Adapters for Enhanced Modeling of Multilingual Knowledge and Text

Yifan Hou[1], Wenxiang Jiao[2], Meizhen Liu[3], Carl Allen[1], Zhaopeng Tu[2], Mrinmaya Sachan[1]
[1]ETH Zürich, [2]Tencent AI Lab, [3]Shandong University
[1]{yifan.hou, carl.allen, mrinmaya.sachan}@inf.ethz.ch
[2]{joelwxjiao, zptu}@tencent.com, [3]meizhen.liu@mail.sdu.edu.cn

Abstract

Large language models appear to learn facts from the large text corpora they are trained on. Such facts are encoded implicitly within their many parameters, making it difficult to verify or manipulate what knowledge has been learned. Language models have recently been extended to multilingual language models (MLLMs), enabling knowledge to be learned across hundreds of languages. Meanwhile, knowledge graphs contain facts in an explicit *triple* format, which require careful and costly curation and are only available in a few high-resource languages, restricting their research and application. To address these issues, we propose to enhance MLLMs with knowledge from multilingual knowledge graphs (MLKGs) so as to tackle language and knowledge graph tasks across many languages, including low-resource ones. Specifically, we introduce a lightweight *adapter set* to enhance MLLMs with cross-lingual entity alignment and facts from MLKGs for many languages. Experiments on common benchmarks show that such enhancement benefits both MLLMs and MLKGs, achieving: (1) comparable or improved performance for knowledge graph completion and entity alignment relative to baselines, especially for low-resource languages (for which knowledge graphs are unavailable); and (2) improved MLLM performance on language understanding tasks that require multilingual factual knowledge; all while maintaining performance on other general language tasks.[1]

1 Introduction

Knowledge graphs serve as a source of explicit factual information for various NLP tasks. However, language models (Devlin et al., 2019; Brown et al., 2020), which capture *implicit* knowledge from vast text corpora, are already being used in knowledge-intensive tasks. Recently, language models have been successfully extended to multilingual language models (MLLMs) that integrate information sourced across hundreds of languages (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020). However, as with most neural networks, the information is encoded in a diffused and opaque manner that is difficult to interpret, verify or utilize (AlKhamissi et al., 2022).

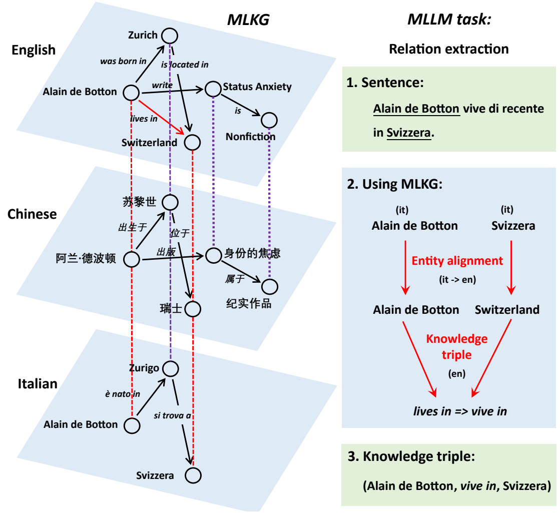

Figure 1: Combining MLLMs and MLKGs benefits both: MLKGs suffer from incompleteness and are limited to few languages, which MLLMs can supplement. MLLMs lack entity alignment and firm facts, which MLKGs can provide.

Meanwhile, multilingual knowledge graphs (MLKGs) require careful curation of explicit facts and annotation of entities that occur in languages (cross-lingual entity alignment), making knowledge graphs expensive and time-consuming to extend to new languages, restricting knowledge graph research to a few high-resource languages. Further, open-source MLKGs such as WordNet (Bond and Foster, 2013) and Wikidata (Vrandečić and Krötzsch, 2014) suffer from *incompleteness* as many true facts (or *triples*) and entity alignments are missing (Chen et al., 2017, 2020).

In this work, we propose to overcome the above limitations of each knowledge source by integrating MLKGs into MLLMs (as shown in Figure 1),

[1] Our code, models, and data (e.g., integration corpus and extended datasets) are available at https://github.com/yifan-h/Multilingual_Space.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3902–3917
December 7-11, 2022 ©2022 Association for Computational Linguistics

to enable (i) the transfer of MLKG knowledge from high-resource languages to low-resource languages; and (ii) explicit knowledge of MLKGs to supplement MLLMs for knowledge-intensive language tasks, one of the key challenges in MLLMs (AlKhamissi et al., 2022).

While this idea seems intuitive, there is no easy way to incorporate the explicit knowledge of MLKGs into the parametrically stored information of MLLMs. Existing knowledge integration methods utilize language models and knowledge graphs in two ways: (1) training knowledge graph embeddings individually and combining the embeddings corresponding to linked entities in sentences with the language model representations (e.g., KnowBERT (Peters et al., 2019) and ERNIE (Zhang et al., 2019)); or (2) absorbing the knowledge in knowledge graphs into the language model's parameters via joint training (e.g., K-BERT (Liu et al., 2020) and K-Adapter (Wang et al., 2021)).

The first method requires embedding knowledge graph entities and accurately extracting entities in sentences across hundreds of languages, which is highly challenging. The second method typically suffers from the *curse of multilinguality* (Conneau et al., 2020; Doddapaneni et al., 2021; Jiao et al., 2022) and *catastrophic forgetting* (Kirkpatrick et al., 2016) due to limited model capacity. Most importantly, both methods integrate knowledge implicitly such that it is difficult to access and extend to low-resource languages (AlKhamissi et al., 2022). Furthermore, both methods require large sets of aligned sentences and knowledge triples, which is costly to gather and accurately annotate across hundreds of languages.

To address above issues, we first collect and clean multilingual data from Wikidata[2] and Wikipedia[3] for the enhancement, where rich factual knowledge and cross-lingual alignments are available. Then, we propose to enhance MLLMs with the MLKG information by using a set of *adapters* (Houlsby et al., 2019), which are lightweight, collectively having only around 0.5% extra parameters than the MLLM. Each adapter integrates information from either MLKG **T**riples (i.e. facts) or cross-lingual **E**ntity alignments, and is trained on either **P**hrase or **S**entence level data. Each of the resulting four adapters (EP/TP/ES/TS) is trained individually to learn information supplemental to that already learned by the MLLM. Adapter outputs are combined by a *fusion* mechanism (Pfeiffer et al., 2021). Training objectives are similar to those for MLKG embedding (Chen et al., 2017) instead of mask language modeling, which are more efficient with large corpus.

We conduct experiments on various downstream tasks to demonstrate the effectiveness of our approach. For MLKG tasks, following the data collection methods of two existing benchmarks (Chen et al., 2020, 2017), we extended them from 2-5 languages to 22 languages, including two rare languages.[4] Results show that our method obtains comparable performance to existing state-of-the-art baselines on the knowledge graph completion benchmark, and significantly better performance on the entity alignment benchmark. More importantly, we can perform these knowledge graph tasks in low-resource languages for which no knowledge graph exists, and achieve comparable results to the high-resource languages. Improvements over baseline MLLMs are significant. The results demonstrate that our proposed method integrates the explicit knowledge from MLKGs into MLLMs that can be used across many languages. Our method also improves existing MLLMs noticeably on knowledge-intensive language tasks, such as cross-lingual relation classification, whilst maintaining performance on general language tasks such as named entity recognition (NER) and question answering (QA).

2 Multilingual Knowledge Integration

In this paper, we fuse knowledge from a MLKG into a MLLM. Following previous works (Wang et al., 2021; Liu et al., 2021), we make use of an entity tagged corpus of text (called a *knowledge integration corpus*) for knowledge integration. We formally introduce these concepts below.

MLLM. A multilingual LM can be thought of as an encoder that can represent text in any language l in a set of languages $\mathcal{L}$. Let $\mathcal{V}$ denote the shared vocabulary over all languages. Let $t^l \in \mathcal{V}$ denote a token in language l. A sentence s^l in a language l can be denoted as a sequence of tokens: $s^l = (t^l_1, t^l_2, ...)$. The output representations of the MLLM for s^l can be denoted by a sequence of vectors: $\mathrm{LM}(s^l) = (\boldsymbol{h}_1, \boldsymbol{h}_2, ...)$. These vectors correspond to representations for each token in the

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page

[3] https://en.wikipedia.org/wiki/Main_Page

[4] The extended datasets as well as KI corpus are published with our code implementation.

sentence, one representation per input token. Various tokenization schemes such as wordpiece or BPE might be considered here. We use the average of the token representations as the representation of the sentence: $\overline{\mathrm{LM}(s^l)} = \mathrm{mean}(\boldsymbol{h}_1, \boldsymbol{h}_2, ...)$. Similarly, for a phrase s^l_{ij} (starting from the i-th token and ending in the j-th token in the sentence), we can obtain its contextualized representation as $\overline{\mathrm{LM}(s^l_{ij})} = \mathrm{mean}(\boldsymbol{h}_i, \boldsymbol{h}_{i+1}, \ldots \boldsymbol{h}_j)$.

MLKG. A multilingual knowledge graph is a graph with entities and knowledge triples in each language $l \in \mathcal{L}$. Let $\mathcal{E}$ denote the set of entities and $\mathcal{T}$ denote the set of knowledge triples. In a MLKG, each entity indexed i might appear in several languages. Let e^l_i denote the entity label of the i-th entity in language l. Furthermore, we denote a knowledge triple in the MLKG as $(e^l_i, r^{l''}_k, e^{l'}_j) \in \mathcal{T}$, where $r^{l''}_k$ is the k^{th} relation. Note that since entities (as well as relations) may appear in various languages under different labels, knowledge triples can be defined across languages.

Knowledge Integration Corpus. For knowledge integration, besides the MLKG, we make use of a corpus of text $\mathcal{C}$ (as shown in the right part of Figure 2). The corpus $\mathcal{C}$ comprises of two kinds of texts. First, we have a set of texts $\mathcal{C}_1$ for the cross-lingual entity alignment, which comprise of sentences with mentions of entities in the MLKG. For example in Figure 2, given the sentence *De Botton spent his early years in Zurich*, we have the aligned entity *Zurich* and its cross-lingual labels. The second set of texts $\mathcal{C}_2$ is for the knowledge triple, which comprises of sentences aligned with knowledge triples in the MLKG. For example in Figure 2, given the sentence *Zurich is the largest city in Switzerland*, we have its aligned knowledge triple *(Zurich, is located in, Switzerland)*.

3 Adapters and Adapter Fusion

In this section, we first describe how we incorporate adapters into language models and how they can be used to enhance them with different sources of knowledge from knowledge graphs.

Adapter. Adapters have become a popular choice for parameter-efficient finetuning of language models on downstream tasks (Houlsby et al., 2019) due to their flexibility, effectiveness, low cost and scalability (Pfeiffer et al., 2021). Adapters are new modules that are added between layers of language models[5], the parameters of which are updated only during finetuning while the language model parameters are frozen. An adapter is a bottleneck layer composed of two feed-forward layers with one non-linear activation function. For $\boldsymbol{h}^m$, the hidden representation of token t^l_i at layer m, the adapter acts as

$$\mathrm{A}(\boldsymbol{h}^m) = \boldsymbol{W}_{\text{up}} \cdot \sigma(\boldsymbol{W}_{\text{down}} \cdot \boldsymbol{h}^m + \boldsymbol{b}_{\text{down}}) + \boldsymbol{b}_{\text{up}}. \quad (1)$$

Here, $\boldsymbol{W}_{\text{down}}$ and $\boldsymbol{W}_{\text{up}}$ are weight matrices, which map the hidden representations to the low-dimensional space and then map them back. $\boldsymbol{b}_{\text{down}}$ and $\boldsymbol{b}_{\text{up}}$ are bias parameters, and σ is a nonlinear activation function.

Adapter Fusion. We follow the architecture of Pfeiffer et al. (2021), but instead of using adapters for finetuning, we use them to enhance MLLMs with knowledge. Our approach is similar to Wang et al. (2021), but our adapters supplement and augment the existing implicit knowledge of MLLMs (into the explicit geometric properties of hidden representations), And our approach is more lightweight, with only c.0.5% additional parameters (*cf* $>10\%$ in Wang et al. (2021)).

As shown in Figure 2 (left), still considering the m-th layer, the output representations of the feedforward layer (denoted $\boldsymbol{h}^m$ as in Eq. 1) are input to the adapters. A fusion layer aggregates all adapter outputs $\mathrm{A}_n(\boldsymbol{h}^m)$ ($n \in \{1...N\}$ indexes each adapter) and the un-adapted representations with a multiplicative attention mechanism:

$$\mathrm{A}_{\text{fusion}}(\boldsymbol{h}^m) = \sum_{n=0}^{N} a^m_n \cdot \boldsymbol{V}^m \cdot \mathrm{A}_n(\boldsymbol{h}^m),$$
$$a^m_n = \text{softmax}(\boldsymbol{h}^m \boldsymbol{Q}^m \otimes \mathrm{A}_n(\boldsymbol{h}^m) \boldsymbol{K}^m).$$

Here, $\mathrm{A}_0(\cdot)$ is the identity function; $\boldsymbol{Q}^m$, $\boldsymbol{K}^m$, $\boldsymbol{V}^m$ are parameters in the multiplicative attention mechanism; and $\otimes$ is the Hadamard product.

The additional knowledge to be learned by the adapters comes from knowledge **T**riples and **E**ntity alignments, each provided in both **P**hrase and **S**entence format (hence $N = 2 \times 2 = 4$). As shown in Figure 2 (center), for a given entity in two languages l and l', Adapter-EP. learns to align the two (multilingual) representations of e^l_i and $e^{l'}_i$, e.g., *Zurich* is aligned with *Zurigo*. Adapter-TP. learns knowledge triples, e.g., predicting *Switzerland* given entity and relation *(Zurich, is located*

[5] Where to insert adapters is flexible but a common choice is after the feedforward layer of a transformer layer.

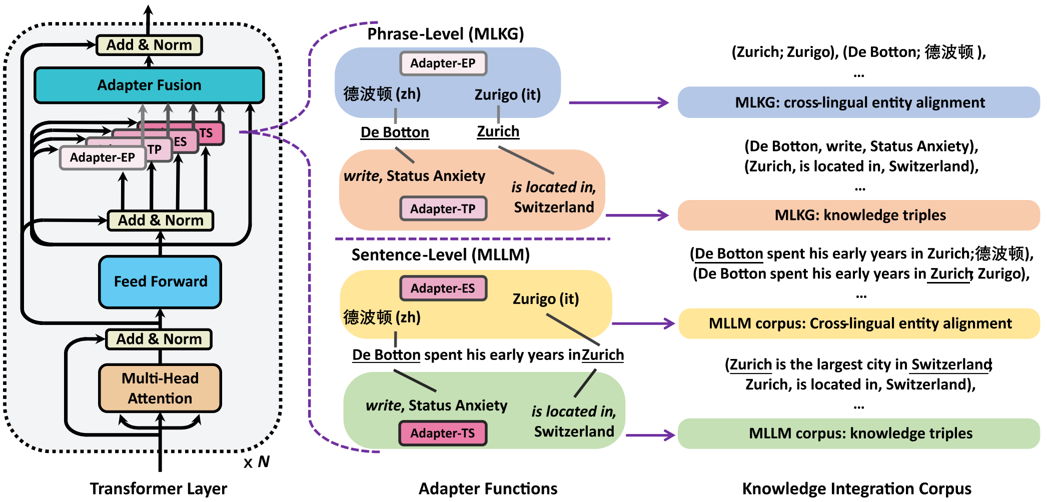
Add & Norm
Adapter Fusion
TS
ES
TP
Adapter-EP
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
x N
Transformer Layer
Phrase-Level (MLKG)
Adapter-EP
德波顿 (zh)
Zurigo (it)
De Botton
Zurich
write, Status Anxiety
is located in, Switzerland
Adapter-TP
Sentence-Level (MLLM)
Adapter-ES
Zurigo (it)
德波顿 (zh)
De Botton spent his early years inZurich
write, Status Anxiety
is located in, Switzerland
Adapter-TS
Adapter Functions
(Zurich; Zurigo), (De Botton; 德波顿),
...
MLKG: cross-lingual entity alignment
(De Botton, write, Status Anxiety),
(Zurich, is located in, Switzerland),
...
MLKG: knowledge triples
(De Botton spent his early years in Zurich;德波顿),
(De Botton spent his early years in Zurich; Zurigo),
...
MLLM corpus: Cross-lingual entity alignment
(Zurich is the largest city in Switzerland;
Zurich, is located in, Switzerland),
...
MLLM corpus: knowledge triples
Knowledge Integration Corpus

Figure 2: The architecture of MLLMs with adapters and their roles. We enhance multilingual and factual knowledge in phrase and sentence levels using different knowledge integration corpus.

in,). Besides these *non-contextualized* settings, entities within context can be considered also (MLLM corpus). Thus, Adapter-ES. and Adapter-TS. have the similar objectives but use contextualized representations from input sentences.

4 Knowledgeable Adapters

Next, we design objectives with corresponding knowledge integration datasets to train a set of adapters. Similar to MLKG embedding (Chen et al., 2017), we aim to encode knowledge into the geometric properties of the *adapted* MLLM representations, i.e., the MLLM and adapters collectively act as an MLKG embedding model. Specifically, we use cosine distance within the contrastive learning loss of InfoNCE (van den Oord et al., 2018):

$$\mathrm{INCE}(\boldsymbol{x},\boldsymbol{x}') = \log \frac{\cos(\boldsymbol{x},\boldsymbol{x}')}{\sum_{\boldsymbol{x}''\in X}\cos(\boldsymbol{x},\boldsymbol{x}'')},$$

where X is a batch that includes the positive sample $\boldsymbol{x}'$ and a number of negative samples.[6]

Adapter-EP. We use Wikidata (Vrandečić and Krötzsch, 2014) to enhance MLLMs with the knowledge of cross-lingual entity alignments. Inspired by the idea that languages are aligned implicitly in a universal space in MLLMs (Wu and Dredze, 2019; Wei et al., 2021), we train the aligned entities to have closer representations. Denoting the MLLM with this adapter as $\mathrm{LM}(\cdot)$, the objective used to train EP is:

$$\mathcal{L}_{\mathrm{EP}} = \sum_{(e_i^l, e_i^{l'})\in\mathcal{E}} \mathrm{INCE}\Big(\overline{\mathrm{LM}(e_i^l)}, \overline{\mathrm{LM}(e_i^{l'})}\Big),$$

where $\overline{\mathrm{LM}(\cdot)}$ means we take the mean of token representations as the entity representation vector.

Adapter-TP. We train this adapter using the knowledge triples in Wikidata. Inspired by previous knowledge graph embedding algorithms (e.g. Bordes et al., 2013), for a given fact triple, we train the (adapted) object entity embedding to be close to the (adapted) joint embedding of the subject entity and relation. The objective used to train TP is quite different from existing mask language modeling-based ones:

$$\mathcal{L}_{\mathrm{TP}} = \sum_{(e_i^l, r_k^{l''}, e_j^{l'})\in\mathcal{T}} \mathrm{INCE}\Big(\overline{\mathrm{LM}([e_i^l; r_k^{l''}])}, \overline{\mathrm{LM}(e_j^{l'})}\Big),$$

where $[;]$ denotes text concatenation. Note that we apply *code-switching* (Liu et al., 2021), and thus entities and relations can be in different languages. This is helpful to capture knowledge triples for low-resource languages.

Adapter-ES. Entity alignment can also be applied to contextualized embeddings produced by the MLLM when entities are input within natural language sentences. For this purpose, we use summaries taken from multilingual Wikipedia. Specifically, we first align the entity in Wikidata with the Wikipedia title, and extract sentences that contain the entity label in its summary. As described earlier, we denoted this corpus as $\mathcal{C}_1$. Thus, similar to Adapter-EP, we train ES by aligning contextualized entity representations of cross-lingually aligned entities with the objective:

$$\mathcal{L}_{\mathrm{ES}} = \sum_{(e^{l'}, s^l)\in\mathcal{C}_1} \mathrm{INCE}\Big(\overline{\mathrm{LM}(s_{ij}^l)}, \overline{\mathrm{LM}(e^{l'})}\Big),$$

[6] We use *in-batch* negative sampling, where entities (with labels in any languages) in the batch are randomly selected.

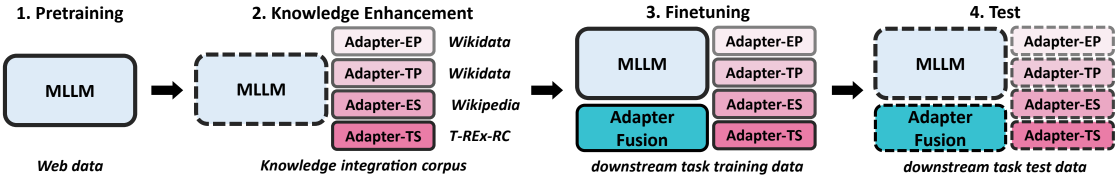

Figure 3: Four stages of using the knowledge adapter set in MLLMs. The dashed outlines mean the parameters are frozen.

where s^l_{ij} means that we input sentence s^l into an MLLM but keep only the representation of entity label e^l (indexed from i-th token to j-th token). As in Figure 2 (right), s^l is: *De Botton spent his early years in Zurich*, and s^l_{ij} here is the entity label of e^l as: *Zurich*. The difference between this adapter and Adapter-EP is that contextual information is included within the entity representation.

Adapter-TS. Knowledge triples can also be learned with contextualized embeddings. This requires paired data in which triples (entities and relations) are annotated in natural sentences. However, no such multilingual corpus exists. Thus, we use the T-REx-RC dataset (Elsahar et al., 2018)[7], which provides aligned data in English and contains sentence and triple pairs. Thus, the objective used to train TS is:

$$\mathcal{L}_{\text{TS}} = \sum_{(s_k,(e_i,r,e_j))\in\mathcal{C}_2} \text{INCE}\Big(\overline{\text{LM}(s_k\backslash e_j)}, \overline{\text{LM}(e_j)}\Big),$$

where $s_k\backslash e_j$ represents the sentence s_k with entity label e_j masked. As the example in Figure 2 (right), $s_k\backslash e_j$ is: *[MASK] is the largest city in Switzerland*, and the aligned triple is: *(Zurich, is located in, Switzerland*. In contrast to Adapter-TP, subject entities and relations occur in natural sentences.

4.1 Enhancement Workflow

We introduce our overall enhancement workflow, which contains four stages. In the first stage, an MLLM is *pretrained* on a large amount of data. In the second stage, the MLLM is frozen while each adapter is trained separately on its particular dataset (knowledge integration corpus) to extract additional information. Adapter outputs are aggregated in the fusion layer to enable their collective knowledge to be pooled (Pfeiffer et al., 2021). For example, we lack knowledge graph data for low-resource languages, however we have two adapters (TP, TS) that learn facts in a particular language (English) and two adapters (EP, ES) that learn cross-lingual alignment. By aggregating them, we can effectively integrate factual knowledge into the representations of low-resource languages. In the third and final stages, all parameters of the MLLM, the adapters, and the fusion module are *finetuned* on a training set for a specific downstream task resulting in a specialized model for the task (see Figure 3).

5 Experiments

This section first introduces the general experimental settings (§5.1). We then show that our adapter set can enhance MLLMs with the knowledge of MLKGs and, in particular, that the enhanced MLLMs generalize well to perform MLKG-related tasks in low-resource languages (§5.2). We also show that enhancing MLLMs with MLKGs improves their performance on knowledge-intensive language tasks (§5.3). We compare our approach with the only existing MLKG integration work (§5.4). Finally, we present an ablation study of the adapter set to demonstrate the effectiveness of each adapter (§5.5).

5.1 MLLMs and Integration Corpus

We select three representative MLLMs implemented by Huggingface[8] and train a set of adapters for each: the base version of mBERT (Devlin et al., 2019), and both the base and large versions of XLMR (i.e., XLM-RoBERTa) (Conneau et al., 2020). Since mBERT and XLMR cover different sets of languages, we consider the intersecting 84 languages supported by both models. All adapters are trained with the same hyperparameters (see Appendix A for details).

Table 1: Statistics of knowledge integration corpora for training adapters. *Align.*: all aligned multilingual entities; *Relat.*: all relations in triples; *Sent.*: sentences.

Module	Source	Statistics
Adapter-EP	Wikidata (MLKG)	Entity / Align.: 1.55M / 63.25M
Adapter-TP	Wikidata (MLKG)	Triple / Relat.: 9.42M / 1422
Adapter-ES	Wikipedia ($\mathcal{C}_1$)	Entity / Sent.: 0.20M / 1.93M
Adapter-TS	T-REx-RC ($\mathcal{C}_2$)	Triple-Sent. Pair: 0.97M

The statistics of the knowledge integration corpora are summarized in Table 1. Next, we introduce their preprocessing steps. The set of entity alignments used to train **Adapter-EP** is extracted from Wikidata by keeping only entities that have more

[7] We denoted this aligned corpus earlier by $\mathcal{C}_2$

[8] https://huggingface.co/

than 10 multilingual entity labels among the 84 considered languages. Knowledge graph triples are used to train **Adapter-TP** if both entities are in that entity set (see Table 8 of Appendix B for further details). For the Wikipedia dataset, we use entities in the Wikidata subset and query their descriptions (the first sentence in the Wikipedia summary that contains the entity label). We remove entities that have less than 2 multilingual descriptions, which results in 1.93 million multilingual sentences to train **Adapter-ES**. For **Adapter-TS**, we use the monolingual dataset T-REx-RC (Elsahar et al., 2018), which has 0.97 million alignments between knowledge triples and sentences in English.

5.2 MLKG Benchmarks

We show that *our knowledge adapter set can enhance MLLM performance at MLKG-related tasks*. We select two popular MLKG benchmarks for evaluation: DBP5L (Chen et al., 2020) for the knowledge graph completion task, and WK3L (Chen et al., 2017) for the cross-lingual entity alignment task. These tasks require the MLLM to identify the correct entity, which is performed by maximizing the similarity of output representations.

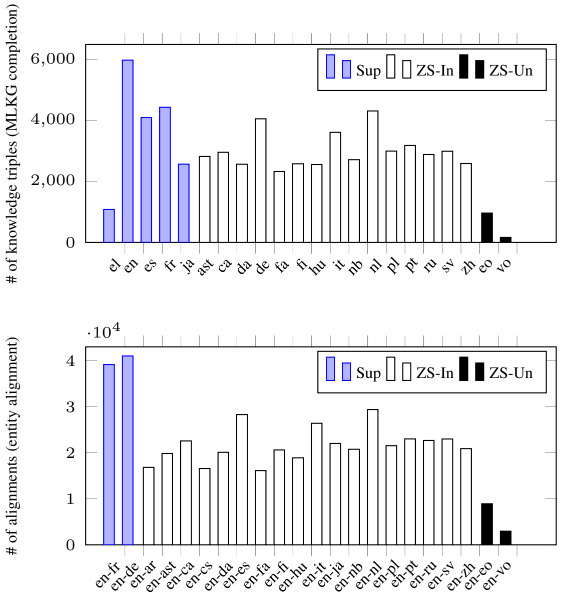

Figure 4: Statistics of the size of test sets for MLKG completion and entity alignment tasks. We can see that extended test sets for zero-shot languages have comparable number of samples as original test sets.

To evaluate MLLMs in a more comprehensive setting, we extend their test sets (from $2-5$ languages) to 22 languages following their data construction settings[9], where languages that contain the most entity labels are selected. Statistics are in Figure 4. We split these languages into three categories to show the generalizability of enhanced MLLMs: **Sup.**: *supervised languages*, which are used to train adapters and for finetuning; **ZS-In**: *zero-shot languages*, which are used for adapter training but not for finetuning; **ZS-Un.**: *unseen languages*, which are unseen in both adapter training and finetuning.

5.2.1 Knowledge Graph Completion

The knowledge graph completion task tests if the model can find the missing triples in different languages. Specifically, for each test triple of a given language, the model is asked to retrieve the correct object entity from the entity set of that language given the subject entity and relation.

Settings. We follow the settings of DBP5L.[10] Specifically, we use the training set of knowledge triples of the five languages (i.e. the Sup. set) to finetune the model, and then use the provided test sets, as well as our extended test sets to evaluate it. For comparison, we select two typical knowledge graph embedding methods, TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015), as baselines and compare the performance of MLLMs and MLLMs-A_{Fusion}, enhanced with the knowledge adapter and fusion mechanism (see Appendix A for further implementation details).

Table 2: Results on the knowledge graph completion task. We attach the number of languages to each type. We can see that for zero-shot languages and unseen languages, using our adapters can significantly improve the performance of LMs on knowledge graph completion.

Model	**Sup. (5)**		**ZS-In (15)**		**ZS-Un. (2')**	
	Hit@1$^{\uparrow}$	MRR$^{\uparrow}$	Hit@1$^{\uparrow}$	MRR$^{\uparrow}$	Hit@1$^{\uparrow}$	MRR$^{\uparrow}$
TransE	**14.5**	**23.7**	-/-	-/-	-/-	-/-
DistMult	8.1	14.3	-/-	-/-	-/-	-/-
mBERT	11.2	13.8	12.8	15.7	48.2	49.1
mBERT-A_{Fusion}	13.1	15.7	**16.1**	**18.8**	**51.8**	**52.4**
XLMR$_{\text{base}}$	5.9	7.8	6.7	9.1	8.2	11.8
XLMR$_{\text{base}}$-A_{Fusion}	9.1	11.8	10.6	13.5	16.6	19.6
XLMR$_{\text{large}}$	7.3	9.7	8.9	11.5	16.8	20.8
XLMR$_{\text{large}}$-A_{Fusion}	13.1	15.6	14.3	17.3	23.9	27.4

Results. Results are summarized in Table 2 (with further detail in Table 9 of Appendix C). We report both Hit@1 score and Mean Reciprocal Rank (MRR) for evaluation. We find that enhancing MLLMs with adapters can improve performance for the *supervised* languages, which is comparable to existing knowledge graph embedding meth-

[9] We follow settings of data collection pipelines described in Chen et al. (2020, 2017) for the extension.

[10] Note that some entity alignments across 5 languages are provided. We only consider the triple data for simplicity and test entity alignment with another benchmark: WK3L.

ods. For the *zero-shot* languages and *unseen* languages, existing (transductive) knowledge graph embedding methods cannot perform the task since entities must be in the training set. Here we find that MLLMs still perform comparably to the supervised languages[11], and the enhanced MLLMs-A_{Fusion} models outperform MLLMs on zero-shot languages by significant margins. This indicates that the adapters allow factual knowledge to be transferred across languages.

5.2.2 Entity Alignment

The entity alignment task is to align entities in different languages. Specifically, given a target language and an entity in a source language (typically English), the model should retrieve that entity from the set of all entities in the target language.

Settings. We follow settings of WK3L.[12] Specifically, we train models using the entity alignments English to German, and English to French. We test models on those two *supervised* languages, as well as our extended 17 *zero-shot* languages and 2 *unseen* languages.[13] We select one typical MLKG embedding method, MTransE (Chen et al., 2017), and a state-of-the-art method, JEANS (Chen et al., 2021), as baselines (see Appendix A for details).

Table 3: Results on multilingual entity alignment tasks. We can find that using our adapters can significantly enhance MLLMs' performance on entity alignment tasks, which also outperforms existing MLKG embedding baselines.

Model	Sup. (2)		ZS-In (18)		ZS-Un (2')	
	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
MTransE	8.7	12.5	-/-	-/-	-/-	-/-
JEANS	40.0	47.5	-/-	-/-	-/-	-/-
mBERT	83.6	83.2	31.8	32.2	50.5	50.8
mBERT-A_{Fusion}	88.9	88.4	77.6	76.2	**91.7**	**89.3**
XLMR$_{\text{base}}$	54.8	54.8	9.4	9.7	10.9	11.1
XLMR$_{\text{base}}$-A_{Fusion}	88.6	88.0	82.4	81.8	85.4	84.3
XLMR$_{\text{large}}$	65.0	65.1	23.9	24.1	28.9	28.9
XLMR$_{\text{large}}$-A_{Fusion}	**90.2**	**89.5**	**90.8**	**89.0**	89.8	88.5

Results. The results are summarized in Table 3 (with further detail in Table 10 of Appendix C). Performance is evaluated again by Hits@1 and MRR. As previously, the (transductive) baselines cannot be extended to languages not in the training set. For the *supervised* languages, we can find that existing MLLMs often outperform classic baselines. However, performance of MLLMs on *zero-shot* languages is noticeably worse. This indicates that existing MLLMs do not transfer entity alignment knowledge well to other languages. However, MLLMs enhanced with the adapter set, MLLMs-A_{Fusion}, generally achieve the best performance, often with significant improvement. The results indicate that our adapter set successfully enhances MLLMs with multilingual knowledge.

5.3 MLLM Benchmarks

Above results show that our adapter set can enhance MLLMs to perform well on MLKG-related tasks on both previously seen and unseen languages. Here, we show that *our knowledge adapter set can allow MLKGs to enhance MLLM performance on language tasks*. In particular, the enhanced MLLMs achieve improved performance on knowledge-intensive language task while maintaining performance on other general language tasks.

5.3.1 Cross-Lingual Relation Classification

We select a popular relation classification benchmark: RELX (Köksal and Özgür, 2020), for which MLLMs must extract relations from sentences in a cross-lingual setting. Models are finetuned on a high-resource corpus, and tested on low-resource languages in a zero-shot setting. For this task, MLLMs are required to transfer the knowledge across languages, as well as capture factual knowledge for the relation classification.

Settings. Our training data is only in English, and test data contains 4 more (zero-shot) languages. We follow the exact setting of Köksal and Özgür (2020) and use the same provided set of hyperparameters to evaluate all MLLMs. We also report the performance of the enhanced BERT model of Köksal and Özgür (2020) called Matching the Multilingual Blanks (MTMB) as a baseline.

Table 4: Results on the multilingual relation classification task (F1 score). We can find that our adapters can effectively enhance MLLMs on the knowledge-intensive downstream tasks, especially for the performance on zero-shot languages.

Model	Sup. (En)	ZS-In (4)	Ave.
mBERT	61.8	57.4	58.3
MTMB	63.6	59.6	60.4
mBERT-A_{Fusion}	**64.0**	**60.9**	**61.5**
XLMR$_{\text{base}}$	**61.4**	56.1	57.1
XLMR$_{\text{base}}$-A_{Fusion}	61.3	**58.0**	**58.6**
XLMR$_{\text{large}}$	63.1	59.1	59.9
XLMR$_{\text{large}}$-A_{Fusion}	**64.2**	**60.4**	**61.1**

Results. Results are summarized in Table 4 (see Table 11 of Appendix D for further detail). We

[11] Note that due the variable size of entity sets, the task difficulty varies across languages (see Table 9 in Appendix C).

[12] Even if side information such as the entity description is provided, we only consider the alignment data for simplicity.

[13] Note that we select the extended language only by the size of test set. The ZS-In set is slightly different from the DBP5L, where *ar* and *cs* are newly added and *el* is not included.

find that for *supervised* languages, mBERT-A$_{\text{Fusion}}$ outperforms both the base version of mBERT as well as the knowledge-enhanced version (MTMB), whereas XLMR with adapters obtains comparable performance. As for *zero-shot* languages, MLLMs-A$_{\text{Fusion}}$ achieve consistent and significant improvements over baselines. This demonstrates that our knowledge adapter set can enhance MLLMs for knowledge-intensive tasks.

5.3.2 General Language Tasks

Besides above knowledge-intensive tasks, we show that our knowledge adapter set can maintain the performance of MLLMs on general multilingual language tasks. We select the popular multilingual benchmark called XTREME (Hu et al., 2020) to evaluate the enhanced MLLMs, which are finetuned on English training data, and tested with many other languages. We select cross-lingual NER and QA as two general tasks. We follow the settings of the XTREME benchmark.

Table 5: Results on the multilingual NER task (F1 score). We can find that our adapters can enhance MLLMs on the performance of NER task for zero-shot languages.

Model	Sup. (En)	ZS-In (39)	Ave.
mBERT	**85.2**	61.6	62.2
mBERT-A$_{\text{Fusion}}$	84.0	**62.3**	**62.9**
XLMR$_{\text{large}}$	84.7	64.9	65.4
XLMR$_{\text{large}}$-A$_{\text{Fusion}}$	**85.0**	**65.3**	**65.8**

NER. We select the WikiAnn dataset (Pan et al., 2017) (under the setting of XTREME) for the NER task, where 40 languages are included for evaluation. The results are summarized in Table 5, and detailed results can be found in Table 12 in Appendix D. We find that MLLMs with our adapter set perform as well as the baseline MLLMs with slight improvements on the zero-shot languages.

Table 6: Results on the multilingual QA tasks. Using our adapters would not reduce the performance on language modeling tasks, while marginal improvement can even achieved.

Model	Sup. (En)		ZS-In (10)		Ave.	
	F1	EM	F1	EM	F1	EM
mBERT	83.5	72.2	**62.6**	47.2	**64.5**	49.4
mBERT-A$_{\text{Fusion}}$	**83.5**	72.0	62.1	**47.2**	62.2	**49.5**
XLMR$_{\text{large}}$	86.5	75.7	75.6	59.3	76.6	60.8
XLMR$_{\text{large}}$-A$_{\text{Fusion}}$	**88.0**	**77.6**	**75.7**	**59.7**	**76.8**	**61.3**

Question Answering. Following the setting of XTREME, We finetune the models on the SQuAD (Rajpurkar et al., 2016) dataset (in English), and evaluate on the test sets of XQuAD (Artetxe et al., 2020) involving 11 languges. Detailed results are in Table 13 in Appendix D. We find that mBERT-A$_{\text{Fusion}}$ maintains the performance as its original version, while XLMR$_{\text{large}}$-A$_{\text{Fusion}}$ can be boosted slightly. In general, MLLMs-A$_{\text{Fusion}}$ with our adapters can obtain comparable or slightly better performance across different language tasks. For those tasks requiring rich knowledge about triples and entity alignments, our adapter set can indeed enhance the MLLMs.

5.4 Comparison with Existing Methods

We compare our approach with the only existing related work (Liu et al., 2021) that attempts to integrate MLKGs into MLLMs. However, it only considers a relatively small set of 10 languages and finetunes the entire MLLM with a joint objective, which is computationally expensive. In contrast, as shown below, our knowledge adapter set can achieve better performance at a much lower cost.

Settings. We follow settings and metrics in Liu et al. (2021), which are slightly different from original settings of RELX and WikiAnn (XTREME) datasets. We only report the performance for MLLMs that are implemented in their study.

Table 7: Comparison with Liu et al. (2021) (denoted by $^{\triangle}$) on RELX, WikiAnn and XQuAD datasets involving 4, 10 and 11 languages, respectively. We can find that our light adapter-based knowledge enhancement method significantly outperforms previous finetuning-based enhancement method.

Model	RELX (4)	WikiAnn (10)	XQuAD (11)	
	Acc.	F1	F1	EM
mBERT	60.1	-/-	-/-	-/-
mBERT$^{\triangle}$	61.1	-/-	-/-	-/-
mBERT-MLKG	**64.7**	-/-	-/-	-/-
XLMR$_{\text{base}}$	56.7	-/-	-/-	-/-
XLMR$^{\triangle}_{\text{base}}$	58.3	-/-	-/-	-/-
XLMR$_{\text{base}}$-A$_{\text{Fusion}}$	**61.7**	-/-	-/-	-/-
XLMR$_{\text{large}}$	61.3	**68.5**	76.6	60.8
XLMR$^{\triangle}_{\text{large}}$	61.9	66.9	76.5	60.6
XLMR$_{\text{large}}$-A$_{\text{Fusion}}$	**64.6**	67.6	**76.8**	**61.3**

Results. In Table 7, for the relation classification task, where Liu et al. (2021) outperforms the MLLM baseline, our method achieves significant further improvement. For NER, only 10 popular *zero-shot* languages (instead of 40 languages in XTREME) are selected for their knowledge integration and evaluation. Even if generally our method achieves better performance for XMLR$_{\text{large}}$-A$_{\text{Fusion}}$ (40 languages) in Table 5, it performs slightly worse than the original version here (10 popular languages). However, the performance of Liu et al. (2021) is worse still. For QA, similar performance is achieved by all three MLLMs, although our enhanced MLLM slightly outperforms other methods.

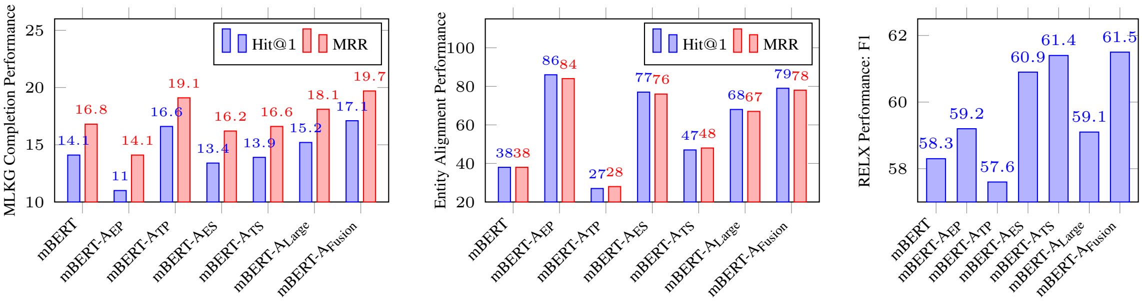

Figure 5: Ablation study results. We select two MLKG-related tasks and the relation classification task for evaluation. We can find that adapters that integrate factual knowledge into MLLMs achieve better performance than others on the MLKG completion task, while adapters that integrate cross-lingual alignments outperform others on the entity alignment task. For the relation classification task, sentence-level adapters achieve better performance. For our adapter set, it can achieve roughly the best performance under all conditions.

5.5 Ablation Study

We conduct ablation studies to understand our knowledge adapters and show that they work as expected.[14] We also compare against a large adapter ($\mathrm{A_{Large}}$) with a comparable total number of parameters (including fusion layers). The large adapter is trained with the same settings as our adapter set and has one set of parameters that integrate all knowledge types at once. As previously, we finetune the original mBERT, mBERT-$\mathrm{A_{Large}}$, and mBERT with our adapters on each downstream task.

In Figure 5, for the knowledge graph completion task (*left*), mBERT-$\mathrm{A_{TP}}$ and mBERT-$\mathrm{A_{TS}}$ perform better than their entity-based counterparts. While mBERT-$\mathrm{A_{Large}}$ also performs well, mBERT-$\mathrm{A_{Fusion}}$ outperforms it significantly. For the entity alignment task (*center*), the situation is reversed such that better performance is achieved by mBERT-$\mathrm{A_{EP}}$ and, mBERT-$\mathrm{A_{ES}}$. Our mBERT-$\mathrm{A_{Fusion}}$ also achieves comparable performance which is much better than mBERT-$\mathrm{A_{Large}}$ with shared parameters. As for the relation classification task (*right*), sentence-level adapters outperform phrase-level adapters, which is intuitive since the task requires sentence-level context. Fusing all four adapters (i.e., mBERT-$\mathrm{A_{Fusion}}$) gives the best performance while mBERT-$\mathrm{A_{Large}}$ performs worse than single smaller adapters. In summary, with our method, we learn different types of knowledge in separate adapters, which can be fused in different proportions according to the downstream task at hand to typically perform better and more consistently than any single adapter-enhanced MLLMs.

6 Other Related Work

MLLM for MLKG. Several works use the implicit knowledge in language models to improve knowledge graph-related tasks (Yao et al., 2019; Niu et al., 2022). However, these approaches are for monolingual knowledge triples and can not easily incorporate cross-lingual entity alignment. Huang et al. (2022) use MLLMs for knowledge graph completion, but language models only encode entities, and the task itself is achieved by graph neural networks. Previous MLKG embedding methods consider entity alignment (Chen et al., 2017, 2020), but are designed for existing MLKGs, and can not generalize to other, e.g. low-resource, languages without the multilingual knowledge in MLLMs (Pires et al., 2019; Wu and Dredze, 2019).

MLKG for MLLM. Liu et al. (2021) propose to synthesize *code-switched* sentences to solve the problem but the resulting MLKG-enhanced MLLMs achieve minimal improvement on language understanding tasks as shown in our experiment, and it cannot benefit the MLKG field. In summary, our work first combine MLKG and MLLM, showing that combining them using our light knowledge adapter set can effectively improve the downstream task performance on both sides.

7 Conclusion

In this paper we propose an approach to enhance MLLMs with MLKGs using a set of knowledge adapters, where explicit knowledge from MLKGs is integrated into the implicit knowledge learned by MLLMs. In experiments, we show that enhanced MLLMs can conduct MLKG-related tasks and achieve better performance on knowledge-intensive tasks, especially on low-resource languages where knowledge graphs are not available.

[14] Since the improvements brought by our adapters are consistent across different MLLMs, we mainly consider mBERT for analysis. We report results on the knowledge graph completion, entity alignment and relation classification tasks, which each require different aspects of knowledge.

Limitations

We point out that there are some limitations of our work. First, even if the adapter set can enhance MLLMs to perform well on various downstream tasks, it is not suitable for tasks with the fully zero-shot setting (without any training data), since the fusion module has to be tuned to suit the task. Second, as shown in our results, the fusion module cannot always outperform all single adapters. For some tasks, a better fusion mechanism could be proposed for the improvement.

Reproducibility Statement

We elaborate the experiment settings and hyperparameters in the paper and in Appendix A. We have published our prepossessed multilingual knowledge integration data, extended MLKG-related task datasets, as well as our code.

Ethics Statement

We do not foresee any significant ethical concerns in this work.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. Yifan Hou is supported by the Swiss Data Science Center PhD Grant (P22-05). Carl Allen is supported by an ETH AI Centre Postdoctoral Fellowship. We also acknowledge support from an ETH Zurich Research grant (ETH-19 21-1) and a grant from the Swiss National Science Foundation (project #201009) for this work.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *CoRR*, abs/2204.06031.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Muhao Chen, Weijia Shi, Ben Zhou, and Dan Roth. 2021. Cross-lingual entity alignment with incidental supervision. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 645–658, Online. Association for Computational Linguistics.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1511–1517. ijcai.org.

Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Uppunda, Yizhou Sun, and Carlo Zaniolo. 2020. Multilingual knowledge graph completion via ensemble knowledge transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3227–3238, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2021. A primer on pretrained multilingual language models. *CoRR*, abs/2107.00676.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022. Multilingual knowledge graph completion with self-supervised adaptive graph alignment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 474–485. Association for Computational Linguistics.

Wenxiang Jiao, Zhaopeng Tu, Jiarui Li, Wenxuan Wang, Jen-tse Huang, and Shuming Shi. 2022. Tencent's multilingual machine translation system for WMT22 large-scale african languages. In *Proceedings of the Seventh Conference on Machine Translation*, Online. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Abdullatif Köksal and Arzucan Özgür. 2020. The RELX dataset and matching the multilingual blanks for cross-lingual relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 340–350, Online. Association for Computational Linguistics.

Linlin Liu, Xin Li, Ruidan He, Lidong Bing, Shafiq R. Joty, and Luo Si. 2021. Knowledge based multilingual language model. *CoRR*, abs/2111.10962.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908. AAAI Press.

Guanglin Niu, Bo Li, Yongfei Zhang, and Shiliang Pu. 2022. CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2867–2877. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Shijie Wu and Mark Dredze. 2019. 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, Hong Kong, China. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Implementation Details

We implement the adapters using the AdapterHub library[15], where all Transformer layers in MLLMs are inserted with adapters.

Adapters in Knowledge Enhancement. To train these knowledgeable adapters, we use 8 GPUs (Tesla V100) with batch size as 128. The learning rate is set as $1e-4$. We use the Adam optimizer with $1e4$ warm-up steps. We train Adapter-EP by randomly sampling entity alignments in different languages. The number of sampled alignments is around 94.2 million. And the training epoch number for Adapter-TP, Adapter-ES, and Adapter-TS is all set as 10. As for the InfoNCE loss, we use the negative sampling within batch. Since we train adapters with sampling strategy and use the contrastive learning loss instead of mask language modeling, it only takes several hours to train one adapter (1-10 hours). The whole enhancement procedure would take around half a day.

Adapters in knowledge graph completion. For MLLM-based methods, we set all hyperparameters as the same to ensure the comparison is fair[16]. We use the average value of word(-piece) representation as the entity embedding. Specifically, we train MLLMs as well as MLLMs-AF (including adapters and the fusion mechanism) to embed entities, where the output representations of the object entities should be close to the context (subject entities with relations) output representations. The similarity is measure by cosine[17]. During the training, the learning rate is set as $1e-8$, and the epoch number is set as 10. The batch size is set as 8. We train MLLMs using the contrastive learning loss similar to the knowledge integration process.

Adapters in Entity Alignment. Similarly, we set all hyperparameters as the same for all MLLM-based methods. Specifically, we set the epoch number as 1 since the overfitting is easy with training data only on 2 languages. Other hyperparameters and settings are the same to that of the MLKG Completion task.

Adapters in Language Tasks. We evaluate our adapter set with MLLMs on the XTREME benchmark. The evaluation settings are the same as theirs.

B Knowledge Integration Dataset Statistics

The detailed statistics can be found in Table 8 below.

C MLKG Dataset Statistics and Detailed Results

The detailed statistics and results can be found in Table 9 and Table 10.

D MLLM Dataset Statistics and Detailed Results

The detailed statistics and results can be found in Table 11 (relation classification), Table 12 (name entity recognition), and Table 13 (question answering).

[15] https://adapterhub.ml/

[16] Note that users may search more fine-grained hyperparameters, but the relative performance would not change.

[17] We also tried different metrics but find that cosine distance works well in this task.

Table 8: Distribution of Wikidata for adapter training. We report the full name and ISO code for all languages. For the entity, relation, and triple, we report the ratio of the label in that specific language to the total number of it.

ISO	Lang.	Entity (%)	Relation (%)	Triple (%)	ISO	Lang.	Entity (%)	Relation (%)	Triple (%)	ISO	Lang.	Entity (%)	Relation (%)	Triple (%)
af	Afrikaans	56.4	20.5	31.8	gu	Gujarati	12.2	14.2	2.1	nn	Norwegian Nynorsk	70.6	44.4	57.9
an	Aragonese	59.8	0.7	10.7	he	Hebrew	25.3	62.2	29.7	no	Norwegian	0.0	-	-
ar	Arabic	33.5	91.0	42.3	hi	Hindi	14.8	13.3	4.4	oc	Occitan	60.9	23.9	32.1
ast	Asturian	84.2	28.3	71.3	hr	Croatian	57.8	23.1	33.5	pl	Polish	92.5	73.0	85.9
az	Azerbaijani	19.3	19.3	9.8	hu	Hungarian	71.9	64.6	70.2	pt	Portuguese	96.4	80.8	91.0
bar	Bavarian	52.4	1.8	10.0	hy	Armenian	21.5	21.4	17.0	ro	Romanian	81.7	32.8	59.3
be	Belarusian	18.0	52.7	11.6	id	Indonesian	65.2	48.2	47.7	ru	Russian	54.4	88.6	64.1
bg	Bulgarian	31.9	22.6	19.4	is	Icelandic	52.3	7.6	15.0	scn	Sicilian	39.6	25.5	17.7
bn	Bengali	18.3	34.6	11.5	it	Italian	97.8	78.2	97.0	sco	Scots	56.8	27.3	27.3
br	Breton	54.5	18.8	30.0	ja	Japanese	37.5	77.5	48.3	sh	Serbo-Croatian	21.7	9.2	6.9
bs	Bosnian	44.7	27.3	18.6	jv	Javanese	41.3	1.6	7.1	sk	Slovak	62.4	25.8	38.4
ca	Catalan	87.2	99.3	88.9	ka	Georgian	16.2	23.8	9.3	sl	Slovenian	69.1	24.8	56.0
ceb	Cebuano	51.5	0.3	0.2	kk	Kazakh	16.7	4.0	2.2	sq	Albanian	73.0	28.1	47.2
cs	Czech	73.5	68.4	66.4	kn	Kannada	13.7	7.8	2.1	sr	Serbian	23.3	92.6	17.5
cy	Welsh	61.4	35.4	43.3	ko	Korean	26.2	58.2	25.3	sv	Swedish	91.7	73.8	90.9
da	Danish	77.3	57.5	75.4	la	Latin	59.9	9.4	23.1	sw	Swahili	50.4	0.6	6.2
de	German	98.5	90.7	98.5	lb	Luxembourgish	55.3	25.2	33.5	ta	Tamil	14.8	18.8	4.8
el	Greek	19.5	46.5	16.5	lt	Lithuanian	52.5	15.7	27.4	te	Telugu	13.2	17.7	3.1
en	English	100.0	100.0	100.0	lv	Latvian	38.2	40.2	25.6	th	Thai	16.2	20.5	7.3
es	Spanish	98.7	94.0	98.6	mk	Macedonian	16.5	95.1	9.3	tl	Tagalog	16.4	7.2	5.3
et	Estonian	60.8	25.9	40.9	ml	Malayalam	15.9	14.6	4.7	tr	Turkish	64.4	81.2	50.9
eu	Basque	74.0	37.4	54.4	mn	Mongolian	11.5	1.3	0.2	tt	Tatar	19.0	35.7	12.4
fa	Persian	32.5	51.1	33.7	mr	Marathi	13.4	17.4	3.7	uk	Ukrainian	45.2	97.7	44.4
fi	Finnish	89.9	56.3	78.8	ms	Malay	56.3	40.9	35.0	ur	Urdu	16.7	28.1	7.8
fr	French	98.5	97.3	99.1	my	Burmese	11.7	5.3	0.9	uz	Uzbek	17.1	3.7	4.6
fy	Western Frisian	41.6	4.7	7.6	nds	Low Saxon	54.1	23.1	29.3	vi	Vietnamese	74.7	32.8	44.4
ga	Irish	78.4	25.2	57.3	ne	Nepali	11.3	7.7	1.1	war	Waray-Waray	61.1	0.1	0.0
gl	Galician	65.2	38.5	45.9	nl	Dutch	98.3	100.0	98.2	zh	Chinese	41.1	64.9	49.7

Table 9: The performance of various models for the MLKG completion task (Hit@1/MRR) across different languages. We also report the number of entities in the test set to show the general difficulty of the completion task in that language.

Language	# of test set	TransE	DisMult	mBERT	mBERT-MLKG	XLM	XLM-MLKG	XLM-R	XLM-R-MLKG
el	1082	13.1 / 24.3	8.9 / 9.8	9.2 / 11.6	8.5 / 11.2	4.8 / 6.9	6.9 / 9.7	5.0 / 7.5	9.3 / 12.8
en	5984	7.3 / 16.9	8.8 / 18.3	15.2 / 17.7	18.5 / 21.3	8.2 / 10.0	11.7 / 14.8	10.4 / 12.5	17.5 / 19.9
es	4101	13.5 / 24.4	7.4 / 13.2	14.3 / 17.2	17.7 / 20.5	7.0 / 9.4	11.7 / 14.9	9.7 / 12.2	18.0 / 20.7
fr	4436	17.5 / 27.6	6.1 / 14.5	12.7 / 15.4	17.4 / 19.9	6.3 / 8.5	12.1 / 14.5	9.2 / 11.6	16.2 / 18.5
ja	2569	21.1 / 25.3	9.3 / 15.8	4.6 / 6.9	3.6 / 5.8	3.1 / 4.4	3.0 / 5.0	2.4 / 4.6	4.7 / 7.4
ast	2823	-	-	13.9 / 16.8	19.1 / 21.8	7.1 / 9.5	13.7 / 16.5	10.6 / 12.9	17.5 / 20.5
ca	2959	-	-	14.8 / 17.6	19.1 / 21.5	7.9 / 10.4	13.8 / 16.5	11.1 / 13.4	17.4 / 20.2
da	2566	-	-	16.1 / 19.2	19.9 / 23.0	8.7 / 11.6	13.3 / 16.9	11.5 / 14.1	17.6 / 21.4
de	4059	-	-	14.1 / 16.8	17.4 / 20.4	8.3 / 11.2	11.4 / 14.6	9.8 / 12.6	15.6 / 18.7
fa	2329	-	-	5.0 / 7.1	5.3 / 6.9	3.9 / 4.8	4.1 / 5.8	5.1 / 7.3	5.2 / 7.2
fi	2582	-	-	11.2 / 14.6	16.1 / 19.1	6.2 / 8.6	9.9 / 13.0	8.2 / 11.1	13.7 / 17.0
hu	2558	-	-	13.7 / 16.7	18.4 / 21.4	6.4 / 9.2	11.4 / 14.8	10.0 / 12.5	15.7 / 18.7
it	3614	-	-	14.4 / 17.0	17.3 / 19.8	7.6 / 9.8	12.2 / 15.2	10.4 / 12.8	15.7 / 18.6
nb	2717	-	-	16.4 / 19.4	19.5 / 23.3	8.9 / 11.6	13.5 / 17.0	11.3 / 13.9	18.0 / 21.4
nl	4316	-	-	14.0 / 16.8	19.1 / 21.7	7.3 / 9.8	13.3 / 15.9	8.6 / 11.5	17.4 / 20.2
pl	2998	-	-	13.4 / 17.2	18.6 / 21.8	6.1 / 8.5	9.7 / 13.3	8.7 / 11.5	14.6 / 18.0
pt	3184	-	-	15.4 / 18.4	18.0 / 20.6	7.3 / 9.7	12.3 / 15.4	9.6 / 12.1	17.5 / 20.6
ru	2887	-	-	9.4 / 11.8	10.3 / 12.1	3.5 / 5.5	4.6 / 6.6	4.8 / 7.4	6.3 / 8.6
sv	2993	-	-	15.7 / 18.5	18.7 / 22.0	9.2 / 11.7	13.0 / 16.4	11.0 / 13.6	17.8 / 21.3
zh	2591	-	-	5.1 / 7.4	4.1 / 6.4	2.2 / 4.2	2.7 / 5.1	3.4 / 5.3	4.3 / 6.8
eo	963	-	-	-	-	8.2 / 11.8	16.6 / 19.6	16.8 / 20.8	23.9 / 27.4
vo	164	-	-	48.1 / 49.1	51.8 / 52.4	-	-	-	-

Table 10: The performance of various models for the entity alignment task (Hit@1/MRR) across different languages. We also report the number of entities in the test set to show the general difficulty of the completion task in that language.

Language	# of test set	MTransE	JEANS	mBERT	mBERT-MLKG	XLM	XLM-MLKG	XLM-R	XLM-R-MLKG
en->fr	39155	14.0 / 17.7	46.3 / 53.8	87.1 / 86.4	92.6 / 92.1	55.3 / 55.3	92.1 / 91.4	65.2 / 65.2	93.5 / 92.8
en->de	41018	3.4 / 7.2	33.7 / 41.2	80.1 / 79.9	85.2 / 84.7	54.3 / 54.3	85.1 / 84.6	64.8 / 64.9	86.8 / 86.2
en->ar	16818	-	-	8.9 / 10.0	68.6 / 67.4	0.7 / 0.9	63.4 / 62.5	0.9 / 1.1	81.8 / 80.0
en->ast	19834	-	-	41.8 / 41.9	85.2 / 83.9	13.0 / 13.2	93.6 / 92.6	33.5 / 33.8	97.3 / 96.3
en->ca	22567	-	-	38.2 / 38.3	81.5 / 80.2	10.8 / 11.1	90.2 / 88.8	29.9 / 30.2	94.5 / 93.4
en->cs	16570	-	-	40.0 / 40.3	82.5 / 81.1	11.9 / 12.2	89.8 / 88.6	30.4 / 30.5	93.9 / 92.8
en->da	20093	-	-	39.2 / 39.4	82.4 / 81.3	12.7 / 12.9	91.7 / 90.5	33.0 / 33.2	95.5 / 94.4
en->es	28288	-	-	40.6 / 40.3	81.8 / 80.2	11.5 / 11.7	90.1 / 88.6	33.2 / 32.3	94.3 / 92.7
en->fa	16120	-	-	10.1 / 11.3	69.4 / 68.2	1.0 / 1.2	67.6 / 66.9	1.8 / 2.2	83.1 / 81.8
en->fi	20608	-	-	39.4 / 39.4	81.3 / 79.9	12.4 / 12.6	90.0 / 88.8	32.2 / 32.4	94.2 / 93.1
en->hu	18896	-	-	36.3 / 36.7	80.5 / 79.4	11.3 / 11.4	89.2 / 88.0	29.6 / 29.9	93.6 / 92.7
en->it	26393	-	-	39.4 / 39.5	80.2 / 78.7	11.5 / 11.8	88.4 / 86.9	31.2 / 31.2	92.4 / 91.0
en->ja	22012	-	-	8.9 / 10.1	64.3 / 63.4	0.7 / 0.8	60.9 / 60.0	1.4 / 1.5	77.8 / 76.4
en->nb	20748	-	-	39.2 / 39.3	82.5 / 81.1	11.5 / 11.8	91.8 / 90.4	32.2 / 32.5	95.6 / 94.4
en->nl	29378	-	-	41.3 / 41.3	82.4 / 80.5	12.2 / 12.4	90.8 / 89.0	34.1 / 34.1	94.6 / 92.8
en->pl	21535	-	-	38.7 / 38.9	80.0 / 78.7	11.2 / 11.4	87.6 / 86.3	30.2 / 30.3	92.6 / 91.2
en->pt	23001	-	-	41.5 / 41.5	82.5 / 81.0	12.3 / 12.6	90.6 / 89.2	33.1 / 33.2	94.4 / 93.1
en->ru	22665	-	-	19.2 / 20.2	74.2 / 72.6	3.6 / 3.7	78.0 / 76.2	10.0 / 10.3	87.9 / 85.9
en->sv	22986	-	-	39.7 / 39.7	81.6 / 80.1	11.8 / 12.1	90.6 / 89.2	32.2 / 32.4	94.4 / 93.1
en->zh	20891	-	-	10.2 / 11.1	55.1 / 54.8	9.0 / 10.6	49.8 / 49.5	1.9 / 1.9	67.3 / 66.2
en->eo	8913	-	-	-	-	10.9 / 11.1	85.4 / 84.3	28.9 / 28.9	89.8 / 88.5
en->vo	2954	-	-	50.5 / 50.8	91.7 / 89.3	-	-	-	-

Table 11: Detailed results of the cross-lingual relation classification task (RELX) evaluated by F1 score.

Language	mBERT	mBERT-MLKG	XLM	XLM-MLKG	XLM-R	XLM-R-MLKG
en	61.8	64.0	61.4	61.3	63.1	64.2
de	57.5	60.0	57.5	56.1	58.0	60.2
es	57.9	63.1	56.9	59.7	59.8	60.7
fr	58.3	61.1	55.7	58.0	59.5	61.5
tr	55.8	59.3	54.1	58.0	59.1	59.0
average	58.3	61.5	57.1	58.6	59.9	61.1

Table 12: Detailed results of the NER task (Wikiann) evaluated by F1 socre.

Language	mBERT	mBERT-MLKG	XLM-R	XLM-R-MLKG	Language	mBERT	mBERT-MLKG	XLM-R	XLM-R-MLKG
en	85.2	84.0	84.7	85.0	ka	64.6	66.9	71.6	69.3
af	77.4	77.2	78.9	79.2	kk	45.8	49.1	56.2	53.3
ar	41.1	40.5	53.0	51.8	ko	59.6	60.2	60.0	61.0
bg	77.0	76.2	81.4	80.6	ml	52.3	53.1	67.8	61.0
bn	70.0	72.8	78.8	78.1	mr	58.2	55.0	68.1	67.2
de	78.0	78.6	78.8	78.5	ms	72.7	68.1	57.1	74.6
el	72.5	70.8	79.5	79.6	my	45.2	55.5	54.3	56.8
es	77.4	74.8	79.6	75.8	nl	81.8	82.3	84.0	83.5
et	75.4	78.6	79.1	78.1	pt	80.8	78.7	81.9	82.5
eu	66.3	68.3	60.9	59.0	ru	64.0	66.8	69.1	70.5
fa	46.2	38.6	61.9	48.9	sw	67.5	70.1	70.5	70.0
fi	77.2	78.3	79.2	79.0	ta	50.7	53.8	59.5	60.8
fr	79.6	78.9	80.5	80.2	te	48.5	48.2	55.8	50.9
he	56.6	54.4	56.8	57.9	th	3.6	0.1	1.3	2.9
hi	65.0	66.3	73.0	73.0	tl	71.7	74.6	73.2	78.0
hu	76.4	78.0	79.8	80.6	tr	71.8	74.4	76.1	80.6
id	53.5	54.6	53.0	55.9	ur	36.9	43.9	56.4	63.2
it	81.5	81.6	81.3	81.2	vi	71.8	70.7	79.4	78.9
ja	29.0	29.2	23.2	23.1	yo	44.9	50.9	33.6	45.4
jv	66.4	65.3	62.5	66.9	zh	42.7	45.0	33.1	28.9

Table 13: Detailed results of the QA task (XQuAD) evaluated by F1/EM score.

Language	mBERT	mBERT-MLKG	XLM-R	XLM-R-MLKG
en	83.5 / 72.2	83.5 / 72.0	86.5 / 75.7	88.0 / 77.6
ar	61.5 / 45.1	61.3 / 44.5	68.6 / 49.0	76.2 / 58.9
de	70.6 / 54.0	70.6 / 54.8	80.4 / 63.4	79.6 / 62.8
el	62.6 / 44.9	63.5 / 47.5	79.8 / 61.7	79.1 / 61.3
es	75.5 / 56.9	74.4 / 57.2	82.0 / 63.9	82.4 / 64.4
hi	59.2 / 46.0	57.2 / 42.9	76.7 / 59.7	75.6 / 59.3
ru	71.3 / 53.3	70.5 / 54.4	80.1 / 64.3	79.7 / 63.6
th	42.7 / 33.5	43.6 / 36.8	74.2 / 62.8	73.3 / 61.2
tr	55.4 / 40.1	53.7 / 38.0	75.9 / 59.3	74.9 / 58.9
vi	69.5 / 49.6	67.7 / 47.9	79.1 / 59.0	80.0 / 60.6
zh	58.0 / 48.3	58.0 / 48.3	59.3 / 50.0	56.0 / 46.7
average	64.5 / 49.4	62.2 / 49.5	76.6 / 60.8	76.8 / 61.3