Finding the Right Language Model Goes a Long Way Towards Accurate Language Identification

Rob van der Goot[1]

[1]NLPnorth, IT University of Copenhagen

Abstract

This paper describes the participation of the NLPnorth team at the CoLI-Dravidian shared task hosted at FIRE2024 [1]. Detecting language on the word level of noisy social media data is still an open challenge. Specifically, for Dravidian languages it is common to code-switch with English in online communication, posing challenges for automatic processing of texts. Starting from a standard language model finetuning, we propose a wide variety of approaches to increase performance on word-level language identification. Our results show that the choice of language model has a large effect on performance, and other methods can lead to even further performance improvements. We experiment with a CRF layer, training on multiple datasets, and language modeling, where each of the methods show different trends across languages/datasets.[1]

Keywords

Language Identification, Named Entity Recognition, Language models

1. Introduction

High performance has been reported for language classification on the sentence level [e.g., 2, 3, 4], especially for canonical language (e.g. Wikipedia, bible). However, in a globalizing world using multiple languages within one utterance is becoming more common, and this is more challenging to detect [e.g., 5, 6, 7, 8]. The "Word-level Code-Mixed Language Identification in Dravidian Languages" shared task targets Dravidian language specifically. Although these languages are originally written in non-latin scripts, due to globalization it became popular to code-switch with English, and use romanization (i.e. latin script) for these languages.

The shared task organizers of the CoLI-Dravidian shared task [1] provided us with data for four Dravidian languages scraped from online platforms: Kannada [9, 10], Tulu [11], Malayalam, and Tamil.[1] As can be seen in an example sentence taken from the data shown in Figure 1, the labels also include symbols, and named entities (in this case 'Banal', which refers to a movie).

As a starting point, we use a standard discriminative transformer-based language model, which we finetune for the task at hand. We compare a wide variety of pre-trained language models. We also propose a variety of approaches to improve performance: 1) train on multiple, related datasets 2) use of a CRF layer 3) task-adaptive pre-training 4) continuous language modeling.

Our findings are:

- Finding the best language model has more influence than any of our proposed modeling improvements.
- The effect of potential improvements is wildly different for different language models and datasets, indicating that we should be careful with conclusions of comparisons done on a single language model.

[1]Code and predictions are available on https://bitbucket.org/robvanderg/coli-dravidian

Forum for Information Retrieval Evaluation, December 12-15, 2024, India

robv@itu.dk (R. v. d. Goot)

https://robvanderg.github.io/ (R. v. d. Goot)

0009-0003-1999-4156 (R. v. d. Goot)

[1]YouTube was reported as the source platform for Kannada and Tulu data, upon manual inspection it seems like the others are from similar platforms

Figure 1: Example from the Malayalam training split, including labels, literal transliteration, and free translation.

Text	Banal	trash	Panavum	nashtam	samayavum	nashtam	.
Labels	OTHER	ENGLISH	MALAYALAM	MALAYALAM	MALAYALAM	MALAYALAM	SYM
Translation	Banal	trash	money	loss	time	loss	.
Free transl.	Banal is trash, a loss of money and time.						

Table 1
Statistics for the included languages (including English). AES level 5: not endangered AES level 4: endangered [15], resource level 1: The Scraping-Bys, resource level 3: The Rising Stars, resource level 5: The Winners [14].

Language	ISO639-3	#Speakers	#Wiki articles	Main Script	AES	Resource
Kannada	kan	43,644,310	31,216	knda	5	1
Malayalam	mal	37,231,970	85,263	mlym	5	1
Tamil	tam	77,456,100	162,659	taml	5	3
Tulu	tcy	1,850,000	2,021	knda	4	1
English	eng	369,935,930	6,780,443	latn	5	5

- Multi-lingual models outperform mono-lingual models in our setups, but this is likely an effect of scale (multi-lingual models are larger, and are trained on more data).

2. Data

We first compare all included languages (the four Dravidian languages and English) from a statistical perspective; we collected their number of speakers [12], number of Wikipedia articles,[2] commonly used scripts, AES endangered status (1-5, 5 is not-endangered) from Glottolog [13], and their resource status according to Joshi et al. [14]. Results in Table 1 show that there are quite many speakers for all languages, and the included languages are mostly not endangered, but are also in the lowest resource level (1: The Scraping-Bys) as defined by Joshi et al. [14].

Since the original data had different labels across the different languages, I first designed a mapping to standardize the labels across languages,[3] which eases the training of multi-dataset models, and simplifies evaluation. Furthermore, the data was originally tokenized on the word level, but sentence boundaries were not annotated. I separated the data on occurrences of '*' and '.' to have shorter chunks of inputs that can more easily be used in length-constrained language models.

After the pre-processing, the resulting label distribution (Table 2) shows that the English label is relatively frequent across all datasets, and that the named entity labels and the mixed labels (a combination of languages within a single word, mostly due to compounds and inflections upon inspection) are more scarce. It should also be noted that the SYM label was much more common in the original data, but it was pre-processed away during the "sentence splitting" (and re-inserted before uploading the test predictions). The only dataset with mixing across Dravidian languages is the Kannada Dataset, which includes a words in Tulu.

3. Methods

We use the MaChAmp toolkit [16] with default hyperparameters for all our experiments (except the statistical baseline). This means we train for 20 epochs, use the adam optimizer with a learning rate of 0.0001, a slanted triangular learning rate [17], and a batch size of 32. MaChAmp uses a language model

[2] https://en.wikipedia.org/wiki/List_of_Wikipedias
[3] Since detailed annotation guidelines were not available the mapping is based on manual inspection of occurrences of labels in the data.

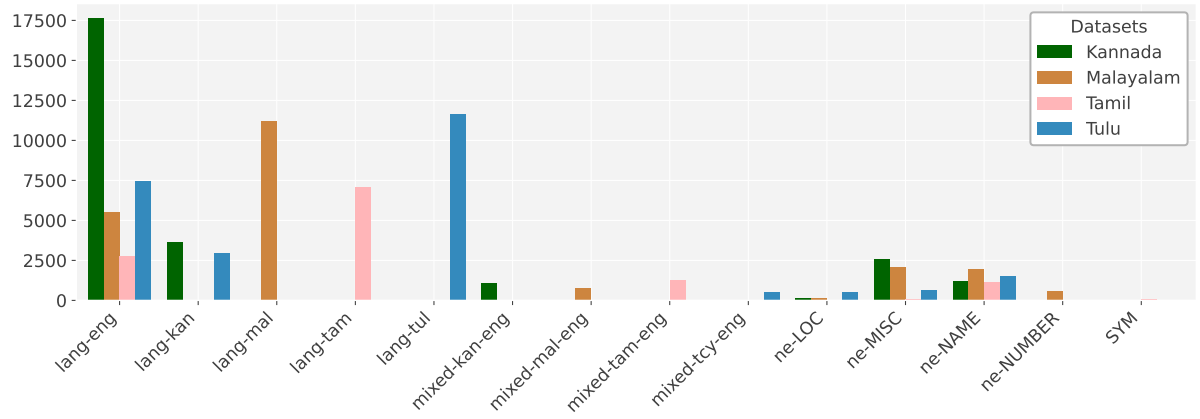

Figure 2: Distribution of the mapped labels for all 4 datasets.

as an encoder, and then adds a feedforward layer on top for classification, and finetunes all weights during training.

3.1. Statistical Baseline

We use character-based profiles as used in textcat [18]. textcat builds character n-gram profiles of texts (which are frequency-ranked lists), which it then uses to compare a new input text to all profiles of the training classes. Since textcat is usually used for sentences and we are classifying on the word level, we re-tuned the hyperparameters where the range of minimum n-gram size is [1,2,3], the maximum [3,4,5,6], and the top-n most frequent n-grams to take into account is [500, 1,000, 10,000, 20,000]. We found that a character n-gram range of 1-6 and the top-n of 20,000 led to the best performance.

3.2. Language models

As a first step, we evaluate a variety of transformer based language models. We use only discriminative language models, and they should be trained on at least one of the included languages. We use the huggingface portal with the language filters and the "fill-mask" task. We excluded language models for which training did not fit on our 40gb GPU's. We pick the best 5 language model based on the average scores, and also the single best language model for each language for further investigations. The following methods are only evaluated on this sub-selection of language models.

3.3. CRF-layer

Upon inspection of the outputs of the initial models, we noticed that many of the cases where the model made an error there was a single label surrounded by other labels. Hence, we add a CRF-layer [19] that incorporates surrounding predictions and models the likelihood of transitioning from a certain label to another label. We also adopt BIO-labels for this setup (and disallow illegal transitions like B-mal ↦ I-eng), as the MaChAmp toolkit enforces this when adding a CRF layer.

3.4. Multi-dataset training

Because the languages are related and annotations are similar, we also attempt to use multi-dataset learning. We first train a single model for all datasets, where we experiment with a separate decoder for each dataset as well as a combined decoder.

Based on this joint model, we also do re-training on each target language. The intuition here is to benefit from all the data while avoiding parameter sharing. For this setup, we also experiment with a

lower learning rate (i.e. *0.1), because the models should have already learned the tasks, and can now focus on learning the more detailed peculiarities of the target language/dataset.

We looked into adding other datasets (for other tasks), but all annotated datasets for the target languages that we could find were in the native (non-Latin) scripts.

3.5. Language modeling

As the larger-sized datasets we could find were all in other scripts than the one used in the shared task, we opted for task-adaptive pretraining [20]. This means that we do language modeling on the training data that is also annotated for the downstream evaluation task. We evaluate the difference between doing language modeling in a sequential setup (first language modeling, then language identification), or in a joint setup (learn both tasks simultaneously). We also evaluate if it is beneficial to see the data only once, or use multiple iterations (up to 20). Note that we keep the amount of epochs and the learning rate stable in the last experiment (i.e. if we see the data only once, the epochs are 20 times smaller), and we use model selection based on the perplexity on the dev set to avoid overfitting.

4. Results

The official metrics for the shared task are macro F1 and weighted F1. Since the task is language identification, and many of the small labels do not refer to languages (i.e. named entities, numbers, and symbols, see Table 1), we use weighted F1 for our evaluations (macro F1 gives equal weight to all labels, so mistakes on smaller labels have a relatively large impact). All reported results (except on the test data) are the average over three seeds.

4.1. Language Model

Table 2 shows the performance of each language model for each dataset. Note that we train a single model for each dataset in this experiment. Interestingly, the mono-lingual models underperform compared to the multi-lingual models, this is probably an effect of the data being romanized, and multilingual models having a larger training data, more training time, and more weights.

4.2. Improvement strategies

We have the exact numbers for all strategies summarized in Appendix A. In this subsection, we will summarize findings for each category of improvements.

CRF-layer The results with an added CRF layer in Figure 3 show that the effect of this differ per language. For Kannada (kan) effects are positive, for Malayalam (mal) negative, and for the other two languages mixed (depending on the language model). Overall, especially when taking into account the standard deviations, differences in performance are relatively small.

Multi-dataset training When training on all datasets simultaneously, the drawback of weight sharing seems to outweigh the benefits of increased training data size as performance is usually lower with higher standard deviations (Figure 4). After re-training on the target dataset/language, we see again that the results differ per language: For Kannadian, this is beneficial for most language models, for Malayalam and Tamil it is negative, whereas for Tulu results are mixed. The lower learning rate has no clear positive effect over the normal re-training. The results of our experiments with a combined decoder classification head showed lower performance for all language models, the scores can be found in Appendix A.

Table 2
Results of our statistical baseline and language models in our baseline setup for all 4 datasets and the average. The language models selected for further experiments are highlighted in bold.

Model	Citation	# weights	Vocab. size	#langs	kan	mal	tam	tcy	Avg.
textcat	[18]	80,000	80,000	4	84.96	83.71	80.99	77.02	81.67
bernice	[21]	277,747,200	250,000	66	96.59	93.32	94.95	90.14	93.75
bert-base-multilingual-cased	[22]	177,853,440	119,547	103	96.23	92.18	93.92	89.79	93.03
bert-base-multilingual-uncased	[22]	167,356,416	105,879	101	96.31	92.64	94.62	90.29	93.47
byt5-small	[23]	299,072,512	256	101	92.37	88.01	89.79	84.14	88.58
byt5-base	[23]	581,063,424	256	101	94.32	90.64	91.24	87.67	90.97
byt5-large	[23]	1,227,593,728	256	101	94.40	89.62	91.37	87.95	90.83
canine-c	[24]	132,082,944	1,114,112	103	88.94	85.04	83.70	81.45	84.78
canine-s	[24]	132,082,944	1,114,112	103	93.03	88.47	88.49	84.76	88.69
distilbert-base-multilingual-cased	[25]	134,734,080	119,547	103	95.76	91.30	93.72	89.62	92.60
glot500-base	[26]	394,121,472	401,145	511	96.17	93.31	94.66	90.07	93.55
infoxlm-base	[27]	278,043,648	250,002	94	96.04	93.22	94.29	89.62	93.29
infoxlm-large	[27]	559,890,432	250,002	94	96.79	93.52	95.14	90.74	94.05
kannada-bert	[28]	237,556,224	197,285	1	96.99	93.20	94.80	90.00	93.74
kannada-bert-scratch	[28]	125,977,344	52,000	1	95.02	91.93	92.42	89.17	92.13
KanBERTo	[29]	83,450,880	52,000	1	92.18	87.51	86.21	84.55	87.61
KooBERT	[30]	184,050,432	128,000	12	69.48	55.34	53.85	59.35	59.50
LaBSE	[31]	470,926,848	501,153	109	96.07	92.32	93.44	89.79	92.90
malayalam-bert	[28]	237,556,224	197,285	1	96.81	92.93	94.80	90.06	93.65
malayalam-bert-scratch	[28]	125,977,344	52,000	1	95.31	92.98	92.71	89.05	92.51
mdeberta-v3-base	[32]	278,218,752	250,101	15	96.84	92.91	94.76	90.33	93.71
mluke-base	[33]	585,839,104	250,002	24	96.58	92.03	94.67	89.78	93.26
mluke-base-lite	[33]	278,639,872	250,002	24	96.58	92.03	94.67	89.78	93.26
mluke-large	[33]	867,884,288	250,002	24	**97.32**	93.45	95.46	90.25	94.12
mluke-large-lite	[33]	560,685,056	250,002	24	96.84	93.43	95.41	90.49	94.04
multilingual-e5-large	[34]	559,890,432	250,002	93	96.54	93.49	94.95	**91.04**	94.00
muril-adapted-local	[35]	237,556,224	197,285	17	90.16	83.88	79.79	82.70	84.13
rembert	[36]	575,920,384	250,300	103	96.62	92.85	94.34	84.62	92.11
sealion-bert-base	[37]	282,649,344	256,000	11	92.94	86.35	85.02	84.57	87.22
tamil-bert	[28]	237,556,224	197,285	1	96.18	92.38	94.48	89.56	93.15
twhin-bert-large	[38]	561,460,736	250,002	89	97.12	93.28	**95.60**	87.72	93.43
twitter-xlm-roberta-base	[39]	278,043,648	250,002	1	96.68	92.64	94.81	90.23	93.59
varta-bert	[40]	184,345,344	128,000	15	35.11	55.75	53.30	59.71	50.97
xlm-mlm-100-1280	[41]	571,496,960	200,000	96	95.66	91.67	93.96	88.23	92.38
xlm-roberta-base	[42]	278,043,648	250,002	93	96.37	92.50	95.17	89.88	93.48
xlm-roberta-large	[42]	559,890,432	250,002	93	96.76	**93.63**	95.45	90.69	**94.13**
xlm-roberta-longformer-base-4096	[43]	280,796,160	250,002	1	95.95	93.00	94.83	89.93	93.43
xlm-v-base	[44]	778,493,184	901,629	93	96.47	93.03	95.28	89.90	93.67

Language modeling For the language modeling experiments we only plot the sequential strategy, as the joint results are consistently substantially lower (Table 5). The remaining results (Figure 5) show that results are mainly positive for Kannadian and Malayam. Also, training on the data 20 times (mlm-20) is not beneficial (in fact, for most models, performance on the dev set was highest in epoch 5-10, so that model was used). Results for twhin-bert-large are worse compared to the other models, probably because its pertaining strategy is the most different compared to standard masked language modeling.

4.3. Results on test data

On the test data, we selected the best 4 models on the average scores over all languages (based on individual seeds), and also submitted the single best models for each language. One interesting observation is that there is a wide variety on what the best five models are, depending on the dataset/language (i.e. the bold numbers in Table 5 do not show clear trends.). This leads us to conclude that we should be careful when claiming generalized findings across different language models in these types of setups.

Results (Table 3) show that our best models performed highly competitively on most languages, except Tamil, which surprisingly was the 2nd highest ranking dataset in our own experiments. It should

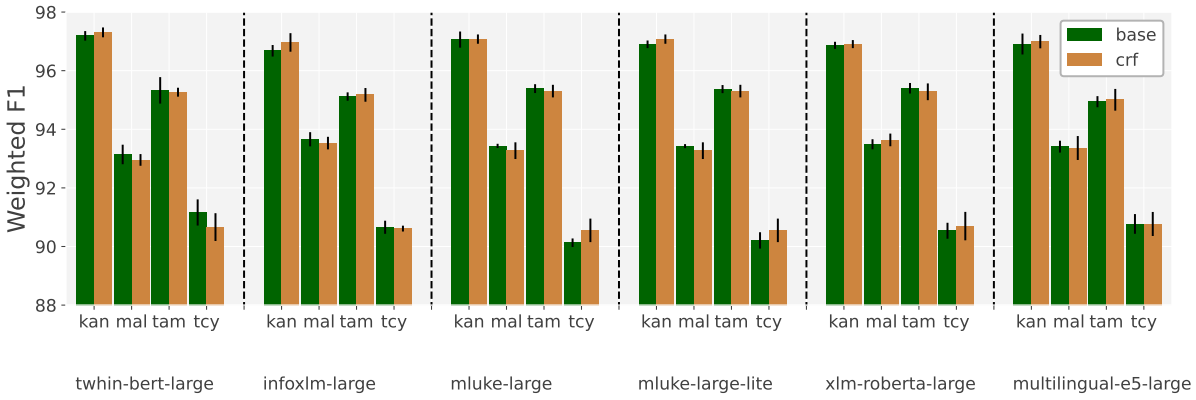

Figure 3: Results of standard MaChAmp model and an added CRF layer, black lines indicate standard deviation.

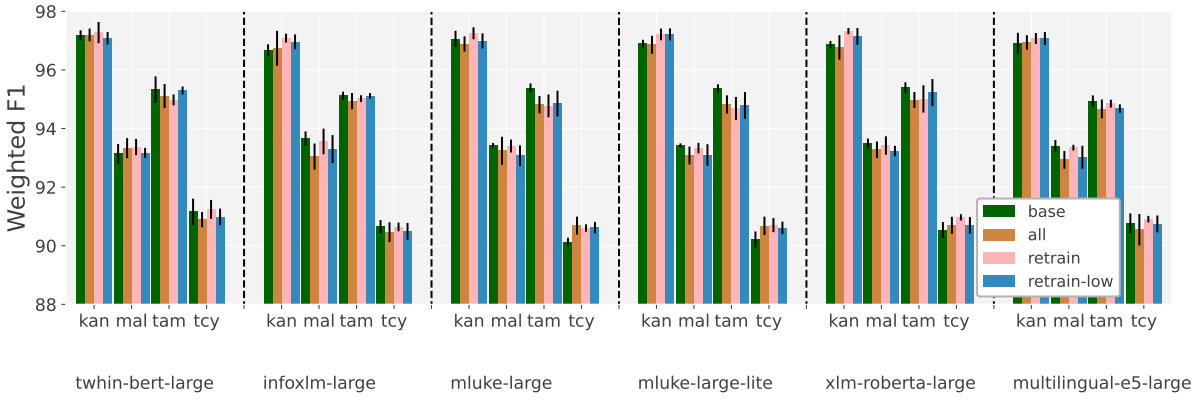

Figure 4: Results of training on all datasets simultaneously (all), and also when re-training on the combined model for individual datasets with the default learning rate (retrain) and a lower learning rate (retrain-low).

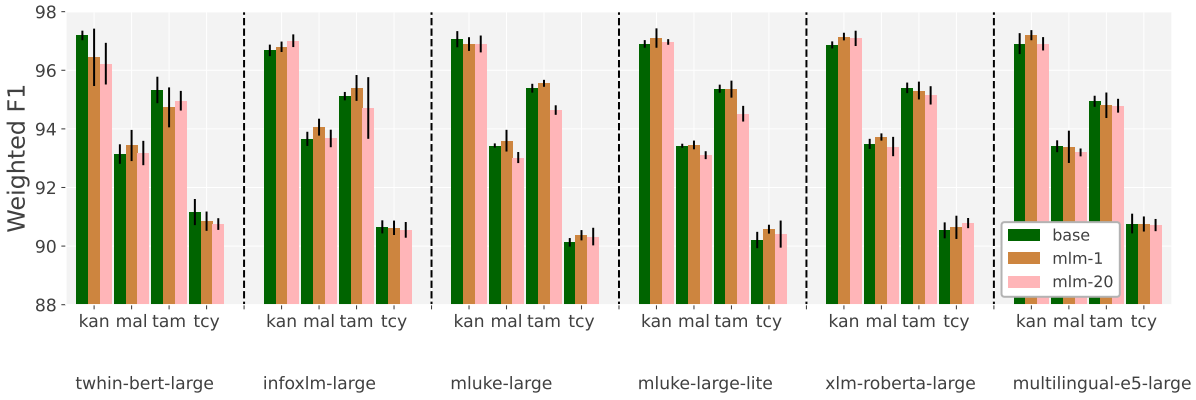

Figure 5: Results when first doing masked language modeling on the original training data for 1 or 20 iterations.

be noted that the official ranking is based on Macro-F1, which I do not report in my paper. Performances are much higher compared to previous shared tasks on Kannada [45] where the winning team achieved weighted F1 of 86, and Tulu [46] where the winning team achieved a macro F1 of 81.3 (our best model has 86.7). However, it is unclear which amount of this change can be ascribed to differences in the data. It can also be seen that performance is slightly lower on the test data compared to the dev data for most

Table 3
Results on the test data, we report the scores of the best run on the test data (we used a total of 5 runs for each dataset).

	Rank	Best model	Dev weighted F1	Test weighted F1
Kannada	2/10	xlm-roberta-large mlm-20	97.15	96.06
Malayalam	1/10	infoxlm-large mlm-20	94.06	93.89
Tamil	4/10	Twitter-twhin-bert-large	95.33	93.55
Tulu	1/9	Twitter-twhin-bert-large	91.16	91.68

Table 4
Pearson correlations of properties of the models and the performance (weighted F1), averaged over all languages. # weights: number of weights in the language model. Vocab size: number of subwords in the vocabulary of the language model. # languages: the number of languages included in the pre-training of the language model. % used: the percentage of subwords in the vocabulary that is used in the datasets. Avg. word len: the average number of subwords per word in the datasets.

Variable	Pearson
# weights	0.2559
Vocab size	0.0693
# languages	0.1668
% used	-0.0153
Avg. word len	-0.1703

languages. This can be due to overfitting, or the test set being more challenging, (dev) results from other teams participating in the shared task might shed more light on this.

5. Analysis

5.1. How to pick the right language model?

Because the choice of language model is an important factor for final performance, we perform a correlation study of different properties of the language models against the final performance. From each model we extract: the number of weights, the size of the vocabulary, the percentage of the vocabulary that is used in a dataset, and the average length of a word (in subwords). We initially also extracted the coverage of the vocabulary for each dataset, but that was almost always 100%, so no usable correlation could be calculated. Results (Table 4) show that none of the weights have a very strong correlation. None of the p-values were $< .05$. Perhaps surprisingly, the percentage of the vocabulary used and the average word length have a negative correlation, although they intuitively could be an indicator of having a better vocabulary. However, this can be explained with the mixed effect to the number of weights; these two variables have a significant ($p < 0.05$) Pearson correlation between 35-36 for all languages.

5.2. What are remaining errors?

To investigate what the remaining errors are, we took the best performing model for each development set, plot a confusion matrix of all combined errors (Figure 6), and manually inspect the errors. It should be noted that this is all done by the first author, who is not speaker of any of the target languages.

Kannada and Tulu are commonly confused, as they occur in the same dataset (the Tulu dataset), some of the cases of confusion are for words that occur in both languages, in other cases it are mostly the context or the individual subwords that occur in the other language that mislead the model. the other main confusion is underprediction of the misc category. As the name suggest, this is likely because it is a less clearly defined category. Upon inspection, we found that for English this is commonly because

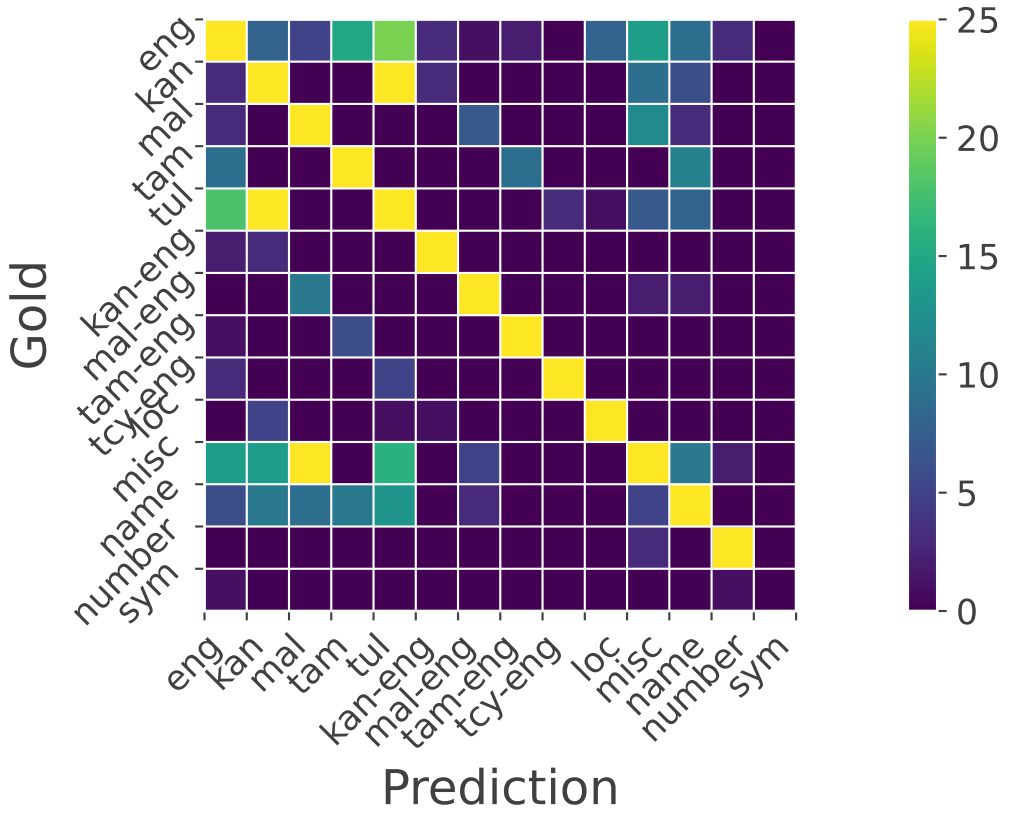

Figure 6: Confusion matrix of frequency of errors by raw counts

standard words are used as part of a name. The eng label also has quite some errors, in both directions (over-prediction and under-prediction). Errors are made here because of interjections (like ah, hahha, which are annotated as eng), typos and slang (Tha is labeled as Tulu by our models) and there seems to be some annotation for the English class which is incorrect (e.g. padike, Bakrid). Finally, the mixed language labels are commonly confused with the dataset languages, in almost all cases this is where only the inflection is done in English, which only leads to 1-2 characters that are different compared to the word in the Dravidian language, hence it is easy for the model to make mistakes.

6. Conclusion

The choice of language model is the most important compared to the other strategies we tested, including a CRF layer, multi-dataset training, and various strategies for including masked language modeling in training. The remaining strategies lead to improved results in certain setups, however, the trends are different across language models and across datasets/languages. Hence, we conclude that future work should be careful with generalizing claims when reporting gains with a limited amount of datasets, languages and/or language models. In our setup, multi-lingual models outperform the mono-lingual models, probably because they are also larger in scale. We evaluated the effect of model size, vocabulary size, vocabulary utility, and average word length with respect to final model performance. Our results show the strongest correlation for model size, and negative correlations for vocabulary utility, but this is probably because of the model size confounder (with an even stronger correlation). An analysis of the errors showed that the remaining cases are often ambiguous words (i.e. their surface form can be used in the annotated and predicted class) or subwords, and interpretation of context is thus still an open challenge.

Acknowledgments

I would like to thank Lottie for maintaining the HPC cluster at the ITU, and the organizers of the shared task for creating the data and sharing it.

Declaration on Generative AI

The author has not employed any generative AI tools.

References

[1] A. Hegde, F. Balouchzahi, S. Butt, S. Coelho, K. G, H. S Kumar, S. D, S. Hosahalli Lakshmaiah, A. Agrawal, Overview of CoLI-Dravidian: Word-level code-mixed language identification in Dravidian languages, in: Forum for Information Retrieval Evaluation FIRE - 2024, 2024.

[2] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: M. Lapata, P. Blunsom, A. Koller (Eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 427–431. URL: https://aclanthology.org/E17-2068.

[3] A. H. Kargaran, A. Imani, F. Yvon, H. Schuetze, GlotLID: Language identification for low-resource languages, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 6155–6218. URL: https://aclanthology.org/2023.findings-emnlp.410. doi:10.18653/v1/2023.findings-emnlp.410.

[4] L. Burchell, A. Birch, N. Bogoychev, K. Heafield, An open dataset and model for language identification, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 865–879. URL: https://aclanthology.org/2023.acl-short.75. doi:10.18653/v1/2023.acl-short.75.

[5] S. Khanuja, S. Dandapat, A. Srinivasan, S. Sitaram, M. Choudhury, GLUECoS: An evaluation benchmark for code-switched NLP, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3575–3585. URL: https://aclanthology.org/2020.acl-main.329. doi:10.18653/v1/2020.acl-main.329.

[6] G. Aguilar, S. Kar, T. Solorio, LinCE: A centralized benchmark for linguistic code-switching evaluation, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 1803–1813. URL: https://aclanthology.org/2020.lrec-1.223.

[7] A. Hegde, F. Balouchzahi, S. Coelho, S. H L, H. A. Nayel, S. Butt, CoLI@FIRE2023: Findings of word-level language identification in code-mixed Tulu text, FIRE '23, Association for Computing Machinery, New York, NY, USA, 2024, p. 25–26. URL: https://doi.org/10.1145/3632754.3633075. doi:10.1145/3632754.3633075.

[8] L. Burchell, A. Birch, R. Thompson, K. Heafield, Code-switched language identification is harder than you think, in: Y. Graham, M. Purver (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 646–658. URL: https://aclanthology.org/2024.eacl-long.38.

[9] F. Balouchzahi, S. Butt, A. Hegde, N. Ashraf, H. Shashirekha, G. Sidorov, A. Gelbukh, Overview of coli-kanglish: Word Level Language Identification in Code-mixed Kannada-English Texts at Icon 2022, in: Proceedings of the 19th International Conference on Natural Language Processing

(ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, 2022, pp. 38–45.
[10] S. H. Lakshmaiah, F. Balouchzahi, M. D. Anusha, G. Sidorov, Coli-machine learning approaches for code-mixed language identification at the word level in kannada-english texts, Acta Polytechnica Hungarica 19 (2022).
[11] A. Hegde, M. D. Anusha, S. Coelho, H. L. Shashirekha, B. R. Chakravarthi, Corpus creation for sentiment analysis in code-mixed tulu text, in: Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages, 2022, pp. 33–40.
[12] Wichmann, Søren, E. W. Holman, C. H. Brown, The ASJP database (version 20), 2022.
[13] H. Hammarström, R. Forkel, M. Haspelmath, S. Bank, Glottolog 5.0., 2024. URL: https://doi.org/10.5281/zenodo.10804357, (Available online at http://glottolog.org, Accessed on 2024-04-24.).
[14] P. Joshi, S. Santy, A. Budhiraja, K. Bali, M. Choudhury, The state and fate of linguistic diversity and inclusion in the NLP world, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6282–6293. URL: https://aclanthology.org/2020.acl-main.560. doi:10.18653/v1/2020.acl-main.560.
[15] H. Hammarström, R. Forkel, M. Haspelmath, S. Bank, GlottoScope, 2024. URL: https://glottolog.org/langdoc/status.
[16] R. van der Goot, A. Üstün, A. Ramponi, I. Sharaf, B. Plank, Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP, in: D. Gkatzia, D. Seddah (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2021, pp. 176–197. URL: https://aclanthology.org/2021.eacl-demos.22. doi:10.18653/v1/2021.eacl-demos.22.
[17] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146 (2018).
[18] W. B. Cavnar, J. M. Trenkle, et al., N-gram-based text categorization, in: Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval, Las Vegas, NV, 1994, p. 14.
[19] J. Lafferty, A. McCallum, F. Pereira, et al., Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Icml, volume 1, Williamstown, MA, 2001, p. 3.
[20] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8342–8360. URL: https://aclanthology.org/2020.acl-main.740. doi:10.18653/v1/2020.acl-main.740.
[21] A. DeLucia, S. Wu, A. Mueller, C. Aguirre, P. Resnik, M. Dredze, Bernice: A multilingual pre-trained encoder for Twitter, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 6191–6205. URL: https://aclanthology.org/2022.emnlp-main.415. doi:10.18653/v1/2022.emnlp-main.415.
[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.
[23] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, C. Raffel, ByT5: Towards a token-free future with pre-trained byte-to-byte models, Transactions of the Association for Computational Linguistics 10 (2022) 291–306. URL: https://aclanthology.org/2022.tacl-1.17. doi:10.1162/tacl_a_00461.
[24] J. H. Clark, D. Garrette, I. Turc, J. Wieting, Canine: Pre-training an efficient tokenization-free encoder for language representation, Transactions of the Association for Computational Linguistics 10 (2022) 73–91. URL: https://aclanthology.org/2022.tacl-1.5. doi:10.1162/tacl_a_00448.

[25] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

[26] A. ImaniGooghari, P. Lin, A. H. Kargaran, S. Severini, M. Jalili Sabet, N. Kassner, C. Ma, H. Schmid, A. Martins, F. Yvon, H. Schütze, Glot500: Scaling multilingual corpora and language models to 500 languages, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1082–1117. URL: https://aclanthology.org/2023.acl-long.61. doi:10.18653/v1/2023.acl-long.61.

[27] Z. Chi, L. Dong, F. Wei, N. Yang, S. Singhal, W. Wang, X. Song, X.-L. Mao, H. Huang, M. Zhou, InfoXLM: An information-theoretic framework for cross-lingual language model pre-training, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 3576–3588. URL: https://aclanthology.org/2021.naacl-main.280. doi:10.18653/v1/2021.naacl-main.280.

[28] R. Joshi, L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages, arXiv preprint arXiv:2211.11418 (2022).

[29] N. Maltesh, Kanberto, 2020. URL: https://huggingface.co/Naveen-k/KanBERTo.

[30] S. Rashinkar, S. Doddapaneni, M. Khapra, KooBERT, https://huggingface.co/KooAI/KooBERT, 2023.

[31] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, W. Wang, Language-agnostic BERT sentence embedding, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 878–891. URL: https://aclanthology.org/2022.acl-long.62. doi:10.18653/v1/2022.acl-long.62.

[32] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, The Eleventh International Conference on Learning Representations (2021).

[33] R. Ri, I. Yamada, Y. Tsuruoka, mLUKE: The power of entity representations in multilingual pretrained language models, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 7316–7330. URL: https://aclanthology.org/2022.acl-long.505. doi:10.18653/v1/2022.acl-long.505.

[34] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report, arXiv preprint arXiv:2402.05672 (2024).

[35] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al., Muril: Multilingual representations for indian languages, arXiv preprint arXiv:2103.10730 (2021).

[36] H. W. Chung, T. Fevry, H. Tsai, M. Johnson, S. Ruder, Rethinking embedding coupling in pre-trained language models, International Conference on Learning Representations (2020).

[37] A. Singapore, Sea-lion (southeast asian languages in one network): A family of large language models for southeast asia, https://github.com/aisingapore/sealion, 2024.

[38] X. Zhang, Y. Malkov, O. Florez, S. Park, B. McWilliams, J. Han, A. El-Kishky, Twhin-bert: A socially-enriched pre-trained language model for multilingual tweet representations at twitter, in: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, 2023, pp. 5597–5607.

[39] F. Barbieri, L. Espinosa Anke, J. Camacho-Collados, XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 258–266. URL: https://aclanthology.org/2022.lrec-1.27.

[40] R. Aralikatte, Z. Cheng, S. Doddapaneni, J. C. K. Cheung, Varta: A large-scale headline-generation dataset for Indic languages, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 3468–3492. URL: https://aclanthology.org/2023.findings-acl.215. doi:10.18653/v1/2023.findings-acl.215.

[41] A. Conneau, G. Lample, Cross-lingual language model pretraining, Advances in neural information processing systems 32 (2019).

[42] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747. doi:10.18653/v1/2020.acl-main.747.

[43] M. Sagen, Large-Context Question Answering with Cross-Lingual Transfer, Master's thesis, Uppsala University, Department of Information Technology, 2021.

[44] D. Liang, H. Gonen, Y. Mao, R. Hou, N. Goyal, M. Ghazvininejad, L. Zettlemoyer, M. Khabsa, XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 13142–13152. URL: https://aclanthology.org/2023.emnlp-main.813. doi:10.18653/v1/2023.emnlp-main.813.

[45] F. Balouchzahi, S. Butt, A. Hegde, N. Ashraf, H. Shashirekha, G. Sidorov, A. Gelbukh, Overview of CoLI-kanglish: Word level language identification in code-mixed Kannada-English texts at ICON 2022, in: B. R. Chakravarthi, A. Murugappan, D. Chinnappa, A. Hane, P. K. Kumeresan, R. Ponnusamy (Eds.), Proceedings of the 19th International Conference on Natural Language Processing (ICON): Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, Association for Computational Linguistics, IIIT Delhi, New Delhi, India, 2022, pp. 38–45. URL: https://aclanthology.org/2022.icon-wlli.8.

[46] A. Hegde, S. Coelho, H. Shashirekha, H. A. Nayel, S. Butt, Overview of coli-tunglish: Word-level language identification in code-mixed tulu text at fire 2023., in: Forum for Information Retrieval Evaluation (FIRE 2023), 2023, pp. 1–12.

A. Full results

The full results for all our experiments and our six selected language models are reported in Table 5.

Table 5
Weighted F1 scores on the dev split for all our evaluated setups.

	kn	mal	tamil	tulu	avg.
textcat	84.96	83.71	80.99	77.02	81.67
twhin-bert-large					
single	97.19	93.14	**95.33**	91.16	94.20
all	97.20	93.33	95.11	90.89	94.13
all-combined	95.88	91.75	93.16	89.68	92.61
retrain	97.27	93.37	94.98	**91.24**	**94.22**
retrain-lowLR	97.08	93.17	95.30	90.98	94.13
bio	**97.31**	92.95	95.27	90.66	94.05
mlm-20.seq	96.44	**93.43**	94.73	90.85	93.86
mlm-1.seq	96.23	93.18	94.96	90.75	93.78
mlm-20.joint	94.66	91.63	94.57	88.43	92.32
mlm-1.joint	96.72	92.87	94.74	90.36	93.67
infoxlm-large					
single	96.68	93.66	95.12	**90.66**	94.03
all	96.74	93.04	94.94	90.46	93.79
all-combined	95.25	92.47	92.84	89.99	92.64
retrain	**97.09**	93.56	95.02	90.64	94.08
retrain-lowLR	96.96	93.30	95.12	90.49	93.97
bio	96.96	93.53	95.17	90.61	94.07
mlm-20.seq	96.80	**94.06**	**95.40**	90.62	**94.22**
mlm-1.seq	97.01	93.67	94.71	90.55	93.99
mlm-20.joint	95.16	92.91	95.00	89.84	93.23
mlm-1.joint	96.98	93.22	94.97	90.36	93.88
mluke-large					
single	97.06	93.43	95.39	90.13	94.00
all	96.89	93.24	94.81	**90.69**	93.91
all-combined	96.46	92.43	92.83	90.31	93.01
retrain	**97.25**	93.40	94.77	90.60	94.01
retrain-lowLR	96.99	93.07	94.85	90.62	93.88
bio	97.08	93.27	95.30	90.55	94.05
mlm-20.seq	96.89	**93.60**	**95.55**	90.37	**94.10**
mlm-1.seq	96.90	93.02	94.64	90.32	93.72
mlm-20.joint	94.66	92.27	94.86	89.17	92.74
mlm-1.joint	96.81	92.97	94.80	90.48	93.76
mluke-large-lite					
single	96.90	93.42	**95.37**	90.21	93.98
all	96.86	93.08	94.82	90.68	93.86
all-combined	96.40	92.50	92.67	90.40	92.99
retrain	97.21	93.33	94.68	**90.70**	93.98
retrain-lowLR	**97.21**	93.10	94.79	90.61	93.93
bio	97.08	93.27	95.30	90.55	94.05
mlm-20.seq	97.10	**93.45**	95.36	90.58	**94.12**
mlm-1.seq	96.97	93.10	94.52	90.41	93.75
mlm-20.joint	95.08	92.28	95.17	88.95	92.87
mlm-1.joint	97.01	92.99	94.79	90.18	93.74
xlm-roberta-large					
single	96.87	93.49	**95.40**	90.54	94.07
all	96.77	93.28	94.98	90.70	93.93
all-combined	96.44	92.01	92.94	90.85	93.06
retrain	**97.32**	93.42	95.01	**90.97**	94.18
retrain-lowLR	97.15	93.23	95.22	90.70	94.07
bio	96.91	93.64	95.28	90.70	94.13
mlm-20.seq	97.15	**93.72**	95.31	90.64	**94.21**
mlm-1.seq	97.09	93.40	95.14	90.79	94.10
mlm-20.joint	95.10	93.11	94.88	89.39	93.12
mlm-1.joint	96.95	93.65	95.22	90.61	94.11
multilingual-e5-large					
single	96.91	**93.41**	94.94	90.77	94.01
all	96.94	92.93	94.67	90.55	93.77
all-combined	96.46	92.90	93.02	**91.01**	93.35
retrain	97.07	93.35	94.85	90.91	**94.04**
retrain-lowLR	97.07	93.02	94.68	90.74	93.88
bio	96.99	93.36	**95.01**	90.77	94.03
mlm-20.seq	**97.20**	93.39	94.81	90.75	94.04
mlm-1.seq	96.90	93.20	94.79	90.72	93.90
mlm-20.joint	96.07	93.11	94.94	90.10	93.55
mlm-1.joint	96.73	93.35	94.78	90.70	93.89