Multilingual Question Answering (MQA)

Group Name – BitsNBytes Subtask ID – 03
Sayan Mahapatra (21CS60R12), Arkapravo Ghosh (21CS60R64), Anima Prasad (21CS60R66)

Abstract

In this project we have experimented with multilingual models (mBERT and XLM-R) and tried to solve the Multilingual Question Answering NLP Task. In this task we are given a passage and a question and we are required to find an answer from the given passage based on the question. For this task we have used multilingual pretrained models from Hugging Face and then finetuned the models for our specific task. The models are trained with 9 languages in total and then we used Bengali and Telugu as the validation set to evaluate the models. We also experimented with training the model only on the English, Telugu and Bengali language and then evaluating on the same Bengali and Telugu evaluation dataset as used earlier.

1 Subtask ID + Group Details (Names, Roll Numbers, Group Name)

- Subtask ID – 03
- Group Name – BitsNBytes
- Members – Sayan Mahapatra, Arkapravo Ghosh, Anima Prasad

2 Individual Contributions of Students

- Sayan Mahapatra – worked on task1 mBERT model and task2 XLM model and data augmentation. Explored the IndicBERT experiment to find if promising results could be obtained. Explored the AI4Bharat dataset and formatted the dataset to match with the existing SQUAD format for data augmentation purposes. Also worked on the report for the final project.
- Arkapravo Ghosh – worked on task1 XLM model and task2 (mBERT + Dutch) model and data augmentation. Explored the AI4Bharat dataset and also tried to choose the subset of data with promising f1-score. Also worked on the report for the final project.
- Anima Prasad – worked on task2 mBERT model and task1 (mBERT + Dutch) model and data augmentation. Explored the AI4Bharat dataset and augmented the chosen subset of data with the squad dataset for self-training the model. Also worked on the report for the final project.

3 Task Description

Multilingual Question Answering is considered one of the more challenging NLP Tasks. Given a context (passage), and a question the task is to extract out the answer to the question from the context.

The Stanford Question Answering Dataset (SQuAD) is benchmarks dataset. In this project we experiment on a multilingual version of the dataset, TyDi QA [1] dataset. mBERT and XLM were used to obtain baseline performance on this dataset and then data augmentation was considered as the next step for improving over the baseline performance. We also tried out two other models – IndicBERT [2], mBERT Multilingual + Dutch Model[6] without the use of translation (unlike MLQA and XQuAD).

4 Approach / Model Architectures

The following two models were used

- BERT multilingual base model (cased) (referred to as mBERT) [3]
- XLM-RoBERTa [4]

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based machine learning technique for natural language

processing pre-training developed by Google. When it was proposed it achieve state-of-the-art accuracy on many NLP and NLU tasks such as text summarization, text classification, semantic similarity, question answering. BERT model is only for English dataset.

MBERT stands for multilingual BERT is a deep learning model that was trained on 104 languages simultaneously and encodes the knowledge of all 104 languages together. It is able to understand the meaning of words in context. MBERT's uses are comparable to BERT's, and it can work with a wide range of languages.

RoBERTa stands for Robustly Optimized BERT Pre-training Approach. The goal of this model was to optimize the training of BERT architecture in order to take lesser time during pre-training. XLM-RoBERTa is a multilingual version of RoBERTa. It is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages. RoBERTa is a transformers model pretrained on a large corpus in a self-supervised fashion.

IndicBERT is a multilingual ALBERT model trained on large-scale corpora, covering 12 major Indian languages. It has much less parameters than other public models like mBERT and XLM-R while it still manages to give state of the art performance on several tasks.
mBERT multilingual + Dutch model is the multilingual model provided by the Google research team with a fine-tuned dutch Q&A downstream task.

TyDi QA is a question answering dataset covering 11 typologically diverse languages with 204K question-answer pairs. The data is collected directly in each language without the use of translation. The languages of TyDi QA are diverse with regard to their typology, the set of linguistic features that each language expresses such that the models are expected to perform well on this set to generalize across a large number of languages.

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.
The AI4Bharat-IndicNLP dataset is an ongoing effort to create a collection of large-scale, general-domain corpora for Indian languages. Currently, it contains 2.7 billion words for 10 Indian languages from two language families.

We tried various approaches. Firstly, we used the whole Tydi-QA gold passage dataset for training and evaluated Tydi-QA dataset dev data (only Telugu and Bengali) on the models. This is done as part of Task 1.

Secondly, as part of task 2 we extracted English, Telugu, and Bengali specific data from the Tydi-QA gold passage train dataset to train the base models, and then evaluated the models on Tydi-QA dataset dev data (only Telugu and Bengali).

We also tried to implement Task1 and Task2 on IndicBERT and mBERT + Dutch models. However, the performance of IndicBERT was not promising so the path was not explored further. Moreover, the performance of mBERT + Dutch model was similar to that of mBERT model. The performance metrics for the models are added below.

These were used to set the baseline performance. After this we tried Data Augmentation approach to improve the performance over the baseline models.

As part of Data Augmentation, we used additional two datasets AI4Bharat Indic Question Answering dataset [5], SQuAD[7]. The SQuAD dataset has the same format as the TyDi QA dataset, although this was not the case with the AI4Bharat dataset. As a result, data preprocessing was performed on the AI4Bharat dataset in order to match the format of TyDi QA. In AI4Bharat Indic Question Answering dataset, only Bengali & Telegu specific data are used. In SQuAD v1 dataset English data is used for data augmentation.

Once the datasets AI4Bharat and SQuAD are in same format, our above baseline mBERT model is used to predict on this augmented dataset to filter out those data which satisfy the threshold f1_score criteria.
Then this filtered data is merged with the TyDi-QA dataset for self-training. Then, the new augmented

dataset is used to train the mBERT base multilingual model. After this, the new trained model is used to evaluate on the Tydi QA validation dataset (only Bengali and Telugu).

5 Code structure

The structure consists of 7 scripts IndicQuestionAnswering.py, run_qa.py, evaluate_qa.py, subset_qa.py, train_qa.py, trainer_qa.py, utils_qa.py. IndicQuestionAnswering.py script contains code for converting AI4Bharat dataset to SQuAD format. Conversion to SQuAD format is necessary since the main dataset TyDi-QA is also in SQuAD format. run_qa.py script contains code to train mBERT model on the entire Tydi QA dataset. This trained model is uploaded in hugging face. evaluate_qa.py script contains code to evaluate a trained mBERT model (downloaded from HF) on the Tydi-QA dataset. subset_qa.py script contains code to find subsets of data from a source dataset (SQuAD, AI4Bharat) which have at least 60% F1-score. The subset dataset obtained is augmented with TyDi-QA data and finally uploaded to HuggingFace (HF) so that in later scripts it can be fetched from HF. train_qa.py script contains code to run training and validation on the above augmented data which is downloaded from HuggingFace. It also writes the F1-scores for each data instance in the Tydi-QA validation set to the file "evaluation.csv". trainer_qa.py script is a helper file used train, evaluate and predict the loaded model on the input dataset. utils_qa.py script contains the post processing the predictions of the question answering model. This helps to find the answers that are substrings of the original context.

The workflow of the script is as follows –

- Run “run_qa.py” script to train the mBERT base model on Tydi QA dataset
- Run “evaluate_qa.py” to obtain validation performance of above trained model on validation set of Tydi QA dataset
- Run “subset_qa.py” to obtain subsets from AI4Bharat and SQuAD datasets and upload them to HuggingFace
- Run “train_qa.py” to train the mBERT base model on Tydi QA + subset of AI4Bharat and SQuAD dataset. Then this trained model is evaluated on the validation set of Tydi QA dataset (only Bengali and Telugu).
- The above steps are followed for multiple epochs.

Command to execute the scripts:

- Train mBERT model for 3 epochs

```
python run_qa.py \
  --model_name_or_path bert-base-multilingual-cased \
  --dataset_name tydiqa \
  --dataset_config_name secondary_task \
  --do_train \
  --do_eval \
  --per_device_train_batch_size 12 \
  --learning_rate 3e-5 \
  --num_train_epochs 3 \
  --max_seq_length 384 \
  --doc_stride 128 \
  --output_dir train_epoch_3
```

- Download the model files and upload them to HuggingFace.
- Evaluate above model on Validation Data

```
python evaluate_qa.py \
  --model_name_or_path <model trained in step 1> \
  --dataset_name tydiqa \
  --dataset_config_name secondary_task \
  --do_eval \
  --per_device_train_batch_size 12 \
  --learning_rate 3e-5 \
  --num_train_epochs 1 \
  --max_seq_length 384 \
  --doc_stride 128 \
  --output_dir evaluate
```

- Prediction on Squad, AIBharat (Bengali and Telugu) dataset and choosing a subset for self-training

```
python subset_qa.py \
  --model_name_or_path <model trained in step 1> \
  --dataset_name augment_data \
  --do_predict \
  --per_device_train_batch_size 12 \
  --learning_rate 3e-5 \
  --num_train_epochs 1 \
  --max_seq_length 384 \
  --doc_stride 128 \
  --save_steps 6000 \
```

--overwrite_output_dir \
--output_dir subset

- Append the selected data and train the model on the new dataset and evaluate on the dev set

python train_qa.py \
--model_name_or_path bert-base-multilingual-cased \
--dataset_name horsbug98/squad_ai4bharat_ben_tel_train \
--do_train \
--do_eval \
--per_device_train_batch_size 12 \
--learning_rate 3e-5 \`
--num_train_epochs 3 \
--max_seq_length 384 \
--doc_stride 128 \
--save_steps 6000 \
--output_dir augment_train_3

Drive link to the baseline models:

Task1-mBERT-Epoch1: https://huggingface.co/horsbug98/Part_1_mBERT_Model_E1

Task1-mBERT-Epoch2: https://huggingface.co/horsbug98/Part_1_mBERT_Model_E2

Task1-XLM-Epoch1: https://huggingface.co/horsbug98/Part_1_XLM_Model_E1

Task2 – mBERT- Epoch 1 : https://huggingface.co/horsbug98/Part_2_mBERT_Model_E1

Task2 – mBERT- Epoch 2 : https://huggingface.co/horsbug98/Part_2_mBERT_Model_E2

Task2 – XLM- Epoch 1: https://huggingface.co/horsbug98/Part_2_XLM_Model_E1

6 Metrics used

The metric used for the performance measurement is F1-score and Exact Match. F1 score is defined as the harmonic mean between precision and recall. It is used as a statistical measure to rate performance. Exact match measures the percentage of predictions that match any one of the ground truth answers exactly.

7 Experiments

We wanted to investigate another multilingual model IndicBERT. Preliminary experiments showed that the model was not performing well for Question Answering task, hence this model was not explored further. We also experimented This model is also considered in self training

8 Results / Discussions

The figure below shows our baseline results. mBERT (trained for 2 epochs) was the best performing model

Results of baselines:

Parts	mBERT		XLM		mBERT Multilingual + Dutch Model	
	Epoch 1	Epoch 2	Epoch 1	Epoch 2	Epoch 1	Epoch 2
Part 1	80.9664	82.2277	81.5198	NA	79.4113	81.0626
Part 2	78.7635	80.8313	77.8194	81.3484	78.8825	80.579

Across all runs our F1 scores improved Epoch over Epoch. NA entries in the table above were for runs which failed due to hardware limitations (we used Kaggle Notebooks)

After data augmentation, validation Accuracy for task1 mBERT model (trained for 2 epochs) improved from 82.2277 to 82.7354. The validation accuracy for model (trained for 2 epochs) evaluated only on Bengali specific Tydi QA dataset changed from 74.842 to 73.1425. The validation accuracy for model (trained for 2 epochs) evaluated only on Telugu specific Tydi QA dataset changed from 83.4752 to 84.3558. We expect that the accuracy would improve further if training is done for more epochs.

9 Difficulty Faced

Data preprocessing, finding good data splits, hardware limitations and processing individual data points to choose subset of a train dataset were the chief difficulties we faced.

Acknowledgments

We would like to thank our course instructor Prof. Pawan Goyal for giving us this project from where we could learn a lot about the multilingual question answering task. We would also like to thank our TA Aniruddha Roy for his immense help and guidance in this project.

References

1. https://github.com/google-research-datasets/tydiqa
2. https://huggingface.co/ai4bharat/indic-bert
3. https://huggingface.co/bert-base-multilingual-cased
4. https://huggingface.co/xlm-roberta-base
5. https://huggingface.co/datasets/ai4bharat/IndicQuestionGeneration/tree/main/data
6. https://huggingface.co/henryk/bert-base-multilingual-cased-finetuned-dutch-squad2
7. https://huggingface.co/datasets/squad