Tidy analysis of TyDi: Analysing knowledge sharing in Multilingual domain

Stanford CS224N Custom Project (**Graded**)

Akshat Jindal
akshatj@stanford.edu

Chetanya Rastogi
chetanya@stanford.edu

Abstract

Recent trends in NLP have shown remarkable success across diverse set of tasks by virtue of pre-training transformer-based architectures [1] across a large corpora. One such architecture, multilingual-BERT (mBERT), has been shown to exhibit surprising cross-lingual abilities even in the absence of any cross-lingual training objective or aligned data. In this project, we work with a new cross-lingual QA dataset (TyDiQA-GoldP [2]) with an aim to investigate the cross-lingual transferability of mBERT by performing various experiments and ablation studies. Our experiments not only reveal excellent cross-lingual abilities of mBERT via zero-shot experiments but also discover that ensembling, gradient accumulation, and task specific knowledge transfer from a different cross-lingual dataset improves mBERT's performance drastically and resulted in a **3.2 points** of absolute improvement on F1 score over baseline. We realise that mBERT's learning ability can be decoupled into *task specific* and *language specific* representations, allowing us to replace the need for expensive low-resource language data with easily available datasets in high-resource languages. Finally, we also explore qualitative improvements brought about by our methods for some languages for a more thorough treatment of our experiments.

1 Key Information to include

- Mentor: Matthew Lamm

2 Introduction

Question answering is a benchmark task in NLP and is considered to be a proxy for a machine's ability to comprehend and reason about a piece a text by answering associated questions. In the last few years, the NLP community has made huge progress in this field both because of the presence of large datasets (such as SQuAD [3], Natural Questions [4], TriviaQA [5] to name a few) as well as recent advancements in transfer learning [6, 7, 8]. But most of the work have been done only for 'English' and hence models developed can only serve English-speakers, a small subset, when deployed in the real world. To enable researchers and practitioners to build impactful solutions in their domains, understanding how our current NLP architectures fare across languages needs to be more than an afterthought.

To overcome this problem, researchers have started to bridge this cross-lingual gap by focusing on developing multi-lingual datasets (such as XQuAD [9], MLQA [10], XQA [11]) to accelerate research in this domain. Needless to say, practitioners have started to pick up the trend by leveraging the power of pre-training a BERT model [8] on a multilingual corpus to generate cross-lingual contextual embeddings (called multilingual-BERT or mBERT). It is interesting that though mBERT had no cross-lingual training objective and neither was it trained on a parallel corpus, it has been shown to generalize well across languages for a variety of downstream tasks [12].

In our work, we focus on further exploring the effectiveness on mBERT on a novel multilingual QA dataset called TyDi QA [2]. We explicitly focus on the gold passage task of this dataset (called TyDiQA-GoldP) which is styled in the same manner as SQuAD v1.1. More specifically, given a context paragraph and a question, the task is to predict the minimal span containing the answer to that question, with the guarantee that the question is answerable given the context. We explore a variety of propositions and perform comprehensive ablation studies to identify the reason behind mBERT's success. Our results suggest that the effectiveness of mBERT's learning can be attributed to two components: *language specific representations* which require a few examples in each of the languages on which the model needs to be tested, and *task specific representations* which can be learned by leveraging a dataset in some high-resource language, thus, reducing the costs and efforts of collecting data in multiple low-resource languages.

3 Related Work

3.1 mBERT: BERT's multilingual sibling

mBERT [8] is a transformer-based [1] model which is pre-trained in the same way as monolingual BERT except that its training corpus consists of Wikipedia text from the top 104 languages. To account for imbalance in the size of Wikipedia pertaining to different languages, some languages were sub-sampled, and some were over-sampled using exponential smoothing. This allows the model to learn deep representations of the subword units [13] across multiple languages in a shared embedding space. It has been shown that the hidden representations learned by mBERT represents useful linguistic information in a language-agnostic way [14]. Similarly, [12] show mBERT's capability to successfully transfer knowledge in a cross-lingual zero-shot setting across a variety of NLP tasks establishing its effectiveness in a multilingual domain.

3.2 (Multilingual) Question Answering

Question Answering (QA) as a task is a classical probe for language understanding. While there exist other multilingual datasets that benchmark models on other NLI tasks[15], it has been shown that QA is less susceptible to annotation artifacts commonly found in other benchmarks [16]. Following this path, researchers have come up with novel datasets and methods to tackle the task of multilingual QA [11, 9, 10]. Our work, though inspired by them, is distinct as we will be evaluating our model on an entirely new dataset [2] which is more challenging and harder to solve then the existing datasets in the following ways:

- The questions in TyDi were written to *seek information*, i.e, the annotators didn't know the context (and hence the answer) while writing the question. This results in question-answer pairs that avoid typical artifacts of QA such as high lexical overlap, which can be exploited by systems to artificially inflate task performance achieving superhuman excellence [17, 18]
- All the questions in TyDi are created by native language-speakers without using any sort of human or machine translation. This makes the data more genuine and real as the translation process tends to introduce problematic artifacts to the output language such as preserving source-language word order or the use of more constrained language by translators which might make the translated text different from purely native text.

The above steps greatly mitigate the risk of sampling bias and hence makes the task more challenging than the previous counterparts.

3.3 Distillation and Ensembling

In the current deep learning paradigm, using more compute typically leads to higher model accuracy [19]. Ensembling is one such way wherein multiple models are trained, and during inference time their outputs are combined in some manner to produce a final prediction for a given input instance [20]. The gain from ensembling is evident from the fact that in many deep learning tasks the top performing models are all ensembles [17, 18].

To make room for the extra compute required for ensembling, we make a trade-off in the size of a single model by using a compressed version of mBERT called distilmBERT [21]. The distilmBERT model is obtained by using a model compression technique called knowledge distillation [22] wherein a compact model (called a student model) is trained to mimic the behaviour of a larger model (called the teacher model) by capturing the inductive biases learned by the teacher. In this case distilmBERT is the student model which is trained to reproduce the behaviour of mBERT which is the teacher model. The details about its architecture and the training methodology can be found in [21].

4 Approach

In this section, we describe our models and training approaches. The core architectures we use are the mBERT (Multilingual BERT) [8] and disitlmBERT (Distilled Multilingual BERT) models [21] augmented with a

QA head as implemented by HuggingFace in their transformers library [23]. The model uses two vectors $S, E \in \mathbb{R}^H$ for Start and End respectively where H is the size of output vectors for each token. Let $T_i \in \mathbb{R}^H$ be the final hidden vector from BERT of the ith token, the probability of each word $i \in \{start, end\}$ is computed as follows:

$$P_i = \frac{e^{KT_i}}{\sum_j e^{KT_j}} \quad | \quad K \in \{S, E\}$$

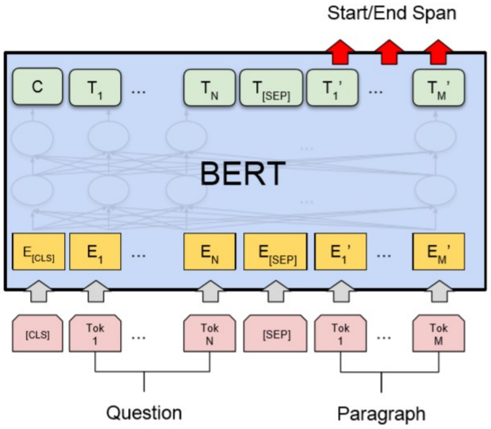

Figure 1: Fine-Tuning BERT on a Question Answering task [8]

We first describe the baseline proposed by the authors of TyDi[2] which we reproduce and then describe the various experiments we run backed by hypotheses that we wanted to test. **We integrate our code changes with the *run_squad.py* script of the transformers library [23] to run all of our experiments in an end-to-end fashion**. We refer the reader to section 5.3 for a thorough treatment of the code setup.

4.1 Baseline

[2] explains two kinds of baselines for the gold passage task. The first baseline is obtained by fine-tuning mBERT on the training samples from all languages and then evaluating on the dev set while the second baseline is obtained by fine-tuning the mBERT model on SQuAD1.1 [3] dataset and evaluating the dev set following the XQuAD zero-shot setting. We reproduced both these baselines, using the hyperparameters as suggested by the authors and treat them as benchmarks for the rest of our experiments.

4.2 Main Techniques

In this section, we describe the various augmentations we do while training a model(s), the hypothesis behind going for the augmentation and the effect it had on performance in the same order as they were performed to give the reader the same sense of progression from the baseline to our final model as we did. The experimental details and results for these are expounded upon in later sections. Our experiments were driven by the following goals :

- Robustly testing the cross-lingual capability of mBERT.
- Incorporating neat techniques prevalent across the NLP community to improve performance on TyDiQA-GoldP task [2].
- Finding ways to reduce the need for expensive annotated data in low resource languages by easily available, cheap resources for high-resource languages.

4.2.1 Zero-Shot Models

After reproducing the baselines, our first approach was to test mBERT's ability to transfer knowledge across languages. For this we trained our QA model on all but one languages in the training dataset and measured the performance on the left out language during evaluation. This was repeated for each of the 9 languages in the dataset.

4.2.2 Ensembling

The zero-shot experiments revealed an interesting byproduct. It was observed that leaving out some languages during training helped **improve** the performance of other languages (that were not left out)! Thus, to fully utilise these zero-shot models, and taking inspiration from [20], we created an ensemble model which averaged the start and end logits predictions obtained from **all the zero-shot models.**

4.2.3 Distillation

To make room for the extra compute required for ensembling, we deacided to make a trade-off in the size of a single model by using a compressed version of mBERT called distilmBERT [21]. With larger and larger models coming up, we felt exploring model compression is always worth the time! Se we performed zero-shot and ensembling experiments with disitlmBERT to see if it might be possible to achieve near similar results as with mBERT.

4.2.4 Pre-finetuning

Using transfer learning for improving performance on QA has been successfully tried by the NLP community. We see various forms of leveraging knowledge from training on a different dataset with the same task to push the performance on the target dataset being used [24] [25]. Taking inspiration from the *modest data augmentation* as done by Devlin et al [8], we use a similar notion here in the multilingual domain of QA. We "*pre-finetune*" mBERT firstly on the Natural Questions(NQ) [4] and KorQuAD(KorQ) [26] individually before fine-tuning on TyDi, and then also try a combined pre-finetuning on both NQ and KorQ before fine-tuning on TyDi. We also employ layer-freezing [9, 12] during the pre- finetuning and fine-tuning stage to expedite the training process and find it aiding the model to learn better.

5 Experiments

5.1 Data

Our dataset corresponds to the gold passage task of the TyDi QA dataset [2] which consists of question-answer pairs across 9 languages. The dataset is in the same format as SQuAD 1.1 [3] and does not contain any unanswerable questions. The dataset consists of ~50k instances in the training set and ~6k instances in the development set. Table 1 provides a complete breakdown of the dataset into the constituent languages with some statistics. Table 2 shows a sample input-output example for the task.

	Train			**Dev**		
Language	No. of samples	Avg. Context Tokens	Median Context Tokens	No. of samples	Avg. Context Tokens	Median Context Tokens
English	3696	141.8	125	641	147.5	132
Arabic	14805	192.1	150	1025	166.7	135
Bengali	2390	277.6	230	130	283.3	246
Finnish	6855	141.0	120	1082	147.9	125
Indonesian	5702	135.0	114	746	131.7	109
Swahili	2755	115.3	75	671	98.1	76
Korean	1625	191.0	142	441	161.8	138
Russian	6490	176.8	136	913	163.3	133
Telugu	5563	239.7	156	712	180.5	118
Total	49881	-	-	6361	-	-

Table 1: Data Statistics for TyDiQA-GoldP task

5.2 Evaluation method

The primary evaluation measure for our task is the macro-averaged F1 score calculated as follows: First, the scores for each example are averaged within a language in the same way as in SQuAD [3] and then the final F1 score is calculated by averaging the score over all non-English languages.

Input	*Context*: The cerebral cortex is folded in a way that allows a large surface area of neural tissue to fit within the confines of the neurocranium. When unfolded, each hemispheric cortex has a total surface area of about 1.3 square feet (0.12m2).[5] The folding is inward away from the surface of the brain, and is also present on the medial surface of each hemisphere within the longitudinal fissure. Most mammals have a cerebral cortex that is convoluted with the peaks known as gyri and the troughs or grooves known as sulci. Some small mammals including some small rodents have smooth cerebral surfaces without gyrification.[3] *Question*: What is the surface area of the human cortex?
Output	*Answer*: 1.3 square feet

Table 2: Sample Example for TyDiQA-GoldP task

5.3 Experimental details

All experiments are performed using the *run_squad.py* script from HuggingFace's transformers library [23] with default parameters unless stated otherwise. Baseline results for mBERT and distilmBERT are obtained by fine-tuning on the entire training set. We add our own functions and integrate them with the base script to support the rest of our experiments. We introduced flags to control the languages being used for training in an end-to-end fashion to facilitate our ensembling and zero-shot experiments without explicitly creating different training files. This helped us in rapid prototyping.

Ensembling involved training 9 independent models, each fine-tuned on all but one language and then averaging the start and end logits of individual models to get the final prediction. We write our own functions for ensembling multiple models and integrate it with the script by adding appropriate flags. The results for the ensembling experiments can be seen in Table 3 marked as **ensemble**. For the results of the individual zero-shot models on both mBERT and distilmBERT, we refer the reader to Table 6 and 7 in the Appendix A.

For pre-fine tuning our models on Natural Questions (NQ) [4] and KorQuAD [26], we freeze the embedding layer and the first 6 encoder layers of the mBERT architecture taking inspiration from [9]. We write our own preprocessing script to convert the NQ dataset to a SQuAD compatible format. We also performed hyper-parameter tuning to determine the optimal number of gradient accumulation steps to be performed. We observed that the performance improves significantly until 8 steps and then provides diminishing returns for larger values. We also ensure that our models are not under or over trained by training them for different epochs and find that the default value of 3 epochs works best.

Since we observed mutually independent factors like pre-finetuning, gradient accumulation, and ensembling contributing to an increase in performance, we created our **Final Model** in the following way :

1. We take the mBERT model and **pre-finetune** it on NQ and KorQ with layer freezing.
2. Then we train **9 zero-shot models** using the above model as the starting point and fine-tuning on all but 1 language of TyDi with layer freezing.
3. In the training of all these 9 models, we use the **value of gradient accumulation steps as 8** with a mini-batch size of 8 to give us a virtual mini-batch size of 64.
4. We **ensemble** these 9 models by averaging the start and end logits predictions to give us the final model. The result can be seen in Table 3 in the last column.

6 Results & Quantitative Analysis

Table 3 shows our final results. Our distillation experiments as expected don't match the mBERT performance, however ensembling improves the performance for both mBERT and distilmBERT. Also, as seen in Table 2, 3 and expounded upon in Section 6.2, we show that expensive annotated data in low-resource languages can be replaced with easily available data in high-resource languages with the same level of performance on our task. Our final model comprising of gradient accumulation, pre-finetuning with layer freezing, and ensembling drastically improves the performance on **all languages** and improves the overall F1 score by **3.2 points**. To see the effects of each of these techniques independently, we present an ablation study below.

	mBERT (baseline 1)	**distillmBERT (baseline 2)**	**ensemble (mBERT)**	**ensemble (distilmBERT)**	**Final Model**
(English)	66.32	61.15	68.03	65.15	**70.85**
Arabic	75.27	73.24	77.48	75.05	**79.1**
Bengali	72.81	58.12	73.23	64.05	**78.1**
Finnish	70.29	66.47	72.57	67.25	**73.4**
Indonesian	76.29	70.06	76.93	72.02	**77.2**
Swahili	80.05	74.34	83.11	77.11	**85.11**
Korean	60.44	52.66	62.53	57.84	**63.34**
Russian	69.92	63.49	71.17	66.04	**72.45**
Telugu	82.86	78.82	83.05	79.44	**84.5**
OVERALL	73.49	67.15	75.01	69.85	**76.75**

Table 3: Performance of various models on TyDiQA-GoldP as measured by F1 score

6.1 Ablation Study

Our ablation study compares the performance of the baseline model against pre-finetuning, gradient accumulation, and ensembling individually. As evident from Table 4 all three of 3 of our modifications boost the performance and the individual gains get compounded which results in the best performance obtained from our final model. We hypothesize our findings as follows:

- We improve upon the baseline by pre fine-tuning on NQ+KorQ. We believe this happens as mBERT is able to utilize the increased data (even though it is only in Korean and English) to gain task-specific knowledge and augments that knowledge across other languages when exposed to them. We also note that the improvement when pre-fine tuned solely on either NQ or KorQuAD is not significant whereas the improvement becomes considerably large when both the datasets are used together. One possible reason for this could be that pre-fine tuning on a monolingual dataset destroys the cross-lingual generalizability of the model (by basically over-fitting to that particular language's typology) which counteracts any gain that additional data might have brought about.
- We improve upon the baseline by using 8 gradient accumulation steps as it virtually increases our batch-size and as [27] show, large mini-batches can both improve optimization speed and end-task performance of the model.
- We also improve upon the baseline by simply ensembling the 9 zero-shot models we trained by leaving out 1 language in each. It is important to note that we also performed the same ensembling experiment using 9 baseline models, trained on all 9 languages (with different seeds) and observed a similar boost in performance (results not included). This shows that it was perhaps the regularization effect of the ensembling principle in general that facilitated the boost.

6.2 Cross-lingual transferability & Cost Saving for Low Resource Languages

Based on the effectiveness of NQ and KorQ prefinetuning results and the reduced performance of zero-shot models on the left-out language, we intuit that the effectiveness of mBERT's learning can be attributed to two components: *language specific representations* which require a few examples in each of the languages on which the model needs to be tested, and *task specific representations* which can be learned by leveraging a dataset in some high-resource language, thus, reducing the costs and efforts of collecting data in multiple low-resource languages.

To test this, we take 2 mBERT models, one pre-finetuned on NQ and KorQ and the other directly off the shelf(control) and then finetune them both on TyDi as follows: retain 100% of the examples for all 8 languages except Swahili for training and incrementally increase the % of Swahili examples kept from 0 to 100 across multiple runs of each model. To demonstrate that this effect is independent of the artefacts of a particular language, we repeat the same exercise for Bengali as well. The two languages were chosen not only because they are low-resource but also that the context length of these languages were at the extremes (Swahili with shortest contexts and Bengali with largest contexts as evident from Table 1). The results for Swahili can be

	mBERT (baseline)	NQ+ baseline	KorQ+ baseline	NQ+KorQ+ baseline	GradientAcc(8)+ baseline	Ensemble+ baseline
(English)	66.32	69.10	67.28	68.62	68.5	68.03
Arabic	75.27	78.23	75.54	77.63	77.9	77.48
Bengali	72.81	69.12	69.19	74.32	75.3	73.23
Finnish	70.29	71.41	70.81	71.66	71.9	72.57
Indonesian	76.29	76.91	73.97	75.84	77.4	76.93
Swahili	80.05	81.28	79.28	83.44	81.9	83.11
Korean	60.44	60.70	63.00	61.11	61.9	62.53
Russia	69.92	70.16	69.83	70.36	70.7	71.17
Telugu	82.86	83.45	82.43	84.77	82.8	83.05
OVERALL	73.49	73.91	73.10	**74.85**	**75.00**	**75.01**

Table 4: Ablation study results

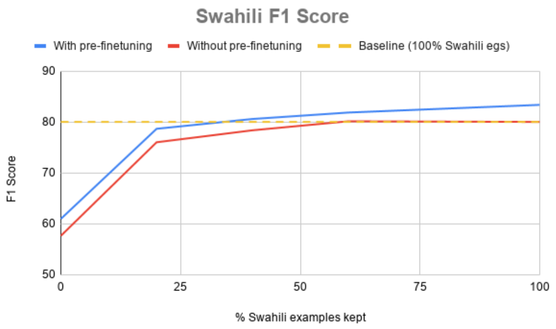

Figure 2: Performance on Swahili

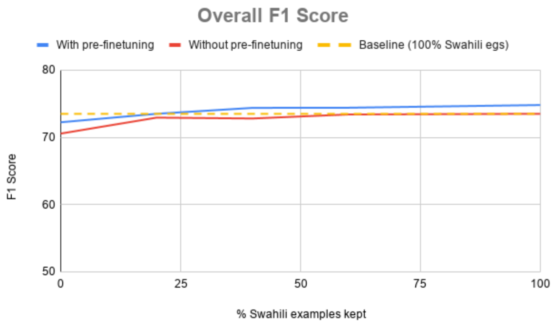

Figure 3: Overall performance

% examples kept for training	Swahili F1 (With Pre-finetuning)	Overall F1 (With Pre-finetuning)	Swahili F1 (No Pre-finetuning)	Overall F1 (No Pre-finetuning)
0 %	61	72.24	57.67	70.56
20%	78.7	73.5	76.05	72.93
40%	**80.62**	**74.38**	78.4	72.8
60%	81.89	74.4	80.17	73.4
100%	83.4	74.8	**80.05**	**73.5**

Table 5: Swahili Low Resource Experiment

seen in Table 5 and visualized in Figures 2 & 3 while the results for Bengali are included in the Appendix A. We make a couple of observations :

1. The increase in performance on Swahili (F1) is linear in % examples kept for both the pre-finetuned and control models.
2. The pre-finetuned model gives much better performance on Swahili and even on the overall average and we are able to meet the Swahili F1 baseline value of 80.05 with just 40% Swahili data with the pre-finetuned model (80.6).

Bengali has a similar result, although a little less flattering. We refer the reader to Table 8 and Figures 4 & 5 in the Appendix for the same. Thus, this confirms our cost-saving hypotheses and we explicitly show it holding for Swahili and Bengali. We intuit that this result would hold for all languages in TyDi but postpone those experiments to our future work.

7 Qualitative Analysis

Here we analyze the output of our final model compared to the baseline vanilla mBERT. Given our limited capacity to be able to analyze examples only from English, we took help from a native Bengali speaker to aid us in analysing examples from Bengali. All the below mentioned examples for English are included in Appendix B.

For English and Bengali, our final model improved upon 27 and 7 instances respectively over the baseline model and increased the F1 scores from 0.0-0.25 to 1.0 for each of these instances. We analysed all these examples and found a common confounding pattern for the baseline model which seem to exist in both of these languages. For example, in English the question-id "3358237151640751403" asks "Who is the **current leader** of South Africa?". The context provides enough evidence suggesting the **President is the leader** and further says that the first **President** was Nelson Mandela and the **incumbent** is Cyril Ramaphosa. The baseline model fails to understand the meaning of the rare word **incumbent** and predicts Nelson Mandela as the answer while our final model correctly produces Cyril Ramaphosa as the answer. Similarly when asked for "When was the **Republican People's Party** *formed* in Turkey?" with question-id "7916249696124721578" the baseline model predicts the date on which the party changed its name to **Republican People's Party** whereas it was *established* in 1919 under a different name and correctly identified by our final model. Again the model went for the lexical overlap rather than understanding the entire context and the question.

To validate whether the above issue existed only in one language or crossed the language boundaries (making the issue inherent to the task rather than arising due to a specific language typology) we evaluated some errors in Bengali and observed similar confounding pattern. The question with question-id "7987011167425321150" asks for the **first film of actress Purnima** and the context contains reference to two films, one with which she **started her journey in the movie world** and the other for which she won her **first national film award**. The baseline model predicts the latter while our final model predicts the former (the correct response). It is clear that the baseline model might have gone for the lexical overlap with the neighbouring tokens (**first film** in question v/s **first national film award** in context) while our final model was able to understand the semantics and produced correct output. Similar issue was observed for question-id "2217820890808407965" which asks for the start date for Kolkata metro and contexts contains two dates, one when it was inaugurated and the other when a particular route was completed. The baseline again gets confused and predicts the latter while our final model produces the correct output.

We also analysed a few examples in English on which our final model was performing worse and to our surprise for many of the instances we found that our final model prediction was more accurate than the gold answers provided by the annotators. For example, question id "4315843272939516169" asks "What percentage of the **U.S. population** suffers from drug addiction?". The context references 3 values, one corresponding to **US youth population**, the second corresponding to **US adult population in the last 12 months** and the third referencing the **population in entirety**. The annotators provide the first value as the answer while our final model predicts the third value as the answer which makes more sense.

The above analysis provides further evidence to the fact that learning on more task-specific data, irrespective of the language, might be helping the model to understand the inherent semantics and challenges of the task and filter out the correct response in the presence of multiple plausible answer choices.

8 Conclusion and Future Work

In this project, we conducted numerous experiments to analyse the performance of mBERT on a novel cross-lingual QA dataset and provided empirical results regarding its cross-lingual transferability. We showed that the learning ability of mBERT can be decoupled into *task specific representations* and *language specific representations* and both of these components can be learned independently to account for lesser data in the joint domain to improve the overall performance. We also provide ablation studies to prove that all our modifications independently help in improving the model performance.

Given the promising nature of our results, we also experimented with reducing the number of training examples for all languages to that with the minimum samples (in this case Korean) so as as to provide equal training opportunity across all languages (results not included as we are still not thorough with these experiments). The initial results seem to suggest that for most of the languages only 20-30% of data was sufficient to achieve the baseline performance with our methods. Another interesting research direction could be to try more comprehensive architectures for the QA head on the top of mBERT which can account for the dependent nature of start/end logits rather than treating them independently.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[2] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 2020.

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[4] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[5] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[6] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[7] Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856, 2019.

[10] Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

[11] Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. XQA: A cross-lingual open-domain question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2358–2368, Florence, Italy, July 2019. Association for Computational Linguistics.

[12] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, 2019.

[13] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[14] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019.

[15] Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*, 2018.

[16] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

[17] SQuAD 2.0: The Stanford Question Answering Dataset. `https://rajpurkar.github.io/SQuAD-explorer/`, 2018. [Online; accessed 10-March-2020].

[18] CoQA: A Conversational Question Answering Challenge. `https://stanfordnlp.github.io/coqa/`, 2018. [Online; accessed 10-March-2020].

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[20] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[21] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[24] Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[25] Jakub Konrád. Transfer learning for question answering on squad, 2019.

[26] Seungyoung Lim, Myungji Kim, and Jooyoul Lee. Korquad1.0: Korean qa dataset for machine reading comprehension, 2019.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

A Supplementary Results

	mBERT Baseline	**-En**	**-Ar**	**-Bn**	**-Fi**	**-Id**	**-Sw**	**-Ko**	**-Ru**	**-Te**	**Ensemble**
(English)	66.32	67	69.13	67.78	65.62	65.33	67.86	66.61	65.42	66.43	68.03
Arabic	75.27	76.33	58.49	77.18	77.11	64.15	77.26	76.61	76.96	76.12	77.48
Bengali	72.81	67.13	71.15	53.9	71.09	70.76	70.8	70.4	66.82	68.48	73.23
Finnish	70.29	70.85	70.47	70.67	53.24	70.2	70.05	70.44	69.98	70.25	72.57
Indonesian	76.29	74.34	73.8	74.07	75.8	61	75.83	75.29	75.28	76.31	76.93
Swahili	80.05	82.01	79.8	81.29	81.75	79.82	57.67	80.83	80.47	81.02	83.11
Korean	60.44	61.71	59.89	60.27	59.48	60.01	61.4	50.73	59.22	57.86	62.53
Russia	69.92	67.55	68.12	69.15	69.63	69.42	69.94	69.27	62.92	69.12	71.17
Telugu	82.86	82.28	81.9	81.31	81.94	81.91	81.55	81.77	82.5	50.99	83.05
Overall	73.49	72.78	70.45	70.98	71.26	69.66	70.56	71.92	71.77	68.77	75.01

Table 6: mBERT Zero shot experiments: Column name shows the language code for the language which was left out during training for zero-shot evaluation

	distilmBERT Baseline	**-En**	**-Ar**	**-Bn**	**-Fi**	**-Id**	**-Sw**	**-Ko**	**-Ru**	**-Te**	**Ensemble**
(English)	61.15	53.78	63.16	64.79	61.86	61.53	63.16	61.85	62.53	61.81	65.15
Arabic	73.24	74.41	57.78	74.46	73.71	73.2	73.82	74.18	74.1	74.71	75.05
Bengali	58.12	57.74	61.82	36.32	55.55	65.69	60.72	59.97	64.33	65.41	64.05
Finnish	66.47	65.92	65.93	65.8	50.57	66.18	65.78	65.23	65.74	67.09	67.25
Indonesian	70.06	70.97	70.28	70.69	69.2	47.93	69.25	69.51	68.84	71.5	72.02
Swahili	74.34	75.3	77.12	76.22	73.71	75.86	47.29	75.92	76.27	76.51	77.11
Korean	52.66	54.38	53.04	52.44	53.83	54.12	55.33	30.89	54.95	54.42	57.84
Russia	63.49	63.17	62.64	63.51	64	63.89	64.28	63.62	55.26	64.07	66.04
Telugu	78.82	78.55	79.08	78.25	78.86	77.81	79.07	77.75	77.56	38.11	79.44
Overall	67.15	67.56	65.96	64.71	64.93	65.58	64.44	64.63	67.13	63.98	69.85

Table 7: distilmBERT Zero shot experiments: Column name shows the language code for the language which was left out during training for zero-shot evaluation

(% examples kept for training)	**Bengali F1 (With Pre-finetuning)**	**Overall F1 (With Pre-finetuning)**	**Bengali F1 (No Pre-finetuning)**	**Overall F1 (No Pre-finetuning)**
0%	54.83	71.82	53.90	70.90
20%	67.40	73.68	64.48	71.80
40%	67.85	73.60	68.10	72.50
60%	69.70	74.25	69.10	73.28
80%	71.00	73.54	70.93	73.08
90%	73.24	74.09	72.20	73.18
100%	74.32	74.85	72.81	73.49

Table 8: Bengali Low Resource Experiment

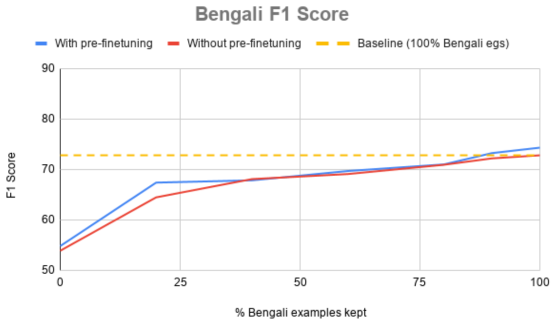

Figure 4: Performance on Bengali

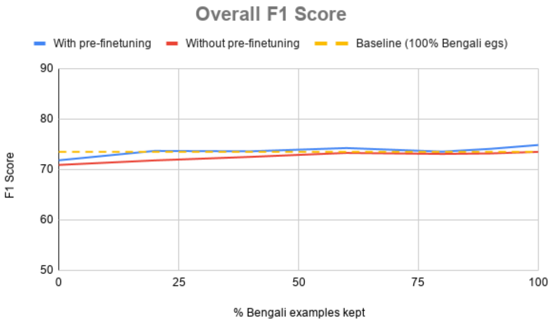

Figure 5: Overall performance

B English Examples used for Qualitative Analysis

Input	*Context*: The President is elected by the National Assembly, the lower house of Parliament, and is usually the leader of the largest party, which has been the African National Congress since the first non-racial elections were held on 27 April 1994. The Constitution limits the president's time in office to two five-year terms.[2] The first president to be elected under the new constitution was Nelson Mandela. The incumbent is Cyril Ramaphosa, who was elected by the National Assembly on 15 February 2018 following the resignation of Jacob Zuma. *Question*: Who is the current leader of South Africa?
Output	*Gold Answer*: Cyril Ramaphosa *Baseline model Answer*: Nelson Mandela *Final Model Answer*: Cyril Ramaphosa

Table 9: Question id: 3358237151640751403

Input	*Context*: The political party was established during the Sivas Congress in 1919 as a union of resistance groups against the Greek invasion of Anatolia. The union represented Turkish people as a unified front during the Turkish War of Independence (1919–1923). On 9 September 1923, the P̈eople's Partyd̈eclared itself to be a political organization and on 29 October 1923, announced the establishment of the Turkish Republic. On 10 November 1924, the People's Party renamed itself the R̈epublican People's Party(̈CHP) as Turkey moved into a one-party period. *Question*: When was the Republican People's Party formed in Turkey?
Output	*Gold Answer*: 1919 *Baseline model Answer*: 10 November 1924 *Final Model Answer*: 1919

Table 10: Question id: 7916249696124721578

Input	*Context*: Based upon representative samples of the US youth population in 2011, the lifetime prevalence[note 10] of addictions to alcohol and illicit drugs has been estimated to be approximately 8% and 2–3% respectively.[15] Based upon representative samples of the US adult population in 2011, the 12 month prevalence of alcohol and illicit drug addictions were estimated at roughly 12% and 2–3% respectively.[15] The lifetime prevalence of prescription drug addictions is currently around 4.7%.[134] *Question*: What percentage of the U.S. population suffers from drug addiction?
Output	*Gold Answer*: 2–3% *Baseline model Answer*: 8% and 2-3% *Final Model Answer*: 4.7%

Table 11: Question id: 4315843272939516169