

MuCoT: Multilingual Contrastive Training for Question-Answering in Low-resource Languages

Gokul Karthik Kumar Abhishek Singh Gehlot
Sahal Shaji Mullappilly Karthik Nandakumar

Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI)
Abu Dhabi, UAE

{gokul.kumar, abhishek.gehlot}@mbzuai.ac.ae
{sahal.mullappilly, karthik.nandakumar}@mbzuai.ac.ae

Abstract

Accuracy of English-language Question Answering (QA) systems has improved significantly in recent years with the advent of Transformer-based models (e.g., BERT). These models are pre-trained in a self-supervised fashion with a large English text corpus and further fine-tuned with a massive English QA dataset (e.g., SQuAD). However, QA datasets on such a scale are not available for most of the other languages. Multi-lingual BERT-based models (mBERT) are often used to transfer knowledge from high-resource languages to low-resource languages. Since these models are pre-trained with huge text corpora containing multiple languages, they typically learn language-agnostic embeddings for tokens from different languages. However, directly training an mBERT-based QA system for low-resource languages is challenging due to the paucity of training data. In this work, we augment the QA samples of the target language using translation and transliteration into other languages and use the augmented data to fine-tune an mBERT-based QA model, which is already pre-trained in English. Experiments on the Google ChAII dataset show that fine-tuning the mBERT model with translations from the same language family boosts the question-answering performance, whereas the performance degrades in the case of cross-language families. We further show that introducing a contrastive loss between the translated question-context feature pairs during the fine-tuning process, prevents such degradation with cross-lingual family translations and leads to marginal improvement. The code for this work is available at <https://github.com/gokulkarthik/mucot>.

1 Introduction

India has a population of 1.4 billion people speaking 447 languages and over 10,000 dialects, making it the country with the fourth-highest number of languages (Chakravarthi, 2020). However, Indian lan-

guages are highly under-represented on the Internet and Natural Language Processing (NLP) systems for Indian languages are in their nascency (Bharathi et al., 2022; Priyadharshini et al., 2021). Tamil is a member of the southern branch of the Dravidian languages, a group of about 26 languages indigenous to the Indian subcontinent. It is also classed as a member of the Tamil language family, which contains the languages of around 35 ethnolinguistic groups, including the Irula and Yerukula languages (Sakuntharaj and Mahesan, 2021, 2017, 2016; Thavareesan and Mahesan, 2019, 2020a,b, 2021). Malayalam is Tamil’s closest significant cousin; the two began splitting during the 9th century AD. Although several variations between Tamil and Malayalam indicate a pre-historic break of the western dialect, the process of separating into a different language, Malayalam, did not occur until the 13th or 14th century (Anita and Subalalitha, 2019b,a; Subalalitha and Poovammal, 2018; Subalalitha, 2019). Even state-of-the-art multilingual NLP systems perform sub-optimally on Dravidian languages (Google, 2021). This can be explained by the fact that multilingual language models are often jointly trained on 100+ languages and Indian languages constitute only a small fraction of their vocabulary and training data (as shown in Figure 2).

Machine learning models and tools have been proposed for many Natural Language Understanding tasks. In this work, we focus on Extractive Question-Answering (QA), where the goal is to localize the answer to a question within a large context (see Figure 1). Specifically, we aim to develop a common multilingual question answering model for multiple Indian languages. A multilingual model has several advantages: (1) learning of cues across different languages, (2) a single model for many languages, and (3) avoiding dependency on English translation during inference. In this work, we start with a pre-trained multilingual Bidi-

Context	Question	Answer	Start Position	Language
ஒரு சாதாரண வளர்ந்த மனிதனுடைய எலும்புகள் (பின்வரும் 206 மார்பெலும்பு மூன்று பகுதிகளாகக் கருதப்பட்டால்) 206 எண்ணிக்கையான எலும்புகளைக் கொண்டிருக்கும் ...	மனித உடலில் எத்தனை எலும்புகள் உள்ளன?	206	53	Tamil
Translation				
A normal adult human skeleton consists of the following 206 (208 if the breast is thought to be three parts) ...	How many bones do you have in your body?	206	56	English
एक सामान्य वयस्क मानव कंकाल में निम्नलिखित 206 होते हैं (यदि स्तन को तीन भाग माना जाता है) ...	आपके शरीर में कितनी हड्डियां हैं?	206	43	Hindi
Transliteration				
Woru sadharan valarnt manidhani elumbukkoodu finwarum 206 (marbelumbu moondau bakudikalagak karudapattal 208) annikkaiyana elumbugalack kontrukkum...	Manit udalil atans elumbues ulsana?	206	54	English

Figure 1: Example of a QA record from the ChAII QA dataset along with the translation and transliteration done on that record.

rectional Encoder Representations from Transformers (mBERT) model and further pre-train it with SQuAD (Rajpurkar et al., 2016), a large-scale question answering dataset in English. The resulting English-language mBERT-QA model is fine-tuned and evaluated for Indian languages Tamil and Hindi using the ChAII dataset (Google, 2021).

Fine-tuning the mBERT-QA model using only the training instances in the ChAII dataset is less effective because of the small number of training samples (1114 records with approximately two-thirds in Hindi and the rest in Tamil). To overcome this problem, we use translation and transliteration to other languages as a data augmentation strategy. The translation is the process of transforming the source content from one language to another, while the transliteration just involves modifying each word from the source content into another script. Both these operations are executed on the training dataset for the contexts, questions, and answers separately; then new locations of transformed answers in the transformed contexts are computed as shown in Figure 1. Using translation and transliteration increases the size of the ChAII dataset manifold.

The choice of languages used for translation and transliteration is critical. Kudugunta et al. (2019) showed that languages under the same family have similar representations in multilingual models. Hence, we put together translations and transliterations from related languages within the same language family to achieve better performance. This will also help with better use of the vo-

cabulary corpora from the low-resource languages. We also study the impact of translation and transliteration on languages outside the family of the target language. Since the cross-family language transfer degraded the QA performance, we introduce a contrastive loss (Radford et al., 2021) between the translated pairs to help retain or improve the original performance by encouraging the embeddings from all languages to be similar regardless of the family group. Thus, the contributions of the paper are three-fold:

- We propose a three-stage training pipeline for question-answering in low-resource languages.
- We evaluate mBERT for question-answering in Tamil and Hindi with translations and transliterations as data augmentation techniques and show that same language family translations improve the performance. In contrast, we show that transliterations do not improve the QA performance on the ChAII dataset, regardless of the language family combinations.
- We propose a contrastive loss between the features of translated pairs to align the cross-family language representations.

2 Related Work

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) is a deep learning model for general-purpose language representations. BERT is often used as the backbone model for several NLP tasks like semantic analysis, question answering, and named entity recognition. The bidirectional transformer used in BERT has a deeper sense of language context and generates intricate semantic feature representations. These representations are learned through a pre-training step using Next Sentence Prediction (NSP) and Masked Language Modelling (MLM) as pretext tasks and transferred to the downstream NLP tasks. The goal of the Next Sentence Prediction task is to identify whether the two input sentences are consecutive or not. In Masked Language Modelling, BERT is trained to predict randomly masked words in a sentence. The Transformer network receives a sequence of tokens as input and utilizes the attention mechanism to learn the contextual relationships between words in a text. These relationships can then be used to extract high-quality

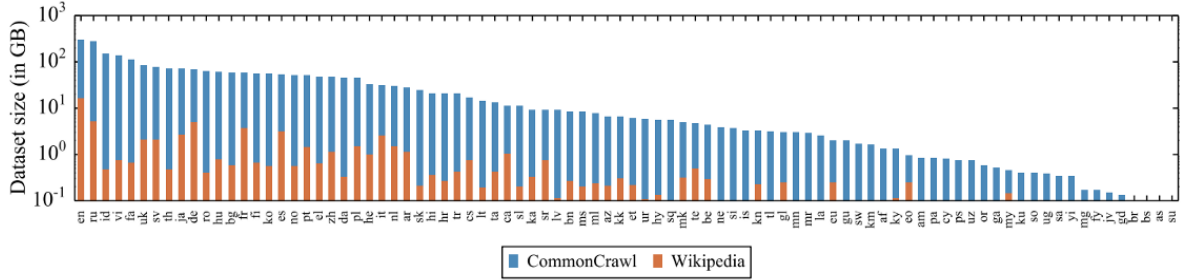


Figure 2: Amount of data in GB (log-scale) for the 88 languages that appear in both the Wiki-100 (Merity et al., 2016) corpus used for mBERT and XLM-100 (Conneau et al., 2020). None of the Indian languages feature among top-25 languages with the largest amount of data.

language features, which can be fine-tuned for applications like semantic analysis and question answering. Multi-lingual-BERT (mBERT) is a BERT model pre-trained using the Wikipedia text corpus (Merity et al., 2016) in more than 100 languages around the world. XLM-RoBERTa (Conneau et al., 2020) scaled this idea with more than 2 terabytes of common crawl data.

Deep models such as Transformers rely heavily on the availability of a large amount of annotated data, which is available only for prominent languages like English, Russian, German or Spanish (Ponti et al., 2019; Joshi et al., 2020). For a majority of other languages with a minimal number of annotations, cross-lingual transfer learning (Prettenhofer and Stein, 2011; Wan et al., 2011; Ruder et al., 2019) has been proposed as a possible solution. This approach can transfer knowledge from the annotation-rich source language to low-resource or zero-resource target languages. Furthermore, multilingual models (Lewis et al., 2019; Clark et al., 2020) can be used to mitigate the data scarcity problem. For example, LASER (Artetxe and Schwenk, 2019) used a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) encoder with a byte pair encoding vocabulary shared between languages. This work showed that joint training of multiple languages helped to improve the model performance for low-resource languages. LaBSE (Feng et al., 2020) used the mBERT (Devlin et al., 2018) encoder pre-trained with masked language modelling and translation language modelling (Lample and Conneau, 2019) tasks. It attempted to optimize the dual encoder translation ranking (Guo et al., 2018) loss during pre-training to achieve similar embedding for the same text in different languages.

The work of Bornea et al. (2020) showed that large pre-trained multilingual models are

not enough for question-answering in under-represented languages and presented several novel strategies to improve the performance of mBERT with translations. This work achieved language-independent embeddings, which improved the cross-lingual transfer performance with additional pre-training on adversarial tasks. It also introduced a Language Arbitration Framework (LAF), which consolidated the embedding representations across languages using properties of translation. Cross-lingual manifold mixup (X-Mixup) (Yang et al., 2021) achieved better cross-lingual transfer by calibrating the representation discrepancy, which resulted in a compromised representation for target languages. It was shown that the multilingual pre-training process can be improved by implementing X-Mixup on parallel data. Contrastive Language-Image pre-training (CLIP) (Radford et al., 2021) introduced an efficient way to learn scalable image representations with natural language supervision. Drawing inspiration from ConVIRT (Zhang et al., 2020), CLIP used a contrastive objective that maximizes the cosine similarity of the correct pairs of images and text, while minimizing the same for incorrect pairs.

Building upon the work of (Bornea et al., 2020), we show that translations of a small-scale dataset into cross-family languages could degrade the QA performance. To overcome this problem, we propose multilingual contrastive training to encourage cross-lingual invariance. Our approach is relatively simpler compared to adversarial training and LAF used in Bornea et al. (2020). Though the proposed contrastive loss has a similar objective to the pre-training loss in (Guo et al., 2018), there are subtle differences because we use it in multi-task learning setup along with the original task loss for fine-tuning.

3 Methodology

3.1 Data Representation and Baseline Model

We adopt the standard data representations that are commonly used in Transformer-based question-answering models. We use the same word-piece Tokenizer of mBERT to tokenize the concatenated input of question-context pairs. For the question answering task, the context is usually very long. In some NLP applications, truncating the input text is a viable choice because it leads to only loss of information. But in the extractive question answering task, removing part of the context may result in loss of answer as well. To overcome this challenge, we follow the popular approach of splitting the long context into parts that fit into the model and regulate this splitting using an additional hyper-parameter called 'max length'. Moreover, to cover for cases where the answer might be distributed over multiple splits of the context, an overlap factor is introduced, which in turn is controlled by another hyper-parameter 'doc stride'.

Our baseline is the mBERT model (Devlin et al., 2018), which is pre-trained using pretext tasks like Masked Language Modelling and Next Sentence Prediction on a multilingual text corpus that includes our target languages, Hindi and Tamil. The default output head of mBERT is replaced with the head for the question-answering task. This is done by adding separate output heads for classifying the start and end positions as shown in Devlin et al. (2018).

3.2 Proposed Framework for Effective Cross-lingual Transfer

We propose a three-stage pipeline called Multilingual Contrastive Training (MuCoT) to effectively train the mBERT model for question-answering in low-resource languages. An illustration of this pipeline for two low-resource languages, namely Tamil and Hindi, is shown in Figure 3. The first stage is pre-training the baseline multilingual model (mBERT). The second stage involves pre-training the QA head using the large-scale dataset(s) in high resource language(s). In Figure 3, English is considered the high-resource language and SQuAD (Rajpurkar et al., 2016) dataset is used to pre-train the QA head and obtain the mBERT-QA model. The final stage involves fine-tuning the mBERT-QA model using both original and augmented samples from the target low-resource languages. In this work, ChAII (Google, 2021) dataset

is used for obtaining training samples in Tamil and Hindi.

Since SQuAD (Rajpurkar et al., 2016) and ChAII (Google, 2021) datasets have similar Wikipedia¹ style contexts, it is possible to train a multilingual QA model jointly using both datasets. However, to take advantage of the engineering and training efforts of publicly available models, we sequentially use both these datasets. After obtaining the mBERT-QA model pre-trained for the English language QA task, we fine-tune it on the ChAII dataset using the following loss function.

$$L_{total} = L_{task} + w_{contrastive} * L_{contrastive}, \quad (1)$$

where L_{task} and $L_{contrastive}$ are the QA task loss and multilingual contrastive loss, respectively, L_{total} is the total loss, and $w_{contrastive}$ is the relative weight assigned to the contrastive loss. Note that fine-tuning using only the QA task loss is often not sufficient to achieve good performance, especially if the dataset used for fine-tuning is small. To mitigate this problem, we translate the training samples into other languages and use both original and translated samples for fine-tuning. While this approach works well for translations into other languages within the same language family, it leads to sub-optimal performance in the case of cross-family language translations, due to divergence in the representations across language families. To solve this issue, we introduce the multi-lingual contrastive loss $L_{contrastive}$.

3.3 Multilingual Contrastive Loss

During fine-tuning, for each data point in the original batch (B_o) of size n , we pick one of its corresponding translations uniformly at random and form a translated batch (B_p) of the same size n . It is important to note that B_o itself is taken from the combined dataset of source instances and translated instances. The two batches that form a pair are denoted as original batch and pair batch, respectively, in Figure 4. We use the same mBERT network up to a specific layer as our encoder (enc) to transform B_o and B_p to get the embeddings, $E_o, E_p \in \mathbb{R}^{n*t*d}$, respectively. Then, we apply a global average pooling (gap) operation to aggregate the vector representations of t tokens into a single vector representation of dimension d for each instance in each batch. This will result in the

¹<https://www.wikipedia.org/>

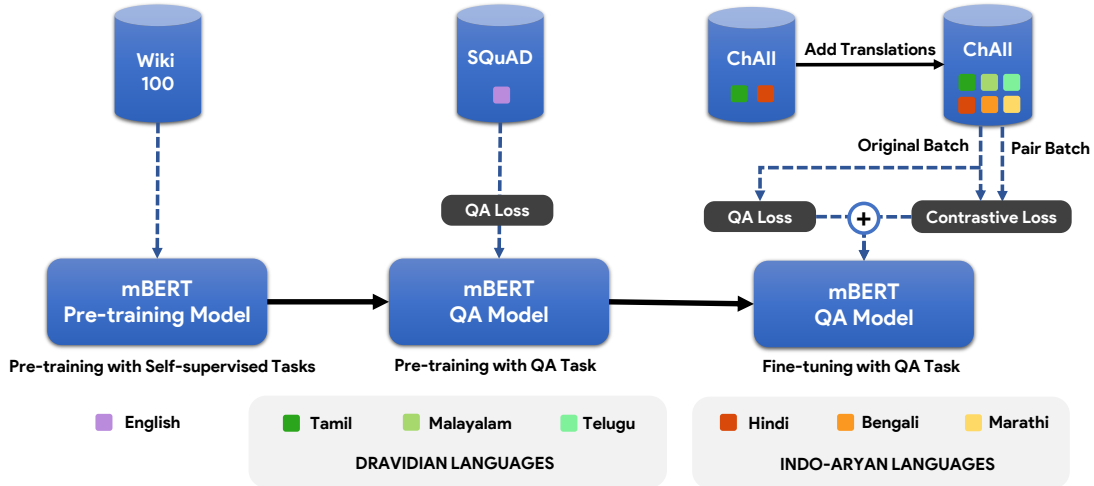


Figure 3: Proposed training pipeline of MuCoT for question answering in low resource languages Tamil and Hindi.

aggregated embeddings $O, P \in \mathbb{R}^{n \times d}$ for B_o and B_p , respectively. With these n feature vectors in the original and the translated batch, we follow the CLIP (Radford et al., 2021) approach and compute the contrastive loss using the cross-entropy loss (L_{ce}). Specifically, we multiply the matrices O and P^T to get the logits matrix $Q \in \mathbb{R}^{n \times n}$. Then, we apply the cross-entropy loss L_{ce} row-wise and column-wise to the logits matrix Q , with its diagonal locations as original classes for each row and column, respectively.

$$O = \text{gap}(\text{enc}(B_o)), \quad (2)$$

$$P = \text{gap}(\text{enc}(B_p)), \quad (3)$$

$$Q = OP^T, \quad (4)$$

$$L_{contrastive} = \frac{L_{ce}^{row}(Q) + L_{ce}^{column}(Q)}{2}. \quad (5)$$

4 Experimental Results

4.1 Datasets

In our experiments, we use ChAII (Google, 2021) question-answering dataset for fine-tuning and evaluation. This dataset was recently released by Google Research India and has 1,114 records of context, question, answer, and its corresponding start position in the context for Tamil and Hindi languages. Hindi is represented predominantly in the dataset with nearly two-thirds of the records. As the ChAII dataset has been published as part of an ongoing Kaggle competition (Google, 2021), the complete test dataset has not been disclosed to the public. Hence, we have used Scikit-learn’s

`train_test_split` method with a test size of 100, stratified on language and with a random seed of 0, to get the *test* split from the training data. Similarly, we applied the same method over the filtered *train* split to get the *validation* split of 100 samples. We also use the translations and transliterations of this training split as augmented samples for fine-tuning the QA model.

Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) is the most popular question-answering dataset in English. This dataset had been crowdsourced to form 100K records of answerable question-answer pairs along with the context. This dataset is used to pre-train the QA head added to the pre-trained mBERT model, which is subsequently fine-tuned using the ChAII dataset.

4.2 Translation and Transliteration Details

We use AI4Bharat’s IndicTrans² (Ramesh et al., 2021) for translation, which is a Transformer-4X model trained on *Samanantar* dataset (Ramesh et al., 2021). In IndicTrans, translation can be done from Indian languages to English and vice versa. Available Indian languages include Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, and Telugu. At first, we translate the ChAII dataset from Hindi and Tamil to English and then to Bengali, Marathi, Malayalam, and Telugu. In the FLORES devset benchmark (Goyal et al., 2021), the BLEU scores of IndicTrans for translating Hindi and Tamil to English are 37.9 and 28.6, respectively. The scores

²<https://indicnlp.ai4bharat.org/indic-trans/>

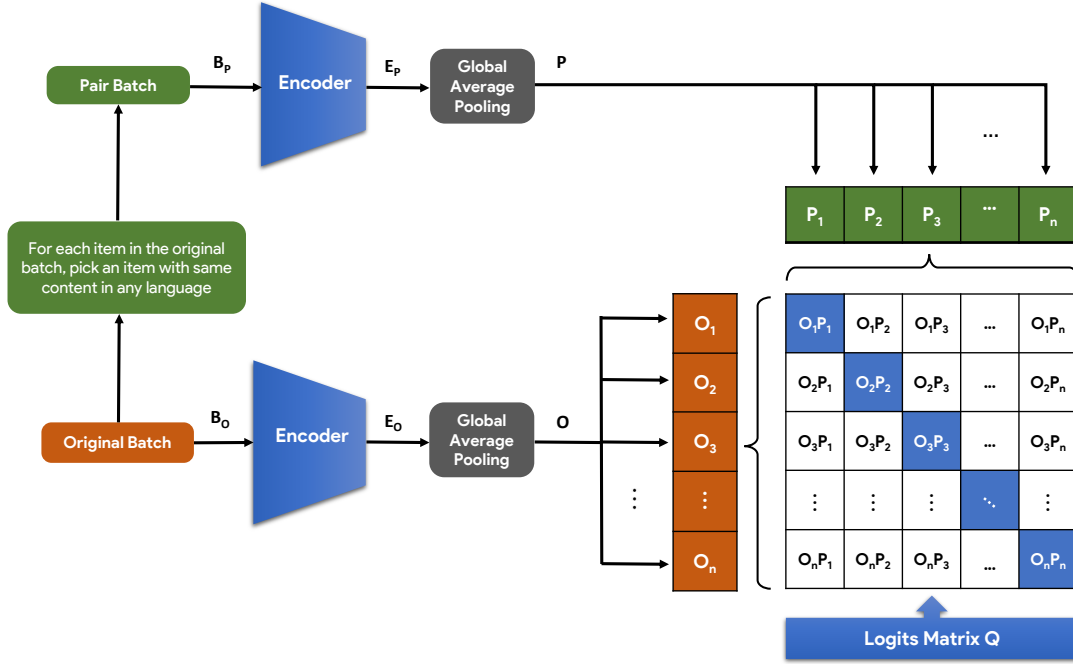


Figure 4: Logits matrix computation for the input to contrastive loss, similar to CLIP (Radford et al., 2021)

for translating English to Bengali, Marathi, Malayalam, and Telugu are 20.3, 16.1, 16.3, and 22.0, respectively. We were not able to translate nearly 500 of the ChAII instances to English as the automatic search for the translated answers in the translated contexts failed. This happened because the same word got translated differently in the context and the answer. For the same reason, we lost nearly another 200 instances when translating from English to other Indian languages.

For transliteration, we use the open-source Indic-trans transliteration module³ (Bhat et al., 2015), which is available for many Indian language scripts including English and Urdu. Here, we directly transliterate from Hindi and Tamil to Bengali, Marathi, Malayalam, and Telugu.

4.3 Model Training Details

We used mBERT⁴ as our baseline model. It is modified for the question-answering task by replacing the output head using HuggingFace’s auto model. At first, we evaluated this model after directly fine-tuning on the train split of the ChAII dataset. Then, we introduced intermediate SQuAD pre-training and fine-tuned on the train split of the ChAII dataset with and without translations or transliterations.

³<https://indic-trans.readthedocs.io/en/latest/index.html>

⁴<https://huggingface.co/bert-base-multilingual-cased>

The hyperparameter settings listed in Table 1 are used for all the experiments. We have experimented with different levels of mBERT layers to compute the contrastive loss. Layer 3 performed consistently well compared to the initial layer 1 and the deeper layers such as 5. Initially, we used contrastive training for all the steps. However, forcing the model to learn exact representations across languages could make the model forget the task-specific patterns learned with intermediate pre-training on a large-scale dataset. Hence, we applied the contrastive loss only for training steps that are a multiple of 500 and picked the best one. Other hyperparameters are tuned based on a standard search over multiple choices.

Hyperparameter	Value
Maximum feature length	128
Document stride	384
Batch size	16
Maximum optimization steps	5000
Learning rate	0.00003
Weight decay	0.01
Contrastive loss layer	3
Contrastive loss weight	0.05
Maximum contrastive steps	1000

Table 1: Hyperparameter configuration of all the models for fine-tuning on ChAII dataset

SQuAD pre-training	No	Yes	Yes		Yes		Yes	
Translations	No	No	Dravidian (ml, te)		Indo-Aryan (bn, mr)		All languages	
Contrastive Training	No	No	No	Yes	No	Yes	No	Yes
Overall	0.44	0.5	0.49	0.53	0.51	0.52	0.49	0.52
Hindi	0.47	0.57	0.52	0.57	0.59	0.58	0.54	0.57
Tamil	0.39	0.37	0.44	0.45	0.35	0.4	0.39	0.41

Table 2: Jaccard scores with translation used as augmentation in different training settings. ml, te, bn, and mr denote Malayalam, Telugu, Bengali, and Marathi, respectively.

SQuAD pre-training	No	Yes	Yes		Yes		Yes	
Transliterations	No	No	Dravidian (ml, te)		Indo-Aryan (bn, mr)		All languages	
Contrastive Training	No	No	No	Yes	No	Yes	No	Yes
Overall	0.44	0.5	0.5	0.49	0.53	0.47	0.49	0.46
Hindi	0.47	0.57	0.52	0.55	0.56	0.53	0.52	0.53
Tamil	0.39	0.37	0.45	0.36	0.44	0.36	0.44	0.32

Table 3: Jaccard scores with transliteration used as augmentation in different training settings. ml, te, bn, and mr denote Malayalam, Telugu, Bengali, and Marathi, respectively.

4.4 Evaluation Metric

Given the noisy nature of the ChAII dataset, we employed the Jaccard score as the evaluation metric. Jaccard similarity coefficient is widely used for determining similarity between sets/intervals and is defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Here, A and B are sets/intervals, and \cap and \cup represent intersection and union, respectively. We compute the evaluation metric for the overall test split as well as for individual language test sets in intervals of 500 optimization steps. For each experiment, we pick the model at a specific optimization step that gives the best overall Jaccard score and reports its performance.

4.5 Performance

As shown in Tables 2 and 3, translation and transliteration affect the performance in different ways. While some data is lost during the translation process due to failed automatic search of translated text in the translated context, transliteration does not cause any such loss. However, to ensure a fair comparison, records lost during translation are dropped from transliterated testing as well. Note that we use the same hyper-parameters from Table 1 for evaluating the models and later stages with additional augmentation and contrastive training.

First, we observe from Table 2 that just having intermediate SQuAD pre-training in English, improves the overall Jaccard score significantly from 0.44 to 0.5. Furthermore, we fine-tune by dividing

translated and transliterated data into Indo-Aryan and Dravidian language families to study how translated and transliterated pairs serve as supervised cross-lingual signals when languages share semantics and structure (Mikolov et al., 2013). Although transliteration improves the Jaccard scores in certain cases compared to the baseline, the trend is not consistent. Moreover, contrastive training does not help in the case of transliteration as shown in Table 3. This could be because the QA model is pre-trained only with regular text and not with transliteration style text.

From Table 2, we observe that grouped translated data in the same language family helps in improving performance. The translated Indo-Aryan data (Bengali and Marathi) increases the Jaccard score of Hindi answers to 0.59 from 0.57. Similarly, Dravidian language data (Telugu and Malayalam) significantly increase the Jaccard similarity of Tamil answers from 0.37 to 0.44. At the same time, the overall Jaccard score did not change much because of the degradation in cross-family language performance. Interestingly, we could observe in Table 2 that the contrastive training helps in preventing such degradation and improves the overall score by encouraging similar representations between languages from across families.

5 Conclusion and Future Work

With Internet usage expanding every day, there is an increasing need to develop better NLP models for a variety of downstream tasks in vernacular lan-

guages. As most of these languages do not have labeled resources that are sufficient to train stand-alone modern deep learning models, we need to rely on multilingual models and enhance them. Our work is a step in this direction and is an attempt to understand and evaluate the impact of cross-lingual knowledge transfer through pre-training and fine-tuning. We utilize modern open-source deep learning models to translate the ChAII dataset into different languages from two language families namely, Dravidian, and Indo-Aryan, and use them to improve the question-answering performance. Our analysis shows an effective way to pick languages for translation, which can be used for fine-tuning. We also showed that introducing a contrastive loss with the original task training loss increases the performance even for cross-family languages.

Despite the inclusion of translations and contrastive loss, we observed that there is only a marginal improvement in the QA performance. This can be attributed to the smaller size of the ChAII dataset with 1114 instances (Tamil and Hindi combined; Train, Validation, and Test combined), which is clearly insufficient to fine-tune a 177M parameter model. Hence, the proposed techniques have to be evaluated on other larger datasets as well as using other multilingual models like XLM-RoBERTa (Conneau et al., 2020), DistillmBERT (Sanh et al., 2019), MURIL (Khanuja et al., 2021) and Indic-BERT (Kakwani et al., 2020). We hope that the proposed techniques will motivate further research in this field, including exploration of the same phenomenon of cross-lingual transfer in other language families and multilingual tasks.

References

- R Anita and CN Subalalitha. 2019a. An approach to cluster Tamil literatures using discourse connectives. In *2019 IEEE 1st International Conference on Energy, Systems and Information Processing (ICESIP)*, pages 1–4. IEEE.
- R Anita and CN Subalalitha. 2019b. Building discourse parser for Thirukkural. In *Proceedings of the 16th International Conference on Natural Language Processing*, pages 18–25.
- Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.
- B Bharathi, Bharathi Raja Chakravarthi, Subalalitha Chinnudayar Navaneethakrishnan, N Sripriya, Arunagiri Pandian, and Swetha Valli. 2022. Findings of the shared task on Speech Recognition for Vulnerable Individuals in Tamil. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.
- Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tamemwar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. *Iit-h system submission for fire2014 shared task on transliterated search*. In *Proceedings of the Forum for Information Retrieval Evaluation, FIRE '14*, pages 48–53, New York, NY, USA. ACM.
- Mihaela Bornea, Lin Pan, Sara Rosenthal, Radu Florian, and Avirup Sil. 2020. Multilingual transfer learning for qa using translation as data augmentation. *arXiv preprint arXiv:2012.05958*.
- Bharathi Raja Chakravarthi. 2020. *HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion*. In *Proceedings of the Third Workshop on Computational Modeling of People’s Opinions, Personality, and Emotion’s in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.
- Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.
- 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.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. 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.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.
- Google. 2021. ChAII - Hindi and Tamil question answering. <https://www.kaggle.com/c/cha-ii-hindi-and-tamil-question-answering> [Accessed: 01-Oct-2021].

- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.
- Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. [Effective parallel corpus mining using bilingual sentence embeddings](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Brussels, Belgium. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, N C Gokul, Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961.
- Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha P. Talukdar. 2021. [Muril: Multilingual representations for indian languages](#). *arXiv preprint arXiv:2103.10730*.
- Sneha Reddy Kudugunta, Ankur Bapna, Isaac Caswell, Naveen Arivazhagan, and Orhan Firat. 2019. Investigating multilingual nmt representations at scale. *arXiv preprint arXiv:1909.02197*.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.
- Patrick Lewis, Barlas Öğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. [Mlqa: Evaluating cross-lingual extractive question answering](#). *arXiv preprint arXiv:1910.07475*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. [Pointer sentinel mixture models](#). *arXiv preprint arXiv:1609.07843*.
- Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.
- Edoardo Maria Ponti, Helen O’horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Computational Linguistics*, 45(3):559–601.
- Peter Prettenhofer and Benno Stein. 2011. Cross-lingual adaptation using structural correspondence learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):1–22.
- Ruba Priyadharshini, Bharathi Raja Chakravarthi, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, and Rahul Ponnusamy. 2021. Overview of the DravidianCodeMix 2021 shared task on sentiment detection in Tamil, Malayalam, and Kannada. In *Forum for Information Retrieval Evaluation*, pages 4–6.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.
- Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2021. [Samanantar: The largest publicly available parallel corpora collection for 11 indic languages](#). *arXiv preprint arXiv:2104.05596*.
- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.
- Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2016. [A novel hybrid approach to detect and correct spelling in Tamil text](#). In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–6.
- Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2017. [Use of a novel hash-table for speeding-up suggestions for misspelt tamil words](#). In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–5.
- Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2021. [Missing word detection and correction based on context of Tamil sentences using n-grams](#). In *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 42–47.

- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- C. N. Subalalitha. 2019. [Information extraction framework for Kurunthogai](#). *Sādhanā*, 44(7):156.
- CN Subalalitha and E Poovammal. 2018. Automatic bilingual dictionary construction for Tirukural. *Applied Artificial Intelligence*, 32(6):558–567.
- Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. [Sentiment analysis in Tamil texts: A study on machine learning techniques and feature representation](#). In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.
- Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. [Sentiment lexicon expansion using Word2vec and fastText for sentiment prediction in Tamil texts](#). In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.
- Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. [Word embedding-based part of speech tagging in Tamil texts](#). In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.
- Sajeetha Thavareesan and Sinnathamby Mahesan. 2021. [Sentiment analysis in Tamil texts using k-means and k-nearest neighbour](#). In *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 48–53.
- Chang Wan, Rong Pan, and Jiefei Li. 2011. Bi-weighting domain adaptation for cross-language text classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2021. Enhancing cross-lingual transfer by manifold mixup. In *International Conference on Learning Representations*.
- Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. 2020. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*.