A Graph Fusion Approach for Cross-Lingual Machine Reading Comprehension

Anonymous ACL submission

Abstract

Although great progress has been made for Machine Reading Comprehension (MRC) in English, scaling out to a large number of languages remains a huge challenge due to the lack of large amounts of annotated training data in non-English languages. To address this challenge, some recent efforts of cross-lingual MRC employ machine translation to transfer knowledge from English to other languages, through either explicit alignment or implicit attention. For effective knowledge transition, it is beneficial to leverage both semantic and syntactic information. However, the existing methods fail to explicitly incorporate syntax information in model learning. Consequently, the models are not robust to errors in alignment and noises in attention. In this work, we propose a novel approach, named *GraFusion-MRC*, which jointly models the cross-lingual alignment information and the mono-lingual syntax information using a graph. We develop a series of algorithms including graph construction, learning, and pre-training. The experiments on two benchmark datasets for cross-lingual MRC show that our approach outperforms all strong baselines, which verifies the effectiveness of syntax information for cross-lingual MRC. The code will be made open-sourced on Github.

1 Introduction

Machine Reading Comprehension (MRC) (Rajpurkar et al., 2016; Joshi et al., 2017), which aims to improve the ability of machines to read and understand human texts, is a challenging task in Natural Language Understanding (NLU) (Rajpurkar et al., 2016). Various large-scale human-annotated corpora, such as SQuAD (Rajpurkar et al., 2016), have greatly advanced the progress in the MRC task (Seo et al., 2017; Yu et al., 2018; Devlin et al., 2019). However, those large-scale human-annotated datasets are mostly in resource-rich languages, such as English. For most languages in

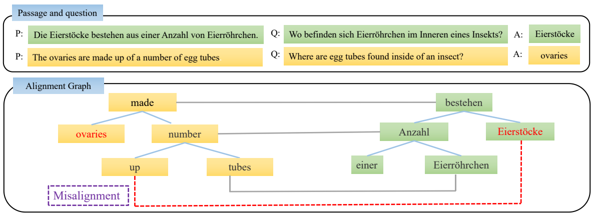


Figure 1: A passage (P), a question (Q), and an answer (A) in English with the translations in German.

the world, there is, however, scarce annotated data for MRC, which limits the corresponding MRC performance.

To tackle the challenge of data scarcity in low-resource languages, recent attempts in cross-lingual NLU adopt machine translation to transfer the knowledge learned from the high quality annotated data in resource-rich languages (i.e., the source languages) to low-resource languages (i.e., the target languages) (Schuster et al., 2019). For example, several methods (Hu et al., 2020; Liang et al., 2020) translate training data in English to target languages, and use the translated data to train the cross-lingual MRC models. Some other methods (Cui et al., 2019; Fang et al., 2021) translate test cases in a target language to English, and use the representation of the translated cases in English to enhance the representations of the original test cases.

For effective knowledge transfer across languages, both semantic and syntactic information is highly valuable and thus should be well represented. However, all previous translation-based approaches carry over knowledge across languages only through unstructured texts, where semantic and syntactic information is implicitly represented and complicatedly entangled. To represent the correlation among words in different languages, previous works either build translation alignments or learn attention matrices. However, it is very challenging to learn the connection between words across languages solely relying on texts only. Admitted by previous studies, misalignments often happen and badly hurt model performance (Xu et al., 2020; Li et al., 2020; Pei et al., 2020). Moreover, deep learning models may pay attention to less relevant words in long text (Zhang et al., 2020).

Can we use syntax information explicitly to enhance knowledge transfer across languages and improve cross-lingual MRC? In this paper, we tackle this challenge. Figure 1 shows a motivating example. Suppose the source training example is in English: the question is "Where are egg tubes found inside of an insect?", and the answer "*ovaries*" is in the sentence "The *ovaries* are made up of a number of egg tubes . . . " After the English example is translated into German, the corresponding answer "*Eierstöcke*" in "Die *Eierstöcke* bestehen aus einer Anzahl von Eierröhrchen . . . " is not correctly identified due to misalignment by an off-the-shelf alignment tool GIZA++ (Och and Ney, 2003). Checking many cases manually, we find misalignments commonly happen in complex sentence structures (e.g., involving passive voice where word orders are different from usual) and usages of rare words (e.g., "ovaries" and "Eierstöcke" belong to the domain of biology). In such cases, syntax information can help the model to figure out the correct alignment. In the example in Figure 1, although "ovaries" and "Eierstöcke" are not correctly aligned, their parents "made/bestehen", and siblings "number/Anzahl" are correctly aligned. Therefore, if we can leverage the syntax structure to propagate the alignment information, we can learn better representation for the target language.

Carrying the above insights, in this paper, we jointly model the cross-lingual alignment information and the mono-lingual syntax information using a graph. We make the following contributions. First, we propose using syntax information to enhance knowledge transfer across languages. Second, we develop a novel graph fusion approach to model the syntax structure as well as the alignment across the source and target inputs. We design a series of algorithms including graph construction, learning, and pre-training. Last, we evaluate our approach on two public cross-lingual MRC benchmarks. The experimental results show that our model effectively transfers knowledge from source language to target language through attention guided by syntax information, and hence outperforms all the strong baselines. The code will be released on Github. The limitation of our work can be found in Appendix.

2 Related Work

Given a question and a passage, the MRC task (Rajpurkar et al., 2016; Joshi et al., 2017) builds a model to find the span of the correct answer for the question from the given passage. Limited by the availability of large-scale annotated data, for most languages in the world, the MRC task relies on cross-lingual MRC models, which transfer knowledge from a resource-rich language to some low-resource languages. As a baseline, some multilingual pre-trained models, such as mBERT (Devlin et al., 2019), XLM (Lample and Conneau, 2019), and XLM-R (Conneau et al., 2020), are fine-tuned by training data in English and then directly applied to other languages.

There are also researches employ machine translators to generate parallel corpus as data augment. For example, some approaches translate training data from English to some target languages, and then adds the translated training sets into the fine-tuning stage (Cui et al., 2019; Hu et al., 2020; Liang et al., 2020; Yuan et al., 2020; Liu et al., 2020). Some other methods translates test cases in a target language into English, and combines the representation of the original test cases and the representation of the translated case to English through the attention mechanism (Cui et al., 2019; Fang et al., 2021).

Although these approaches improve the MRC results substantially, one weakness remained is the alignment quality between the example in original language and the translated example. Previous studies (Xu et al., 2020; Li et al., 2020; Pei et al., 2020) indicate that misalignments often happen and can badly degrade the model performance. Inspired by the observation in SG-Net (Zhang et al., 2020) that the syntax information can prevent a model from attending to some dispensable words and show significant gains in the English MRC task, we propose to use the syntax information to guide the correlation between the inputs in source and target languages.

3 Methodology

3.1 Overview of Network Architecture

Figure 2 shows the overview of our approach. The backbone is a stack of bidirectional Transformers (Vaswani et al., 2017) with $N + 2$ layers. The first layer encodes the inputs, and the last layer learns the final representations for decoding.

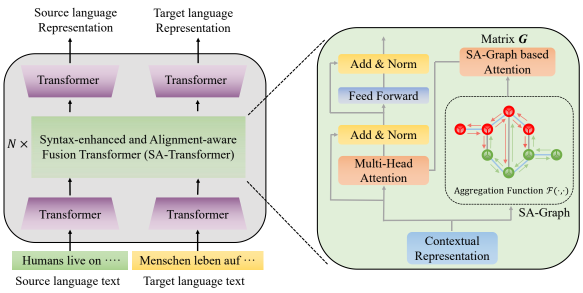


Figure 2: The overview of our model *GraFusionMRC*, where the red and green nodes represent the words in the source and target language, respectively.

Our major technical contribution is in the middle N layers, where a graph neural network is constructed and trained to model both syntax and alignment information. Such information jointly contributes to the knowledge transfer across languages and results in better representation for the target language through enhanced attention matrices.

Given an instance $\boldsymbol{S}$ in the source language, we first apply a machine translator to translate it to an instance $\boldsymbol{T}$ in the target language (or given an instance $\boldsymbol{T}$ in the target language, we can translate it to $\boldsymbol{S}$ in English). We then unify the length of $\boldsymbol{S}$ and $\boldsymbol{T}$ to be l by padding or truncating operations, and input $\boldsymbol{S} \in \mathbb{R}^{l\times d}$ and $\boldsymbol{T} \in \mathbb{R}^{l\times d}$ in parallel to our model, where d is the dimensionality of the token embedding vectors. The input is then encoded by a Transformer as follows:

$$\boldsymbol{A}_0^s = \textit{Transformer}(\boldsymbol{S}), \quad (1)$$

$$\boldsymbol{A}_0^t = \textit{Transformer}(\boldsymbol{T}). \quad (2)$$

We then take the concatenation of $\boldsymbol{A}_0^s$ and $\boldsymbol{A}_0^t$, and apply N *Syntax-enhanced and Alignment-aware Fusion Transformer* layers (or SA-Transformer for short) to produce the representation by

$$[\boldsymbol{A}_n^s; \boldsymbol{A}_n^t] = \textit{Transformer}_{sa}([\boldsymbol{A}_{n-1}^s; \boldsymbol{A}_{n-1}^t]), \quad (3)$$

where the subscripts $n \in [1, N]$ indicate that the variables are at the n-th SA-Transformer layer, and $[\boldsymbol{A}_n^s; \boldsymbol{A}_n^t] \in \mathbb{R}^{2l\times d}$ is the concatenation of the representations of the parallel sentences in the source and the target languages.

Each SA-Transformer layer applies a multi-head self-attention operation (Vaswani et al., 2017) followed by a feed-forward layer. Specifically, the multi-head self-attention operation first obtains a triplet consisting of the query $\boldsymbol{Q}_i$, the key $\boldsymbol{K}_i$ and the value $\boldsymbol{V}_i \in \mathbb{R}^{2l\times d_h}$ for each $head_i$ by applying linear transformations $\boldsymbol{W}_i^q$, $\boldsymbol{W}_i^k$, and $\boldsymbol{W}_i^v \in \mathbb{R}^{d\times d_h}$ on the input matrix $[\boldsymbol{A}_{n-1}^s; \boldsymbol{A}_{n-1}^t]$, respectively, where d_h is the dimensionality of each head, and matrices $\boldsymbol{W}_i^q$, $\boldsymbol{W}_i^k$, and $\boldsymbol{W}_i^v$ are parameters to be learned. Then, each $head_i$ conducts the following attention operation:

$$head_i = \textit{softmax}\left(\frac{\boldsymbol{Q_i}\boldsymbol{K_i}^\top}{\sqrt{d_k}} + \boldsymbol{G}\right)\boldsymbol{V}_i, \quad (4)$$

where i denotes the i-th head of the multi-head operation, and $\boldsymbol{G} \in \mathbb{R}^{2l\times 2l}$ is the attention matrix to be described in Section 3.2.

After obtaining the representation $[\boldsymbol{A}_n^s; \boldsymbol{A}_n^t]$ via (3), we further add another Transformer layer to separately project $\boldsymbol{A}_n^s$ and $\boldsymbol{A}_n^t$ back to the individual language spaces and obtain $\boldsymbol{A}^s$ and $\boldsymbol{A}^t \in \mathbb{R}^{l\times d}$, respectively, since our final goal is to predict the labels in the individual languages.

We then use $\boldsymbol{A}^s$ and $\boldsymbol{A}^t$ to predict the answer span in the source and target languages, respectively. Let us take $\boldsymbol{A}^t$ as an example to elaborate. Following LBMRC (Liu et al., 2020), we feed $\boldsymbol{A}^t$ to two separate linear layers, each followed by a softmax operation to produce the final span prediction $\boldsymbol{p}_{str}^t$ and $\boldsymbol{p}_{end}^t \in \mathbb{R}^l$, i.e., the predictions of the start and the end positions, respectively. For example, $\boldsymbol{p}_{str}^t$ is calculated by $\boldsymbol{p}_{str}^t = \textit{softmax}(\boldsymbol{A}^t \cdot \boldsymbol{u}_{str} + \boldsymbol{b}_{str})$, where $\boldsymbol{u}_{str} \in \mathbb{R}^d$ and $\boldsymbol{b}_{str} \in \mathbb{R}^l$ are two trainable parameters. We then calculate the standard cross entropy loss for the predicted start and end positions in the target language by

$$\mathcal{L}^t = -\frac{1}{\|\mathcal{D}\|}\sum_{i=1}^{\|\mathcal{D}\|}(\boldsymbol{y}_{str,i}^t \cdot \log(\boldsymbol{p}_{str,i}^t) + \boldsymbol{y}_{end,i}^t \cdot \log(\boldsymbol{p}_{end,i}^t)), \quad (5)$$

where $\|\mathcal{D}\|$ is the total number of training examples, $\boldsymbol{y}_{str,i}^t$ and $\boldsymbol{y}_{end,i}^t \in \mathbb{R}^l$ are the ground-truth labels for the start and end positions of the i-th training example.

3.2 Syntax-Enhanced and Alignment-Aware Graph (SA-Graph)

To incorporate the syntax and alignment information into Transformer, we learn an attention matrix $\boldsymbol{G}$, where an element $\boldsymbol{G}_{i,j}$ in $\boldsymbol{G}$ is the attention score indicating the attention that word i pays to the word j. To learn the matrix $\boldsymbol{G}$, we first construct the *syntax-enhanced and alignment-aware graph* (or SA-Graph for short), where each node corresponds to a word, and the edges represent the syntax and alignment information. Given a pair of parallel sentences as input, we build a graph to represent the relations among the words in the sentences. Each word in the parallel sentences corresponds to a node in the graph, and the edges between the nodes are based on the relations between the words. As introduced in Section 1, we consider two types of relations of words, cross-lingual word alignment and mono-lingual syntactic dependency. We build edges for those two relations.

In machine translation, the corresponding words in source and target languages can be aligned with each other. Taking German sentence "*Wir sollten die Umwelt schützen*" and its parallel sentence "*We should protect the environment*" in English as an example, we can apply some off-the-shelf alignment tools, such as GIZA++[1] (Och and Ney, 2003), to compute the word alignment. The aligned words often share similar semantic meaning, for example, "*Wir*" and "*We*", "*sollten*" and "*should*", "*die*" and "*the*", "*Umwelt*" and "*environment*", as well as "*schützen*" and "*protect*". We then add *word-alignment edges* between the nodes corresponding to those words.

In addition to the edges between words across languages, we also consider the syntactic structures of sentences and build edges between words within the same language. Specifically, we first split a given passage into sentence-level and then apply the Stanza toolkit[2] (Qi et al., 2020) to extract the dependency between words for each sentence. Two words are connected by a *word-dependency edge* if there exists a dependency between them. We also add a special word-dependency edge between the same words in a passage.

Based on the graph, the representation $\boldsymbol{f}_i$ of a word i is derived by

$$\boldsymbol{f}_{i,n} = \mathcal{F}(\boldsymbol{h}_{i,n}, \mathcal{N}(i)), \tag{6}$$

where $\boldsymbol{h}_{i,n}$ is the representation of word i from the n-th layer, $\mathcal{N}(i)$ denotes the neighbors of word i in the SA-Graph, and $\mathcal{F}(\cdot,\cdot)$ is the aggregation function of word i and its neighbors that will be described in Equation (8), Section 3.3. Once $\boldsymbol{f}_{i,n}$ is computed, the attention matrix $\boldsymbol{G}$ is obtained by

$$\boldsymbol{G}^n_{i,j} = (\boldsymbol{W}^n_{att} \cdot \boldsymbol{f}_{i,n} + \boldsymbol{b}^n_{att}) \cdot (\boldsymbol{W}^n_{att} \cdot \boldsymbol{f}_{j,n} + \boldsymbol{b}^n_{att}), \tag{7}$$

where $\boldsymbol{W}^n_{att} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b}^n_{att} \in \mathbb{R}^d$ are trainable parameters. For convenient representation, we use $\boldsymbol{f}_i$ instead of $\boldsymbol{f}_{i,n}$ in the following. Next, we present the learning process of the representation $\boldsymbol{f}_i$.

[1]https://github.com/moses-smt/giza-pp

[2]https://github.com/stanfordnlp/stanza

3.3 Graph Learning

After we construct the SA-Graph, we perform a learning algorithm over the graph. For each node i, we want to learn a better representation $\boldsymbol{f}_i = \mathcal{F}(\boldsymbol{h}_i, \mathcal{N}(i))$ than its original representation $\boldsymbol{h}_i$ by aggregating the information from its neighbors $\mathcal{N}(i)$. As described in Section 3.2, there are two types of edges in the graph. Correspondingly, the node representation $\boldsymbol{f}_i$ consists of two parts:

$$\boldsymbol{f}_i = \frac{1}{2}(\boldsymbol{f}^a_i + \boldsymbol{f}^d_i), \tag{8}$$

where $\boldsymbol{f}^a_i$ is the representation of word i aggregated from the alignment information, i.e., $\boldsymbol{f}^a_i = \mathcal{F}_a(\boldsymbol{h}_i, \mathcal{N}_a(i))$, where $\mathcal{N}_a(i)$ is the set of neighbors of word i that are connected by word-alignment edges. Similarly, $\boldsymbol{f}^d_i$ aggregates the dependency information, i.e., $\boldsymbol{f}^d_i = \mathcal{F}_d(\boldsymbol{h}_i, \mathcal{N}_d(i))$, where $\mathcal{N}_d(i)$ is the set of dependency neighbors. In Equation (8), in addition to the average function, other combination operators, such as weighted sum or max-pooling, may also be considered. Here we choose the simple but effective average method based on our empirical study. Experiments with other combination operators are presented and discussed in Appendix.

To learn aggregation by alignment $\mathcal{F}_a(\cdot,\cdot)$, for a word i, the representation $\boldsymbol{f}^a_i$ aggregates the information from its neighbors $\mathcal{N}_a(i)$ connected by the word-alignment edges. As indicated in the previous studies (Xu et al., 2020; Li et al., 2020; Pei et al., 2020), word alignment is a challenging task and misalignments may exist in results produced by existing methods. To mitigate the alignment errors, we develop a gate mechanism to guard against irrelevant alignment.

$$\begin{aligned} \boldsymbol{g}_i &= \sigma(\boldsymbol{V}_1 \cdot \boldsymbol{h}_i + \boldsymbol{W}_1 \cdot \bar{\boldsymbol{h}}_j), \\ \boldsymbol{f}^a_i &= (1 - \boldsymbol{g}_i) \odot (\boldsymbol{V}_2 \cdot \boldsymbol{h}_i) + \boldsymbol{g}_i \odot (\boldsymbol{W}_2 \cdot \bar{\boldsymbol{h}}_j), \end{aligned} \tag{9}$$

where $\bar{\boldsymbol{h}}_j = \text{avg}\{\boldsymbol{h}_j | \boldsymbol{h}_j \in \mathcal{N}_a(i)\}$ is the average of the representations of the nodes in the neighbor set $\mathcal{N}_a(i)$, σ is the sigmoid function, $\odot$ denotes element-wise multiplication, $\boldsymbol{g}_i \in \mathbb{R}^d$ serves as the role of gating, and matrices $\boldsymbol{V}_1, \boldsymbol{W}_1, \boldsymbol{V}_2$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d \times d}$ are model parameters. $\boldsymbol{g}_i$ is the gate to control whether the aligned information should contribute to the representation of word i. If the nodes connected by the alignment edge bear very different semantic meanings, the weights in the gate are close to zero, which switch off the information flow.

In addition to cross-lingual word alignment information, the mono-lingual syntax information discloses the inherent dependency among words and thus also benefits the representation of words. The representation $\boldsymbol{f}_i^d$ aggregates the syntax information for node i using a graph attention network (Velickovic et al., 2018) as follows.

$$\boldsymbol{f}_i^d = \sigma(\sum(\alpha_{iu} \boldsymbol{W}_3 \boldsymbol{h}_u, \forall u \in \mathcal{N}_d(i))), \quad (10)$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{d \times d}$ is a model parameter, σ is the sigmoid function, and $\alpha_{iu} \in \mathbb{R}$ is the attention coefficient that indicates the importance of word i to its neighbor u, calculated as follows:

$$\boldsymbol{\alpha}_{iu} = \frac{\exp(LR(\boldsymbol{W}_4[\boldsymbol{h}_i; \boldsymbol{h}_u]))}{\sum_{k \in \mathcal{N}_d(i)} \exp(LR(\boldsymbol{W}_4[\boldsymbol{h}_i; \boldsymbol{h}_k]))}, \quad (11)$$

where LR is the Leaky ReLU activate function, and $\boldsymbol{W}_4 \in \mathbb{R}^{2d}$ is a model parameter.

3.4 Pre-training SA-Graph

To enhance the representation power of the SA-Graph, we use translated parallel data to pre-train the graph (Reid and Artetxe, 2021). The basic idea is to randomly mask some nodes in the graph and use the representations of its semantic and syntactic neighbors to recover it. As can be seen in Fig. 3, for each masked node i, we aggregate the representations of its neighbors. The aggregated representation is then fed into a linear classifier, which outputs the probabilities over the whole vocabulary. Cross entropy is used to compute the recovery loss as $\mathcal{L}_{SA}(i) = -\log P(i|\mathcal{N}(i))$. During the pre-training stage, we propose two masking strategies with each adopted half of the time.

Mono-lingual Masking: Given a source sentence $\boldsymbol{S}$ and the translated sentence $\boldsymbol{T}$, the first masking strategy constrains all the masked tokens to be within only one language, i.e., either the source or the target language. For those masked tokens, since the corresponding words in the other language should not be masked according to the masking constraint, the model can learn from the alignment information to predict the masked ones. In other words, this masking strategy encourages the model to explore the semantic correlation from the alignment information. At each iteration in our implementation, we first choose a language, and then randomly mask 15% of nodes belonging to the chosen language in SA-Graph are masked at random.

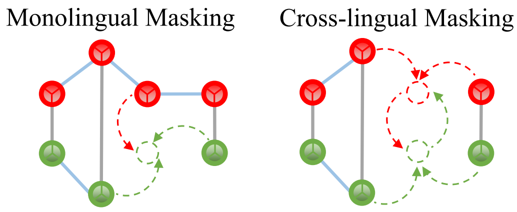


Figure 3: Illustration of the graph masking strategies.

Cross-Lingual Masking: The above mono-lingual masking strategy would make the model tend to ignore the word-dependency edges. To facilitate the model to leverage the syntax information, we further develop a cross-lingual masking strategy: whenever a node is masked, its aligned node must be masked together. In this way, we cut off the alignment information flow, and the model is forced to learn from word-dependency edges to recover the masked nodes.

Besides the above two masking strategies, we also employ translation language modeling (TLM) in our pre-training process, which has shown strong performance in XLM pre-trained model (Lample and Conneau, 2019). For each masked word i, we compute the recovery loss as $\mathcal{L}_{TLM}(i) = -\log P(i|\boldsymbol{h}_i)$. The final loss for the pre-training is the sum of the loss of translation language modeling and our graph masking tasks, i.e., $L(i) = \mathcal{L}_{SA}(i) + \mathcal{L}_{TLM}(i)$.

4 Experiments

We evaluate the proposed *GraFusionMRC* approach on two benchmark datasets. In this section, we first describe the experiment setup. We then report and analyze the experimental results. We also illustrate how SA-Graph affects the attention weights through a case study. The further analysis of our model is presented in Appendix.

4.1 Datasets, Evaluation and Baselines

Datasets: MLQA (Lewis et al., 2020) and TyDiQA-GoldP dataset (Clark et al., 2020) are two recent public benchmark datasets for cross-lingual machine reading comprehension. They contain a total of 14 languages, including English (en), Arabic (ar), German (de), Spanish (es), Hindi (hi), Vietnamese (vi), simplified Chinese (zh), Bengali (bn), Finnish (fu), Indonesian (id), Korean (ko), Russian (ru), Swahili (sw), and Telugu (te). The details of these two datasets are given in the Appendix. Although MLQA and TydiQA provide sufficient test data, their training data is quite limited. Following FILTER (Fang et al., 2021), we use SQuAD v1.1 (Rajpurkar et al., 2016) English training data as additional data during the fine-tuning stage of our model. Moreover, the English training data in SQuAD v1.1 is further translated into the target languages in the MLQA and TyDiQA-GoldP test data via the translation system[3]. Besides, to pre-train our SA-Graph model, we further collect additional parallel sentences following Lample and Conneau (2019); Huang et al. (2019). There are one million pairs of parallel sentences in English and each target language.

Evaluation Metrics: We adopt the standard evaluation metrics from the SQuAD dataset (Rajpurkar et al., 2016), including F1 and Exact Match (EM) scores. The F1 score is used to measure the overlap of tokens between the predicted and ground-truth answer spans, while the EM score only counts the cases where the predicted answer spans exactly match the ground-truth answer spans. We run the official evaluation script provided by MLQA (Lewis et al., 2020) and TyDiQA (Clark et al., 2020) to report the results.

Baselines: We compare *GraFusionMRC* with the following two groups of approaches.

1) Fine-tuning with English training data only: In this group of baselines, we pick the existing cross-lingual models, including **mBERT** (Devlin et al., 2019), **XLM** (Lample and Conneau, 2019), **MMTE** (Siddhant et al., 2020) and **XLM-R** (Conneau et al., 2020). These models are fine-tuned using English training data only.

2) Models using translation: In this group, we first select XLM-R as the representative for cross-lingual models, since it performs the best among all the models in the first group in our experiments for the cross-lingual MRC task. We then fine-tune the XLM-R model with the combined translated training data of all languages jointly, which is denoted as **XLM-R (translate-train)**. We also include the **FILTER** (Fang et al., 2021) as baseline, which leverages the intrinsic cross-lingual correlation between different languages.

4.2 Implementation Details

We implement on top of HuggingFace's Transformers (Wolf et al., 2019) and report results on two both base and large models, i.e., $GraFusionMRC_{base}$ and $GraFusionMRC_{large}$. We initialize our base model by the pre-trained XLM-R base model released by HuggingFace[4], which contains 12 layers; and use XLM-R large model for initializing our large model, which contains 24 layers. We set the number of intermediate Transformer layers, i.e., the Syntax-Enhanced and Alignment-Aware Transformer layers, to 10 in the base model and to 22 in the large model. The first bottom Transformer layer is used for encoding the raw input sentences and the top layer converts the joint representation of the sentences in the source and target languages back to individual language spaces.

To make a fair comparison, we reproduce FILTER based on the XLM-R model. During the fine-tuning stage, following LBMRC (Liu et al., 2020) and XLM-R (Conneau et al., 2020), we pair the passage and question for each language as $[<cls>\ ,question,</s>\ ,</s>\ ,passage,</s>]$ as the input. We then concatenate the input of the target language with the translated input in the source language. Since we employ individual Transformer layers to encode the inputs for each language at the bottom of our model, the supporting max sequence length of each language can be 512. We empirically set the max sequence length to 384 for each language to balance efficiency. Then we concatenate the source and target input and feed a sequence of length 768 to the SA-Transformer to extract their correlation. Limited by space, the fine-tuning details are given in Appendix.

4.3 Experimental Results

We conduct experiments with three variants of our *GraFusionMRC* approach: (1) *GraFusionMRC*+a: only the word-alignment edges are used in graph learning; (2) *GraFusionMRC*+ad: both the word-alignment and word-dependency edges are included in the graph learning stage to obtain the node representation; and (3) *GraFusionMRC*+adp: the pre-training stage is added before the graph learning stage to enhance the representation power of the SA-Graph.

The results on MLQA and TyDiQA-GoldP datasets are presented in Table 1 and Table 2, respectively. In the first group of baselines, the $XLM\text{-}R_{base}$ model consistently outperforms all other baselines in most of the target languages, demonstrating itself as a strong baseline for the

[3] https://console.cloud.google.com/storage/browser/xtreme_translations

[4] https://huggingface.co/xlm-roberta-base

Model	en	ar	de	es	hi	vi	zh	Avg.
mBERT	80.2 / 67.0	52.3 / 34.6	59.0 / 43.8	67.4 / 49.2	50.2 / 35.3	61.2 / 40.7	59.6 / 38.6	61.4 / 44.2
XLM	68.6 / 55.2	42.5 / 25.2	50.8 / 37.2	54.7 / 37.9	34.4 / 21.1	48.3 / 30.2	40.5 / 21.9	48.5 / 32.7
MMTE	78.5 / -	56.1 / -	58.4 / -	64.9 / -	46.2 / -	59.4 / -	58.3 / -	60.3 / 41.4
XLM-R_{base}	78.5 / 65.3	56.1 / 36.8	61.7 / 47.1	66.0 / 48.7	60.1 / 42.4	63.6 / 43.5	60.1 / 35.5	63.7 / 45.6
XLM-R_{base} (translate-train)	**77.8 / 64.4**	58.0 / 38.1	63.4 / 49.1	68.7 / 51.9	62.8 / 46.1	65.3 / 45.9	61.8 / 36.9	65.4 / 47.5
$FILTER_{base}$	77.2 / 63.9	60.2 / 41.2	66.9 / 52.7	70.5 / 53.2	64.5 / 47.2	66.8 / 47.7	63.4 / 42.1	67.1 / 49.7
$GraFusionMRC_{base}$+a	77.7 / 64.0	61.2 / 42.2	67.6 / 53.4	72.9 / 55.7	66.0 / 48.4	67.1 / 48.6	64.6 / 43.3	68.2 / 50.8
$GraFusionMRC_{base}$+ad	77.5 / 63.8	61.9 / 42.8	68.9 / 54.9	73.4 / 56.4	66.6 / 49.2	68.4 / 49.1	65.2 / 44.0	68.8 / 51.5
$GraFusionMRC_{base}$+adp	77.4 / 63.8	**62.8 / 43.4**	**69.3 / 55.3**	**74.0 / 56.8**	**67.1 / 49.5**	**68.8 / 49.5**	**65.6 / 44.3**	**69.3 / 51.8**
XLM-R_{large} (translate-train)	83.5 / 70.6	66.6 / 47.1	70.1 / 54.9	74.1 / 56.6	70.6 / 53.1	74.0 / 52.9	62.1 / 37.0	71.6 / 53.2
$FILTER_{large}$	84.0 / 70.8	72.1 / 51.1	74.8 /60.0	78.1 / 60.1	76.0 / 57.6	78.1 /57.5	70.5 / 47.0	76.2 / 57.7
$GraFusionMRC_{large}$+a	**84.2 / 71.5**	73.0 / 52.0	75.4 / 60.3	78.8 / 60.9	77.9 / 58.4	79.0 / 57.8	71.4 / 48.6	77.1 / 58.5
$GraFusionMRC_{large}$+ad	83.9 / 71.0	73.7 / 52.5	75.9 / 61.2	79.6 / 61.2	78.6 / 58.7	79.9 / 59.8	72.4 / 48.8	77.7 / 59.0
$GraFusionMRC_{large}$+adp	83.5 / 70.7	**74.2 / 52.7**	**76.2 / 61.7**	**80.1 / 62.0**	**79.2 / 59.0**	**80.4 / 60.1**	**73.0 / 49.3**	**78.1 / 59.4**

Table 1: MLQA results (F1 / EM) for each language.

Model	en	ar	bn	fi	id	ko	ru	sw	te	Avg.
mBERT	75.3 / 63.6	62.2 / **42.8**	49.3 / 32.7	59.7 / 45.3	64.8 / 45.8	58.8 / 50.0	60.0 / 38.8	57.5 / 37.9	49.6 / 38.4	59.7 / 43.9
XLM	66.9 / 53.9	59.4 / 41.2	27.2 / 15.0	58.2 / 41.4	62.5 / 45.8	14.2 / 5.1	49.2 / 30.7	39.4 / 21.6	15.5 / 6.9	43.6 / 29.1
MMTE	62.9 / 49.8	**63.1** / 39.2	55.8 / 41.9	53.9 / 42.1	60.9 / 47.6	49.9 / 42.6	58.9 / 37.9	63.1 / 47.2	54.2 / 45.8	58.1 / 43.8
XLM-R_{base}	71.9 / 57.1	54.3 / 32.1	57.8 / 44.6	63.9 / 50.4	68.5 / 51.3	61.2 / 38.7	60.4 / 33.8	65.2 / 55.6	64.9 / 47.6	63.1 / 45.7
XLM-R_{base} (translate-train)	**71.6 / 56.4**	57.8 / 34.5	60.8 / 48.6	67.1 / 53.0	71.9 / 53.7	63.3 / 40.4	62.2 / 34.7	62.8 / 53.7	67.3 / 50.9	65.0 / 47.3
$FILTER_{base}$	68.4 / 55.5	58.3 / 34.6	61.3 / 46.7	67.7 / 54.0	72.2 / 54.5	65.5 / 41.1	63.3 / 35.4	72.3 / 63.1	67.1 / 49.6	66.2 / 48.3
$GraFusionMRC_{base}$+a	71.2 / 55.7	60.1 / 37.1	**62.4 / 48.8**	69.0 / 54.6	73.5 / 57.2	67.2 / 43.5	64.6 / 38.0	**73.9 / 64.1**	68.1 / 53.4	67.8 / 50.3
$GraFusionMRC_{base}$+ad	70.8 / 55.4	61.4 / 38.6	/	69.7 / 55.1	74.9 / 59.0	68.1 / 44.7	64.8 / 36.7	/	70.4 / 53.2	68.6 / 49.0
$GraFusionMRC_{base}$+adp	70.6 / 55.1	62.6 / 39.5	/	**70.4 / 55.7**	**75.7 / 59.3**	**69.0 / 46.1**	**65.5 / 37.2**	/	**71.0 / 53.6**	**69.3 / 49.5**
XLM-R_{large} (translate-train)	**75.1 / 62.0**	66.9 / 39.8	63.8 / 47.5	70.1 / 52.8	77.1 / 61.7	67.8 / 43.4	66.5 / 41.8	65.7 / 47.8	69.6 / 43.4	69.2 / 48.9
$FILTER_{large}$	72.4 / 59.1	72.8 / 50.8	70.5 / 56.6	73.3 / 57.2	76.8 / 59.8	68.9 / 45.7	68.9 / 46.6	77.4 / 65.7	69.9 / 50.4	72.3 / 54.7
$GraFusionMRC_{large}$+a	74.1 / 61.3	73.4 / 51.6	**71.7 / 57.5**	74.1 / 58.0	77.8 / 62.4	69.5 / 46.2	69.8 / 46.7	**78.0 / 65.7**	70.3 / 53.0	73.2 / 55.8
$GraFusionMRC_{large}$+ad	73.9 / 61.2	74.2 / 53.2	/	74.9 / 59.5	79.2 / 64.2	70.5 / 47.5	70.4 / 47.6	/	72.0 / 54.9	73.6 / 55.4
$GraFusionMRC_{large}$+adp	73.5 / 60.8	**75.1 / 53.8**	/	**76.2 / 61.0**	**79.8 / 64.2**	**71.3 / 48.3**	**71.3 / 48.2**	/	**72.5 / 55.7**	**74.2 / 56.0**

Table 2: TyDiQA-GoldP results (F1 / EM) for each language. As the Stanza toolkit does not support languages *Bengali* and *Swahili*, we don't report results on these two languages in the syntactic fusion setting. We correct the ko text segment module of FILTER and bring its performance back from 33.1 to 68.9 of the F1 score.

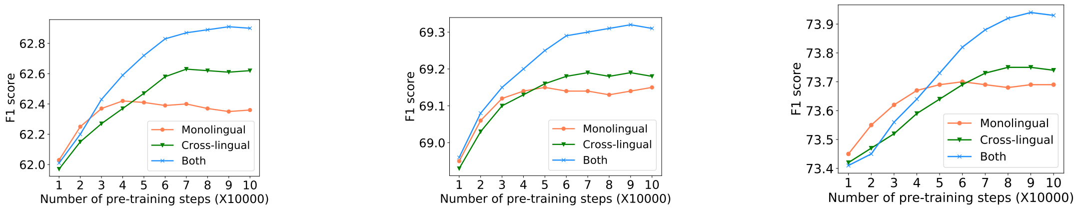
cross-lingual MRC task. Based on this observation, we use XLM-R as the representative for cross-lingual models, and further fine-tune this model with translated training data in target languages.

As shown in the first row ("**XLM-R (translate-train)**") of the second and third group of baselines, adding translated data in target languages substantially improves the model performance, which suggests that the translated data strengthen knowledge transfer effectively. We also observe the FILTER method performs better than the strong baseline XLM-R (translate-train) on both datasets. It indicates that the attention between the source sentence and the translated target sentence leads to better representation of words, and further contributes to the cross-lingual MRC task.

All three variants of our *GraFusionMRC* approach outperform the XLM-R and FILTER models on both datasets. In particular, the *GraFusionMRC*+adp method achieves an average improvement of 2 points over the FILTER model in both MLQA and TyDiQA-GoldP. The major difference of *GraFusionMRC* from FILTER is that we enhance the learning of the attention matrix between the inputs in the source and target languages through explicit syntax and alignment information. Moreover, our gate mechanism and graph attention network increase the model robustness against the errors in alignment and syntactic parsing.

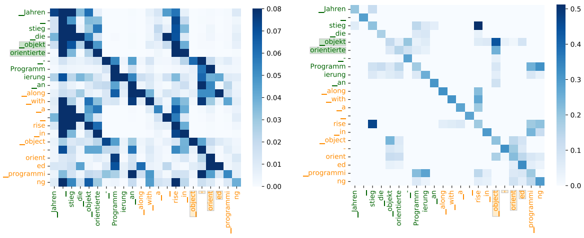
When we compare the three variants of the *GraFusionMRC* approach, the general trend is that using both the syntax edges and alignment edges is better than using alignment edges alone. This justifies the effectiveness of injecting syntax information into representation learning. Moreover, the pre-training using large-scale parallel data also boosts the model performance with a clear gain.

An interesting observation is that our base *$GraFusionMRC_{base}$* model even outperforms the large XLM-R_{large} model in the TyDiQA-GoldP dataset among languages *fi*, *ko*, *sw*, and *te*. The use of alignment and syntactic information successfully bridges the model performance gap caused by the number of parameters, and once again confirms the effectiveness of utilizing SA-Graph.



(a) Arabic (b) German (c) Spanish

Figure 4: Effectiveness of graph masking strategies of *Arabic*, *German* and *Spanish* on MLQA dataset.



(a) FILTER (b) GraFusionMRC

Figure 5: Visualization of different attention matrices. The German and English sentence is presented in green and orange color, respectively. We also highlight the correct answer spans in both German and English.

4.4 Impacts of Different Masking Strategies

In this section, we provide a detailed analysis to better understand the effectiveness of different graph masking strategies. Specifically, we further explore the impacts on *GraFusionMRC*+ad of different choices of graph masking strategies, including using either mono-lingual or cross-lingual masking strategies or employ them both. Fig. 4 shows the F1 scores of *Arabic*, *German*, and *Spanish* on MLQA datasets under different choices of strategies.

It can be observed that in all three languages, the cross-lingual masking strategy performs better, but converges slower than the mono-lingual masking strategy. This may be attributed to the reason that the cross-lingual masking strategy is more complex than mono-lingual, and thus needs more training steps to fit the training data, and provide stronger alignment capture capabilities. Meanwhile, combining both strategies can obtain the best performance, which indicates that those two strategies might focus on different alignment information and thus be complementary to each other.

4.5 Visualization

To showcase the effectiveness of our SA-Graph, we compare the attention distributions from the last fusion layer of the FILTER model with that of our proposed *GraFusionMRC* in Figure 5. The triple of (P,Q,A) in English is (". . . along with a rise in object-oriented programming . . . ", "In the 1990s, what type of programming changed the handling of databases?", "object-oriented"). The original answer in German "objektorientierte" is misaligned to the word "were" in English by the GIZA++ toolkit. With the help of syntactic information, our model is able to learn a higher attention weight between "objektorientierte" and the correct parallel word "object-oriented". The visualization of the example illustrates the benefit of the SA-Graph, which improves knowledge transfer through the enhanced attention matrix.

5 Limitation

The demand for syntactic information of our model limits its application to broader low-resource languages, that are not supported by Stanza. Besides, we propose to use syntactic information to enhance the correlation between different languages because we manually find some alignment errors by solely using the GIZA++ toolkit. Practically, recent works, *e.g.*, AWEsoME (Dou and Neubig, 2021), can provide more robust performance on aligning different language pairs. However, those methods are more sophisticated and have higher computational complexity, this requires a balance between performance and efficiency.

6 Conclusion

In this paper, we develop a novel *GraFusionMRC* approach that leverages both cross-lingual alignment information and mono-lingual syntactic information for cross-lingual MRC. To the best of our knowledge, we are the first to explicitly inject both information to enhance the representation learning in the cross-lingual MRC task. We develop a systematic approach including the construction of the Syntax-Enhanced and Alignment-Aware Graph, the learning algorithms, as well as the pre-training strategies. The experimental results show that our approach outperforms all strong baselines on two public cross-lingual MRC benchmarks.

References

J. Clark, Eunsol Choi, M. Collins, Dan Garrette, T. Kwiatkowski, V. Nikolaev, and J. Palomaki. 2020. Tydiqa: 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 *ACL*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019. Cross-lingual machine reading comprehension. *EMNLP-IJCNLP*, pages 1586–1595.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. *ArXiv*, abs/2101.08231.

Yuwei Fang, Shuohang Wang, Zhe Gan, S. Sun, and Jingjing Liu. 2021. Filter: An enhanced fusion method for cross-lingual language understanding. In *AAAI*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *ICML*.

H. Huang, Yaobo Liang, N. Duan, Ming Gong, Linjun Shou, D. Jiang, and M. Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *EMNLP/IJCNLP*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. Mlqa: Evaluating cross-lingual extractive question answering. In *ACL*.

Xin Li, Lidong Bing, Wenxuan Zhang, Zheng Li, and Wai Lam. 2020. Unsupervised cross-lingual adaptation for sequence tagging and beyond. *ArXiv*, abs/2010.12405.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*.

Junhao Liu, Linjun Shou, Jian Pei, Ming Gong, Min Yang, and Daxin Jiang. 2020. Cross-lingual machine reading comprehension with language branch knowledge distillation. In *COLING*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2020. Rea: Robust cross-lingual entity alignment between knowledge graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2175–2184.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Machel Reid and Mikel Artetxe. 2021. Paradise: Exploiting parallel data for multilingual sequence-to-sequence pretraining. *ArXiv*, abs/2108.01887.

Sebastian Schuster, S. Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *NAACL*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Aditya Siddhant, Melvin Johnson, Henry Tsai, N. Arivazhagan, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. 2020. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. *ArXiv*, abs/1909.00437.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Petar Velickovic, Guillem Cucurull, A. Casanova, A. Romero, P. Liò, and Yoshua Bengio. 2018. Graph attention networks. *ArXiv*, abs/1710.10903.

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

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *EMNLP*, pages 5052–5063, Online.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Fei Yuan, Linjun Shou, Xuanyu Bai, Ming Gong, Yaobo Liang, Nan Duan, Yan Fu, and Daxin Jiang. 2020. Enhancing answer boundary detection for multilingual machine reading comprehension. In *ACL*.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020. Sg-net: Syntax-guided machine reading comprehension. In *AAAI*.

A More Details for the Datasets

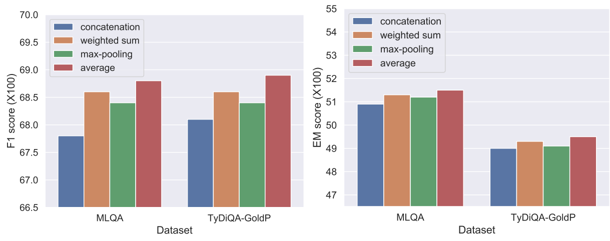
We evaluate our model on two public cross-lingual machine reading comprehension datasets: MLQA (Lewis et al., 2020) and TyDiQA-GoldP dataset (Clark et al., 2020).

- **MLQA** (Lewis et al., 2020), is a cross-lingual machine reading comprehension benchmark that covers 7 languages, including *English, Arabic, German, Spanish, Hindi, Vietnamese* and *Simplified Chinese*. The number of question-answering instances in the test set for those languages is 11590, 5335, 4517, 5254, 4918, 5495, and 5137, respectively.

- **TyDiQA-GoldP** (Clark et al., 2020), is another cross-lingual machine reading comprehension benchmark covering 9 typologically diverse languages, including *English, Arabic, Bengali, Finnish, Indonesian, Korean, Russian, Swahili*, and *Telugu*. The number of question-answering instances in the development set for those languages is 440, 921, 113, 782, 565, 276, 812, 499, and 669, respectively.

B Comparison of Different Combinations

In this subsection, we study the effect of different aggregation functions in our *GraFusionMRC*+ad model. For each node i in SA-Graph, to aggregate the $\boldsymbol{f}_i^a$ and $\boldsymbol{f}_i^d$ vector, we experiment with three different types of aggregation functions: *concatenation*, *weighted sum*, and *max-pooling*. Please note that, for concatenation operation, we need an extra trainable matrix $\boldsymbol{W}_{cat} \in \mathbb{R}^{2d \times d}$ to project the vector back to the dimension of d for consistency. For better comparison, we also report the results of the average operation, which is used in our model.



(a) Average F1 scores (b) Average EM scores

Figure 6: Further analysis on the choice of different aggregation functions using the *GraFusionMRC*+ad model.

As shown in Figure 6, the average function tends to show the best performance on both MLQA and TyDiQA-GoldP datasets, which indicates that the average operation is a simple but effective method to aggregate the cross-lingual and mono-lingual correlation. We can also see that the concatenation operation shows the worst. A possible explanation could be that the introduction of the matrix $\boldsymbol{W}_{cat}$ increases the computational complexity of the aggregation process, making it more difficult to find the global optimal solution. Another interesting observation is that the performance of the weighted sum operation outperforms the max-pooling operation. This may be because the weighted sum operation could capture the fine-grained correlation between vectors while the max-pooling operations only capture the significant information.

C Fine-tuning Details

We select the hyperparameters from batch size: {16, 32, 64}, learning rate: {1e-5, 3e-5, 5e-5, 1e-6, 5e-6}, and warmup rate: {5%, 10%, 15%}. All experiments are conducted on 4 16G NVIDIA P100 GPUs. Each experiment is repeated 10 times and the average results are reported. The number of parameters of FILTER, *GraFusionMRC*+a and *GraFusionMRC*+ad model are 1.02, 1.07 and 1.15 times that of XLM-R. For MLQA dataset, the best performance of *GraFusionMRC*+adp model is achieved at batch size=32, learning rate=5e-5, warmup rate=10%. It takes about 8 hours to get the best result running on 4 16G P100. As for the TyDiQA-GoldP dataset, we achieve the best results at batch size=64, learning rate=3e-5, warmup rate=5%. It takes about 10 hours to get the best result running on 4 16G P100.