Adapters for Enhanced Modeling of Multilingual Knowledge and Text

Yifan Hou¹, Wenxiang Jiao², Meizhen Liu³, Carl Allen¹, Zhaopeng Tu², Mrinmaya Sachan¹
¹ETH Zürich, ²Tencent AI Lab, ³Shandong University
¹{yifan.hou, carl.allen, mrinmaya.sachan}@inf.ethz.ch
²{joelwxjiao, zptu}@tencent.com, ³meizhen.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 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 (AIKhamissi et al., 2022).

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),

![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.](https://github.com/yifan-h/Adapters_for_Enhanced_Modeling_of_Multilingual_Knowledge_and_Text/blob/master/images/figure1.png)
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., Know-BERT (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 Wikidata2 and Wikipedia3 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 Triples (i.e. facts) or cross-lingual Entity alignments, and is trained on either Phrase or Sentence 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. 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: $\text{LM}(s^l) = (h_1, h_2, ...)$. These vectors correspond to representations for each token in the

---

2https://www.wikidata.org/wiki/Wikidata:Main_Page
3https://en.wikipedia.org/wiki/Main_Page
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: \( \text{LM}(s_i) = \text{mean}(h_{i_1}, h_{i_2}, \ldots) \). Similarly, for a phrase \( s_{ij} \) (starting from the \( i \)-th token and ending in the \( j \)-th token in the sentence), we can obtain its contextualized representation as \( \text{LM}(s_{ij}) = \text{mean}(h_i, h_{i+1}, \ldots, h_j) \).

**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 \( e_i \) denote the entity label of the \( i \)-th entity in language \( l \). Furthermore, we denote a knowledge triple in the MLKG as \( (e_i, r_{ik}^l, e_j^l) \in \mathcal{T} \), where \( r_{ik}^l \) 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 (Houlhsby 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 \( h_m \), the hidden representation of token \( t_i^l \) at layer \( m \), the adapter acts as

\[
A(h_m) = W_{\text{up}} \cdot \sigma(W_{\text{down}} \cdot h_m + b_{\text{down}}) + b_{\text{up}}. \tag{1}
\]

Here, \( W_{\text{down}} \) and \( W_{\text{up}} \) are weight matrices, which map the hidden representations to the low-dimensional space and then map them back. \( b_{\text{down}} \) and \( b_{\text{up}} \) are bias parameters, and \( \sigma \) is a non-linear 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 \( h_m \) as in Eq. 1) are input to the adapters. A fusion layer aggregates all adapter outputs \( A_n(h_m) \ (n \in \{1,...,N\} \text{ indexes each adapter}) \) and the un-adapted representations with a multiplicative attention mechanism:

\[
A_{\text{fusion}}(h_m) = \sum_{n=0}^{N} a_n^m \cdot V_n^m \cdot A_n(h_m),
\]

\[
a_n^m = \text{softmax}(h_n^m Q_m^m \otimes A_n(h_m) K_m^m).
\]

Here, \( A_0(\cdot) \) is the identity function; \( Q_m^m, K_m^m, V_m^m \) 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 Triples and Entity alignments, each provided in both Phrase and Sentence 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 \( e_i^l \) and \( e_j^{l'} \), e.g., *Zurich* is aligned with *Zurigo*. **Adapter-TP** learns knowledge triples, e.g., predicting Switzerland given entity and relation (*Zurich, is located in, Switzerland*).

---

\(^5\)Where to insert adapters is flexible but a common choice is after the feedforward layer of a transformer layer.
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):

\[
\text{INCE}(x, x') = \log \frac{\cos(x, x')}{\sum_{x'' \in X} \cos(x, x'')},
\]

where \(X\) is a batch that includes the positive sample \(x'\) 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 \(\text{LM}(\cdot)\), the objective used to train EP is:

\[
\mathcal{L}_{\text{EP}} = \sum_{(e_i', s_l) \in C_1} \text{INCE}\left(\text{LM}(s_l), \text{LM}(e_i')\right),
\]

**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:

\[
\mathcal{L}_{\text{TP}} = \sum_{(e_i', r_l'') \in T} \text{INCE}\left(\text{LM}([e_i'; r_l'']), \text{LM}(e_j')\right),
\]

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 \(C_1\). Thus, similar to Adapter-EP, we train ES by aligning contextualized entity representations of cross-lingually aligned entities with the objective:

\[
\mathcal{L}_{\text{ES}} = \sum_{(e_i', s_l) \in C_1} \text{INCE}\left(\text{LM}(s_l), \text{LM}(e_i')\right),
\]
where $s^i_j$ means that we input sentence $s^i$ into an MLLM but keep only the representation of entity label $e^j$ (indexed from $i$-th token to $j$-th token). As in Figure 2 (right), $s^i$ is: De Botton spent his early years in Zurich, and $s^j$ here is the entity label of $e^j$ 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:

$$\mathcal{L}_{TS} = \sum_{(s_k,e_j) \in C_2} \text{INCE}\left(\text{LM}\left(s_k \setminus e_j\right), \text{LM}(e_j)\right),$$

where $s_k \setminus e_j$ represents the sentence $s_k$ with entity label $e_j$ masked. As the example in Figure 2 (right), $s_k \setminus e_j$ 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 the aligned triple is:

$$(Zurich, is located in, Switzerland)$$

used to train Adapter-EP is extracted from Wikidata by keeping only entities that have more

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

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).

| 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 (C₁)          | Entity / Sent.: 0.20M / 1.93M |
| Adapter-TS   | T-REx-RC (C₂)           | 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 $C_2$

$^8$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.](image)

We follow settings of data collection pipelines described in Chen et al. (2020, 2017) for the extension. 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. 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-AFusion, enhanced with the knowledge adapter and fusion mechanism (see Appendix A for further implementation details).

| Model              | Sup. (5) | ZS-In (15) | ZS-Un. (2') |
|--------------------|----------|------------|-------------|
|                    | Hit@1 (%)| MRR (%)    | Hit@1 (%)   | MRR (%)    |
| TransE DistMult    | 8.1      | 14.3       | 8.6         | 16.6       |
| mBERT              | 11.2     | 13.8       | 12.8        | 15.7       |
| mBERT-Afused       | 13.1     | 15.7       | 16.1        | 18.8       |
| XLMRlarge          | 5.9      | 7.8        | 6.7         | 9.1        |
| XLMRlarge-Afused   | 9.1      | 11.8       | 10.0        | 13.5       |
| XLMRsmall          | 7.3      | 9.7        | 8.9         | 11.5       |
| XLMRsmall-Afused   | 13.1     | 15.6       | 14.3        | 17.3       |

**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 methods in Chen et al. (2020, 2017) for the extension.

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, and the enhanced MLLMs-\(A_{\text{Fusion}}\) 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.\(^\text{12}\) 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.\(^\text{13}\) 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        |
| MTransE     | 8.7      | 12.5       | -/=/-      | -/=/-      |
| JEANS       | 40.0     | 47.5       | -/=/-      | -/=/-      |
| mBERT\(-\text{base}\) | 83.6  | 83.2       | 31.8       | 32.2       | 50.3 | 50.8 |
| XLM\(\text{Base}\)-\(A_{\text{Fusion}}\) | 54.8  | 54.8       | 9.4        | 9.7        | 10.9 | 11.1 |
| XLM\(\text{Base}\)-\(A_{\text{Fusion}}\) | 65.0  | 65.1       | 23.9       | 24.1       | 28.9 | 28.9 |
| XLM\(\text{Large}\)-\(A_{\text{Fusion}}\) | 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-\(A_{\text{Fusion}}\) 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\(-\text{base}\) | 61.8  | 57.4       | 58.3      |
| MTransE     | 63.6      | 59.6       | 60.4      |
| XLM\(\text{Base}\)-\(A_{\text{Fusion}}\) | 64.0  | 60.9       | 61.5      |
| XLM\(\text{Large}\)-\(A_{\text{Fusion}}\) | 61.4  | 58.1       | 57.1      |
| XLM\(\text{Large}\)-\(A_{\text{Fusion}}\) | 61.3  | 58.0       | 58.6      |
| XLM\(\text{Large}\)-\(A_{\text{Fusion}}\) | 63.1  | 59.1       | 59.9      |
| XLM\(\text{Large}\)-\(A_{\text{Fusion}}\) | 64.2  | 60.4       | 61.1      |

**Results.** Results are summarized in Table 4 (see Table 11 of Appendix D for further detail).
find that for supervised languages, mBERT-A\textsubscript{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\textsubscript{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 fine-tuned 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.

| Model | ZS-In (39) | Sup. (En) | Ave. |
|-------|------------|----------|------|
| mBERT | 6.1 | 85.2 | 62.2 |
| mBERT-A\textsubscript{Fusion} | 62.3 | 84.0 | 62.9 |
| XLMR\textsubscript{large} | 65.8 | 84.7 | 65.3 |
| XLMR\textsubscript{large}-A\textsubscript{Fusion} | 65.8 | 85.0 | 65.3 |

**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.

| Model | ZS-In (10) | Sup. (En) | Ave. |
|-------|------------|----------|------|
| mBERT | 62.6 | 83.5 | 47.2 |
| mBERT-A\textsubscript{Fusion} | 49.4 | 83.5 | 47.2 |
| XLMR\textsubscript{large} | 49.5 | 86.5 | 59.3 |
| XLMR\textsubscript{large}-A\textsubscript{Fusion} | 60.8 | 88.0 | 59.7 |

**Question Answering.** Following the setting of XTREME, We fine-tune 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 languages. Detailed results are in Table 13 in Appendix D. We find that mBERT-A\textsubscript{Fusion} maintains the performance as its original version, while XLMR\textsubscript{large}-A\textsubscript{Fusion} can be boosted slightly. In general, MLLMs-A\textsubscript{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.

| Model | RELX (4) | WikiAnn (10) | XQuAD (11) |
|-------|----------|--------------|------------|
| Acc. | F1 | F1 | EM |
| mBERT | 60.1 | -/- | -/- |
| mBERT-A\textsubscript{Fusion} | 61.1 | -/- | -/- |
| mBERT-MLKG | 64.7 | -/- | -/- |
| XLMR\textsubscript{base} | 58.7 | -/- | -/- |
| XLMR\textsubscript{base}-A\textsubscript{Fusion} | 61.7 | -/- | -/- |
| XLMR\textsubscript{large} | 61.3 | 68.5 | 76.6 |
| XLMR\textsubscript{large}-A\textsubscript{Fusion} | 61.9 | 66.9 | 76.5 |
| XLMR\textsubscript{large}-A\textsubscript{Fusion} | 64.6 | 67.6 | 76.8 |

**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 XLMR\textsubscript{large}-A\textsubscript{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.
We conduct ablation studies to understand our knowledge adapters and show that they work as expected.\footnote{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.} We also compare against a large adapter (\(A_{\text{Large}}\)) 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-\(A_{\text{Large}}\), and mBERT with our adapters on each downstream task.

In Figure 5, for the knowledge graph completion task (left), mBERT-A\(_{\text{TP}}\) and mBERT-A\(_{\text{TS}}\) perform better than their entity-based counterparts. While mBERT-\(A_{\text{Large}}\) also performs well, mBERT-A\(_{\text{Fusion}}\) outperforms it significantly. For the entity alignment task (center), the situation is reversed such that better performance is achieved by mBERT-A\(_{\text{EP}}\) and, mBERT-A\(_{\text{ES}}\). Our mBERT-A\(_{\text{Fusion}}\) also achieves comparable performance which is much better than mBERT-\(A_{\text{Large}}\) 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-A\(_{\text{Fusion}}\)) gives the best performance while mBERT-\(A_{\text{Large}}\) 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 cannot 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.
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 Alkhassi, 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 multilingual 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 multirelational 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, Gretchin 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 Up-punda, 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.
A Implementation Details

We implement the adapters using the AdapterHub library\textsuperscript{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\textsuperscript{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\textsuperscript{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).

\textsuperscript{15}https://adapterhub.ml/

\textsuperscript{16}Note that users may search more fine-grained hyperparameters, but the relative performance would not change.

\textsuperscript{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 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 (%) |
|-------|-------|------------|--------------|------------|-------|-------|------------|--------------|------------|
| af    | Afrikaans | 36.4       | 20.5         | 51.8       | ba    | Basque   | 8.0         | 10.7         | 8.7         |
| am    | Amharic  | 59.8       | 0.7          | 10.7       | bs    | Bosnian  | 64.2       | 28.3         | 71.3        |
| ar    | Arabic   | 35.5       | 91.0         | 42.3       | ca    | Catalan  | 53.5       | 91.0         | 42.3        |
| as    | Assamese | 84.2       | 28.3         | 71.3       | de    | German   | 84.2       | 28.3         | 71.3        |
| az    | Azerbaijani | 19.5     | 19.3         | 9.8        | el    | Greek    | 52.4       | 1.8          | 10.0        |
| ba    | Bashkir  | 52.4       | 1.8          | 10.0       | es    | Spanish  | 68.0       | 32.7         | 11.6        |
| be    | Belarusian | 18.0     | 32.7         | 11.6       | fi    | Finnish  | 11.9       | 23.6         | 19.4        |
| bg    | Bulgarian | 51.9       | 23.6         | 19.4       | fr    | French   | 18.3       | 34.6         | 11.5        |
| bn    | Bengali   | 54.8       | 18.3         | 34.6       | hr    | Croatian | 54.8       | 18.3         | 34.6        |
| br    | Breton    | 54.8       | 18.3         | 34.6       | hu    | Hungarian | 54.8       | 18.3         | 34.6        |
| bs    | Bosnian   | 64.7       | 27.3         | 18.6       | hy    | Armenian | 54.8       | 18.3         | 34.6        |
| ca    | Catalan   | 87.2       | 99.3         | 88.9       | id    | Indonesian | 39.0      | 54.4         | 43.3        |
| co    | Corsican  | 51.3       | 0.3          | 0.2        | ko    | Korean   | 64.7       | 27.3         | 18.6        |
| cs    | Czech     | 73.5       | 64.4         | 66.4       | ko    | Korean   | 39.6       | 54.4         | 43.3        |
| cy    | Welsh     | 61.4       | 54.4         | 43.3       | la    | Latin    | 77.5       | 57.5         | 74.4        |
| da    | Danish    | 98.5       | 90.7         | 98.5       | lb    | Luxembourgish | 77.5  | 57.5         | 74.4        |
| de    | German    | 98.5       | 90.7         | 98.5       | lt    | Lithuanian | 77.5      | 57.5         | 74.4        |
| el    | Greek     | 98.5       | 90.7         | 98.5       | lv    | Latvian  | 77.5       | 57.5         | 74.4        |
| en    | English   | 100.0      | 100.0        | 100.0      | nb    | Norwegian | 77.5      | 57.5         | 74.4        |
| es    | Spanish   | 100.0      | 100.0        | 100.0      | nl    | Dutch    | 77.5       | 57.5         | 74.4        |
| et    | Estonian  | 100.0      | 100.0        | 100.0      | nn    | Norwegian | 77.5      | 57.5         | 74.4        |
| eu    | Basque    | 74.0       | 37.4         | 54.4       | no    | Norwegian | 77.5      | 57.5         | 74.4        |
| fa    | Persian   | 58.5       | 37.4         | 54.4       | pl    | Polish   | 74.0       | 37.4         | 54.4        |
| fi    | Finnish   | 58.5       | 37.4         | 54.4       | pt    | Portuguese | 74.0     | 37.4         | 54.4        |
| fr    | French    | 58.5       | 37.4         | 54.4       | pt    | Portuguese | 74.0     | 37.4         | 54.4        |
| ga    | Irish     | 70.8       | 25.2         | 57.7       | ro    | Romanian | 39.6       | 54.4         | 43.3        |
| gl    | Galician  | 65.2       | 38.5         | 45.9       | ru    | Russian  | 39.6       | 54.4         | 43.3        |

Table 9: The performance of various tasks 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   | 9.2        |
| en       | 5984          | 7.3    | 16.9    | 8.8   | 18.3       | 15.2| 17.7     | 18.5  | 21.3       |
| es       | 4101          | 13.5   | 24.4    | 7.4   | 13.2       | 14.3| 17.2     | 17.7  | 20.5       |
| fr       | 4436          | 17.5   | 27.6    | 6.1   | 14.5       | 12.7| 15.4     | 17.4  | 19.9       |
| ja       | 25699         | 21.1   | 25.3    | 9.3   | 15.8       | 12.7| 15.4     | 17.4  | 19.9       |
| ast      | 2823          | -      | -       | -     | -          | -   | -        | -     | -          |
| ca       | 2959          | -      | -       | -     | -          | -   | -        | -     | -          |
| da       | 2566          | -      | -       | -     | -          | -   | -        | -     | -          |
| de       | 4059          | -      | -       | -     | -          | -   | -        | -     | -          |
| fi       | 2329          | -      | -       | -     | -          | -   | -        | -     | -          |
| fr       | 2582          | -      | -       | -     | -          | -   | -        | -     | -          |
| hu       | 2558          | -      | -       | -     | -          | -   | -        | -     | -          |
| it       | 3614          | -      | -       | -     | -          | -   | -        | -     | -          |
| nb       | 2717          | -      | -       | -     | -          | -   | -        | -     | -          |
| nl       | 4316          | -      | -       | -     | -          | -   | -        | -     | -          |
| pl       | 2998          | -      | -       | -     | -          | -   | -        | -     | -          |
| pt       | 3184          | -      | -       | -     | -          | -   | -        | -     | -          |
| ru       | 2887          | -      | -       | -     | -          | -   | -        | -     | -          |
| sv       | 2993          | -      | -       | -     | -          | -   | -        | -     | -          |
| zh       | 2591         | -      | -       | -     | -          | -   | -        | -     | -          |
| eo       | 963           | -      | -       | -     | -          | -   | -        | -     | -          |
| vo       | 164           | -      | -       | -     | -          | -   | -        | -     | -          |
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->de   | 16120         | 78.9    | 72.5  | 76.5  | 75.6       | 77  | 74.3    | 75.5  | 75.8       |
| en->es   | 6720          | 72.5    | 72.5  | 72.5  | 72.5       | 72  | 72.5    | 72.5  | 72.5       |
| en->fr   | 6720          | 72.5    | 72.5  | 72.5  | 72.5       | 72  | 72.5    | 72.5  | 72.5       |
| en->de   | 16120         | 78.9    | 72.5  | 76.5  | 75.6       | 77  | 74.3    | 75.5  | 75.8       |
| en->es   | 6720          | 72.5    | 72.5  | 72.5  | 72.5       | 72  | 72.5    | 72.5  | 72.5       |
| en->fr   | 6720          | 72.5    | 72.5  | 72.5  | 72.5       | 72  | 72.5    | 72.5  | 72.5       |

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       | 51.2   | 51.2       | 51.2  | 51.2     | 51.2  | 51.2       |
| de       | 51.2   | 51.2       | 51.2  | 51.2     | 51.2  | 51.2       |
| es       | 51.2   | 51.2       | 51.2  | 51.2     | 51.2  | 51.2       |
| fr       | 51.2   | 51.2       | 51.2  | 51.2     | 51.2  | 51.2       |
| tr       | 51.2   | 51.2       | 51.2  | 51.2     | 51.2  | 51.2       |
| average  | 51.2   | 51.2       | 51.2  | 51.2     | 51.2  | 51.2       |

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

| Language | mBERT  | mBERT-MLKG | XLM   | XLM-MLKG | XLM-R | XLM-R-MLKG |
|----------|--------|------------|-------|----------|-------|------------|
| en       | 41.5   | 41.5       | 41.5  | 41.5     | 41.5  | 41.5       |
| de       | 41.5   | 41.5       | 41.5  | 41.5     | 41.5  | 41.5       |
| es       | 41.5   | 41.5       | 41.5  | 41.5     | 41.5  | 41.5       |
| fr       | 41.5   | 41.5       | 41.5  | 41.5     | 41.5  | 41.5       |
| tr       | 41.5   | 41.5       | 41.5  | 41.5     | 41.5  | 41.5       |
| average  | 41.5   | 41.5       | 41.5  | 41.5     | 41.5  | 41.5       |
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 |