baselines and our predictions under the single model and multi model setting (missing values correspond to settings not applicable to the task). All results are from k-fold ($k = 5$) evaluations averaged over 10 random runs.

Even so, our multi-model predictor still outperforms the model-wise baseline.

The results nicely imply that for a wide range of tasks, our predictor is able to reasonably estimate left-out slots in a partly populated table given results of other experiment records, without actually running the system.

We should note that RMSE scores across different tasks should not be directly compared, mainly because the scale of each evaluation metric is different. For example, a BLEU score (Papineni et al., 2002) for MT experiments typically ranges from 1 to 40, while an accuracy score usually has a much larger range, for example, BLI accuracy ranges from 0.333 to 78.2 and TSF-POS accuracy ranges from 1.84 to 87.98, which consequently makes the RMSE scores of these tasks higher.

Comparison to Expert Human Performance We constructed a small scale case study to evaluate whether NLPERF is competitive to the performance of NLP sub-field experts. We focused on the TED-MT task and recruited 10 MT practitioners,[2] all of whom had published at least 3 MT-related papers in ACL-related conferences.

In the first set of questions, the participants were presented with language pairs from one of the k data folds along with the dataset features and were asked to estimate an eventual BLEU score for each data entry. In the second part of the questionnaire, the participants were tasked with making estimations on the same set of language pairs, but this time they also had access to features, and BLEU scores from all the other folds.[3]

[2]None of the study participants were affiliated to the authors' institutions, nor were familiar with this paper's content.

[3]The interested reader can find an example questionnaire (and make estimations over one of the folds) in the A.

Predictor	RMSE
Mean Baseline	12.64
Human (w/o training data)	9.38
Human (w/ training data)	7.29
NLPERF	**6.04**

Table 4: Our model performs better than human MT experts on the TED-MT prediction task.

The partition of the folds is consistent between the human study and the training/evaluation for the predictor. While the first sheet is intended to familiarize the participants with the task, the second sheet fairly adopts the training/evaluation setting for our predictor. As shown in Table 4, our participants outperform the mean baseline even without information from other folds, demonstrating their own strong prior knowledge in the field. In addition, the participants make more accurate guesses after acquiring more information on experimental records in other folds. In neither case, though, are the human experts competitive to our predictor. In fact, only one of the participants achieved performance comparable to our predictor.

Feature Perturbation Another question of interest concerning predicting performance is "how will the model perform when trained on data of a different size" (Kolachina et al., 2012a), or other varieties of extrapolating to unseen feature combinations. To test NLPERF's extrapolation ability in this regard, we conduct an array of experiments on one language pair with various data sizes on the Wiki-MT task. We pick two language pairs, Turkish to English (TR–EN) and Portuguese to

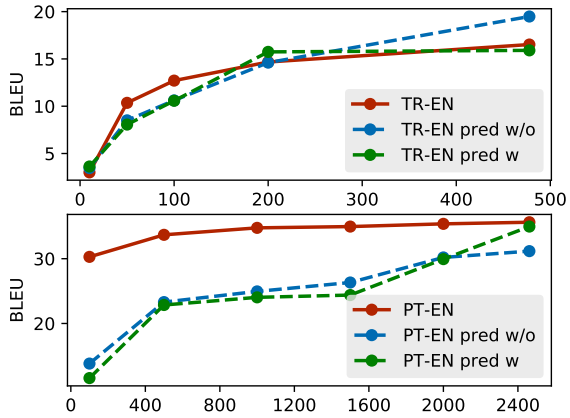


Figure 1: Our model's predicted BLEU scores and true BLEU scores, on sampled TR–EN datasets (sizes 10k/50k/100k/200k/478k) and PT–EN datasets (sizes 100k/500k/1000k/2000k/2462k). Pred w/ denotes that the predictor is trained with one data point from the target language pair (full size), achieving an rmse score of 1.53 and 10.99 on the tr–en and pt–en pair respectively. Pred w/o denotes that the predictor is trained without any data points from the target language pair, achieving an rmse score of 1.83 and 9.97.

English (PT–EN) as our testbed for the Wiki-MT task. We down-sample parallel datasets with different sizes and train MT models with each sampled dataset to obtain the true BLEU scores. We then collect the features of all sampled datasets and use our predictor to obtain predictions. We train two varieties of predictor for each language pair: one is trained with experimental records from all other languages, and the other is trained with a single additional experimental record of the tested language pair using the full dataset without down-sampling. The plot of true BLEU scores and predicted BLEU scores are shown in Figure 1. Our predictor achieves a very low average RMSE of 1.83/1.53 for TR–EN pair but a relatively higher RMSE of 9.97/10.99 for PT–EN pair. The favorable performance on the tr-en pair demonstrates the possibility of our predictor to do feature extrapolation over data set size. In contrast, the predictions on the pt-en pair are significantly less accurate. This is due to the fact that there are only two other experimental settings scoring as high as 34 BLEU score, and both have data sizes of 3378k (en-es) and 611k (gl-es), leading to the predictor inadequately predicting high BLEU scores for low-resourced settings during extrapolation. This reveals the fact that while the predictor is able to extrapolate performance on settings similar to what it has seen in the data, it may be less successful under circumstances unlike its training inputs. What's more, adding one data point for the tested language pair improves the performance for the sampled data point, but the predictor seems to rely more on experimental records of other language pairs for the datapoints not covered in the data. It is possible that another functional form for the predictor (such as that explored by Rosenfeld et al. (2020)) may be more effective in this regard.

5 What Datasets Should We Test On?

As shown in Table 1, it is common practice to test models on a subset of all available datasets. The reason for this is practical – it is computationally prohibitive to evaluate on all settings. However, if we pick test sets that are not *representative* of the data as a whole, we may mistakenly reach unfounded conclusions about how well models perform on other data with distinct properties. For example, models trained on a small-sized dataset may not scale well to a large-sized one, or models that perform well on languages with a particular linguistic characteristic may not do well on languages with other characteristics (Bender and Friedman, 2018).

Here we ask the following question: if we are only practically able to test on a small number of experimental settings, which ones should we test on to achieve maximally representative results? Answering the question could have practical implications: organizers of large shared tasks like SIGMORPHON (McCarthy et al., 2019) or UD (Zeman et al., 2018a) could create a minimal subset of settings upon which they would ask participants to test to get representative results; similarly, participants could possibly expedite the iteration of model development by testing on the representative subset only. A similar avenue for researchers and companies deploying systems over multiple languages could lead to not only financial savings, but potentially a significant cut-down of emissions from model training (Strubell et al., 2019).

We present an approximate explorative solution to the problem mentioned above. Formally, assume that we have a set $\mathcal{N}$, comprising experimental records (both features and scores) of n datasets for one task. We set a number m ($< n$) of datasets that we would like to select as the representative subset. By defining $\mathrm{RMSE}_{\mathcal{A}}(\mathcal{B})$ to be the RMSE score derived from evaluating on one subset $\mathcal{B}$ the

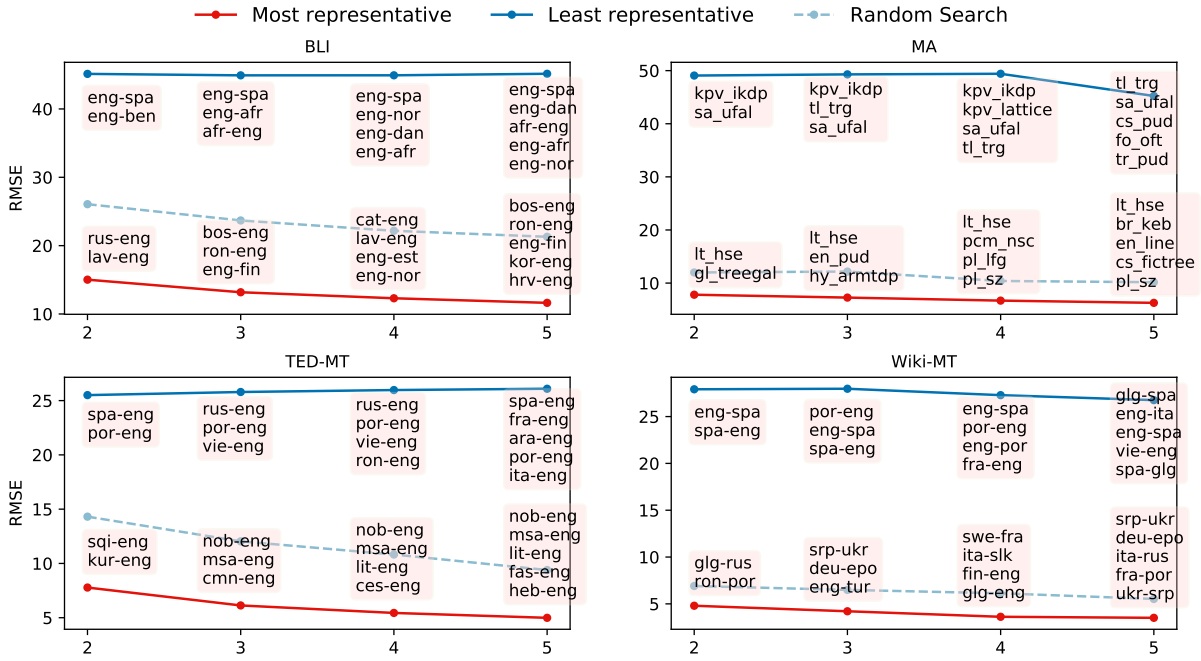


Figure 2: Beam search results (beam size=100) for up to the 5 most (and least) representative datasets for 4 NLP tasks. We also show random search results averaged over 100 random runs.

predictor trained on another subset of experimental records $\mathcal{A}$, we consider the *most representative* subset $\mathcal{D}$ to be the one that minimizes the RMSE score when predicting all of the other datasets:

$$\arg\min_{\mathcal{D}\subset\mathcal{N}} \mathrm{RMSE}_{\mathcal{D}}(\mathcal{N}\setminus\mathcal{D}). \quad (5)$$

Naturally, enumerating all $\binom{n}{m}$ possible subsets would be prohibitively costly, even though it would lead to the optimal solution. Instead, we employ a beam-search-like approach to efficiently search for an approximate solution to the best performing subset of arbitrary size. Concretely, we start our approximate search with an exhaustive enumeration of all subsets of size 2. At each following step t, we only consider the best k subsets $\{\mathcal{D}_t^{(i)}; i \in 1,\ldots,k\}$ into account and discard the rest. As shown in Equation 6, for each candidate subset, we expand it with one more data point,

$$\{\mathcal{D}_t^{(i)} \cup \{s\}; \forall i \in 1\ldots k, s \in \mathcal{N}\setminus\mathcal{D}_t^{(i)}\}. \quad (6)$$

For tasks that involve multiple models, we take experimental records of the selected dataset from all models into account during expansion. Given all expanded subsets, we train a predictor for each to evaluate on the rest of the data sets, and keep the best performing k subsets $\{\mathcal{D}_{t+1}^{(i)}; i \in 1,\ldots,k\}$ with minimum RMSE scores for the next step. Furthermore, note that by simply changing the $\arg\min$ to an $\arg\max$ in Equation 5, we can also find the *least* representative datasets.

We present search results for four tasks[4] as beam search progresses in Figure 2, with corresponding RMSE scores from all remaining datasets as the y-axis. For comparison, we also conduct random searches by expanding the subset with a randomly selected experimental record. In all cases, the most representative sets are an aggregation of datasets with diverse characteristics such as languages and dataset sizes. For example, in the Wiki-MT task, the 5 most representative datasets include languages that fall into a diverse range of language families such as Romance, Turkic, Slavic, etc. while the least representative ones include duplicate pairs (opposite directions) mostly involving English. The phenomenon is more pronounced in the TED-MT task, where not only the 5 most representative source languages are diverse, but also the dataset sizes. Specifically, the Malay-English (msa-eng) is a tiny dataset (5k parallel sentences), and Hebrew-English (heb-eng) is a high-resource case (212k parallel sentences).

Notably, for BLI task, to test how representative the commonly used datasets are, we select the most frequent 5 language pairs shown in

[4]Readers can find results on other tasks in Appendix B.

Table 1, namely en-de, es-en, en-es, fr-en, en-fr for evaluation. Unsurprisingly, we get an RMSE score as high as 43.44, quite close to the performance of the worst representative set found using beam search. This finding indicates that the standard practice of choosing datasets for evaluation is likely unrepresentative of results over the full dataset spectrum, well aligned with the claims in Anastasopoulos and Neubig (2020).

A particularly encouraging observation is that the predictor trained with only the 5 most representative datasets can achieve an RMSE score comparable to k-fold validation, which required using all of the datasets for training.[5] This indicates that one would only need to train NLP models on a small set of *representative* datasets to obtain reasonably plausible predictions for the rest.

6 Can We Extrapolate Performance for New Models?

In another common scenario, researchers propose new models for an existing task. It is both time-consuming and computationally intensive to run experiments with all settings for a new model. In this section, we explore if we can use past experimental records from other models and a minimal set of experiments from the new model to give a plausible prediction over the rest of the datasets, potentially reducing the time and resources needed for experimenting with the new model to a large extent. We use the task of UD parsing as our testbed[6] as it is the task with most unique models (25 to be exact). Note that we still only use a single categorical feature for the model type.

To investigate how many experiments are needed to have a plausible prediction for a new model, we first split the experimental records equally into a sample set and a test set. Then we randomly sample n $(0 \leq n \leq 5)$ experimental records from the sample set and add them into the collection of experiment records of past models. Each time we re-train a predictor and evaluate on the test set. The random split repeats 50 times and the random sampling repeats 50 times, adding up to a total of 2500 experiments. We use the mean value of the results from other models, shown in Equation 7 as the prediction baseline for the left-out model, and because experiment results of other models reveal significant information about the dataset, this serves as a relatively strong baseline:

$$\hat{s}_k = \frac{1}{n-1} \sum_{i=1}^{n} \mathbb{1}(i \in \mathcal{M}/\{k\}) \cdot s_i. \quad (7)$$

$\mathcal{M}$ denotes a collection of models and k denotes the left-out model.

We show the prediction performance (in RMSE) over 8 systems[7] in Figure 3. Interestingly, the predictor trained with no model records (0) outperforms the mean value baseline for the 4 best systems, while it is the opposite case on the 4 worst systems. Since there is no information provided about the new-coming model, the predictions are solely based on dataset and language features. One reason might explain the phenomenon - the correlation between the features and the scores of the worse-performing systems is different from those better-performing systems, so the predictor is unable to generalize well (ONLP).

In the following discussion, we use RMSE@n to denote the RMSE from the predictor trained with n data points of a new model. The relatively low RMSE@0 scores indicate that other models' features and scores are informative for predicting the performance of the new model even without new model information. Comparing RMSE@0 and RMSE@1, we observe a consistent improvement for almost all systems, indicating that NLPERF trained on even a single extra random example achieves more accurate estimates over the test sets. Adding more data points consistently leads to additional gains. However, predictions on worse-performing systems benefit more from it than for better-performing systems, indicating that their feature-performance correlation might be considerably different. The findings here indicate that by extrapolating from past experiments, one can make plausible judgments for newly developed models.

7 Related Work

As discusssed in Domhan et al. (2015), there are two main threads of work focusing on predicting performance of machine learning algorithms. The first thread is to predict the performance of a method as a function of its training time, while the second thread is to predict a method's performance as a function of the training dataset size. Our work belongs in the second thread, but could easily be extended to encompass training time/procedure.

[5] to be accurate, $k-1$ folds of all datasets.

[6] MA and BLI task results are in Appendix C

[7] The best and worst 4 systems from the shared task.

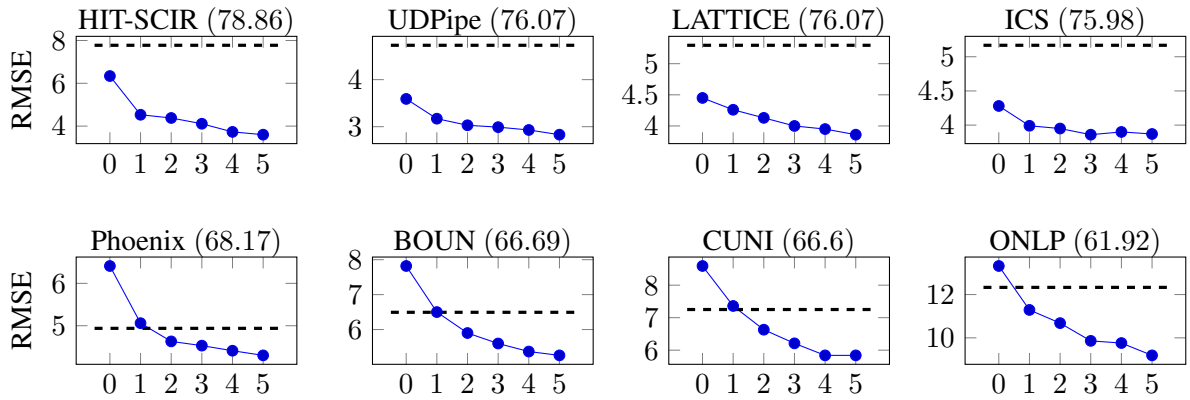


Figure 3: RMSE scores of **UD** task from dataset-wise mean value predictor (the dashed black line in each graph) and predictors trained with experimental records of other models and 0–5 records from a new model.

In the first thread, Kolachina et al. (2012b) attempt to infer learning curves based on training data features and extrapolate the initial learning curves based on BLEU measurements for statistical machine translation (SMT). By extrapolating the performance of initial learning curves, the predictions on the remainder allows for early termination of a bad run (Domhan et al., 2015).

In the second thread, Birch et al. (2008) adopt linear regression to capture the relationship between data features and SMT performance and find that the amount of reordering, the morphological complexity of the target language and the relatedness of the two languages explains the majority of performance variability. More recently, Elsahar and Gallé (2019) use domain shift metrics such as $\mathcal{H}$-divergence based metrics to predict drop in performance under domain-shift. Rosenfeld et al. (2020) explore the functional form of the dependency of the generalization error of neural models on model and data size. We view our work as a generalization of such approaches, appropriate for application on any NLP task.

8 Conclusion and Future Work

In this work, we investigate whether the experiment setting itself is informative for predicting the evaluation scores of NLP tasks. Our findings promisingly show that given a sufficient number of past training experimental records, our predictor can 1) outperform human experts; 2) make plausible predictions even over new-coming models and languages; 3) extrapolate well on features like dataset size; 4) provide a guide on how we should choose representative datasets for fast iteration.

While this discovery is a promising start, there are still several avenues on improvement in future work.

First, the dataset and language settings covered in our study are still limited. Experimental records we use are from relatively homogeneous settings, e.g. all datasets in Wiki-MT task are sentencepieced to have 5000 subwords, indicating that our predictor may fail for other subword settings. Our model also failed to generalize to cases where feature values are out of the range of the training experimental records. We attempted to apply the predictor of Wiki-MT to evaluate on a low-resource MT dataset, translating from Mapudungun (arn) to Spanish (spa) with the dataset from Duan et al. (2019), but ended up with a poor RMSE score. It turned out that the average sentence length of the arn–spa data set is much lower than that of the training data sets and our predictors fail to generalize to this different setting.

Second, using a categorical feature to denote model types constrains its expressive power for modeling performance. In reality, a slight change in model hyperparameters (Hoos and Leyton-Brown, 2014; Probst et al., 2019), optimization algorithms (Kingma and Ba, 2014), or even random seeds (Madhyastha and Jain, 2019) may give rise to a significant variation in performance, which our predictor is not able to capture. While investigating the systematic implications of model structures or hyperparameters is practically infeasible in this study, we may use additional information such as textual model descriptions for modeling NLP models and training procedures more elaborately in the future.

Lastly, we assume that the distribution of training and testing data is the same, which does not

consider domain shift. On top of this, there might also be a domain shift between data sets of training and testing experimental records. We believe that modeling domain shift is a promising future direction to improve performance prediction.

Acknowledgement

The authors sincerely thank all the reviewers for their insightful comments and suggestions, Philipp Koehn, Kevin Duh, Matt Post, Shuoyang Ding, Xuan Zhang, Adi Renduchintala, Paul McNamee, Toan Nguyen and Kenton Murray for conducting human evaluation for the TED-MT task, Daniel Beck for discussions on Gaussian Processes, Shruti Rijhwani, Xinyi Wang, Paul Michel for discussions on this paper. This work is generously supported from the National Science Foundation under grant 1761548.

References

Antonios Anastasopoulos and Graham Neubig. 2020. Should all cross-lingual embeddings speak english? In *Proc. ACL*. To appear.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. Bilingual lexicon induction through unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007, Florence, Italy. Association for Computational Linguistics.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 745–754. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Xilun Chen and Claire Cardie. 2018. Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270, Brussels, Belgium. Association for Computational Linguistics.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Mingjun Duan, Carlos Fasola, Sai Krishna Rallabandi, Rodolfo M. Vega, Antonios Anastasopoulos, Lori Levin, and Alan W Black. 2019. A resource for computational experiments on mapudungun. In *Proc. LREC*. To appear.

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. 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 2163–2173.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Geert Heyman, Bregt Verreet, Ivan Vulić, and Marie-Francine Moens. 2019. Learning unsupervised multilingual word embeddings with incremental multilingual hubs. 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 1890–1902.

Holger Hoos and Kevin Leyton-Brown. 2014. An efficient approach for assessing hyperparameter importance. In *International conference on machine learning*, pages 754–762.

Jiaji Huang, Qiang Qiu, and Kenneth Church. 2019. Hubless nearest neighbor search for bilingual lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4072–4080, Florence, Italy. Association for Computational Linguistics.

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

Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. 2012a. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 22–30, Jeju Island, Korea. Association for Computational Linguistics.

Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. 2012b. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 22–30. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

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.

Pranava Madhyastha and Rishabh Jain. 2019. On model stability as a function of random seed. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 929–939, Hong Kong, China. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie

Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. 2019. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 20(53):1–32.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, New Orleans, USA.

Brian Richards. 1987. Type/token ratios: What do they really tell us? *Journal of child language*, 14(2):201–209.

Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime Carbonell. 2019. Zero-shot neural transfer for cross-lingual entity linking. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, Honolulu, Hawaii.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. 2020. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Christopher KI Williams and Carl Edward Rasmussen. 1996. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520.

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474.

Pengcheng Yang, Fuli Luo, Peng Chen, Tianyu Liu, and Xu Sun. 2019. Maam: A morphology-aware alignment model for unsupervised bilingual lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3190–3196.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018a. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018b. Conll 2018 shared task: multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945.

Appendix

A Questionnaire

An example of the first questionnaire from our user case study is shown below. The second sheet also included the results in 44 more language pairs. We provide an answer key after the second sheet.

Please provide your prediction of the BLEU score based on the language pair and dataset features (the domain of the training and test sets is TED talks). After you finish, please go to sheet v2.

idx	Source Language	Target Language	Parallel Sentences (k)	Source vocab size (k)	Source subword vocab size (k)	Target vocab size (k)	Target subword vocab size(k)	BLEU
1	Basque (eus)	English	5	20	8	9	6	
2	Slovak (slk)	English	61	134	8	36	8	
3	Burmese (mya)	English	21	101	8	21	8	
4	Korean (kor)	English	206	386	9	67	8	
5	Lithuanian (lit)	English	42	108	8	29	8	
6	Arabic (ara)	English	214	308	8	69	8	
7	Czech (ces)	English	103	181	8	47	8	
8	Esperanto (epo)	English	7	21	8	10	6	
9	Finnish (fin)	English	24	77	8	22	8	
10	Albanian (sqi)	English	45	93	8	30	8	
11	Vietnamese (vie)	English	172	66	8	61	8	

Please provide your prediction of the BLEU score in the yellow area given all the information in this sheet. Note that all experiments are trained with the same model.

idx	Source Language	Target Lang.	Parallel Sentences (k)	Source vocab size (k)	Source subword vocab size (k)	Target vocab size (k)	Target subword vocab size(k)	BLEU
1	Basque (eus)	English	5	20	8	9	6	
2	Slovak (slk)	English	61	134	8	36	8	
3	Burmese (mya)	English	21	101	8	21	8	
4	Korean (kor)	English	206	386	9	67	8	
5	Lithuanian (lit)	English	42	108	8	29	8	
6	Arabic (ara)	English	214	308	8	69	8	
7	Czech (ces)	English	103	181	8	47	8	
8	Esperanto (epo)	English	7	21	8	10	6	
9	Finnish (fin)	English	24	77	8	22	8	
10	Albanian (sqi)	English	45	93	8	30	8	
11	Vietnamese (vie)	English	172	66	8	61	8	
12	French (fra)	English	192	158	8	65	8	37.74
13	Estonian (est)	English	11	39	8	14	7	9.9
14	Macedonian (mkd)	English	25	61	8	23	8	21.75
15	Bosnian (bos)	English	6	23	8	9	6	32.42
16	Swedish (swe)	English	57	84	8	34	8	33.92
17	Polish (pol)	English	176	267	8	63	8	21.51
18	Persian (fas)	English	151	148	8	57	8	24.5
19	Kurdish (kur)	English	10	39	8	14	7	6.86
20	Hungarian (hun)	English	147	305	8	56	8	22.67
21	Slovenian (slv)	English	20	58	8	20	8	14.18
22	Romanian (ron)	English	181	205	8	63	8	32.42
23	Russian (rus)	English	208	291	8	68	8	22.6
24	Serbian (srp)	English	137	239	8	54	8	30.41
25	Tamil (tam)	English	6	27	8	10	6	1.82
26	Kazakh (kaz)	English	3	15	8	7	5	2.05
27	Marathi (mar)	English	10	29	8	13	7	3.68
28	Ukrainian (ukr)	English	108	191	8	48	8	24.09
29	Thai (tha)	English	98	323	8	45	8	20.34
30	Belarusian (bel)	English	5	20	8	8	5	2.85
31	Turkish (tur)	English	182	304	8	63	8	22.52
32	Azerbaijani (aze)	English	6	23	8	9	6	3.1
33	German (deu)	English	168	194	8	61	8	33.15
34	Bulgarian (bul)	English	174	216	8	62	8	35.78
35	Norwegian (nob)	English	16	36	8	17	7	29.63
36	Georgian (kat)	English	13	44	8	15	7	4.94
37	Danish (dan)	English	45	72	8	31	8	37.73
38	Armenian (hye)	English	21	56	8	20	8	13.97
39	Mandarin (cmn)	English	200	481	9	67	8	17.0

idx	Source Language	Target Language	Parallel Sentences	Source vocab	Source subword	Target vocab	Target subword	BLEU
40	Indonesian (ind)	English	87	76	8	43	8	27.27
41	Galician (glg)	English	10	28	8	13	7	16.84
42	Portuguese (por)	English	185	165	8	64	8	41.67
43	Urdu (urd)	English	6	13	6	10	6	3.38
44	Italian (ita)	English	205	195	8	67	8	35.67
45	Spanish (spa)	English	196	179	8	66	8	39.48
46	Greek (ell)	English	134	171	8	54	8	34.94
47	Bengali (ben)	English	5	18	8	9	6	2.79
48	Japanese (jpn)	English	204	584	9	67	8	11.42
49	Malay (msa)	English	5	13	7	9	6	3.68
50	Dutch (nld)	English	184	172	8	63	8	34.27
51	Croatian (hrv)	English	122	191	8	52	8	31.84
52	Hebrew (heb)	English	212	276	8	68	8	33.89
53	Mongolian (mon)	English	8	21	8	11	6	2.96
54	Hindi (hin)	English	19	31	8	19	7	14.25

Answer Key: eus: 3.37, slk: 25.36, mya: 3.93, kor: 16.23, lit: 13.75, ara: 28.38, ces: 25.07, epo: 3.28, fin: 13.79, sqi: 29.6, vie: 24.67.

B Representative datasets

In this section, we show the searching results of most/least representative subsets for the rest of the five tasks.

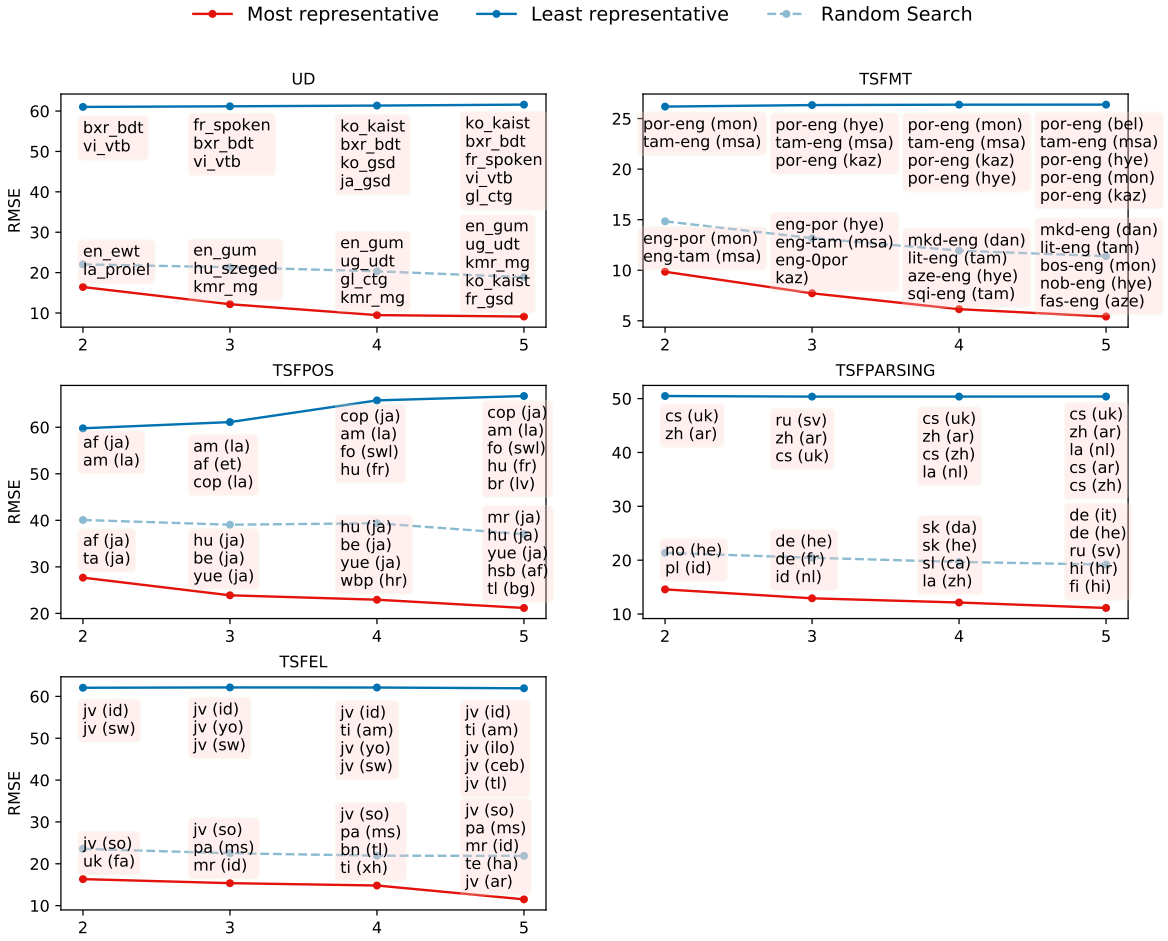


Figure 4: Beam search results (beam size=100) for up to the 5 most (and least) representative datasets for the remaining NLP tasks. We also show random search results of corresponding sizes.

C New Model

In this section, we show the extrapolation performance for new models on BLI, MA and the remaining systems of UD.

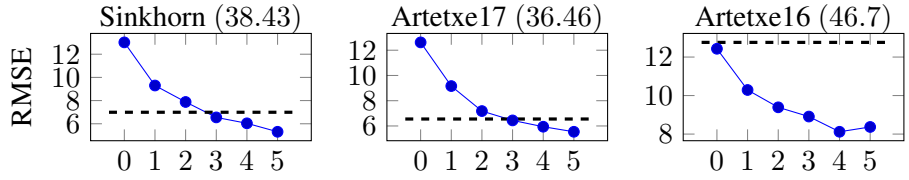


Figure 5: RMSE scores of **BLI** task from dataset-wise mean value predictor (the dashed black line in each graph) and predictors trained with experimental records of other models and 0–5 records from a new model (as indicated by the title of each graph).

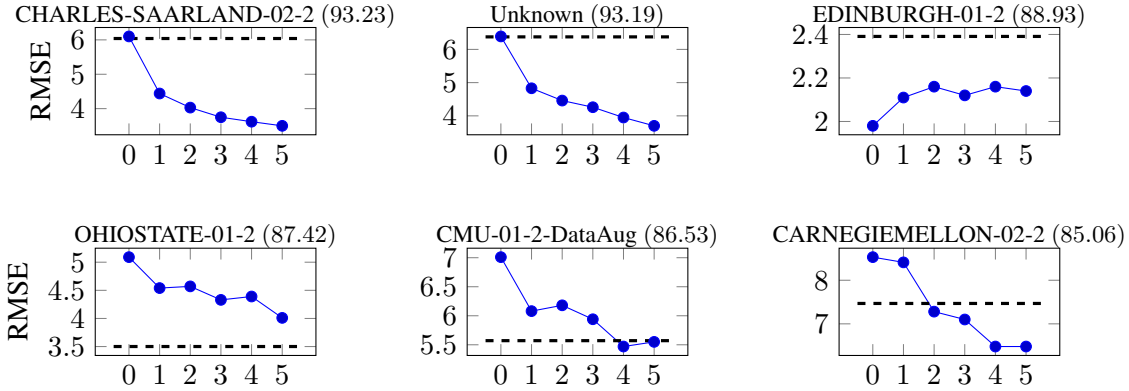


Figure 6: RMSE scores of **MA** task from dataset-wise mean value predictor (the dashed black line in each graph) and predictors trained with experimental records of other models and 0–5 records from a new model (as indicated by the title of each graph)

.

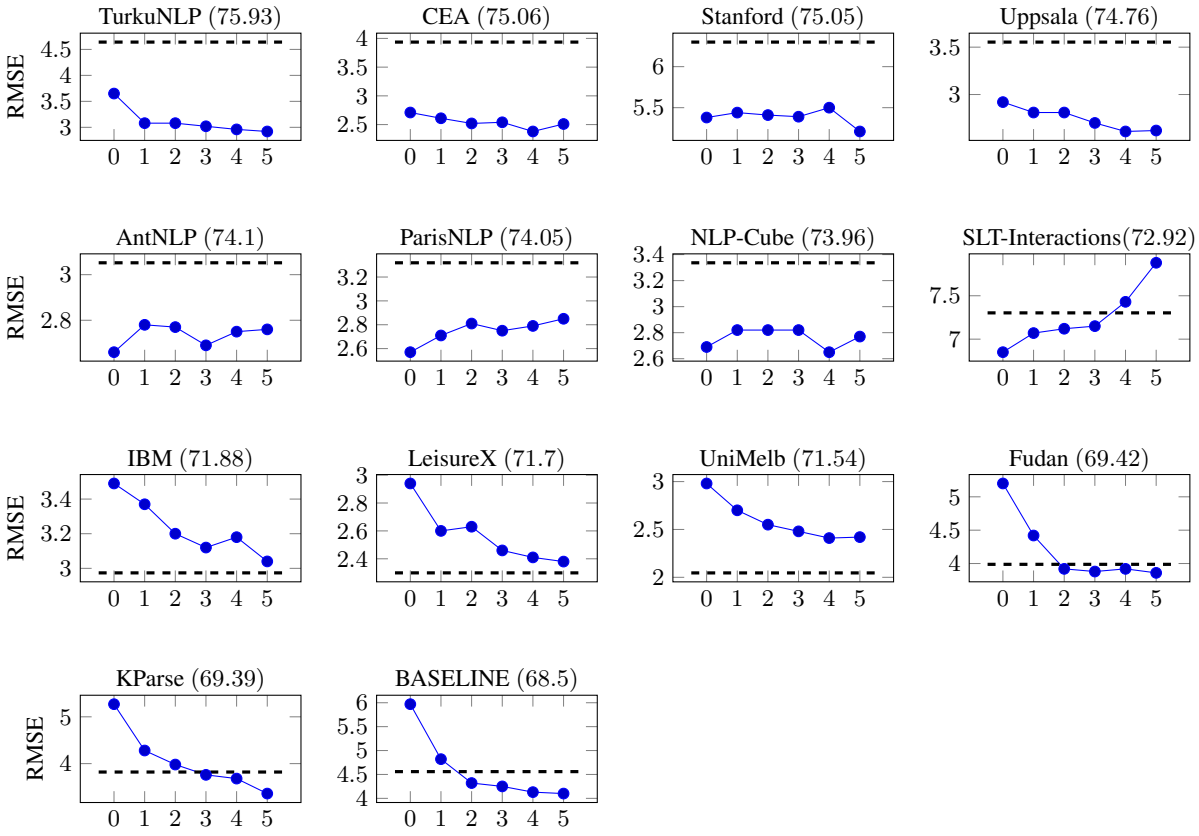
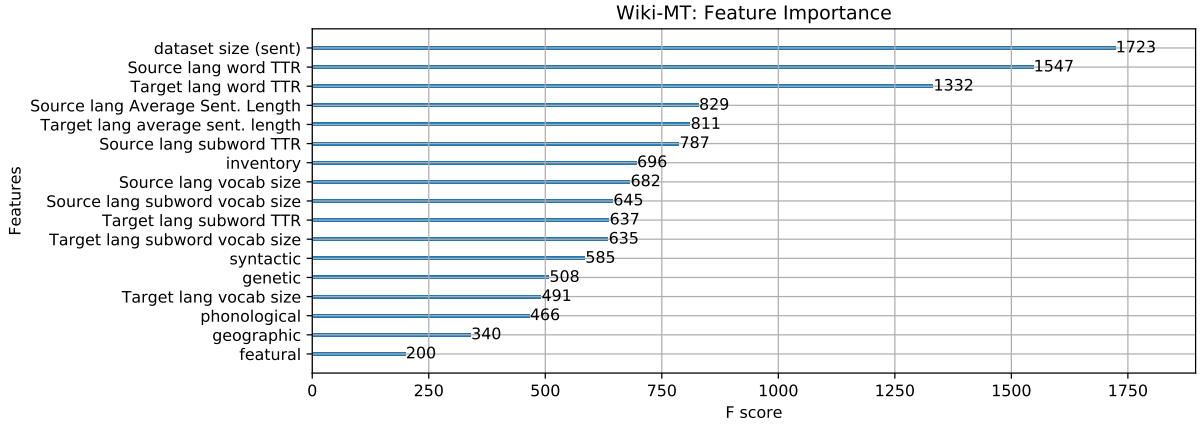
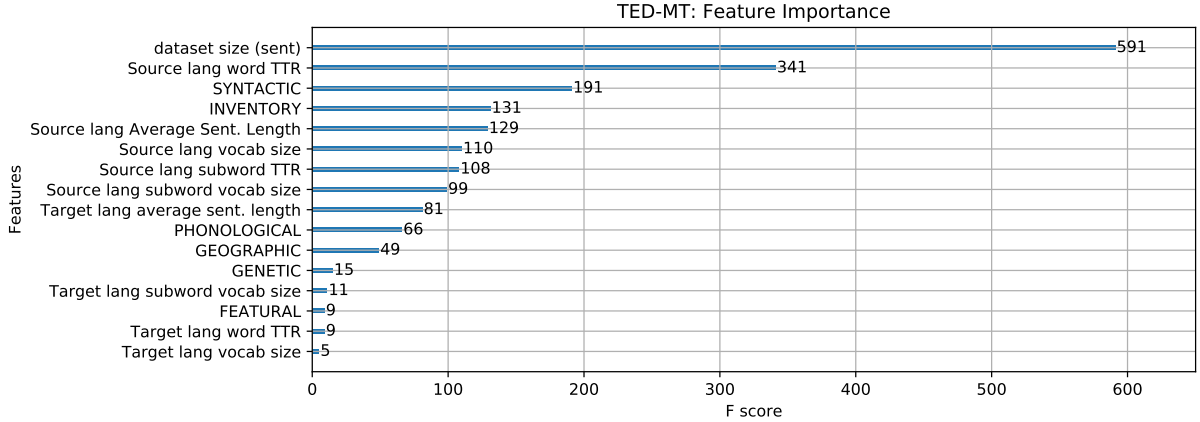


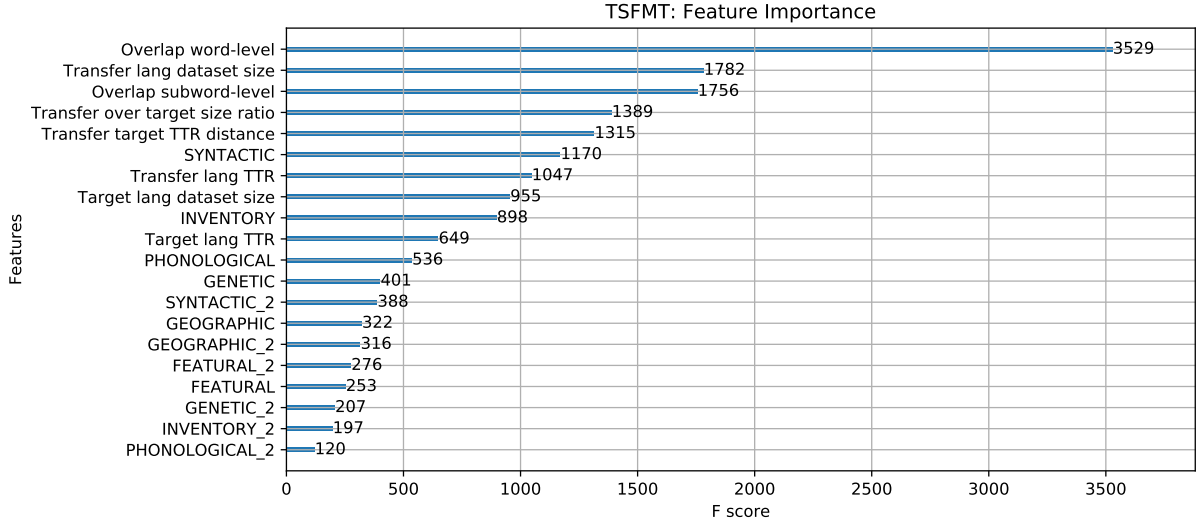
Figure 7: RMSE scores of **UD** task from dataset-wise mean value predictor (the dashed black line in each graph) and predictors trained with experimental records of other models and 0–5 records from a new model (as indicated by the title of each graph).

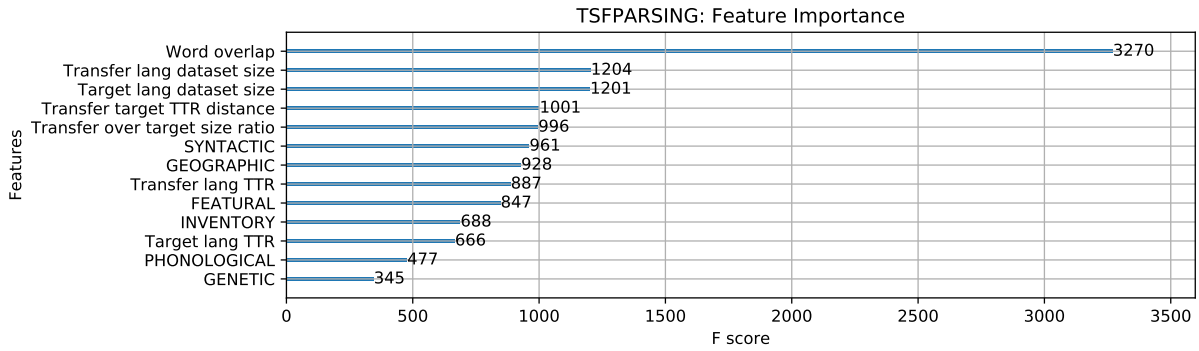
D Feature importance

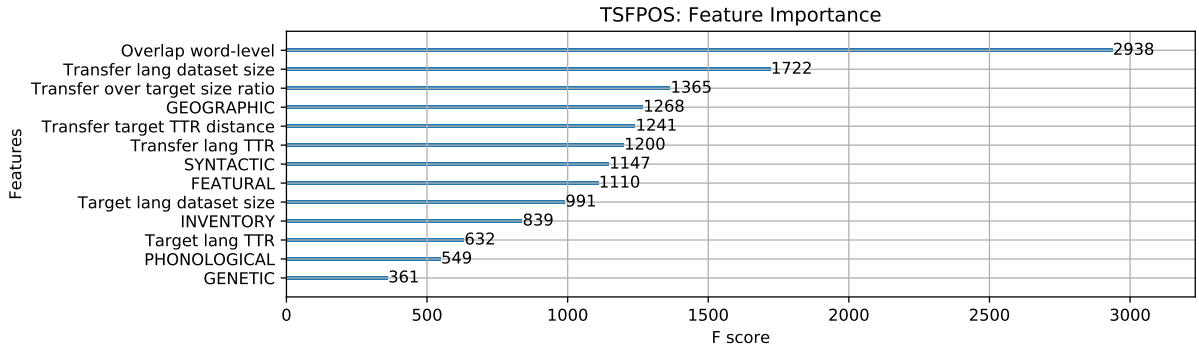
In this section, we show the plots of feature importance for all the tasks.

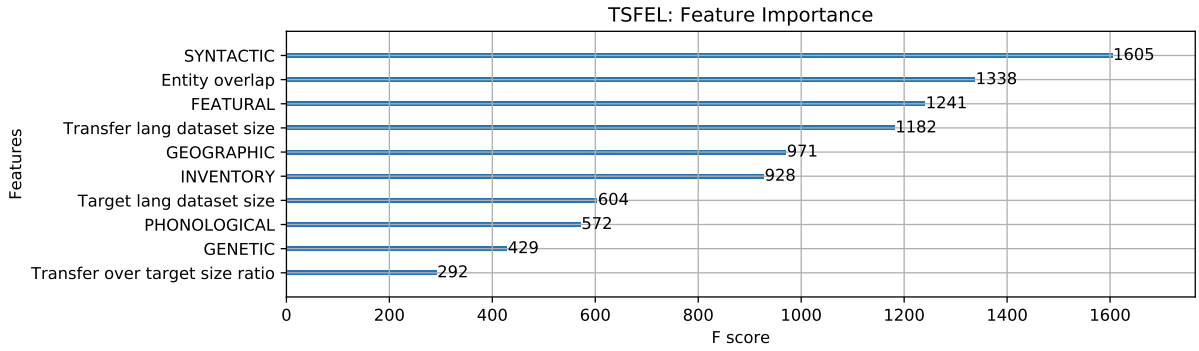


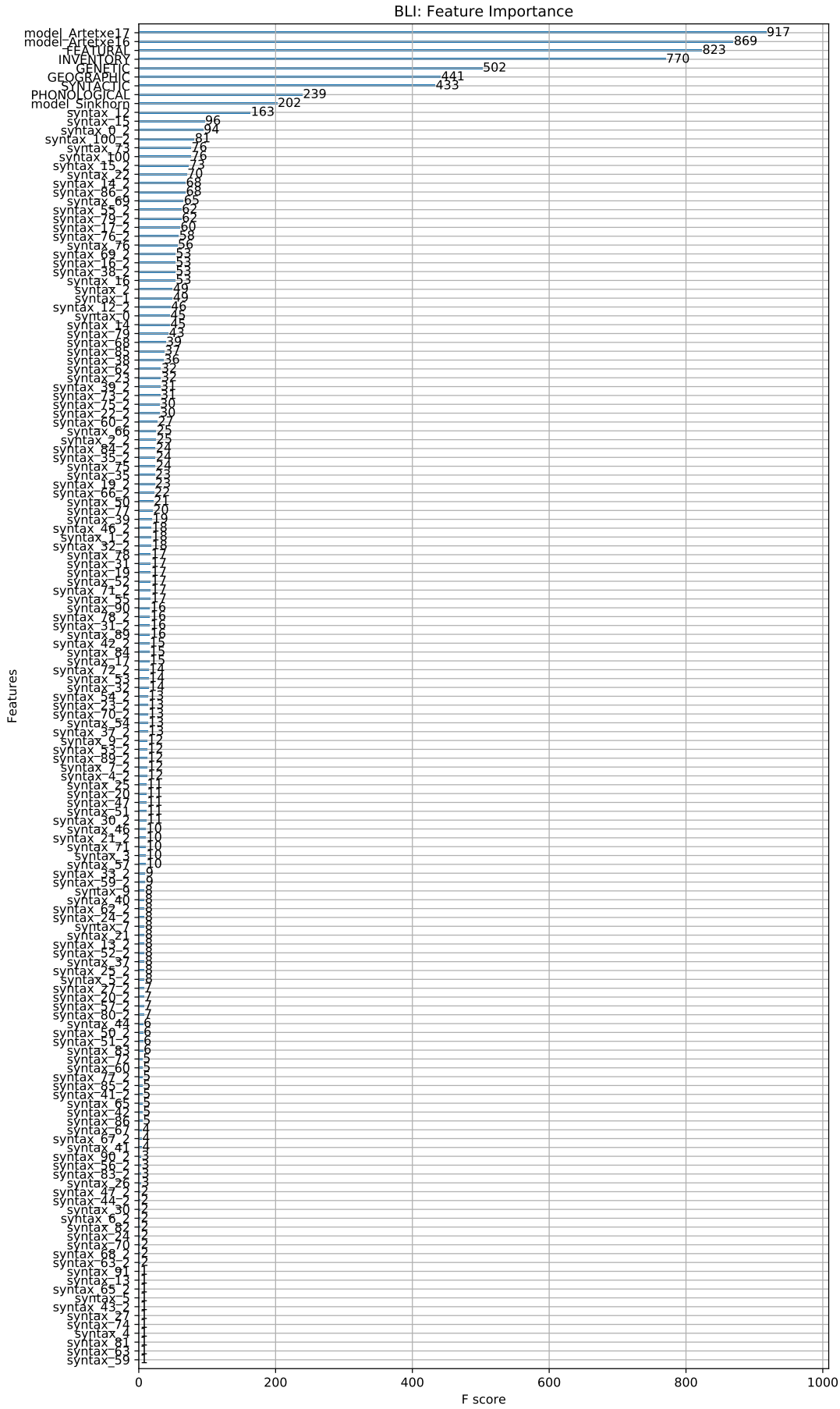


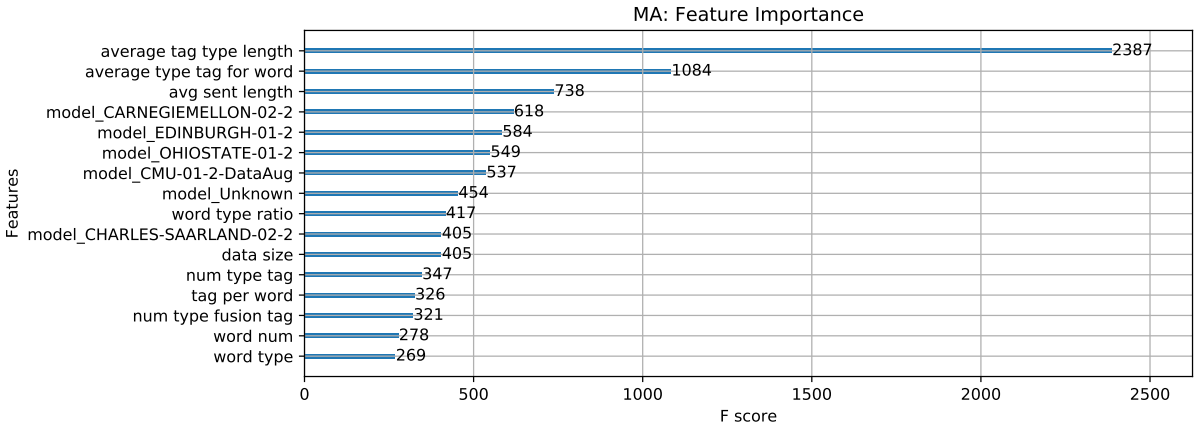


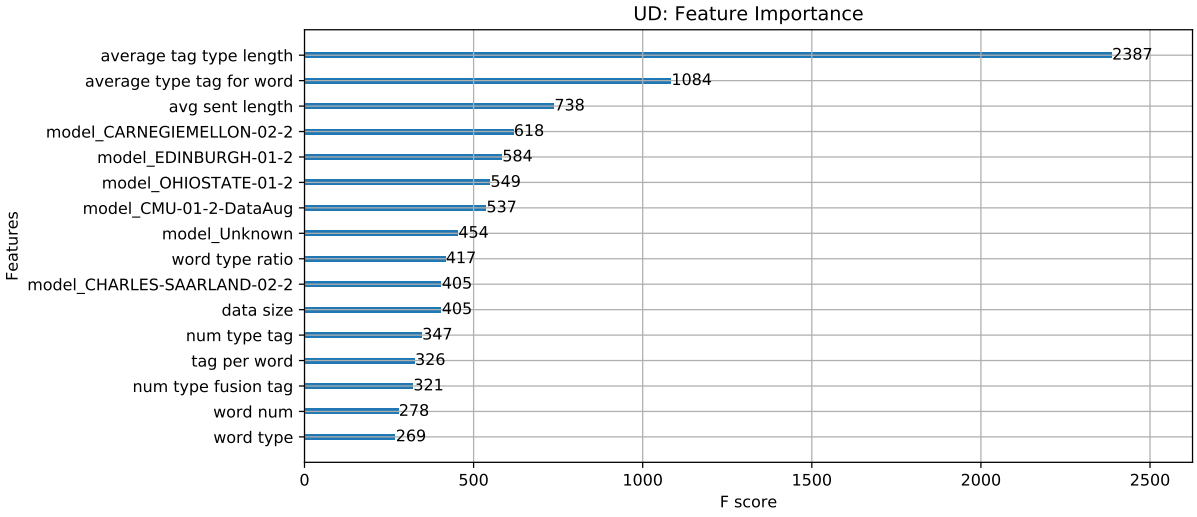

TSFPARSING: Feature Importance
Word overlap
Transfer lang dataset size
Target lang dataset size
Transfer target TTR distance
Transfer over target size ratio
SYNTACTIC
GEOGRAPHIC
Transfer lang TTR
FEATURAL
INVENTORY
Target lang TTR
PHONOLOGICAL
GENETIC
3270
1204
1201
1001
996
961
928
887
847
688
666
477
345
Features
0
500
1000
1500
2000
2500
3000
3500
F score


TSFPOS: Feature Importance
Overlap word-level
Transfer lang dataset size
Transfer over target size ratio
GEOGRAPHIC
Transfer target TTR distance
Transfer lang TTR
SYNTACTIC
FEATURAL
Target lang dataset size
INVENTORY
Target lang TTR
PHONOLOGICAL
GENETIC
2938
1722
1365
1268
1241
1200
1147
1110
991
839
632
549
361
Features
0
500
1000
1500
2000
2500
3000
F score


TSFEL: Feature Importance
SYNTACTIC
Entity overlap
FEATURAL
Transfer lang dataset size
GEOGRAPHIC
INVENTORY
Target lang dataset size
PHONOLOGICAL
GENETIC
Transfer over target size ratio
1605
1338
1241
1182
971
928
604
572
429
292
Features
0
200
400
600
800
1000
1200
1400
1600
F score


BLI: Feature Importance
Features
F score
model_Artetxe17
model_Artetxe16
FEATURAL
INVENTORY
GENETIC
GEOGRAPHIC
SYNTACTIC
PHONOLOGICAL
model_Sinkhorn
syntax_12
syntax_15
syntax_0
syntax_100_2
syntax_73
syntax_100
syntax_15_2
syntax_22
syntax_14_2
syntax_86_2
syntax_69
syntax_55_2
syntax_79_2
syntax_17_2
syntax_76_2
syntax_76
syntax_69_2
syntax_16_2
syntax_38_2
syntax_16
syntax_2
syntax_1
syntax_12_2
syntax_0_2
syntax_14
syntax_79
syntax_68
syntax_85
syntax_48
syntax_62
syntax_23
syntax_39_2
syntax_73_2
syntax_75_2
syntax_22_2
syntax_60_2
syntax_66
syntax_2_2
syntax_84_2
syntax_35_2
syntax_75
syntax_35
syntax_19_2
syntax_66_2
syntax_50
syntax_77
syntax_39
syntax_46_2
syntax_1_2
syntax_32_2
syntax_78
syntax_31
syntax_19
syntax_52
syntax_71_2
syntax_55
syntax_90
syntax_78_2
syntax_31_2
syntax_89
syntax_42_2
syntax_84
syntax_17
syntax_72_2
syntax_53
syntax_32
syntax_54_2
syntax_23_2
syntax_70_2
syntax_54
syntax_37_2
syntax_9_2
syntax_53_2
syntax_89_2
syntax_7_2
syntax_4_2
syntax_25
syntax_20
syntax_47
syntax_51
syntax_30_2
syntax_46
syntax_21_2
syntax_71
syntax_3
syntax_57
syntax_33
syntax_58_2
syntax_59_2
syntax_6
syntax_40
syntax_62_2
syntax_24_2
syntax_7
syntax_21
syntax_13_2
syntax_52_2
syntax_37
syntax_25_2
syntax_5_2
syntax_27_2
syntax_20_2
syntax_37_2
syntax_80_2
syntax_44
syntax_50_2
syntax_51_2
syntax_83
syntax_72
syntax_60
syntax_77_2
syntax_85_2
syntax_41_2
syntax_65
syntax_42
syntax_86
syntax_67
syntax_67_2
syntax_41
syntax_90_2
syntax_58_2
syntax_83_2
syntax_26
syntax_47_2
syntax_44_2
syntax_30
syntax_6_2
syntax_82
syntax_24
syntax_70
syntax_68_2
syntax_63_2
syntax_91
syntax_13
syntax_65_2
syntax_5
syntax_43_2
syntax_27
syntax_74
syntax_4
syntax_81
syntax_63
syntax_59
917
869
823
770
502
441
433
239
202
163
96
94
81
76
76
73
70
68
68
65
62
62
60
58
56
53
53
53
53
49
49
46
45
45
43
39
37
36
32
32
31
31
30
30
30
27
25
25
24
24
23
23
22
21
20
19
18
18
18
17
17
17
17
17
17
16
16
16
16
15
15
15
14
14
14
13
13
13
13
12
12
12
12
12
11
11
11
11
11
10
10
10
10
9
9
8
8
8
8
8
8
8
8
8
8
8
8
8
7
7
7
7
6
6
6
6
5
5
5
5
5
5
5
4
4
4
3
3
3
3
2
2
2
2
2
2
2
2
2
2
2
1
1
1
1
1
1
1
1
1
1
1
1
1
0
200
400
600
800
1000


MA: Feature Importance
average tag type length
average type tag for word
avg sent length
model_CARNEGIEMELLON-02-2
model_EDINBURGH-01-2
model_OHIOSTATE-01-2
model_CMU-01-2-DataAug
model_Unknown
word type ratio
model_CHARLES-SAARLAND-02-2
data size
num type tag
tag per word
num type fusion tag
word num
word type
2387
1084
738
618
584
549
537
454
417
405
405
347
326
321
278
269
Features
0
500
1000
1500
2000
2500
F score


UD: Feature Importance
average tag type length
average type tag for word
avg sent length
model_CARNEGIEMELLON-02-2
model_EDINBURGH-01-2
model_OHIOSTATE-01-2
model_CMU-01-2-DataAug
model_Unknown
word type ratio
model_CHARLES-SAARLAND-02-2
data size
num type tag
tag per word
num type fusion tag
word num
word type
2387
1084
738
618
584
549
537
454
417
405
405
347
326
321
278
269
Features
0
500
1000
1500
2000
2500
F score