Multilingual Transfer Learning for QA Using Translation as Data Augmentation

Mihaela Bornea, Lin Pan, Sara Rosenthal, Radu Florian, Avirup Sil
IBM Research AI, Thomas J. Watson Research Center, Yorktown Heights, NY 10598
{mabornea,panl,sjrosenthal,raduf,avij}@us.ibm.com

Abstract
Prior work on multilingual question answering has mostly focused on using large multilingual pre-trained language models (LM) to perform zero-shot language-wise learning: train a QA model on English and test on other languages. In this work, we explore strategies that improve cross-lingual transfer by bringing the multilingual embeddings closer in the semantic space. Our first strategy augments the original English training data with machine translation-generated data. This results in a corpus of multilingual silver-labeled QA pairs that is 14 times larger than the original training set. In addition, we propose two novel strategies, language adversarial training and language arbitration framework, which significantly improve the (zero-resource) cross-lingual transfer performance and result in LM embeddings that are less language-variant. Empirically, we show that the proposed models outperform the previous zero-shot baseline on the recently introduced multilingual MLQA and TYDi QA datasets.

Introduction
Recent advances in open domain question answering (QA) have mostly revolved around machine reading comprehension (MRC) where the task is to read and comprehend a given text and then answer questions based on it. However, most recent work in MRC has only been in English e.g. SQuAD (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018), HotpotQA (Yang et al. 2018) and Natural Questions (Kwiatkowski et al. 2019). Significant performance gains and the state-of-the-art (SOTA) on these datasets are credited to large pre-trained language models (Devlin et al. 2019; Radford et al. 2019; Yang et al. 2019b).

Multilingual BERT (mBERT), which is trained on Wikipedia articles from 104 languages and equipped with a 120k shared wordpiece vocabulary, has encouraged a lot of progress on cross-lingual tasks e.g. XNLI (Conneau et al. 2018), NER (Keung, Lu, and Bhardwaj 2019; Wu and Dredze 2019) and QA (Artetxe, Ruder, and Yogatama 2019; Cui et al. 2019; He et al. 2018) by performing zero-shot training: train on one language and test on unseen target languages.

In this work, we focus on multilingual QA and, in particular, on two recent large-scale datasets: MLQA (Lewis et al. 2020) and TYDi QA\(^1\) (Clark et al. 2020). Both datasets contain English QA pairs but also examples from 13 other diverse languages.

Some examples are shown in Figure 1. MLQA evaluates two challenging scenarios: 1) Cross-Lingual Transfer (XLT) when the question and the context are in the same language, and 2) Generalized Cross-lingual Transfer (G-XLT) when the question is in one language (e.g. En) and the context is in another language (e.g. De). TYDi QA is designed for XLT only. The two datasets are challenging for multilingual QA due to the large number of languages and the variety of linguistic phenomena they encompass (e.g. word order, re-duplication, grammatical meanings).

Ideally, we want to build QA systems for all existing languages but it is impractical to collect manually labeled training data for all of them. In the absence of labeled data, (Clark et al. 2020) suggested several research directions for pushing the boundaries in multilingual QA, including zero-shot QA, exploring data augmentation with machine

---

\(^1\)TYDi QA in our paper refers to the Gold Passage task.
translation, as well as effective transfer learning. These are avenues we explore in our work in addition to asking the following research questions:

1. Is a large pre-trained LM sufficient for zero-shot multilingual QA?

Prior work proposes zero-shot transfer learning from English SQuAD data (Rajpurkar et al. 2016) to other languages using only a pre-trained LM and competitive results are achieved on MLQA (Lewis et al. 2020) and TyDi QA (Clark et al. 2020). We venture beyond zero-shot training by first exploring data augmentation (Albieti et al. 2019) on top of their underlying model. We achieve this by using translation methodologies (Yarowsky, Ngai, and Wicentowski 2001) to augment the English training data. We use machine translation to obtain additional silver labeled data allowing us to improve cross-lingual transfer at a low cost. Our approach introduces several multilingual extensions to the SQuAD training data: translating the context in English, translating just the context but keeping the question in English, and translating the question and the context to other languages. This enables us to augment the original English human-labeled training examples with 14 times more multilingual silver-labeled QA pairs.

2. Can we bring language-specific embeddings in multilingual LMs closer for effective cross-lingual transfer?

Our hypothesis is that we can make the cross-lingual QA transfer more effective if we can bring the embeddings in a multilingual pre-trained LM closer to each other in the same semantic space. To answer a question in French it should suffice to train the system on Hindi and not be necessary to train a system on the target language: hence, French and Hindi should look as if they are the same language. We propose two approaches to explore cross-lingual transfer:

In our first approach, we propose a novel strategy based on adversarial training (AT) (Miyato, Dai, and Goodfellow 2017; Chen et al. 2018; Yang et al. 2019a). We investigate how the addition of a language-adversarial task during QA finetuning for a pre-trained LM can significantly improve the cross-lingual transfer performance while causing the embeddings in the LM to become less language-dependent.

In our second approach, we develop a novel Language Arbitration Framework (LAF) to consolidate the embedding representation across languages using properties of the translation. We train additional auxiliary tasks e.g., making sure an English question and its translation in Arabic produce the same answer when they see the same input context in Spanish. The intuition behind language arbitration is that while we are training the model on English and translated examples, the proposed multilingual objectives bring the language-specific embeddings closer to the English embeddings.

Overall, our main contributions in this paper are as follows:

- We create a new translation dataset which has 14 times more multilingual silver-labeled QA pairs than SQuAD.
- We present an adversarial training approach and a language arbitration framework to bring the LM embeddings closer to each other to improve cross-lingual QA transfer.
- We achieve statistically significant improvements compared to prior work (Lewis et al. 2020; Clark et al. 2020) with all of our models.

Multilingual Question Answering

In this section, we briefly discuss the LM and QA models. These are the foundations applied to our approach.

Pre-trained Language Model

Given a token sequence \( \mathbf{X} = [x_1, x_2, \ldots, x_T] \), we choose mBERT, a deep Transformer (Vaswani et al. 2017) network, which outputs a sequence of contextualized token representations \( \mathbf{H} = [h_1, h_2, \ldots, h_T] \).

\[
h_1, \ldots, h_T = \text{mBERT}(x_1, \ldots, x_T) \quad (1)
\]

mBERT has 12 layers each with 12 heads and \( h_t \in \mathbb{R}^{768} \). It is pre-trained on 104 languages and produces SOTA results on many cross-lingual tasks (Conneau et al. 2018; Keung, Lu, and Bhardwaj 2019).

Underlying QA model: mBERT\textsubscript{QA}

We build mBERT\textsubscript{QA}, our underlying QA model, as described in (Lewis et al. 2020; Devin et al. 2019). To create the input sequence we concatenate the [CLS], question, [SEP] and context tokens. mBERT\textsubscript{QA} adds two dense layers followed by a softmax on top of mBERT for answer extraction:

\[
\alpha_y = \text{softmax}(\mathbf{W}_1 \mathbf{H}); \quad \alpha_t = \text{softmax}(\mathbf{W}_2 \mathbf{H}),
\]

where \( \mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{768 \times 1} \), and \( \mathbf{H} \in \mathbb{R}^{T \times 768} \). \( \alpha_y \) and \( \alpha_t \) denote the probability of the \( t \)th token in the sequence being the answer start and end, respectively. These two layers are trained during the finetuning stage using the cross entropy loss:

\[
\mathcal{L}_{QA} = -\frac{1}{T} \sum_{t=1}^{T} \mathbb{1}(b_t) \log \alpha_y^t + \sum_{t=1}^{T} \mathbb{1}(e_t) \log \alpha_t^t \quad (2)
\]

where \( \mathbb{1}(\mathbf{b}) \) and \( \mathbb{1}(\mathbf{e}) \) are one-hot vectors from the ground truth offsets of the answer start and end.

Prior work - Zero-shot (ZS) Learning: Both (Lewis et al. 2020) and (Clark et al. 2020) propose zero-shot learning for multilingual QA by training on English QA data (SQuAD v1.1) and testing on all other languages. This is our basic model and the baseline setting. We train mBERT\textsubscript{QA} with examples of the form \((Q_{En}, C_{En}, A_{En})\) where \( A_{En} \subset C_{En} \). During inference, we use the trained model to extract the answer span \( A_y \) from \( C_y \) where \( y \) is any language.

Models

In this section, we outline our improvements on top of the prior work on MLQA.
Table 1: Comparing our original training data SQuAD v1.1 with our augmented training data using translation techniques. The Question Type is based on the first word in the question.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Avg. # of Words</th>
<th>Why</th>
<th>How</th>
<th>Question Type Frequency</th>
<th># Q-A Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ques.</td>
<td>Ans.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQuAD v1.1</td>
<td>10.1</td>
<td>3.2</td>
<td>1,194</td>
<td>8,082</td>
<td>87,599</td>
</tr>
<tr>
<td>T (Q)</td>
<td>11.9</td>
<td>3.2</td>
<td>7,164</td>
<td>48,492</td>
<td>525,594</td>
</tr>
<tr>
<td>T (C)</td>
<td>10.1</td>
<td>4.4</td>
<td>5,742</td>
<td>39,668</td>
<td>441,690</td>
</tr>
<tr>
<td>T (Q+C)</td>
<td>12.1</td>
<td>4.4</td>
<td>5,742</td>
<td>39,668</td>
<td>441,690</td>
</tr>
<tr>
<td>T (All)</td>
<td>11.6</td>
<td>4.1</td>
<td>16,260</td>
<td>111,664</td>
<td>1,233,776</td>
</tr>
</tbody>
</table>

--

Algorithm 1 Pseudo-code for adversarial training on the multilingual QA task.

**Require:** \( \langle Q'_{En}, C_{En} \rangle, (b, e), L, D, \text{mBERT}_{QA}, \eta \).
\( \langle Q'_l, C_{En} \rangle \) is the translated question (or English), \( C_{En} \) is the English context. \( b \) and \( e \) are the correct answer start and end positions. \( L \) is the language label for the question, \( D \) is discriminator, and \( h(\text{mBERT}_{QA}) \) is the question representation from the \( \text{mBERT}_{QA} \) model, and learning rate \( \eta \).

1. **for** epochs **do**
2. **for** steps **do**
3. \( (Q'_l, C_{En}), (b, e), L_t \)
4. \( (\alpha, \beta) \leftarrow \text{mBERT}_{QA}(\langle Q'_l, C_{En} \rangle) \)
5. \( p_e \leftarrow D(h(\text{mBERT}_{QA})) \)
6. \( \eta \nabla_{\theta_{QA}} L_{QA} \)
7. \( \eta \nabla_{\theta_{D}} L_D \)
8. **end for**
9. **end for**

--

Data Augmentation with Translation

Our first approach beyond zero-shot QA is to introduce data-augmentation (Yu et al. 2018; Alberti et al. 2019) based models. Since we only have English examples to train our system on, we expand our training data and explore several translation-based data augmentation models for MLQA. Table 1 shows statistics for the different datasets. We use the IBM Watson Language Translator\(^2\) to:

1. **Translate (Q+C):** We pick a language \( l \in L \) where \( L = \{De, En, Ar, Hi, Zh\} \)\(^3\) and translate \( (Q_{En}, C_{En}, A_{En}) \) to create examples \( \langle Q'_l, C'_{En}, A'_{En} \rangle \) in that language. We do this for each of the 5 languages. Note, \( Q'_l \) and \( C'_{En} \) are the translations of \( Q_{En} \) and \( C_{En} \) and \( A'_{En} \) is the translated answer, all in language \( l \). In order to obtain the alignment of the gold answer \( A_{En} \) in the translated context \( C'_{En} \), we place pseudo HTML tags around \( A_{En} \) and then translate \( C_{En} \). Note that the main challenge of this strategy is the answer alignment step and we only keep the translated examples where this succeeds. The number of translated examples we obtained is 87,062 for German, 77,759 for Spanish, 84,185 for Arabic, 20,981 for Hindi and 84,104 for Chinese. The final data set including English has 441,690 examples. The percentage of reduced question type ranges from 14% (Why) to 20% (Why).

2. **Translate(Q):** Only \( Q_{En} \) is translated to other languages leaving \( C_{En} \) intact to create examples \( \cup_l \langle Q'_l, C_{En}, A_{En} \rangle \).

\(^2\)https://www.ibm.com/watson/services/language-translator/
\(^3\)Our translation api does not support Vietnamese and Swahili.

This data augmentation strategy produces a more accurate dataset since it does not require the answer alignment stage which can be error-prone. We translate every \( Q_{En} \) to 5 other languages and we obtain a dataset of 525,594 examples, which is 6 times larger than SQuAD v1.1. T(Q) increases the average number of words in the question by 1.8.

3. **Translate (C):** We only translate \( C_{En} \) to other languages to create \( \cup_l \langle Q_{En}, C'_l, A_{En} \rangle \). We use the same answer alignment strategy as in **Translate (Q+C)** to generate the gold answer \( A_{En} \) for the translated examples in \( L \). We obtain 441,690 examples (same as Translate (Q+C)). T(C) increases the average number of words in the question by 1.2.

4. **Translate(ALL):** We combine the data from all the 3 strategies together to create a meta-translation model with 1,233,776 examples, 14 times larger than SQuAD.

Adversarial Training

Translation-based strategies provide ample scope for \( \text{mBERT}_{QA} \) to train on plenty of \( \langle Q, C, A \rangle \) examples where \( Q \) and \( C \) can be in different languages. However, it can still be challenging as new languages can continuously be added to the model requiring optimal MT systems in all languages. Therefore, it is important to explore bringing the embeddings of different languages in \( \text{mBERT} \) close to each other to achieve effective cross-lingual transfer. For this purpose, we introduce a novel multilingual adversarial training (AT) method inspired by (Goodfellow et al. 2014). The goal is to fine-tune \( \text{mBERT} \) so that its embeddings become language-invariant as possible. Algorithm 1 provides an overview of this approach.
Concretely, we use the \textit{Translate(Q)} strategy outlined in the previous section, and for every \{\text{\textit{Q}_{En}, C_{En}, A_{En}}\}, we derive examples of \{\text{\textit{Q}_{En}, C_{En}, A_{En}}\}, where the question is translated. All the examples are added to the training data. The discriminator \(D\) of the AT model is trained to classify the question representation in different languages to the correct language label \(L \in \{\text{En, De, Es, Ar, Hi, Zh}\}\). We use the `[CLS]` token to derive a single question representation as input for \(D\) and train with cross-entropy loss:

\[
\mathcal{L}_D = - \sum_{l=1}^{L} 1(g_l) \log p_l, \quad (3)
\]

where \(1(g)\) is a one-hot vector for the ground-truth language labels, and \(p\) are the language predictions from the model. \(D\) is implemented as a multilayer perceptron.

Under the AT objective, the underlying QA model, in addition to the QA objective, is trained to also minimize the KL-divergence between the uniform distribution, and the language labels predicted by the discriminator.

\[
\mathcal{L}_{adv} = - \sum_{l=1}^{L} KL[U(g_l)|| \log p_l], \quad (4)
\]

\(L_{adv}\) encourages the LM embeddings to appear uniform to the discriminator, across all languages. In contrast, \(L_D\) drives the discriminator to recognize the language. During training, in each step, we first update mBERT\(_{QA}\) with \(L_{QA} + L_{adv}\) (See Eq 2 for \(L_{QA}\)) while fixing the parameters of the discriminator (Alg. 1 line 6), and then update the discriminator with \(L_D\) fixing those of mBERT\(_{QA}\) (line 8).

In addition to performing AT using all 6 languages, \textbf{AT (en-all)}, we also conduct experiments picking just one random language (e.g. \(l = \text{Zh}\)) to perform \textbf{AT (en-zh)}.

**Language Arbitration Framework**

In this section, we explore an alternative approach for bringing the language-specific embeddings closer to each other using a novel Language Arbitration Framework (LAF) to train a multilingual QA model. Just like a regular human arbitrator, LAF’s job at the end of training is to make sure the same question in different languages produce the same answer while maintaining that the underlying representation of the questions are the same. Similar to the AT method, \textbf{Translate(Q)} is used to generate our training examples. For every \{\text{\textit{Q}_{En}, C_{En}, A_{En}}\} in the original English dataset, we derive an augmented training set with example pairs \{\text{\textit{Q}_{En}, C_{En}, A_{En}}\}, where the question is translated to language \(l \in L\). Training of LAF proceeds with such example pairs and exploits properties of the translation to consolidate the LM embeddings. In addition to training the base mBERT\(_{QA}\) model on English and the translation, using the standard objective from Eq (2), LAF also performs the following objectives during training:

1. \textbf{Produce the same answer (PSA)}: PSA encourages the translation \(\langle\textit{Q}'_{En}, C_{En}\rangle\) to produce the same answer as the original example \(\langle\textit{Q}_{En}, C_{En}\rangle\), for all languages \(l \in L\). We run mBERT\(_{QA}\) on English and the translation. Then, in addition to computing \(L_{QA}\) (Equation 2) we compute the additional loss:

\[
\mathcal{L}_{PSA} = -\frac{1}{2} \left( \sum_{t=1}^{T} \mathbb{1}(b_t^{En}) \log \alpha_t^{p} + \sum_{t=1}^{T} \mathbb{1}(e_t^{En}) \log \alpha_t^{e} \right) \quad (5)
\]

\(\mathbb{1}(b_t^{En})\) and \(\mathbb{1}(e_t^{En})\) are one-hot vectors indicating the answer start and end positions predicted by the mBERT\(_{QA}\) for \(\langle\textit{Q}_{En}, C_{En}\rangle\). \(\alpha_t^{p}\) and \(\alpha_t^{e}\) denote the answer begin and end probability predicted by mBERT\(_{QA}\) for \(\langle\textit{Q}_{En}, C_{En}\rangle\). While \textbf{Translate(Q)} optimizes the standard \(L_{QA}\) objective on translated data, \(L_{PSA}\) uses the English predictions for additional supervision and brings the LM embeddings closer by maintaining agreement between English and the translation. This is beneficial in cases where there is partial overlap between the English predicted answer and the gold label.

2. \textbf{Produce the same answer and question similarity (PSA+QS)}: In this approach, in addition to the PSA loss, we also compute the \textit{cosine-similarity} between \(\textit{Q}_{En}\) and \(\textit{Q}_{En}'\) in all languages \(l \in L\). The intuition is that the cosine similarity of translations should be high, encouraging the embeddings to move even closer to each other.

To obtain a single question representation, \(\bar{h}_{Q_{En}}\) for \(\textit{Q}_{En}\)
and $\tilde{h}_{Q_{\text{en}}}^l$ for language $l$, we perform average pooling over the hidden vectors for the question tokens from mBERT.

$$L_{QS} = 1 - \cosine(\tilde{h}_{Q_{\text{en}}}, \tilde{h}_{Q_{\text{en}}})$$

(6)

In addition to performing PSA and PSA+QS in all the languages, we also apply them in a single language, $l = Z$ as PSA(en-zh) and PSA+QS(en-zh).

**Experiments**

**Data and Evaluation Metric**

**MLQA:** We first evaluate our techniques on MLQA (Lewis et al. 2020) which is a large multilingual QA dataset that covers 7 languages as listed in Table 2. The dataset is 4-ways language-parallel with parallel passages from Wikipedia articles on the same topic. Questions are originally asked in English and they are translated to other target languages.

The dataset provides a development set (1,148 parallel instances) that is significantly smaller than the blind test (11,590 parallel instances). Hence, we train our models on the SQuAD v1.1 dataset (details in Table 1). We also create a much larger multilingual training corpus with the help of machine translation. To provide a comprehensive evaluation of our techniques we run all experiments on the MLQA dataset since it was designed for both G-XLT and XLT task.

**TyDi QA:** We choose the best models based on our MLQA experiments and run them on the TyDi QA (Clark et al. 2020) GoldP dataset. The GoldP task was designed only for XLT evaluation and is similar to MLQA. There are 9 languages of which English (en) and Arabic (ar) are the only ones in common between TyDi QA and MLQA. Although TyDi QA has a multilingual training set, in this work we train our models on SQuAD v1.1 in order to test the cross-lingual transfer ability of our proposed models. We also create a separate training corpus by translating the questions to the language in the LAF setting improves over the ZS baseline

**Evaluation Metric:** We use the official evaluation metric from both datasets and report the mean token F1 as opposed to Exact Match (EM) as the latter severely penalizes a system if it adds functions words.

**Hyper-parameters**

We perform hyper-parameter selection on the SQuAD and MLQA dev split. We use $3 \times 10^{-5}$ as the learning rate, 384 as maximum sequence length, and a doc stride of 128. Everything except ZS was trained for 1 epoch. We use the same hyper-parameter values on the MLQA test set and TyDi QA experiments. The best question representation is achieved with the [CLS] token for AT and average pooling for LAF (PSA+QS). Other methods tried were the concatenation of [CLS] and [SEP].

The discriminator is implemented as a multilayer perceptron with 2 hidden layers and a hidden size of 768 * 4. For both AT and LAF, in addition to (en-zh),

3We report token-level F1 as opposed to Exact Match (EM) as the latter severely penalizes a system if it adds functions words.

which was chosen at random, we also experimented with German, the language closest to English. Both achieve similar performance.

**MLQA Results**

Table 2 shows the performance of various competing strategies for MLQA. For each language of the context we report the G-XLT performance averaged across questions in all the 7 languages. The final two columns show the overall G-XLT and the XLT performance across all the 7 languages.

**Zero-shot:** We report the results of our re-implementation of the ZS setting of mBERT $Q_{\text{A}}$ (Lewis et al. 2020) which is the underlying QA model and show our improvements on top it.

**Translation:** T(Q) provides the biggest improvement out of all the competing translation techniques T(C), T(Q+C) with an overall gain (on average) of 6 points on G-XLT and 3.5 points on XLT. We believe that this degradation is due to answer alignment errors when translating the context. The alignment also causes a loss in training examples compared to the case when just the questions are translated. Note that the T(C) model is the weakest as it is the most affected by the alignment strategies and has the highest standard deviation among all the models. Combining all the strategies together provides a tiny improvement on G-XLT but at a cost to XLT performance: we believe that the T(C) data hurts this model and the parameters of mBERT alone are not sufficient to bring embeddings of different languages close to each other even with translation data. As we add more languages, the per-language capacity of the QA system decreases. This impacts the performance (known as the curse of multilinguality (Conneau et al. 2019)).

**Adversarial Training:** We first experiment with the AT (en-zh) model and noticed that adding a single language to the training data significantly improves performance over ZS. However AT (en-zh) is not strong (56.5 (G-XLT), 62.8 (XLT)) compared to T(Q), T(Q+C) and T(All). During training the discriminator is tasked to make a binary classification between En and Zh in this case. We hypothesize that this task may be too easy to balance the overall system training, since (Sønderby et al. 2017) showed that making the discriminator work harder is beneficial for training AT models. We leave training AT individually with each of the 6 other languages as part of our future work. When we extend the scope of the model to look at all languages together, we get the best performing MLQA system so far with {61.2 (G-XLT), 65.2 (XLT)}.

**Language Arbitration Framework:** Similar to AT, for LAF, we first start with an ‘en-zh’ model and then move on to an ‘en-all’ model. Our PSA+QS is weaker than just doing PSA on ‘en-zh’ suggesting again that choosing only one extra language in the LAF setting improves over the ZS baseline but is not as beneficial as adding all languages together. By choosing all the languages, we get the best performing overall model on the test split. PSA (en-all) does not lag behind but PSA+QS (en-all) provides an overall improvement of 10.2 and 4 points and 0.8 and 1.5 points improvement in G-XLT and XLT respectively over the ZS baseline and the best translation system ‘T(All)’. It is more beneficial to bring
We observe that the best LAF model is consistently better than the previous baseline (Lewis et al. 2020) achieving a significant 4 point improvement against the ZS baseline.

Statistical Significance: We compute statistical significance via the Fisher randomization test. The best LAF model (PSA+QS (en-all)) is statistically significantly better than the best AT and Translation model ($p < 0.05$). The best model for all three methods (T(Q), AT (en-all) and PSA+QS (en-all)) is significantly better than the ZS baseline.

### Table 2: Our results on MLQA test averaged over 3 runs. We compare our models against the previous baseline (Lewis et al. 2020): ZS setting with mBERTQA. Best numbers within the method are in bold. The best LAF and AT models are statistically significantly better than the best Trans model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Method</th>
<th>ar</th>
<th>de</th>
<th>en</th>
<th>es</th>
<th>hi</th>
<th>vi</th>
<th>zh</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>T(Q)</td>
<td>53.8</td>
<td>68.0</td>
<td>73.5</td>
<td>65.4</td>
<td>53.2</td>
<td>63.2</td>
<td>56.7</td>
<td>60.9 (±0.2)</td>
</tr>
<tr>
<td></td>
<td>(C)</td>
<td>44.8</td>
<td>51.7</td>
<td>62.0</td>
<td>57.6</td>
<td>42.7</td>
<td>55.8</td>
<td>50.8</td>
<td>52.2 (±1.0)</td>
</tr>
<tr>
<td></td>
<td>(Q+C)</td>
<td>48.9</td>
<td>58.4</td>
<td>70.4</td>
<td>63.6</td>
<td>46.8</td>
<td>61.4</td>
<td>54.4</td>
<td>57.7 (±0.1)</td>
</tr>
<tr>
<td></td>
<td>(ALL)</td>
<td>52.6</td>
<td><strong>61.1</strong></td>
<td><strong>73.6</strong></td>
<td><strong>66.3</strong></td>
<td>50.6</td>
<td><strong>64.8</strong></td>
<td><strong>58.2</strong></td>
<td><strong>61.1</strong> (±0.1)</td>
</tr>
<tr>
<td>AT</td>
<td>(en-zh)</td>
<td>50.5</td>
<td>56.7</td>
<td>68.0</td>
<td>60.8</td>
<td>50.7</td>
<td>57.4</td>
<td>51.5</td>
<td>56.5 (±0.1)</td>
</tr>
<tr>
<td></td>
<td>(en-all)</td>
<td><strong>54.1</strong></td>
<td><strong>61.1</strong></td>
<td><strong>73.6</strong></td>
<td><strong>65.5</strong></td>
<td><strong>54.2</strong></td>
<td><strong>63.4</strong></td>
<td><strong>56.8</strong></td>
<td><strong>61.2</strong> (±0.1)</td>
</tr>
<tr>
<td>LAF</td>
<td>PSA (en-zh)</td>
<td>50.8</td>
<td>56.9</td>
<td>68.6</td>
<td>61.3</td>
<td>51.0</td>
<td>57.8</td>
<td>51.6</td>
<td>56.9 (±0.1)</td>
</tr>
<tr>
<td></td>
<td>PSA+QS (en-zh)</td>
<td>50.7</td>
<td>56.6</td>
<td>68.5</td>
<td>61.2</td>
<td>51.0</td>
<td>57.7</td>
<td>51.8</td>
<td>56.8 (±0.4)</td>
</tr>
<tr>
<td></td>
<td>PSA (en-all)</td>
<td>54.5</td>
<td>61.4</td>
<td>74.2</td>
<td>66.0</td>
<td>54.6</td>
<td>64.2</td>
<td>57.5</td>
<td>61.8 (±0.1)</td>
</tr>
<tr>
<td></td>
<td>PSA+QS (en-all)</td>
<td><strong>54.8</strong></td>
<td><strong>61.5</strong></td>
<td><strong>74.3</strong></td>
<td><strong>66.1</strong></td>
<td><strong>54.9</strong></td>
<td><strong>64.3</strong></td>
<td><strong>57.6</strong></td>
<td><strong>61.9</strong> (±0.1)</td>
</tr>
</tbody>
</table>

| AVG | 54.8 | 61.5 | 74.3 | 66.1 | 54.9 | 64.3 | 57.6 | 61.9 |

Table 3: G-XLT F1 scores of the LAF:PSA+QS (en-all) model on the overall test set for individual cross languages performance. XLT F1 is 65.7 averaged across the diagonal, as shown with the G-XLT results in the last row of Table 2.

Table 4: XLT F1 scores of ZS and LAF with mBERT.

<table>
<thead>
<tr>
<th>Model</th>
<th>ar</th>
<th>de</th>
<th>en</th>
<th>es</th>
<th>hi</th>
<th>vi</th>
<th>zh</th>
<th>XLT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS</td>
<td>51.7</td>
<td>60.6</td>
<td>80.4</td>
<td>66.8</td>
<td>50.5</td>
<td>61.4</td>
<td>60.1</td>
<td>61.7</td>
</tr>
<tr>
<td>LAF</td>
<td><strong>58.0</strong></td>
<td><strong>65.5</strong></td>
<td><strong>80.2</strong></td>
<td><strong>70.2</strong></td>
<td><strong>58.9</strong></td>
<td><strong>64.0</strong></td>
<td><strong>63.1</strong></td>
<td><strong>65.7</strong></td>
</tr>
</tbody>
</table>

Table 5: Our results on TYD1 QA dev. We compare our models against the previous baseline (Clark et al. 2020): ZS setting with mBERTQA. T(Q)*, AT*, LAF* are the MLQA models. The LAF and AT models are statistically significantly better than the T(Q) model and ZS.

<table>
<thead>
<tr>
<th>Model</th>
<th>en</th>
<th>bn</th>
<th>ko</th>
<th>in</th>
<th>te</th>
<th>sw</th>
<th>ar</th>
<th>ru</th>
<th>fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS</td>
<td>75.0</td>
<td>62.8</td>
<td>55.3</td>
<td>61.6</td>
<td>49.9</td>
<td>57.8</td>
<td>61.6</td>
<td>65.0</td>
<td>58.5</td>
</tr>
<tr>
<td>LAF*</td>
<td>74.1</td>
<td>59.9</td>
<td>56.5</td>
<td>63.0</td>
<td>49.0</td>
<td>63.8</td>
<td>67.0</td>
<td>64.6</td>
<td>56.7</td>
</tr>
<tr>
<td>AT*</td>
<td>74.1</td>
<td>59.9</td>
<td>55.3</td>
<td>64.1</td>
<td>49.1</td>
<td>63.8</td>
<td><strong>68.4</strong></td>
<td><strong>65.9</strong></td>
<td>57.3</td>
</tr>
<tr>
<td>T(Q)*</td>
<td>73.2</td>
<td>59.4</td>
<td>56.7</td>
<td>61.1</td>
<td>47.8</td>
<td>62.4</td>
<td>67.5</td>
<td>63.8</td>
<td>54.6</td>
</tr>
</tbody>
</table>

TYD1 QA Results

Table 5 shows the results on TYD1 QA. We first experiment with the same models that we trained for MLQA by translating SQuAD to the MLQA languages. In this setting, we evaluate cross-lingual transfer beyond translation, since en and ar are the only languages the two datasets have in common. Our best MLQA translation strategy T(Q), improves the F1 significantly on ar but it is slightly detrimental for the other target languages. On average the translation baseline shows no improvement over ZS. The best performing model is LAF with 1.5 F1 gains over ZS. LAF also has the best cross-lingual transfer performance, improving in, sw, ru as well as ar compared to the ZS baseline. We also
tested our models trained by translating SQuAD to the TyDi QA languages. In this case, we notice consistent trends with the MLQA results. All techniques improve the cross-lingual transfer across all languages. Data augmentation with MT shows large improvement over ZS increasing the F1 by 3.4 points. AT is better compared to T(Q) and the best results are obtained with cross-lingual LAF with an average increase of 5.3 F1 points compared to ZS. Our improvements over ZS and T(Q) are statistically significant and we used the Fisher randomization test.

Error Analysis
We take a random sample of our dev data and perform error analysis on the output to provide insights into our contributions. The correct answer predicted by the better model is shown with underline and the incorrect answer predicted by the poorer model is shown with italics.

Translation is better than ZS:
C(En): Stephen William Kuffler is known for his research on neuromuscular junctions in frogs, presynaptic inhibition, and the neurotransmitter GABA.
Q(Zh): 他以什么神经递质的名字而闻名
Explanation: Data augmentation helps.

AT is better than Translation:
C(De): Heftiger Regen verursachte auf Hawai‘i geringere Schäden durch örtliche Überflutungen auf der Nordhalbkugel die stärksten Winde und...
Q(En): Where were heavy rains?
Explanation: Adversarial training makes the mBERT embeddings more language-invariant.

LAF is better than AT:
C(Es): La película, que combina animación por computadora con acción en vivo, fue dirigida por Michael Bay, con Steven Spielberg como productor ejecutivo.
Q(Vi): Ai là đạo diễn sản xuất bộ phim Transformers năm 2007?
Explanation: LAF makes the mBERT embeddings even more language-invariant than AT.

LAF & AT are better than Translation:
C(En): Berlin is a world city of culture, politics, media and science...serves as a continental hub...metropolis is a popular tourist destination.
Q(De): Wofür war Berlin bekannt?
Explanation: See previous explanations.

Related Work
A large number of recent QA/ MRC datasets such as SQuAD (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018), TriviaQA (Joshi et al. 2017), NewsQA (Trischler et al. 2017) and Natural Questions (Kwiatkowski et al. 2019) have focused on English and have not explored multilingual QA.

There are plenty of non-English QA datasets (Gao et al. 2016; He et al. 2018; Shao et al. 2018; Mozannar et al. 2019; Gupta et al. 2018; Lee et al. 2018; Li et al. 2018; Asai et al. 2018; Croce, Zelenanska, and Basili 2019) in Chinese, Arabic, Hindi, Korean, French, Japanese and Italian. These datasets are 2-3 way parallel or mono-lingual. XQuAD (Artetxe, Ruder, and Yogatama 2019) is a translated subset of SQuAD v1.1 into 10 languages. The most competitive multilingual datasets are MLQA and TyDi QA due to their scale and use of the original contexts as they appear in Wikipedia rather than manual translation from English.

Prior work has explored back-translation for data-augmentation (Yu et al. 2018), multi-task learning (McCann et al. 2018; Bonbadim, Uva, and Moschitti 2017; Chen et al. 2017), adversarial learning (Wallace et al. 2019; Yang et al. 2019a; Wang and Bansal 2018; Zhu et al. 2020; Keung, Lu, and Bhardwaj 2019; Chen et al. 2018) either for mono-lingual QA or for other NLP tasks. None of these have explored multilingual techniques similar to ours that make the embeddings in the LM become language-agnostic.

Contrary to our approach, (Yuan et al. 2020) present results on MLQA but assume access to a commercial search engine and web queries to create their specialized training data for their answer boundary detection task. They only report XLT results on 3/7 MLQA languages, whereas, we evaluate on all 7 languages and report both XLT and G-XLT performance. We also note that access to a search engine is not always feasible and since the authors do not provide the web queries it is unclear how to extend their technique to other languages.

Perhaps, the closest work to ours is (Cui et al. 2019a), their approach relies on back-translation and an ensemble of two QA systems one on source (context) and one on target (question) language. Our proposed methods 1. do not rely on back-translation, 2. we introduce more diverse translation models and 3. we introduce two novel strategies for multilingual QA based on language arbitration and adversarial learning. Most importantly their ensemble approach relies on training data in the target language whereas we do not.

Choosing which of the multilingual LMs (e.g. mBERT (Devlin et al. 2019), XLM-R (Conneau et al. 2019) and M4 (Arivazhagan et al. 2019)) to use for MLQA is a separate thread of work that involves comparing pre-training objectives and which large corpora to train on and is not the main focus of this paper. Due to the large number of experiments we ran we focus on one framework and we chose mBERT.

Conclusion
In this work, we highlight open challenges in the existing multilingual approach by (Lewis et al. 2020) and (Clark et al. 2020). Specifically, we show that large pre-trained multilingual LMs are not enough for this task. We produce several novel strategies for multilingual QA that go beyond zero-shot training and outshine the previous baseline built on top of mBERT. We present a translation model that has 14 times more training data. Further, our AT and LAF strategies utilize translation as data augmentation to bring the language-specific embeddings of the LM closer to each other. These approaches help us significantly improve the cross-lingual transfer. Empirically, our models demonstrate strong results and all approaches improve over the previous ZS strategy. We hope these techniques spur further research in the field such as exploring other multilingual LMs and invoking additional networks on top of large LMs for multilingual NLP.
Acknowledgments
We thank Graeme Blackwood for his help with the machine translation api. We are grateful to Salim Roukos and the IBM MNLP team for the helpful discussions. We also thank the anonymous reviewers for their suggestions that helped us improve this paper.

References


