Multilingual ELMo and the Effects of Corpus Sampling

Vinit Ravishankar, Andrey Kutuzov, Lilja Øvrelid, Erik Velldal
Language Technology Group
Department of Informatics
University of Oslo
{vinitr,andreku,liljao,erikve}@ifi.uio.no

Abstract

Multilingual pretrained language models are rapidly gaining popularity in NLP systems for non-English languages. Most of these models feature an important corpus sampling step in the process of accumulating training data in different languages, to ensure that the signal from better resourced languages does not drown out poorly resourced ones. In this study, we train multiple multilingual recurrent language models, based on the ELMo architecture, and analyse both the effect of varying corpus size ratios on downstream performance, as well as the performance difference between monolingual models for each language, and broader multilingual language models. As part of this effort, we also make these trained models available for public use.

1 Introduction

As part of the recent emphasis on language model pretraining, there also has been considerable focus on multilingual language model pretraining; this is distinguished from merely training language models in multiple languages by the creation of a multilingual space. These have proved to be very useful in 'zero-shot learning'; i.e., training on a well-resourced language (typically English), and relying on the encoder's multilingual space to create reasonable priors across languages.

The main motivation of this paper is to study the effect of corpus sampling strategy on downstream performance. Further, we also examine the utility of multilingual models (when constrained to monolingual tasks), over individual monolingual models, one per language. This paper therefore has two main contributions: the first of these is a multilingual ELMo model that we hope would see further use in probing studies as well as evaluative studies, downstream; we train these models over 13 languages, namely Arabic, Basque, Chinese, English, Finnish, Hebrew, Hindi, Italian, Japanese, Korean, Russian, Swedish and Turkish. The second contribution is an analysis of sampling mechanism on downstream performance; we elaborate on this later.

In Section 2 of this paper, we contextualise our work in the present literature. Section 3 describes our experimental setup and Section 4 our results. Finally, we conclude with a discussion of our results in Section 5.

2 Prior work

Multilingual embedding architectures (static or contextualised) are different from cross-lingual ones (Ruder et al., 2019; Liu et al., 2019) in that they are not products of aligning several monolingual models. Instead, a deep neural model is trained end to end on texts in multiple languages, thus making the whole process more straightforward and yielding truly multilingual representations (Pires et al., 2019). Following Artetxe et al. (2020), we will use the term 'deep multilingual pretraining' for such approaches.

One of the early examples of deep multilingual pretraining was BERT, which featured a multilingual variant trained on the 104 largest language-specific Wikipedias (Devlin et al., 2019). To counter the effects of some languages having overwhelmingly larger Wikipedias than others, Devlin et al. (2019) used exponentially smoothed data weighting; i.e., they exponentiated the probability of a token being in a certain language by a certain α, and re-normalised. This has the effect of 'squashing' the distribution of languages in their training data; larger languages become smaller, to avoid drowning out the signal from smaller languages. One can also look at this technique as a sort of sampling. Other multilingual models,

such as XLM (Lample and Conneau, 2019) and its larger variant, XLM-R (Conneau et al., 2020), use different values of α for this sampling (0.5 and 0.3 respectively). The current paper is aimed at analysing the effects of different α choices; in spirit, this work is very similar to Arivazhagan et al. (2019); where it differs is our analysis on downstream tasks, as opposed to machine translation, where models are trained and evaluated on a very specific task. We also position our work as a resource, and we make our multilingual ELMo models available for public use.

3 Experimental setup

3.1 Background

When taken to its logical extreme, sampling essentially reduces to truncation, where all languages have the same amount of data; thus, in theory, in a truncated model, no language ought to dominate any other. Of course, for much larger models, like the 104-language BERT, this is unfeasible, as the smallest languages are too small to create meaningful models. By selecting a set of languages such that the smallest language is still reasonably sized for the language model being trained, however, we hope to experimentally determine whether truncation leads to truly neutral, equally capable multilingual spaces; if not, we attempt to answer the question of whether compression helps at all.

Our encoder of choice for this analysis is an LSTM-based ELMo architecture introduced by Peters et al. (2018). This might strike some as a curious choice of model, given the (now) much wider use of transformer-based architectures. There are several factors that make ELMo more suitable for our analysis. Our main motivation was, of course, resources – ELMo is far cheaper to train, computationally. Next, while pre-trained ELMo models already exist for several languages (Che et al., 2018; Ulčar and Robnik-Šikonja, 2020), there is, to the best of our knowledge, no multilingual ELMo. The release of our multilingual model may therefore also prove to be useful in the domain of probing, encouraging research on multilingual encoders, constrained to recurrent encoders.

3.2 Sampling

Our initial starting point for collecting the language model training corpora were the CoNLL 2017 Wikipedia/Common Crawl dumps released as part of the shared task on Universal Dependencies parsing (Ginter et al., 2017); we extracted the Wikipedia portions of these corpora for our set of 13 languages. This gives us a set of fairly typologically distinct languages, that still are not entirely poorly resourced. The smallest language in this collection, Hindi, has $\sim$ 91M tokens, which we deemed sufficient to train a reasonable ELMo model.

Despite eliminating Common Crawl data, this gave us, for our set of languages, a total corpus size of approximately 35B tokens, which would be an unfeasible amount of data given computational constraints. We therefore selected a baseline model to be somewhat synthetic – note that this is a perfectly valid choice given our goals, which were to compare various sampling exponents. Our 'default' model, therefore, was trained on data that we obtained by weighting this 'real-world' Wikipedia data. The largest α we could use, that would still allow for feasible training, was $\alpha = 0.4$ (further on, we refer to this model as M0.4); this gave us a total corpus size of $\sim$4B tokens. Our second, relatively more compressed model, used $\alpha = 0.2$ (further on, M0.2); giving us a total corpus size of $\sim$2B tokens; for our final, most compressed model (further on, TRUNC), we merely truncated each corpus to the size of our smallest corpus (Hindi; 91M), giving us a corpus sized $\sim$1.2B tokens. Sampling was carried out as follows: if the probability of a token being sampled from a certain language i is p_i, the adjusted probability is given by $q_i = \frac{p_i}{\sum_{j=1}^{N} p_j}$. Note that this is a similar sampling strategy to the one followed by more popular models, like mBERT. We trained an out-of-the box ELMo encoder for approximately the same number of steps on each corpus; this was equivalent to 2 epochs for M0.4 and 3 for M0.2.

Detailed training hyperparameters and precise corpus sizes are presented in Appendices A and B.

3.3 Tasks

While there is a dizzying array of downstream evaluation tasks for monolingual models, looking to evaluate multilingual models is a bit harder. We settled on a range of tasks in two different groups:

1. **Monolingual tasks**: these tasks directly test the monolingual capabilities of the model, per language. We include PoS tagging and

dependency parsing in this category. In addition to our multilingual models, we also evaluate our monolingual ELMo variants on these tasks.

2. **Transfer tasks**: these tasks involve leveraging the model's multilingual space, to transfer knowledge from the language it was trained on, to the language it is being evaluated on. These tasks include natural language inference and text retrieval; we also convert PoS tagging into a transfer task, by training our model on English and asking it to tag text in other languages.

In an attempt to illuminate precisely what the contribution of the different ELMo models is, we ensure that our decoder architectures – that translate from ELMo's representations to the task's label space – are kept relatively simple, particularly for lower-level tasks. We freeze ELMo's parameters: this is not a study on fine-tuning.

The tasks that we select are a subset of the tasks mentioned in XTREME (Hu et al., 2020); i.e., the subset most suitable to the languages we trained our encoder on. A brief description follows:

PoS tagging: For part-of-speech tagging, we use Universal Dependencies part-of-speech tagged corpora (Nivre et al., 2020). Built on top of our ELMo-encoder is a simple MLP, that maps representations onto the PoS label space.

PoS tagging (transfer): We use the same architecture as for regular PoS tagging, but train on English and evaluate on our target languages.

Dependency parsing: We use dependency-annotated Universal Dependencies corpora; our metrics are both unlabelled and labelled attachment scores (UAS/LAS). Our parsing architecture is a biaffine graph-based parser (Dozat and Manning, 2018).

XNLI: A transfer-based language inference task; we use Chen et al.'s 2017 ESIM architecture, train a tagging head on English, and evaluate on the translated dev portions of other languages (Conneau et al., 2018).

Tatoeba: The task here is to pick out, for each sentence in our source corpus (English), the appropriate translation of the sentence in our target language corpus. This, in a sense, is the most 'raw' tasks; target language sentences are ranked based on similarity. We follow Hu et al. (2020) and use the Tatoeba dataset.

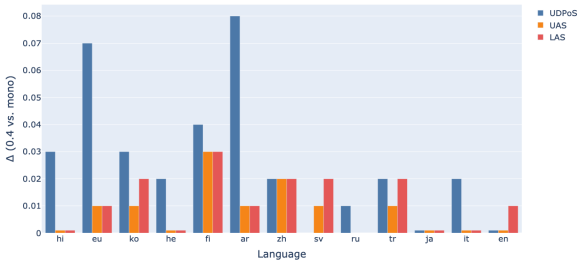


Figure 1: Performance difference between monolingual and multilingual models, on our monolingual tasks. Absent bars indicate that the language was missing.

We tokenize all our text using the relevant UDPipe (Straka et al., 2019) model, and train/evaluate on each task three times; the scores we report are mean scores.

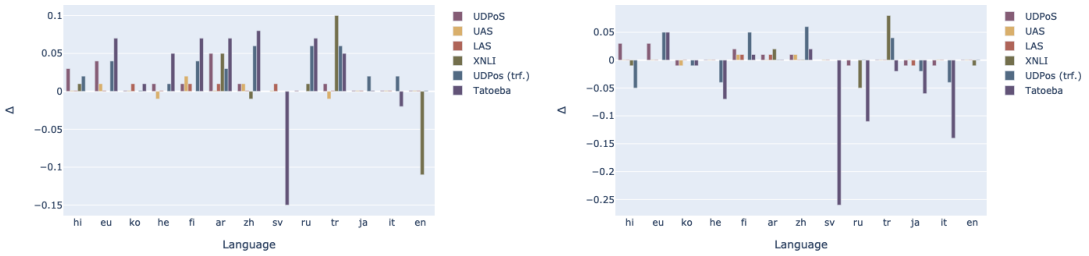
4 Results

First, we examine the costs of multilingualism, as far as monolingual tasks are concerned. We present our results on our monolingual tasks in Figure 1. Monolingual models appear to perform consistently better, particularly PoS tagging; this appears to be especially true for our under-resourced languages, strengthening the claim that compression is necessary to avoid drowning out signal. For PoS tagging, the correlation between performance difference (monolingual vs. M0.4) and corpus size is highly significant ($\rho = 0.74; p = 0.006$).

	PoS	UAS	LAS	PoS (trf.)	XNLI	Tatoeba
MONO	0.86	0.86	0.81	-	-	-
M0.4	0.83	0.85	0.80	0.36	0.45	0.18
M0.2	**0.84**	0.85	0.80	**0.39**	**0.46**	**0.21**
TRUNC	0.83	0.85	0.80	0.36	0.45	0.13

Table 1: Average scores for each task and encoder; non-monolingual best scores in bold.

We find that compression appears to result in visible improvements, when moving from $\alpha = 0.4$ to $\alpha = 0.2$. These improvements, while not dramatic, apply across the board (see Table 1), over virtually all task/language combinations; this is visible in Figure 2a. Note the drop in performance on certain tasks for English, Swedish and Italian –



(a) M0.2 vs. M0.4

(b) TRUNC vs. M0.4

Figure 2: Performance differences between our models on our selected tasks.

we hypothesise that this is due to Swedish and Italian being closer to English (our most-sampled language), and therefore suffering from the combination of the drop in their corpus sizes, as well as the more significant drop in English corpus size. The Pearson correlation between the trend in performance for PoS tagging and the size of a language's corpus is statistically significant ($\rho = 0.65; p = 0.02$); note that while this is over multiple points, it is single runs per data point.

Figure 2b also shows the difference in performance between the truncated model, TRUNC, and M0.4; this is a lot less convincing than the difference to M0.2, indicating that no additional advantage is to be gained by downsampling data for better-resourced languages.

We include full, detailed results in Appendix C.

Cross-lingual differences Finally, in an attempt to study the differences in model performance across languages, we examine the results of all models on Tatoeba. This task has numerous advantages for a more detailed analysis; i) it covers all our languages, bar Hindi, ii) the results have significant variance across languages, and iii) the task does not involve any additional training. We present these results in Figure 3.

We observe that M0.2 consistently appears to perform better, as illustrated earlier. Performance does not appear to have much correlation with corpus size; however, the languages for which M0.4 performs better are Swedish and Italian, coincidentally, the only other Latin-scripted Indo-European languages. Given the specific nature of Tatoeba, which involves picking out appropriate translations, these results make more sense: these languages receive not only the advantage of having more data for themselves, but also from the additional data available to English, which in turn optimises their biases solely by virtue of language similarity.

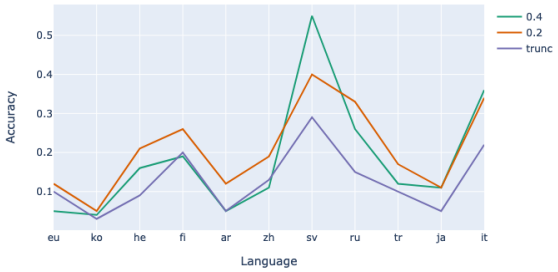


Figure 3: Accuracy on Tatoeba per model

5 Discussion

Our results allow us to draw conclusions that come across as very 'safe': some compression helps, too much hurts; when compression does help, however, the margin appears rather moderate yet significant for most tasks, even given fewer training cycles. Immediately visible differences along linguistic lines do not emerge when ratios differ, despite the relative linguistic diversity of our language choices; we defer analysis of this to a future work, that is less focused on downstream analysis, and more on carefully designed probes that might illuminate the difference between our models' internal spaces. Note that a possible confounding factor in our results is also the complexity of the architectures we build on top of mELMO: they also have significant learning capacity, and it is not implausible that whatever differences there are between our models, are drowned out by highly parameterised downstream decoders.

To reiterate, this study is not (nor does it aim to be) a replication of models with far larger parameter spaces and more training data. This is something of a middle-of-the-road approach; future work could involve this sort of evaluation on downscaled transformer models, which we shy away from in order to provide a usable model release. We hope that the differences between these models provide some insight, and pave the way for further research, not only specifically addressing the question of sampling from a perspective of performance, but also analytically. There has already been considerable work in this direction on multilingual variants of BERT (Pires et al., 2019; Chi et al., 2020), and we hope that this work motivates papers applying the same to recurrent mELMo, as well as comparing and contrasting the two. The ELMo models described in this paper are publicly released via NLPL Vector Repository.[1]

Acknowledgements

Our experiments were run on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway, under the NeIC-NLPL umbrella.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. *arXiv:1907.05019 [cs]*. ArXiv: 1907.05019.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online. Association for Computational Linguistics.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. ArXiv: 1609.06038.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. 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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

[1] `http://vectors.nlpl.eu/repository/`

Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2019. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 33–43, Hong Kong, China. Association for Computational Linguistics.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Milan Straka, Jana Straková, and Jan Hajic. 2019. UDPipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, Florence, Italy. Association for Computational Linguistics.

Matej Ulčar and Marko Robnik-Šikonja. 2020. High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4731–4738, Marseille, France. European Language Resources Association.

A Hyperparameters

Param	**Value**
Layers	2
Output dimensionality	2048
Batch size	192
Negative samples per batch	4096
Vocabulary size	100,000
Number of epochs	2 (M0.4); 3(M0.2)

Table 2: Models were bidirectional LSTMs. Monolingual models were trained on individual sizes given at $\alpha = 0.4$.

B Corpus sizes

C Detailed results

Language	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH	Total
M0.4	242.29	585.52	113.42	239.57	208.46	91.74	468.45	460.53	184.63	379.9	366.86	396.01	282.76	4020.14
M0.2	149.09	231.76	102.01	148.25	138.29	91.74	207.3	205.54	130.15	186.68	183.45	190.6	161.06	2125.92
TRUNC	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	1192.62

Table 3: Corpus sizes, in million tokens

Language		AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
POS	MONO	0.89	0.89	0.88	0.82	0.84	0.9	0.91	0.94	0.67	0.88	-	0.83	0.86
	0.4	0.81	0.89	0.81	0.78	0.82	0.87	0.89	0.94	0.64	0.87	-	0.81	0.84
	0.2	0.86	0.89	0.85	0.79	0.83	0.9	0.89	0.94	0.64	0.87	-	0.82	0.85
	TRUNC	0.82	0.89	0.84	0.8	0.82	0.9	0.88	0.93	0.63	0.86	-	0.81	0.85
UAS	MONO	0.86	0.89	0.84	0.88	0.89	0.94	0.93	0.95	0.8	-	0.85	0.69	0.8
	M0.4	0.85	0.89	0.83	0.85	0.89	0.94	0.93	0.95	0.79	-	0.84	0.68	0.78
	M0.2	0.85	0.89	0.84	0.87	0.88	0.94	0.93	0.95	0.79	-	0.84	0.67	0.79
	TRUNC	0.85	0.89	0.83	0.86	0.89	0.94	0.93	0.95	0.78	-	0.84	0.68	0.79
LAS	MONO	0.79	0.86	0.79	0.84	0.84	0.9	0.9	0.94	0.74	-	0.81	0.59	0.74
	0.4	0.78	0.85	0.78	0.81	0.84	0.9	0.9	0.94	0.72	-	0.79	0.57	0.72
	0.2	0.79	0.85	0.78	0.82	0.84	0.9	0.9	0.94	0.73	-	0.8	0.57	0.72
	TRUNC	0.79	0.85	0.78	0.82	0.84	0.9	0.9	0.93	0.72	-	0.79	0.57	0.72
POS (trf.)	0.4	0.23	0.89	0.25	0.43	0.36	0.31	0.52	0.22	0.18	0.49	-	0.23	0.22
	0.2	0.26	0.89	0.29	0.47	0.37	0.33	0.54	0.24	0.18	0.55	-	0.29	0.28
	TRUNC	0.23	0.89	0.3	0.48	0.32	0.26	0.48	0.2	0.17	0.49	-	0.27	0.28
XNLI	M0.4	0.41	0.67	-	-	-	0.44	-	-	-	0.48	-	0.35	0.35
	M0.2	0.46	0.56	-	-	-	0.45	-	-	-	0.49	-	0.45	0.34
	TRUNC	0.43	0.66	-	-	-	0.43	-	-	-	0.43	-	0.43	0.35
Tatoeba	0.4	0.05	-	0.05	0.19	0.16	-	0.36	0.11	0.04	0.26	0.55	0.12	0.11
	0.2	0.12	-	0.12	0.26	0.21	-	0.34	0.11	0.05	0.33	0.4	0.17	0.19
	TRUNC	0.05	-	0.1	0.2	0.09	-	0.22	0.05	0.03	0.15	0.29	0.1	0.13

Table 4: Full score table across all languages, tasks and models