Zero-Shot Dependency Parsing with Worst-Case Aware Automated Curriculum Learning

Anonymous ACL submission

Abstract

Large multilingual pretrained language models such as mBERT and XLM-RoBERTa have been found to be surprisingly effective for cross-lingual transfer of syntactic parsing models (Wu and Dredze, 2019), but only between related languages. However, source and training languages are rarely related, when parsing truly low-resource languages. To close this gap, we adopt a method from multi-task learning, which relies on automated curriculum learning, to dynamically optimize for parsing performance on *outlier* languages. We show that this approach is significantly better than uniform and size-proportional sampling in the zero-shot setting.

1 Introduction

The field of multilingual NLP is booming (Agirre, 2020). This is due in no small part to large multilingual pretrained language models (PLMs) such as mBERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020), which have been found to have surprising cross-lingual transfer capabilities in spite of receiving no cross-lingual supervision.[1] Wu and Dredze (2019), for example, found mBERT to perform well in a zero-shot setting when fine-tuned for five different NLP tasks in different languages. There is, however, a sharp divide between languages that benefit from this transfer and languages that do not, and there is ample evidence that transfer works best between typologically similar languages (Pires et al., 2019). This means that the majority of world languages that are *truly low-resource* are still left behind and inequalities in access to language technology are increasing.

Large multilingual PLMs are typically fine-tuned using training data from a sample of languages that is supposed to be representative of the languages that the models are later applied to. However, this is difficult to achieve in practice, as multilingual datasets are not well balanced for typological diversity and contain a skewed distribution of typological features (Ponti et al., 2021). This problem can be mitigated by using methods that sample from skewed distributions in a way that is robust to outliers.

Zhang et al. (2020) recently developed such a method. It uses curriculum learning with a worst-case-aware loss for multi-task learning. They trained their model on a subset of the GLUE benchmark (Wang et al., 2018) and tested on outlier tasks. This led to improved zero-shot performance on these outlier tasks. This method can be applied to multilingual NLP where different languages are considered different tasks. This is what we do in this work, for the case of multilingual dependency parsing. Multilingual dependency parsing is an ideal test case for this method, as the Universal Dependency treebanks (Nivre et al., 2020) are currently the manually annotated dataset that covers the most typological diversity (Ponti et al., 2021). Our research question can be formulated as such: *Can worst-case aware automated curriculum learning improve zero-shot dependency parsing?*

2 Worst-Case-Aware Curriculum Learning

In multi-task learning, the total loss is generally the average of losses of different tasks:

$$\min_{\theta} \ell(\theta) = \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta) \qquad (1)$$

[1] In the early days, cross-lingual transfer for dependency parsing relied on projection across word alignments (Spreyer and Kuhn, 2009; Agić et al., 2016) or *delexicalized transfer* of abstract syntactic features (Zeman and Resnik, 2008; McDonald et al., 2011; Søgaard, 2011; Cohen et al., 2011). Delexicalized transfer was later 're-lexicalized' by word clusters (Täckström et al., 2012) and word embeddings (Duong et al., 2015), but with the introduction of multilingual contextualized language models, transfer models no longer rely on abstract syntactic features, removing an important bottleneck for transfer approaches to scale to truly low-resource languages.

where l_i is the loss of task i. The architecture we use in this paper is adapted from Zhang et al. (2020), which is an automated curriculum learning (Graves et al., 2017) framework to learn a worst-case-aware loss in a multi-task learning scenario. The architecture consists of a sampler, a buffer, a trainer and a multilingual dependency parsing model. The two main components are the sampler, which adopts a curriculum sampling strategy to dynamically sample data batches, and the trainer which uses worst-case-aware strategy to train the model. The framework repeats the following steps: (1) the sampler samples data batches of different languages to the buffer; (2) the trainer uses a worst-case strategy to train the model; (3) the automated curriculum learning strategy of the sampler is updated.

Sampling data batches We view multilingual dependency parsing as multi-task learning where parsing in each individual language is considered a task. This means that the target of the sampler at each step is to choose a data batch from one language. This is a typical multi-arm bandit problem (Even-Dar et al., 2002). The sampler should choose bandits that have higher rewards, and in our scenario, data batches that have a higher loss on the model are more likely to be selected by the sampler and therefore, in a later stage, used by the trainer. Automated curriculum learning is adopted to push a batch with its loss into the buffer at each time step. The buffer consists of n first-in-first-out queues, and each queue corresponds to a task (in our case, a language). The procedure repeats k times and, at each round, k data batches are pushed into the buffer.

Worst-case-aware risk minimization In multilingual and multi-task learning scenarios, in which we jointly minimize our risk across n languages or tasks, we are confronted with the question of how to summarize n losses. In other words, the question is how to compare two loss vectors α and β containing losses for all tasks $l_i, \dots l_n$:

$$\alpha = [\ell_1^1, \dots, \ell_n^1]$$

and

$$\beta = [\ell_1^2, \dots, \ell_n^2]$$

The most obvious thing to do is to minimize the mean of the n losses, asking whether $\sum_{\ell \in \alpha} \ell < \sum_{\ell \in \beta} \ell$. We could also, motivated by robustness (Søgaard, 2013) and fairness (Williamson and Menon, 2019), minimize the maximum (supremum) of the n losses, asking whether $\max_{\ell \in \alpha} \ell < \max_{\ell \in \beta} \ell$. Mehta et al. (2012) observed that these two loss summarizations are extremes that can be generalized by a family of multi-task loss functions that summarize the loss of n tasks as the L^p norm of the n-dimensional loss vector. Minimizing the average loss then corresponds to computing the L^1 norm, i.e., asking whether $|\alpha|^1 < |\beta|^1$, and minimizing the worst-case loss corresponds to computing the L^∞ (supremum) norm, i.e., asking whether $|\alpha|^\infty < |\beta|^\infty$.

Zhang et al. (2020) present a stochastic generalization of the L^∞ loss summarization and a practical approach to minimizing this family of losses through automated curriculum learning (Graves et al., 2017): The core idea behind their generalization is to optimize the worst-case loss with a certain probability, otherwise optimize the average (loss-proportional) loss with the remaining probability. The hyperparameter ϕ is introduced by the worst-case-aware risk minimization to trade off the balance between the worst-case and the loss-proportional losses. The loss family is formally defined as:

$$\min \ell(\theta) = \begin{cases} \min\ \max_i(\ell_i(\theta)), & p < \phi \\ \min \ell_{\tilde{i}}(\theta), & p \geq \phi, \tilde{i} \sim P_\ell \end{cases} \tag{2}$$

where $p \in [0, 1]$ is a random generated rational number, and $P_\ell = \frac{\ell_i}{\sum_{j \leq n} \ell_j}$ is the normalized probability distribution of task losses. If $p < \phi$ the model choose the maximum loss among all tasks, otherwise, it randomly chooses one loss according to the loss distribution. If the hyperparameter ϕ equals 1, the trainer updates the model with respect to the worst-case loss. On the contrary, if $\phi = 0$, the trainer loss-proportionally samples one loss.

Sampling strategy updates The model updates its parameters with respect to the loss chosen by the trainer. After that, the sampler updates its policy according to the behavior of the trainer. At each round, the policy of the task that is selected by the trainer receives positive rewards and the policy of all other tasks that have been selected by the sampler receive negative rewards.

The multilingual dependency parsing model We use a standard biaffine graph-based dependency parser (Dozat and Manning, 2017). The model takes token representations of words from a contextualized language model (mBERT or XLM-R)

Language	Treebank	Genus	Lang. family
Arabic	PADT	Semitic	Afro-Asiatic
Basque	BDT	Basque	Basque
Chinese	GSD	Chinese	Sino-Tibetan
English	EWT	Germanic	IE
Finnish	TDT	Finnic	Uralic
Hebrew	HTB	Semitic	Afro-Asiatic
Hindi	HDTB	Indic	IE
Italian	ISDT	Romance	IE
Japanese	GSD	Japanese	Japanese
Korean	GSD	Korean	Korean
Russian	SynTagRus	Slavic	IE
Swedish	Talbanken	Germanic	IE
Turkish	IMST	Turkic	Altaic

Table 1: 13 training languages

as input and classifies head and dependency relations between words in the sentence. The Chu-Liu-Edmonds algorithm (Chu and Liu, 1965; Edmonds, 1967) is then used to decode the score matrix into a tree. All languages share the same encoder and decoder in order to learn features from different languages, and more importantly to perform zero-shot transfer to unseen languages.

3 Experiments

We base our experimental design on Üstün et al. (2020), a recent paper doing zero-shot dependency parsing with good performance on a large number of languages. They fine-tune mBERT for dependency parsing using training data from a sample of 13 typologically diverse languages from Universal Dependencies (UD; Nivre et al., 2020), listed in Table 1. For testing, they use 30 test sets from treebanks whose language has not been seen at fine-tuning time. We use the same training and test sets and experiment both with mBERT and XLM-R as PLMs. It is important to note that not all of the test languages have been seen by the PLMs.

We test worst-case aware learning with different values of ϕ and compare this to three main baselines: *size-proportional* samples batches proportionally to the data sizes of the training treebanks, *uniform* samples from different treebanks with equal probability, thereby effectively reducing the size of the training data, and *smooth-sampling* uses the smooth sampling method developed in van der Goot et al. (2021) which samples from multiple languages using a multinomial distribution. These baselines are competitive with the state-of-the-art when using mBERT, they are within 0.2 to 0.4 LAS points from the baseline of Üstün et al. (2020) on the same test sets. When using XLM-R, they are largely above the state-of-the-art.

		mBERT	XLM-R
OURS	ϕ=0	**36.4**	42.1
	ϕ=0.5	36.1	**42.3**
	ϕ=1	36.1	**42.3**
BASELINES	size-proportional	35.0	41.9
	smooth-sampling	35.2	41.7
	uniform	35.2	41.4

Table 2: **Zero-shot performance:** Average LAS scores on the test sets of the 30 unseen (zero-shot) languages in the language split in Üstün et al. (2020).

We implement all models using MaChAmp (van der Goot et al., 2021), a library for multi-task learning based on AllenNLP (Gardner et al., 2018). The library uses transformers from HuggingFace (Wolf et al., 2020). We make our code publicly available.

Our main results are in Table 2 where we report average scores across test sets, for space reasons. Tables with results broken down by test treebank can be found in Appendix A. We can see that worst-case-aware training outperforms all of our baselines in the zero-shot setting, highlighting the effectiveness of this method. This answers positively our research question *Can worst-case aware automated curriculum learning improve zero-shot dependency parsing?*

Our results using mBERT are more than 1 LAS point above the corresponding baselines. Our best model with mBERT comes close to Udapter (36.5 LAS on the same test sets) while being a lot simpler and not using external resources such as typological features, which are not always available for truly low-resource languages.

The results with XLM-R are much higher in general[2] but the trends are similar: all our models outperform all of our baselines albeit with smaller differences (there is only a 0.4 LAS difference between our best model and the best baseline). This highlights the robustness of the XLM-R model itself. Our results with XLM-R outperform Udapter by close to 7 LAS points.

4 Varying the homogeneity of training samples

We investigate the interaction between the effectiveness of worst-case learning and the represen-

[2] Note, however, that the results are not directly comparable since different subsets of test languages have been seen by the two PLMs.

sample	BASE	OURS	δ	RER
13LANG	35.2	36.4	1.2	1.9
GERMANIC	30.7	31.4	0.7	1.0
SLAVIC	30.4	31.7	1.3	1.9
ROMANCE	31.3	32.5	1.2	1.7
ROM+EU	33.3	34.8	1.5	2.2
ROM+AR	32.0	32.2	0.2	0.3
ROM+TR	32.2	33.0	0.8	1.2
ROM+ZH	33.4	34.1	0.7	1.1

Table 3: LAS of best baseline (BASE) and best worst-case training (OURS). Absolute difference (δ) and relative error reduction (RER) between OURS and BASE.

tativeness of the sample of training languages. It is notoriously difficult to construct a sample of treebanks that is representative of the languages in UD (de Lhoneux et al., 2017; Schluter and Agić, 2017; de Lhoneux, 2019). We can, however, easily construct samples that are **not** representative, for example, by taking a sample of related languages. We expect worst-case aware learning to lead to larger improvements in cases where some language types are underrepresented in the sample. We can construct an extreme case of underrepresentation by selecting a sample of training languages that has one or more clear outliers. For example we can construct a sample of related languages, add a single unrelated language in the mix, and then evaluate on other unrelated languages. We also expect that with a typologically diverse set of training languages, worst-case aware learning should lead to larger relative improvements than with a homogeneous sample, but perhaps slightly smaller improvements than with a very skewed sample.

We test these hypotheses by constructing seven samples of training languages in addition to the one used so far (13LANG). We construct three different homogeneous samples using treebanks from three different genera: GERMANIC, ROMANCE and SLAVIC. We construct four skewed samples using the sample of romance languages and a language from a different language family, an *outlier* language: Basque (eu), Arabic (ar), Turkish (tr) and Chinese (zh). Since we keep the sample of test sets constant, we do not include training data from languages that are in the test sets. The details of which treebanks are used for each of these samples can be found in Table 5 in Appendix B.

We can see first that, as expected, our typologically diverse sample performs best overall. This indicates that it is a good sample. We can also see that, as expected, the method works best with a skewed sample: the largest gains from using worst-case learning, both in terms of absolute LAS difference and relative error reduction, are seen for a skewed sample (ROM+EU). However, contrary to expectations, the lowest gains are obtained for another skewed sample (ROM+AR). The gains are also low for ROM+TR, ROM+ZH and for GERMANIC. Additionally, there are slightly more gains from using worst-case aware learning with the SLAVIC sample than for our typologically diverse sample. These results could be due to the different scripts of the languages involved both in training and testing.

Looking at results of the different models on individual test languages (see Figure 1 in Appendix C), we find no clear pattern of the settings in which this method works best. We do note that the method always hurts Belarusian, which is perhaps unsurprising given that it is the test treebank for which the baseline is highest. Worst-case aware learning hurts Belarusian the least when using the SLAVIC sample, indicating that, when using the other samples, the languages related to Belarusian are likely downsampled in favour of languages unrelated to it. Worst-case learning consistently helps Breton and Swiss German, indicating that the method might work best for languages that are underrepresented within their language family but not necessarily outside of it. For Swiss German, worst-case learning helps least when using the GERMANIC sample where it is less of an outlier.

5 Conclusion

In this work, we have adopted a method from multitask learning which relies on automated curriculum learning to the case of multilingual dependency parsing. This method allows to dynamically optimize for parsing performance on *outlier* languages. We found this method to improve dependency parsing on a sample of 30 test languages in the zero-shot setting, compared to sampling data uniformly across treebanks from different languages, or proportionally to the size of the treebanks. We investigated the impact of varying the homogeneity of the sample of training treebanks on the usefulness of the method and found conflicting evidence with different samples. This leaves open questions about the relationship between the languages used for training and the ones used for testing.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Eneko Agirre. 2020. Cross-Lingual Word Embeddings. *Computational Linguistics*, 46(1):245–248.

Yoeng-Jin Chu and Tseng-hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 50–61, Edinburgh, Scotland, UK. Association for Computational Linguistics.

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

Miryam de Lhoneux. 2019. *Linguistically Informed Neural Dependency Parsing for Typologically Diverse Languages*. Ph.D. thesis, Uppsala University.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017. Old School vs. New School: Comparing Transition-Based Parsers with and without Neural Network Enhancement. In *Proceedings of the 15th Treebanks and Linguistic Theories Workshop (TLT)*, pages 99–110.

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

Timothy Dozat and Christopher Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of the 5th International Conference on Learning Representations.*

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Cross-lingual transfer for unsupervised dependency parsing without parallel data. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 113–122, Beijing, China. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum Branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2002. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Nishant Mehta, Dongryeol Lee, and Alexander G Gray. 2012. Minimax multi-task learning and a generalized loss-compositional paradigm for mtl. In *Advances in Neural Information Processing Systems*, pages 2150–2158.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

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.

Edoardo Maria Ponti, Rahul Aralikatte, Disha Shrivastava, Siva Reddy, and Anders Søgaard. 2021. Minimax and neyman–Pearson meta-learning for outlier languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1245–1260, Online. Association for Computational Linguistics.

Natalie Schluter and Željko Agić. 2017. Empirically sampling Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 117–122, Gothenburg, Sweden. Association for Computational Linguistics.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 682–686, Portland, Oregon, USA. Association for Computational Linguistics.

Anders Søgaard. 2013. Part-of-speech tagging with antagonistic adversaries. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 640–644, Sofia, Bulgaria. Association for Computational Linguistics.

Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 12–20, Boulder, Colorado. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Robert Williamson and Aditya Menon. 2019. Fairness risk measures. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6786–6797. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Sheng Zhang, Xin Zhang, Weiming Zhang, and Anders Søgaard. 2020. Worst-case-aware curriculum learning for zero and few shot transfer. *arXiv preprint arXiv:2009.11138*.

A Results by treebank

Results by language of the test treebanks are in Table 4.

B Training samples

The training samples are summarized in Table 5.

C Results by treebank with the different samples

Relative error reduction between our best worst-case aware result and the best baseline for each training sample used, with mBERT, in Figure 1.

	mBERT						XLM-R					
iso	ϕ=0	ϕ=0.5	ϕ=1	S-P	S-S	U	ϕ=0	ϕ=0.5	ϕ=1	S-P	S-S	U
aii	8	**11.3**	10.8	1.6	6.4	6.0	2	3.3	3.1	2.9	**3.5**	3.1
akk	1.5	1.4	1.6	2.5	**3.0**	1.9	2.5	2.5	**2.8**	1.9	2.2	2.3
am	**16.5**	10.9	13.2	6.6	10.8	10.6	68.0	68.6	68.3	68.4	**68.8**	68.1
be	78.5	79.4	79.6	**82.0**	80.9	80.5	85.6	85.5	85.6	86.4	86.8	**86.8**
bho	**38.1**	37.8	37.9	37.0	36.7	36.7	37.3	37.4	37.1	37.4	**37.6**	37.2
bm	**9.0**	8.7	8.7	6.9	6.7	6.9	6.0	6.4	6.2	**6.5**	6.3	6.4
br	**62.9**	62.6	62.0	60.3	60.3	59.6	59.5	59.6	**60.5**	59.9	59.5	58.9
bxr	25.9	**26.0**	25.6	24.6	25.5	25.4	27.7	**28.2**	28.0	27.2	27.2	26.2
cy	**55.5**	55.0	55.2	55.1	54.4	54.2	59.8	60.1	59.9	60.2	**60.6**	59.6
fo	67.4	67.8	**68.0**	66.3	67.2	66.4	73.5	72.8	**73.5**	72.6	72.4	73.0
gsw	48.3	**48.8**	48.2	44.9	42.2	42.3	46.0	**46.5**	**46.5**	43.6	42.2	44.3
gun	8.2	8.5	**8.7**	7.3	8.0	8.3	6.8	6.8	**7.6**	6.5	5.8	5.6
hsb	50.8	51.3	**51.4**	49.4	49.2	49.1	**62.6**	61.9	62.0	61.4	61.6	60.0
kk	**60.1**	58.9	58.4	58.5	59.0	58.2	63.0	62.7	62.5	**63.7**	62.3	61.5
kmr	9.3	9.2	8.9	8.6	**9.6**	9.5	**53.5**	53.1	53.2	51.8	51.7	52.0
koi	19.3	18.8	**19.8**	15.8	15.8	16.0	17.0	**20.1**	19.1	17.8	17.8	16.0
kpv	16.8	17.0	**17.2**	15.6	16.2	15.8	18.3	19.1	**19.5**	17.0	17.8	16.3
krl	46.6	46.4	46.3	46.5	**47.1**	46.4	61.0	61.2	60.7	62.0	**62.1**	61.8
mdf	**26.1**	24.3	24.3	22.5	24.5	25.4	20.4	**20.7**	19.6	18.4	18.4	16.8
mr	60.6	**61.2**	60.1	56.9	57.7	57.7	69.2	69.7	**70.0**	67.8	**70.0**	69.7
myv	**20.2**	19.9	19.8	18.5	19.3	19.9	16.8	**17.2**	16.9	16.0	16.3	15.5
olo	40.7	**41.7**	41.0	41.0	40.9	40.5	56.5	**56.7**	56.1	55.8	54.3	54.4
pcm	33.9	32.8	33.0	32.5	34.3	**35.4**	**39.2**	39.2	38.9	38.0	37.6	37.8
sa	**22.5**	21.9	22.3	21.1	21.0	20.6	50.2	49.7	**50.9**	**50.9**	50.1	50.0
ta	52.3	**54.7**	54.3	53.2	52.0	51.6	54.9	**55.0**	54.8	53.8	53.8	54.0
te	69.9	69.8	70.0	69.4	**70.6**	68.7	76.0	76.0	76.7	76.3	**77.1**	76.3
tl	65.4	57.5	56.5	**65.8**	59.3	65.4	77.1	75.7	75.7	**78.1**	76.7	76.4
wbp	5.9	8.8	**9.2**	7.5	7.5	7.2	7.8	**9.5**	7.5	8.5	5.2	8.8
yo	37.8	37.9	38.5	**39.7**	38.0	37.5	3.3	**3.6**	3.2	2.3	2.7	1.8
yue	**33.0**	32.5	32.5	32.4	32.4	32.4	41.9	41.7	42.0	**42.9**	42.4	42.8
average	**36.4**	36.1	36.1	35.0	35.2	35.2	42.1	**42.3**	**42.3**	41.9	41.7	41.4

Table 4: **Zero-shot performance:** LAS scores on the test sets of the 30 unseen (zero-shot) languages in the language split in Üstün et al. (2020) using mBERT and XLM-R. S-P=size-proportional, S-S = smooth-sampling, U=uniform. Bold indicates the best performance across models using the same PLM.

	GERMANIC	SLAVIC	ROMANCE	ROM+EU	ROM+AR	ROM+TR	ROM+ZH	13LANG
Afrikaans-AfriBooms	✓							
Danish-DDT	✓							
Dutch-Alpino	✓							
English-EWT	✓							✓
German-HDT	✓							
Gothic-PROIEL	✓							
Icelandic-IcePaHC	✓							
Norwegian-Bokmaal	✓							
Swedish-Talbanken	✓							✓
Czech-PDT		✓						
Old_Church_Slavonic-PROIEL		✓						
Old_Russian-TOROT		✓						
Polish-LFG		✓						
Russian-SynTagRus		✓						✓
Serbian-SET		✓						
Slovak-SNK		✓						
Ukrainian-IU		✓						
French-GSD			✓	✓	✓	✓	✓	
Italian-ISDT			✓	✓	✓	✓	✓	✓
Portuguese-GSD			✓	✓	✓	✓	✓	
Romanian-RRT			✓	✓	✓	✓	✓	
Spanish-AnCora			✓	✓	✓	✓	✓	
Basque-BDT				✓				✓
Arabic-PADT					✓			✓
Chinese-GSD							✓	✓
Turkish-IMST						✓		✓
Finnish-TDT								✓
Hebrew-HTB								✓
Hindi-HDTB								✓
Japanese-GSD								✓
Korean-GSD								✓

Table 5: Treebanks included in the different samples

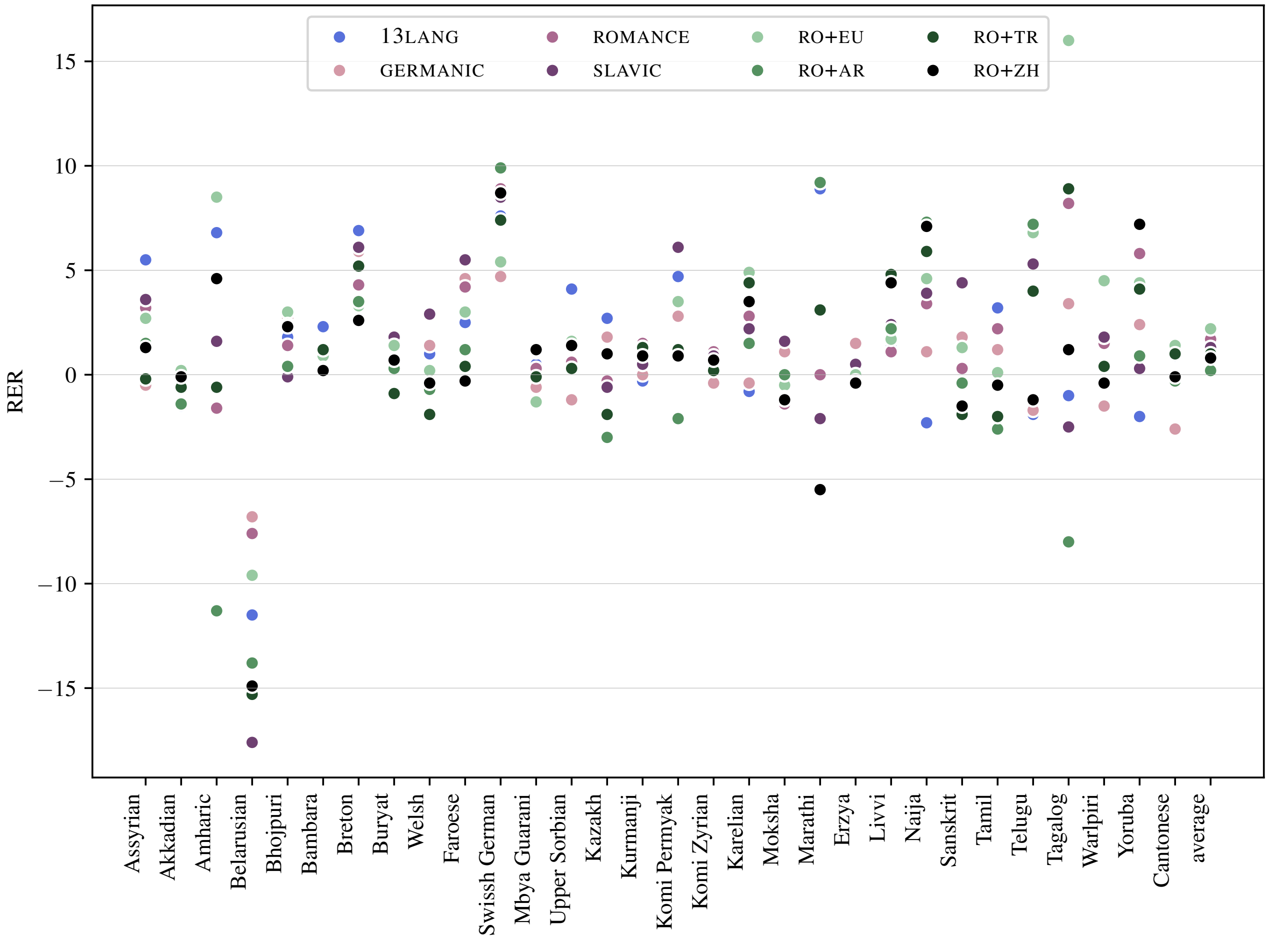

Figure 1: Relative error reduction (RER) in LAS points between our best worst-case aware result and the best baseline for each training sample used on test sets in the 30 languages.