Leveraging MuRIL for Word-Level Language Identification in Code-Mixed Dravidian Text

Akhil Bhogal[1,*,†], Soumyadeep Sar[1,*,†], Dwaipayan Roy[1,*] and Kripabandhu Ghosh[1,*]

[1]IISER Kolkata, Kalyani

Abstract

Language Identification (LI) is essential for many NLP tasks, particularly in multilingual and code-mixed contexts like social media in India. Dravidian languages, widely spoken in southern India, present unique challenges for LI due to their rich morphology and the frequent use of Roman or hybrid scripts. This report details our approach to addressing word-level LI in Kannada, Tamil, Malayalam, and Tulu as part of a shared task. We fine-tuned MuRIL, a BERT based model from Google, which is trained on transliterated and translated Indian Languages. It performed significantly better than mBERT on multi-lingual benchmarks like XNLI and XTREME, making it a suitable choice for the task. Our models performed exceptionally well, securing 1^{st} place in Kannada, 2^{nd} in Tulu, 3^{rd} in Tamil, and 2^{nd} in Malayalam (with the latter submitted late). Our results demonstrate the effectiveness of our approach in tackling the complexities of LID in Dravidian languages.

Keywords

LLMs, Code-mixed, BERT, Deep learning, NLP, Dravidian Languages, Language identification (LI)

1. Introduction

Language Identification (LI) is a natural language processing (NLP) research problem that aims to determine the language(s) in which a given text is written. It can be applied at various levels: document, sentence, word, or sub-word level. In multilingual environments, LI is crucial for applications like machine translation, information retrieval, and sentiment analysis. LI becomes more complex in cases of code-mixed text, where multiple languages are used within the same sentence or word. This is particularly common in social media or informal communication in multilingual regions. In India we have regionally enriched diverse languages, leading to serious concern in how to deal with them. This can be greatly solved by Language Identification.

LI in multilingual and code-mixed contexts, particularly within Dravidian languages, has emerged as a critical area of research. This is driven by the increasing importance of code-mixed texts in social media, where local languages like Tamil, Kannada, Malayalam, and Tulu are intermixed with English. Such mixing occurs at various levels, including within individual words, posing significant challenges for conventional LI models.

Recent studies have highlighted the complexities associated with Dravidian languages, which, despite their rich morphological structures, are under-resourced in terms of linguistic resources and technological tools. For instance, Chakravarthi et al. [1] emphasize the difficulties in handling offensive language detection in code-mixed texts, showing that traditional machine learning models like Naive Bayes and Logistic Regression outperformed more advanced models like BERT in specific scenarios involving Dravidian languages. This suggests that tailored approaches may be necessary for effective language identification in these contexts.

The ongoing challenges have given rise to initiatives like the CoLI-Dravidian@FIRE 2024 shared task [2, 3, 4, 5, 6], which will lead to advance word-level LI models for these languages. By providing curated

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

*Corresponding author.

†These authors contributed equally.

akhilbhogal25@gmail.com (A. Bhogal); Soumyadeepsar26@gmail.com (S. Sar); dwaipayan.roy@iiserkol.ac.in (D. Roy); kripaghosh@iiserkol.ac.in (K. Ghosh)

0009-0006-9265-6729 (S. Sar)

datasets and fostering competition, such tasks aim to push the boundaries of current LI methodologies and improve the handling of under-resourced languages in code-mixed scenarios.

This research is vital for applications across various domains, including sentiment analysis, machine translation, and social media monitoring, where accurate language identification is foundational.

Declaration of Generative AI

The Author(s) have not employed any Generative AI tools.

2. Related Works

The study from Hegde et al. [7] focuses on identifying the language of individual words in sentences containing a mix of Tulu, Kannada, and English. This task is crucial for processing code-mixed texts, especially for under-resourced languages like Tulu, which lack annotated data. The authors introduced an open-source dataset and organized a shared task where participants categorized words into predefined classes like Tulu, Kannada, English, Mixed, Name, Location, and Other. Machine learning models using TF-IDF of character n-grams were commonly employed, with the best model achieving a weighted F1 score of 0.89 and a macro F1 score of 0.81. This work lays the groundwork for future language identification tools for under-resourced languages.

The research work from E. Ojo et al. [8] tackles the challenge of word-level language identification in code-mixed texts, particularly focusing on Kannada-English (Kn-En) language pairs. The study addresses the complexities of code-mixing, where languages can be intermingled at the sentence, word, or sub-word level. By employing a combination of machine learning and deep neural networks, the authors utilized both character-based and word-based text features. The dataset, comprising YouTube video comments, was pre-processed and categorized into classes such as "Kannada," "English," "Mixed-Language," "Name," "Location," and "Other." Their CK-Keras model, enhanced with pre-trained Word2Vec embeddings, demonstrated superior performance, achieving the highest F1 scores among the evaluated methods. This work contributes significantly to the development of language identification models for code-mixed languages, showcasing the effectiveness of combining traditional machine learning with deep learning techniques.

3. Dataset and pre-processing

The dataset was derived from two pivotal works done in Dravidian code-mixed text. The first one being, the work from Hegde et al. [9], where work was done to promote sentiment analysis in code-mixed YouTube comments, especially for low-resource languages like Tulu, which could be very challenging. The second work which contributed to the dataset was the code-mixed Kannada-English dataset from Shashirekha et al. [10]. For 4 different Dravidian languages, the details of each of these datasets is listed in the Table 1. The main categories of tags in the data are listed below:

- Dravidian Language
- English
- Mixed Language(Dravidian language + English mixed terms)
- Name
- Location(Name of places)
- Other

The distribution of word-level tags(labels) across different languages is shown in Fig. 1. Tokens “.” and “*” having ‘SYM’ or ‘sym’ tags were excluded from the dataset, before training our models. The words tagged as their respective language tag, are code-mixed words, words written in roman script.

Table 1
Details of datasets

Language	No. of Tokens in Train/Valid/Test	No. of Categories
Tulu	29523/3006/3283	6
Kannada	30015/2484/2502	6
Tamil	13556/1984/2024	6
Malayalam	24995/2504/2401	7

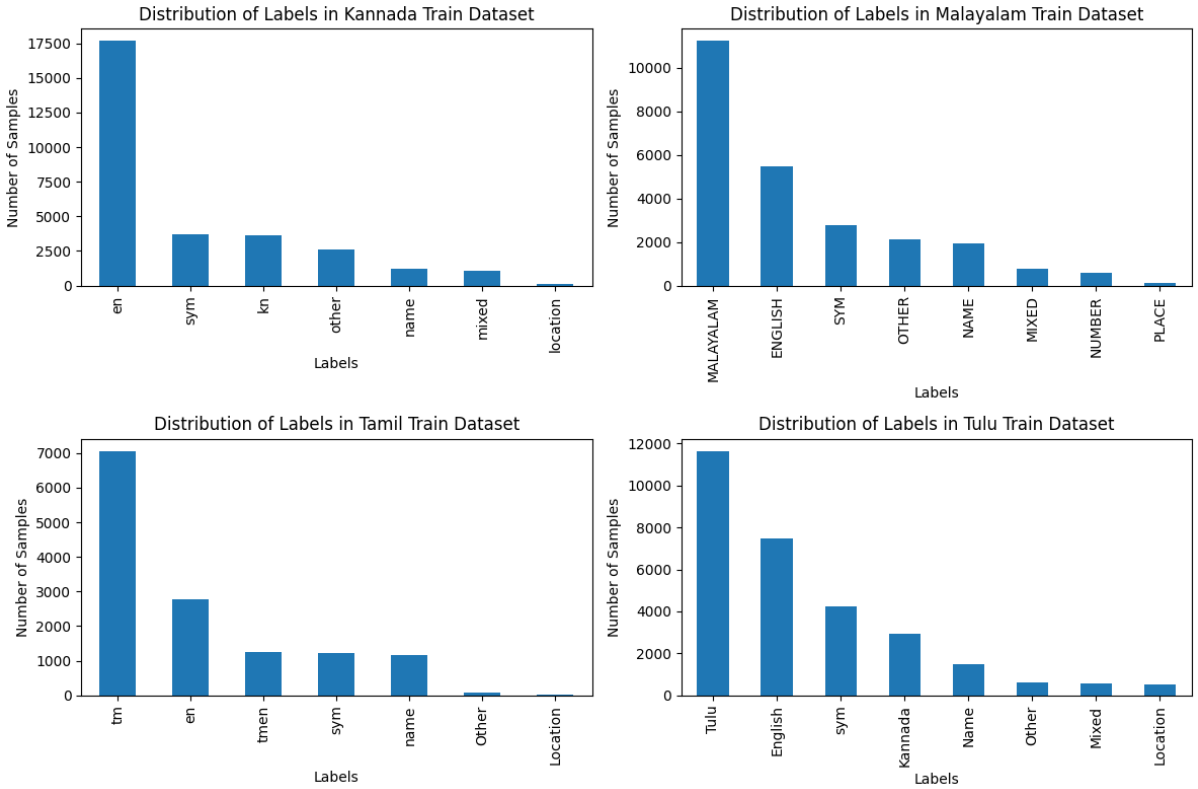

Figure 1: Tag distribution across different languages

The mixed words are usually english words merged with other language extensions, making it a mixed word, written in Roman script.

In the pre-processing stage, we handle tokenization and label alignment for inputs where words may be split into subwords. Here's a general approach:

3.1. Tokenization with Subword Splitting:

Input data is tokenized, with each word potentially being split into multiple subword tokens. This is necessary for models like BERT, which use subword tokenization. Handling Special Tokens and Label Alignment:

- The tokenizer introduces special tokens (e.g., [CLS], [SEP]) that do not correspond to any label. We assign a label of -100 to these tokens so they are ignored during training.
- For words that split into multiple subword tokens, only the first token retains the original label. Subsequent subword tokens are also assigned a label of -100.
- Realigning Tokens and Labels:
  We map each token back to its original word using the tokenizer's word ids method. Labels are realigned accordingly, ensuring that only meaningful token-label pairs are retained.

Table 2
Evaluation of MuRIL and mBERT on XTREME(IN) benchmarks

Model	PANX F 1	UDPOS F1	XNLI Acc.	Tatoeba Ācc.	XQuAD $\mathbf{F1/EM}$	MLQA $\mathbf{F1}/EM$	TyDiQA-GoldP $\mathbf{F1/EM}$	Avg.
mBERT	58.0	71.2	66.8	18.4	71.2/58.2	65.3/51.2	63.1/51.7	59.1
MuRIL	77.6	75.0	74.1	25.2	79.1/65.6	73.8/58.8	75.4/59.3	68.6

- Batch Processing and Dynamic Padding:
  The dataset is pre-processed in batches to improve efficiency. During batching, sequences are dynamically padded to the length of the longest sequence in the batch, optimizing memory usage and computational efficiency. This process ensures that the input data is correctly formatted for training, with proper handling of subword tokenization and label alignment.

4. Methodology

The research problem demands to classify word-level entities into different language classes as well as names of people and location. This problem clearly encompasses two types of task in one, the first being language identification and second one is Named Entity Recognition (NER) task. So we address this task as a Token classification problem. We use `AutoModelForTokenClassification`[1] from transformers [11] to use particularly the token classification configuration of pre-trained BERT-based models along with the task-oriented layers. We experimented with various multilingual transformer models, which is discussed below.

4.1. Model used

MuRIL's (Khanuja et al. [12]) unique training approach involves the use of translated and transliterated data, which enhances its ability to handle multilingual text, especially for Indian languages. Unlike mBERT (Devlin et al. [13]), which only uses monolingual corpora, MuRIL incorporates cross-lingual translation tasks to improve understanding of code-mixed and transliterated content.

Due to its different pre-training approach, MuRIL outperforms mBERT on the XTREME (Hu et al. [14]) benchmark, specifically excelling in XNLI (Conneau et al. [15], ,cross-lingual natural language inference), NER, and question-answering tasks. We can see their performances on the benchmarks in Table 2, where we can see MuRIL outperforming mBERT by a reasonable margin, leading to our choice for fine-tuning on the problem task.

MuRIL-base-cased has been used for all languages except for Malayalam where MuRIL-large-case has been used.

4.2. Training process

We primarily used muril-base-cased model, for finetuning on each language task, which was used to make predictions on the test dataset. The details of fine-tuning are given below as per each language:

4.2.1. Tulu

For the Tulu language model, we fine-tuned the base model for 30 epochs with a learning rate of 2e-5 and a weight decay of 0.001. Both the training and evaluation were conducted with a batch size of 16. During this process, we monitored the validation loss to identify the best-performing model. After the single training run, the model with the lowest validation loss was selected as the final model for this language.

[1] https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#transformers.AutoModelForTokenClassification

Table 3
Result across different languages

Language	M_{F1}	M_{Pr}	M_{Re}	W_{F1}	W_{Pr}	W_{Re}/Acc
Tulu	0.85852938	0.89882938	0.82815216	0.91790522	0.91913179	0.91867195
Kannada	0.92936992	0.92419394	0.93591665	0.96689675	0.96709914	0.96682654
Tamil	0.72329027	0.71897978	0.72901181	0.93381769	0.93215274	0.93577075
Malayalam	0.87637428	0.90565098	0.85539625	0.93395478	0.93500179	0.93722511

4.2.2. Kannada

We used similar hyperparameters as above for this language as well: a learning rate of 2e-5, a weight decay of 0.001, and batch sizes of 16 for both training and evaluation. The model was trained for 20 epochs, and to ensure robustness, we trained the model three times. Each time, we evaluated the model on the validation set and selected the model with the lowest evaluation loss as the final model. Additionally, we enabled the load-best-model-at-end as *True* parameter to ensure that the best model checkpoint was restored at the end of training. This avoids over-fitting on the training data.

4.2.3. Tamil

We initially fine-tuned the model for 20 epochs with a learning rate of 2e-5, a weight decay of 0.001, and a batch size of 16. However, the initial results were not satisfactory, indicating that further training could improve performance. Therefore, the output model from the first round of fine-tuning was further trained for an additional 20 epochs using the same hyper-parameters. Among the models generated, the one with the best macro F1 score on the validation set was selected as the final model for Tamil.

4.2.4. Malayalam

In case of malayalam we particularly experimented and tried the "muril-large-cased" model, which is much larger than the base version. The model was fine-tuned for 15 epochs with a learning rate of 1.2e-5, a smaller learning rate compared to other language training. The other parameters were kept the same as the above 3 training setups.

5. Result and Discussion

From table 3, we observe outstanding result in case of Kannada language, compared to all other languages. This can be attributed to the fact that we have a lot of words belonging from english language making the task much easier(refer to fig 1). Tamil was the language on which our models did not performed quite upto the mark. The reason being Tamil's smallest training dataset, which caused a fall in performance for the fine-tuned model's performance. For Tulu language, the performance deteriorate was due to the fact that it is a low-resource language, hence no models were pre-trained on the language. While for other Languages Muril performed reasonably good, owing to its special pre-training on transliterated text and its counter-part. Also constant monitoring of validation loss on validation data for each fine-tuning process helped us to obtain the model checkpoint which is not over-fitted, even though we trained model for so many epochs.

6. Additional Experiments

6.1. Seq2Seq Modelling

In this additional experiment, we explored the use of Flan-T5-Small(Chung et al. 16) for this language identification (LID) and named entity recognition (NER) task. We fine-tuned the Flan-T5-Small model

Table 4
Result from Flan-T5 small

Technique	Language	M_{F1}	M_{Pr}	M_{Re}	W_{F1}	W_{Pr}	W_{Re}/Acc
Seq2Seq	Tamil	0.72812546	0.73125305	0.72545729	0.93779794	0.93746167	0.93853074
Token Classifier	Tamil	0.57876845	0.57198737	0.58707348	0.91158505	0.91422529	0.91004997

using a prefix-based approach where each input was preceded by the prompt "Do LID:". The model was trained for 10 epochs with a learning rate of 5e-4, a batch size of 8 for training, and 2 for evaluation. The training was designed to maximize performance on word-level classification tasks. The model was evaluated using the validation dataset, and the epoch with the lowest validation loss was selected for inference.

During inference, we encountered two major issues that impacted the performance:

- **Omitted Words Issue:** In some cases, the number of words predicted by the model did not match the actual number of words in the input sentence. This discrepancy led to the omission of certain words from the prediction output. For instance, in the sentence:
  "*Super star na adhu rajini sir mattum tha vera yarunaalayum antha patta tha vanga muditadhu da*"
  , the output using the model was:
  {'Super': 'en', 'star': 'en', 'na': 'tm', 'adhu': 'tm', 'rajini': 'name', 'sir': 'en', 'mattum': 'tm', 'tha': 'tm', 'vera': 'tm', 'yarunaalayum': 'tm', 'antha': 'tm', 'vanga': 'tm', 'muditadhu': 'tm', 'da': 'tm'}
  While the output is in correct format, some of the words (*'patta', 'tha'*) from within have been omitted. Though, for the predicted language for the remaining words is correct. In total, 39 sentences from the validation dataset had to be omitted due to this mismatch.
- **Inappropriate Substitution Issue:** The second issue was the inappropriate substitution of certain words with unrelated terms. For example, in the sentence "*Irudhiyil ivarum MGR labellil dhaan Avar Cinemavil Herovoda Vaazhkaiyai ottuvaara*", the word "*Cinemavil*" (which should have been tagged as a code-mixed word 'tmen') was substituted by "*Movievil*" and incorrectly tagged as 'en' (English) in the output.

Despite these issues, the model achieved a weighted F1-score of 0.9378 and an accuracy of 0.9385 on the remaining 144 sentences. The macro F1-score was 0.7281, indicating a relatively balanced performance across the different classes. However, these problems with word omission and inappropriate substitutions suggest areas where further refinement of the model or preprocessing steps could improve accuracy, especially in complex code-mixed scenarios.

These findings highlight the challenges in using a general-purpose model like Flan-T5-Small for specialized tasks involving code-mixed language and suggest the need for more tailored approaches or enhanced training data to handle such cases effectively.

6.2. Flan-T5 as token classifier:

We also fine-tuned the pre-trained 'google/flan-t5-small' model for token classification, using *AutoModelForTokenClassification* from *transfomers* library of HuggingFace. The model was trained with a learning rate of 2e-5, with a batch size of 16 for both training and evaluation. The training process consisted of 30 epochs, with a weight decay of 0.001 to regularize the model. To manage model checkpoints, we set the *save_total_limit* to 5, retaining only the most recent five model saves. This training configuration was executed three times to ensure reliability and consistency of the results. The final result is mention in Table 4

Due to the aforementioned problems in this method, we discontinued the use after Tamil, and did not fine-tune it for other languages.

7. Conclusion

For this task of LID in Code-Mixed text, we fine-tuned a seq2seq (encoder-decoder) model Flan-T5-small and MuRIL, a BERT-based encoder-only model. We found that FLan-T5-small is not suitable for this task, while on the other hand MuRIL which has also been pre-trained on the roman text of the many Dravidian languages performed quite well, with having 0.9294 Macro-F1 score for Kannada. We also found that there is a correlation between the size of training dataset and the Macro-F1 score. We attributed the score of Tulu being average despite a large dataset to the pre-training of the model (as Tulu is low-resource language).

8. Future Directions

Other highly accurate and heavy large language models like XLM-RoBERTa(large and base) were not fine-tuned and tested in this report. Their performance can also give us crucial insights on the research problem. Apart from just fine-tuning a LLM, we can use techniques like data augmentation using synthetic data obtained from SOTA models like Llama-3 , GPT and so on. The Augmented data can train models on a much more diverse representations, which in turn can lead to even better performances. Furthermore we can try to understand how LI can help in other NLP tasks in code-mixed text paradigm like sentiment analysis, bias detection, fake news, etc.

References

[1] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada, in: B. R. Chakravarthi, R. Priyadharshini, A. Kumar M, P. Krishnamurthy, E. Sherly (Eds.), Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 133–145. URL: https://aclanthology.org/2021.dravidianlangtech-1.17.

[2] F. Balouchzahi, S. Butt, A. Hegde, N. Ashraf, S. Hosahalli Lakshmaiah, G. Sidorov, A. Gelbukh, Overview of CoLI-Kanglish: Word Level Language Identification in Code-mixed Kannada-English Texts at ICON 2022, in: 19th International Conference on Natural Language Processing Proceedings, 2022.

[3] 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.

[4] 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).

[5] 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.

[6] 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.

[7] A. Hegde, F. Balouchzahi, S. Coelho, H. L. Shashirekha, H. A. Nayel, S. Butt, Overview of coli-tunglish: Word-level language identification in code-mixed tulu text at FIRE 2023, in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation (FIRE-WN 2023), Goa, India, December 15-18, 2023, volume 3681 of *CEUR*

Workshop Proceedings, CEUR-WS.org, 2023, pp. 179–190. URL: https://ceur-ws.org/Vol-3681/T4-1.pdf.

[8] O. E. Ojo, A. Gelbukh, H. Calvo, A. Feldman, O. O. Adebanji, J. Armenta-Segura, Language identification at the word level in code-mixed texts using character sequence and word embedding, 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. 1–6. URL: https://aclanthology.org/2022.icon-wlli.1.

[9] A. Hegde, M. D. Anusha, S. Coelho, H. L. Shashirekha, B. R. Chakravarthi, Corpus creation for sentiment analysis in code-mixed Tulu text, in: M. Melero, S. Sakti, C. Soria (Eds.), Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages, European Language Resources Association, Marseille, France, 2022, pp. 33–40. URL: https://aclanthology.org/2022.sigul-1.5.

[10] H. L. Shashirekha, F. Balouchzahi, M. D. Anusha, G. Sidorov, Coli-machine learning approaches for code-mixed language identification at the word level in kannada-english texts, 2022. URL: https://arxiv.org/abs/2211.09847. arXiv:2211.09847.

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[12] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, P. Talukdar, Muril: Multilingual representations for indian languages, 2021. arXiv:2103.10730.

[13] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[14] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, M. Johnson, Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization, 2020. URL: https://arxiv.org/abs/2003.11080. arXiv:2003.11080.

[15] A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, V. Stoyanov, Xnli: Evaluating cross-lingual sentence representations, 2018. URL: https://arxiv.org/abs/1809.05053. arXiv:1809.05053.

[16] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling instruction-finetuned language models, 2022. URL: https://arxiv.org/abs/2210.11416. doi:10.48550/ARXIV.2210.11416.