Abstract

In this work, we study whether multilingual language models (MultiLMs) can transfer logical reasoning abilities to other languages when they are fine-tuned for reasoning in a different language. We evaluate the cross-lingual reasoning abilities of MultiLMs in two schemes: (1) where the language of the context and the question remain the same in the new languages that are tested (i.e., the reasoning is still monolingual, but the model must transfer the learned reasoning ability across languages), and (2) where the language of the context and the question is different (which we term code-switched reasoning). On two logical reasoning datasets, RuleTaker and LeapOfThought, we demonstrate that although MultiLMs can transfer reasoning ability across languages in a monolingual setting, they struggle to transfer reasoning abilities in a code-switched setting. Following this observation, we propose a novel attention mechanism that uses a dedicated set of parameters to encourage cross-lingual attention in code-switched sequences, which improves the reasoning performance by up to 14% and 4% on the RuleTaker and LeapOfThought datasets, respectively.  

1 Introduction

Recent studies show that language models (LMs) are capable of logically reasoning over natural language statements (Clark et al., 2020b), reasoning with their implicit knowledge (Talmor et al., 2020), and performing multi-step reasoning via chain-of-thought prompting when the model size is large enough (Wei et al., 2022b; Kojima et al., 2022; Wei et al., 2022a). A separate line of work has focused on pre-training language models on multilingual corpora to enable knowledge transfer across different languages. These efforts led to multilingual language models (MultiLM) such as mBERT (Devlin et al., 2019), mT5 (Xue et al., 2021), and XLM-R (Conneau et al., 2020), which have been shown to generalize in a zero-shot cross-lingual setting (Pires et al., 2019a; Conneau and Lample, 2019). The cross-lingual transfer is often enabled by fine-tuning the MultiLM on a high-resource language (typically English) and then evaluating it on other target languages.

However, as most of the recent efforts on reasoning-related tasks have been centered around English, our knowledge of the multilingual reasoning capabilities of language models remains limited. In this work, we investigate the logical reasoning capabilities of MultiLMs, especially in monolingual and structured code-switched settings (Figure 1). In the monolingual setting, the context and the question are in the same language. In the structured code-switched setting, we refer to a setting where the context and question are in two different languages. The code-switched setting can be found in many realistic scenarios, such as when non-English speakers may ask questions about information that is unavailable in their native language (Asai et al., 2021).

For both reasoning settings, we conduct experiments using the RuleTaker dataset (Clark et al., 2020b), which contains artificial facts and rules, and the LeapOfThought dataset (Talmor et al., 2020), which incorporates real-world knowledge.  

Figure 1: An example of monolingual and code-switched reasoning. In code-switched reasoning, the context and question are in different languages.
into the reasoning context. Our results show that although MultiLMs perform well when fine-tuned in different languages (i.e., high in-language performance when fine-tuning and testing on the same language), their cross-lingual transfer can vary considerably, especially in the code-switched setting. We posit that the lack of code-switched data in MultiLM pre-training data makes fine-tuning on code-switched data inconsistent with pre-training.

To improve the code-switched reasoning capabilities of MultiLMs, we propose two methods. First, we propose a dedicated cross-lingual query matrix (section 4.1) to better model cross-lingual attention when the MultiLMs receive code-switched sequences as input. This query matrix is pre-trained on unsupervised code-switched data, either shared across all language pairs or specific to a single one. Then, we propose a structured attention dropout (see section 4.1), in which we randomly mask attention between tokens from different languages (i.e., context-question attentions) during training. This masking makes the fine-tuning phase more consistent with the pre-training by regularizing cross-lingual attention.

By mixing the two methods, we also experiment with an interfered variant of the cross-lingual query, which considerably improves cross-lingual generalization, especially in code-switched settings. We evaluate our methods for the code-switched setting and show they improve the cross-lingual transfer of MultiLMs by 14% and 4% for the RuleTaker and LeapOfThought datasets, respectively.

2 Motivation

Most prior work on reasoning with language models remains limited to monolingual (English) systems (Han et al., 2021; Sanyal et al., 2022; Shi et al., 2023; Tang et al., 2023). In this work, we investigate the reasoning abilities of MultiLMs, formulating an analysis in formal reasoning that evaluates MultiLMs on their ability to resolve logical statements. Given a set of facts and rules as context (in natural language sentences), the task is to predict whether a given statement is true.

In our multilingual reasoning setting, we assume a given set of languages \( \{L_1, L_2, ..., L_N\} \), and define \( L_c \) and \( L_q \) as the context and statement languages, respectively. Typically, MultiLMs are evaluated in a monolingual setup where \( L_c = L_q \). However, if MultiLMs are truly multilingual, we posit that they should also be able to reason in a scenario where \( L_c \neq L_q \). Thus, to evaluate the multilingual reasoning ability of MultiLMs, we first define four different evaluation setups based on the language of context or statement: (1) both the context and statement are always in one language (monolingual reasoning); (2) the context is always in one language, and the statement can be in any language; (3) the context can be in any language, but the statement is always in one language; and (4) both the context and statement can be in any language.

To have a reasonable baseline to compare with the code-switched setups, we first focus on the monolingual evaluation (1), in which we evaluate the reasoning ability of MultiLMs for nine typologically different languages. Then, by fine-tuning the models on code-switched data, we evaluate their performance for setups (2) and (3) where either the language of the context or the language of the question is different from the training data. This evaluation aims to study the possibility of teaching models to reason across languages in a code-switched setting, and to investigate the extent they can transfer their reasoning to other code-switched datasets. Finally, we hypothesize that in order to succeed in setup (4), the model would have to be strong in setups (2) and (3). Since our experimental results show that the MultiLMs struggle in these two setups, we focus on improving their performance for setups (2) and (3).

3 Multilingual Reasoning

In this section, we describe our evaluation of the logical reasoning capabilities of MultiLMs for monolingual and code-switched settings.

3.1 Analysis Setup

We run our experiments on two datasets focusing on multi-hop logical reasoning over natural language knowledge:

RuleTaker. This is a set of five datasets, each constrained by the maximum depth of inference required to prove the facts used in its questions (Clark et al., 2020b). This dataset is generated with the closed-world assumption (CWA), assuming a statement is false if it is not provable. Each example consists of facts and rules (i.e., context) and a statement (more details in Appendix A.1).

LeapOfThought (LoT). This dataset comprises ~30K true or false hypernymy inferences, verbalized using manually written templates (Talmor
et al., 2020). The hypernymy relations and properties are derived from WORDNET (Fellbaum, 1998) and CONCEPTNET (Speer et al., 2017). This dataset contains two main test sets: EXPLICIT REASONING which performs inference over explicit natural language statements, and IMPLICIT REASONING where the model must reason by combining the context with missing information that should be implicitly encoded by the model. We create a modified version of LoT, and use the IMPLICIT REASONING test set in our evaluation. The dataset modification pipeline and the reason behind using only the IMPLICIT evaluation setting is further discussed in Appendix A.2.

**Models.** We conduct all our experiments using the cased version of multilingual BERT (mBERT; Devlin et al. 2019) and the base version of XLM-R (Conneau et al., 2020). We train a binary classifier on top of the model’s classification token (e.g., [CLS] in mBERT) to predict whether a given statement/question is true or false. The model’s input is [CLS] context [SEP] statement [SEP] and the [CLS] output token is used for the classification. For evaluation, we measure the model’s accuracy. We use full fine-tuning for these experiments. The random baseline is 50% (binary classification).

**Languages.** Both RuleTaker and LoT datasets are only available in English. We translated these two datasets into eight languages using the Google Translate API. We have chosen typologically diverse languages covering different language families: Germanic, Romance, Indo-Aryan, and Semitic, and including both high- and medium-resource languages from the NLP perspective. These languages include French (fr), German (de), Chinese (zh), Russian (ru), Spanish (es), Farsi (fa), Italian (it), and Arabic (ar).

### 3.2 Reasoning Over Monolingual Data

The average in-language and cross-lingual zero-shot performance of mBERT for each source language are depicted in Table 1. For the cross-lingual zero-shot performance, we first fine-tune models on a single source language, test it on other languages, and then take an average of these results.

On the RuleTaker dataset, the model is able to learn the task for the Depth-0 subset nearly perfectly for almost all the languages, exhibiting relatively high cross-lingual transfer performance (~87%). However, for models trained on higher depths (i.e., requiring more reasoning hops), the model’s performance drops for both in-language and cross-lingual evaluation settings, and the performance gap between different source languages increases. Moreover, when increasing the depth, zero-shot cross-lingual performance suffers more compared to in-language performance, showing that as the complexity of the task increases, the harder it becomes to generalize to other languages.

For the LoT dataset, the model must learn to reason by combining its implicit knowledge of hypernyms with the given explicit knowledge. However, the model’s performance differs for different languages, suggesting that the model’s ability to access and use the implicit knowledge is not the same for all languages. We also observe that a language with high in-language performance does not necessarily have a high zero-shot cross-lingual performance. We hypothesize that for some languages, the model starts learning in-language noises that are not generalizable to other languages.

We generally observe the same patterns for the XLM-R model (see Appendix B) when fine-tuned on the monolingual RuleTaker and LoT datasets.

### 3.3 Reasoning Over Code-Switch Data

When we fine-tune the model using a code-switched data format, the context is in one language and the statement is in another. In our experiments, we use English as an anchor language for the context (i.e., en-X) or for the statement (i.e., X-en). In the fine-tuning phase, we learn the task using the en-X data format, and evaluate it on both en-X and X-en data formats. The models’ average in-language and zero-shot cross-lingual performance are shown in Table 2.

For Depth-0 of the RuleTaker dataset, mBERT is able to learn the reasoning task almost perfectly for most languages. As the depth of the task increases, the performance of the code-switched reasoning declines. This decline is more pronounced at higher depths compared to the monolingual scenario. While the model is capable of learning reasoning within this framework, its zero-shot generalization to other code-switched data, such as en-X (where the context language remains English but the statement language changes), is poor. Reasoning over two languages poses a greater challenge than reasoning within monolingual data due to the need for information alignment across languages. Consequently, the transferability of such tasks to other language pairs becomes more challenging.
The two samples are semantically the same, only having different statement languages. Cross-lingual performance is the average performance of the model being fine-tuned on a single source language and then zero-shot transferred to other languages.

Table 1: Monolingual Setting: In-language and cross-lingual zero-shot performance (accuracy) of the mBERT model for the RuleTaker and LeapOfThought datasets. Cross-lingual performance is the average performance of the model in the same language as the training data.

<table>
<thead>
<tr>
<th></th>
<th>RuleTaker</th>
<th></th>
<th>LeapOfThought</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Depth 0</td>
<td>Depth 1</td>
<td>Depth 2</td>
</tr>
<tr>
<td>en</td>
<td>100.00</td>
<td>87.96</td>
<td>93.37</td>
</tr>
<tr>
<td>fr</td>
<td>99.40</td>
<td>87.06</td>
<td>90.50</td>
</tr>
<tr>
<td>de</td>
<td>99.99</td>
<td>87.39</td>
<td>90.04</td>
</tr>
<tr>
<td>ar</td>
<td>99.41</td>
<td>89.57</td>
<td>90.77</td>
</tr>
<tr>
<td>es</td>
<td>99.48</td>
<td>80.20</td>
<td>90.35</td>
</tr>
<tr>
<td>zh</td>
<td>99.12</td>
<td>87.48</td>
<td>92.43</td>
</tr>
<tr>
<td>it</td>
<td>99.97</td>
<td>89.61</td>
<td>90.54</td>
</tr>
<tr>
<td>zh</td>
<td>99.81</td>
<td>90.09</td>
<td>93.14</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>99.78</td>
<td>87.67</td>
</tr>
</tbody>
</table>

Table 2: Code-Switched Setting: In-language and cross-lingual zero-shot performance (accuracy) of the mBERT model for the RuleTaker and LeapOfThought datasets. In-language performance corresponds to evaluating the model in the same language as the training data.

<table>
<thead>
<tr>
<th></th>
<th>RuleTaker</th>
<th></th>
<th>LeapOfThought</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Depth 0</td>
<td>Depth 1</td>
<td>Depth 2</td>
</tr>
<tr>
<td>en-fr</td>
<td>99.29</td>
<td>54.82</td>
<td>53.47</td>
</tr>
<tr>
<td>en-fa</td>
<td>97.46</td>
<td>54.04</td>
<td>52.05</td>
</tr>
<tr>
<td>en-de</td>
<td>99.63</td>
<td>54.26</td>
<td>52.69</td>
</tr>
<tr>
<td>en-ar</td>
<td>85.93</td>
<td>53.73</td>
<td>52.36</td>
</tr>
<tr>
<td>en-es</td>
<td>99.99</td>
<td>57.25</td>
<td>56.29</td>
</tr>
<tr>
<td>en-zh</td>
<td>100.00</td>
<td>52.68</td>
<td>51.81</td>
</tr>
<tr>
<td>en-ru</td>
<td>98.03</td>
<td>58.40</td>
<td>52.06</td>
</tr>
<tr>
<td>en-it</td>
<td>99.91</td>
<td>56.26</td>
<td>54.68</td>
</tr>
<tr>
<td>Average</td>
<td>97.53</td>
<td>55.18</td>
<td>53.18</td>
</tr>
</tbody>
</table>

On the LoT dataset, the model performs quite well on the code-switched data, outperforming the monolingual scenario for nearly all languages. The relatively high code-switched performance shows that the language of the context plays an important role in accessing the implicit knowledge encoded in the model’s parameters, as the model must rely on this knowledge to solve the task. Providing the context in English facilitates access to implicit knowledge compared to other languages. This is also inline with the empirical observation that generalization to X-en is considerably worse than en-X. We generally observe the same pattern for the XLM-R (see Appendix B) when fine-tuned on the monolingual RuleTaker and LoT datasets.

Following the empirical observations showing MultiLMs struggle to transfer reasoning abilities in a code-switched setting, we propose a novel attention mechanism to mitigate the lack of code-switched transfer in these models.

## 4 Cross-lingual-aware Self-Attention

Although MultiLMs have been pre-trained on multilingual corpora, individual inputs to the model stay mostly monolingual (Devlin et al., 2019; Conneau et al., 2020). When these models are fine-tuned on a code-switched downstream task, unlike the pre-training phase, tokens from different languages can attend to each other, which, as demonstrated in Tables 2 and 8, results in poor generalization to other code-switched language pairs. We also observe that self-attention patterns considerably change when we compare code-switched in-language and cross-lingual samples’ attention patterns⁴ (see Figure 4).

### 4.1 Approach

In order to make the fine-tuning phase more consistent with the pre-training, we propose two sets of methods to better handle the cross-lingual interactions of tokens.

**Cross-lingual Query** To better model the cross-lingual attention for code-switched tasks, we pre-train a cross-lingual query matrix $Q_{cross}$ (while keeping all other parameters frozen) on code-switched unsupervised data (more experimental...

---

⁴The two samples are semantically the same, only having different statement languages.
Structured Attention Dropout  

As mentioned earlier, poor generalization of MultiLMs in code-switched settings can be attributed to inconsistency between the pre-training and fine-tuning phases, where the former mostly deals with monolingual attention while the latter needs to handle cross-lingual attention as well. We propose that the consistency can be improved by limiting the cross-lingual attention in the fine-tuning phase (i.e., regularizing computational interactions between languages). As demonstrated in Figure 2, this can be achieved by randomly masking attention scores (i.e., attention dropout), with probability $P_{\text{mask}}$, when tokens from different languages attend to each other. Moreover, to ensure a reliable bridge between context and question, we never mask the attention scores of the first token (e.g., [CLS] in mBERT) to help the model better flow information between two sections. Table 11 demonstrates the importance of structured attention dropout for better generalization in code-switched settings.

Interfering Cross-lingual Query  

Given the promising performance of the attention dropout for code-switched tasks, we experiment with a variation of cross-lingual query, where queries $Q$ and $Q_{\text{cross}}$ also partially handle cross-lingual and monolingual attentions, respectively (see Figure 3b). We empirically observe that having attention masks that could randomly interfere with each other generally results in better performance (see Table 12) compared to the attention masks proposed in Figure 3a. In this scheme, $M_1$ and $M_2$ are generated randomly and online, but once sampled, the same masks will be used for all the attention heads in all layers (more details in Appendix D). Due to better empirical performance, this variation of the cross-lingual query will be used for all the following experiments.

4.2 Experimental Setup  

All models are trained with the AdamW optimizer (Loshchilov and Hutter, 2017) using the HuggingFace Transformers (Wolf et al., 2020) implementation for Transformer-based (Vaswani et al., 2017) models. The hyperparameters used for performing different experiments can be found in Appendix C. All the reported scores are averaged over three different seeds.

4A given sample can have different attention masks in different epochs.
Fine-tuning Setup. As Bitfit fine-tuning outperforms full fine-tuning for all our experiments, we only report the Bitfit results here (Zaken et al., 2021). In Bitfit tuning, only biases are tuned in the MultiLM encoder, together with classifier and pooler parameters.

Language Pairs. To show the effectiveness of the proposed method, we fine-tune the models on four typologically diverse languages (language of the statement), namely fr, de, zh, and ru. Our analysis shows that combining monolingual and code-switched data in the fine-tuning step improves the reasoning performance. Moreover, a multilingual reasoner should be able to reason over both monolingual and code-switched data. So, for this set of evaluations, we use a combination of English and en-X (half of each) as the training dataset, which we denote mix(en, en-X).

Pre-training Cross-lingual Query. We train a shared (Shared $Q_{\text{cross}}$) or language-pair specific (Pair $Q_{\text{cross}}$) cross-lingual query matrix. For Shared $Q_{\text{cross}}$, a shared cross-lingual query is trained on a parallel code-switched variant of XNLI (Conneau et al., 2018a), where an English premise is followed by the same premise but in another language. For Pair $Q_{\text{cross}}$, we train a cross-lingual query for each en-X language pair again using the XNLI dataset. In both cases, only the cross-lingual query matrix is trained and the rest of the parameters are frozen. The training happens for 500K iterations.

Baselines. We compare the performance of the proposed method against two baselines: (1) The pre-trained model (original) (2) a model pre-trained on code-switched data (CS-baseline). For the CS-baseline, we pre-train the model on the parallel code-switched variant of XNLI (similar to the data we use to learn the shared cross-lingual query matrix) for 500K iterations to adapt the model to the code-switched setting.

Cross-lingual Evaluation. For all the experiments, we evaluate the zero-shot performance of the model on (1) a monolingual setting (where both context and question are in one language), (2) an en-X code-switched setting (where the context is in English and the question is in other languages), and (3) a X-en code-switched setting (where the question is in English and the context is in other languages). For the case when we Bitfit fine-tune the model using a language-specific query matrix (Pair $Q_{\text{cross}}$), we use the query matrix of the target language during the inference (only the weights). For example, while doing the zero-shot evaluation on en-zh, we use the en-zh cross-lingual query matrix instead of the one from the fine-tuned model.

4.3 Experimental Results
Table 3 shows the average zero-shot transfer performance (accuracy) for the RuleTaker dataset. For both mBERT and XLM-R, introducing a shared cross-lingual query matrix (Shared $Q_{\text{cross}}$) improves the reasoning accuracy. These results underscore the significance of maintaining consistency between the pre-training and fine-tuning phases in code-switched downstream tasks to facilitate effective transfer learning.

Using a specific query matrix for each language pair (Pair $Q_{\text{cross}}$) further boosts the cross-lingual transfer performance across most tested settings (up to 18%). In this scenario, there is a dedicated set of parameters to learn the attention patterns for a language pair rather than having them share the same number of parameters among many different language pairs. In other words, dedicated parameters help the model learn attention patterns for specific language pairs.5

Interestingly, in many cases, our approach also improves the transfer performance for monolingual data (mono). We hypothesize that, by having a separate cross-lingual query matrix, the model does not need to learn the cross-lingual attention pattern using the same parameters, reducing the chance of overfitting to the code-switched format.

We also conducted a comparison with a code-switched baseline in which the MultiLM is pre-trained on a code-switched version of XNLI. The code-switched baseline (CS-baseline) showed improved transfer results for en-X format and, in some cases, performed competitively with the Pair $Q_{\text{cross}}$ approach. However, it negatively affected performance in monolingual and X-en scenarios, particularly for the mBERT model. In essence, the model exhibited overfitting to the language pairs in en-X format it was trained on, making it unable to generalize effectively to monolingual and other code-switched formats. On the other hand, both Shared $Q_{\text{cross}}$ and Pair $Q_{\text{cross}}$ demonstrated the ability to generalize their reasoning to

5There is no Pair $Q_{\text{cross}}$ for en-fa and en-it (as they are not part of the XNLI dataset), and all the transfer results for these two languages are fully zero-shot.
To study the effect of the cross-lingual query matrix on an implicit reasoning task, we expand our experimental setup to include the LeapOfThought (LoT) dataset. Table 4 illustrates the average zero-shot transfer performance for this dataset. For this dataset, our proposed method also enhances the reasoning ability of the models for all examined language pairs. However, the degree of improvement observed is smaller compared to the RuleTaker dataset. In the case of the implicit reasoning task within the LoT dataset, the model must rely on both contextual cues and implicit knowledge to successfully solve the task. Conversely, for the RuleTaker dataset, the model is required to fully reason over the context. Consequently, for implicit reasoning, the model only partially uses contextual information, resulting in a lesser impact on performance when improving cross-lingual context-question attentions.

### 4.4 Generalization to Other Reasoning Tasks

So far, our experiments have focused on the logical reasoning ability of MultiLMs, either in monolingual samples (mono) and code-switched language pairs (en-X and X-en). The original is the pre-trained model and the CS-baseline is the model that pre-trained on code-switched data. Shared Q\text{cross} and Pair Q\text{cross}, refer to cases where the cross-lingual query matrix Q\text{cross} is either shared across many language pairs or is specific to each language pair, respectively.
Table 4: Average cross-lingual transfer of mBERT and XLM-R on LoT dataset to monolingual samples (mono) and code-switched language pairs (en-X and X-en). The original is the pre-trained model and CS-baseline refers to the model pre-trained on code-switched data. Shared $Q_{\text{cross}}$ and Pair $Q_{\text{cross}}$, refer to cases where cross-lingual query is either shared across many language pairs or is specific to each language pair, respectively.

Table 5: Performance (accuracy) of mBERT model for the XNLI dataset in both monolingual and code-switched evaluation settings.

5 Related Work

Reasoning in NLP. Language models (LMs) have demonstrated their ability to perform logical reasoning over natural language statements (Clark et al., 2020b; Chen et al., 2023). They can also leverage their implicit knowledge for reasoning purposes (Talmor et al., 2020) and exhibit multi-step reasoning capabilities by utilizing chain-of-thought prompting, even with minimal demonstrations or instructions, when the model size is sufficiently large (Wei et al., 2022b; Kojima et al., 2022; Wei et al., 2022a; Tang et al., 2023). In parallel to English-centric efforts on reasoning tasks, there have been attempts to create multilingual reasoning datasets to evaluate the cross-lingual abilities of pre-trained MultiLMs (Conneau et al., 2018b; Artetxe et al., 2020; Clark et al., 2020a; Hu et al., 2020; Shi et al., 2022). Recent pre-trained large MultiLMs like BLOOM (Scao et al., 2022), BLOOMZ (Muenchhoff et al., 2022), and XGLM (Lin et al., 2021), exhibited promising few-shot performance on a variety of cross-lingual reasoning datasets using in-context learning (Brown et al., 2020). Prior works studied the reasoning ability of MultiLMs in the context of open-retrieval answer generation (Asai et al., 2021), and mathematical problem-solving in a multilingual setting via chain-of-thought reasoning (Shi et al., 2022). This work conducts the first investigation of the logical reasoning capability of MultiLMs and proposes a cross-lingual-aware attention mechanism to improve their performance.

Cross-lingual Transfer. MultiLMs such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), mT5 (Xue et al., 2021), and XGLM (Lin et al., 2021) have achieved state of the art results in cross-lingual understanding tasks by jointly pre-training Transformer models (Vaswani et al., 2020), mT5 (Xue et al., 2021), and XGLM (Lin et al., 2021) on many languages. These models have shown effective cross-lingual transfer for many tasks, including named entity recognition (Pires et al., 2019b; Wu and Dredze, 2019; Foroutan et al., 2022), cross-lingual natural language inference (Conneau et al., 2018a; Hu et al., 2020), question answering (Lewis et al., 2019), and commonsense reasoning (Tikhonov and Ryabinin, 2021). This study focuses on the cross-lingual transfer performance of MultiLMs in the context of logical reasoning.

Code-switched NLP. Code-switching is a linguistic phenomenon of alternating between two
or more languages within a single conversation or text. In recent years, code-switching-related research has been growing in the NLP community. The growth is motivated by the increasing need for NLP systems to handle code-switched data and call to pay more attention to multilingualism and low-resource languages (Doğruöz et al., 2021; Winata et al., 2022; Jose et al., 2020; Sitaram et al., 2019). Previous research has been done for a diverse range of tasks such as Language Identification, Part of Speech Tagging, Sentiment Analysis, and Automatic Speech Recognition (Winata et al., 2021; Khanuja et al., 2020; Ostapenko et al., 2022; Tarunesh et al., 2021). To the best of our knowledge, this work is the first to study logical reasoning in the context of code-switched NLP. Furthermore, a majority of prior studies have focused on word-level code-switching, where the language of certain words in a text randomly changes. However, our investigation delves into the realm of “structured code-switching”, wherein language transitions occur at a section level.

6 Discussion

In this study, we explored the effectiveness of MultiLMs in a code-switched setting and found that while these models exhibit strong reasoning capabilities in monolingual settings, they struggle when it comes to code-switching. To address this, we first proposed the structured attention dropout, which encourages the model to rely less on cross-lingual attention when dealing with code-switched data. This simple method considerably improved cross-lingual transfer to other code-switched languages, demonstrating the importance of structured attention for this setting. We then proposed a novel structured attention mechanism, incorporating the cross-lingual query, that helps the model to better handle cross-lingual attention in the code-switched setting. The proposed cross-lingual query matrix, pre-trained on unsupervised code-switched data, significantly improved the cross-lingual transfer to other code-switched language pairs in all studies settings, demonstrating the importance of code-switched alignment for MultiLMs. We also observed better cross-lingual code-switched performance for the LeapOfThought dataset (real-world knowledge contexts) compared to RuleTaker (utilizing artificial facts and rules). We attribute LeapOfThought’s better code-switched performance to the usage of real-world knowledge in the reasoning context (compared to artificial facts and rules in RuleTaker), in line with Tang et al. (2023) observation that language models perform better when provided with commonsense-consistent context, and struggle with artificial ones.

7 Limitations

In this work, we evaluate our proposed method on encoder-only language models, and the impact of this method on autoregressive models and encoder-decoder-only models has not been explored, leaving room for further investigation and evaluation. Moreover, our experiments are limited to relatively small language models (less than one billion parameters), and the results and our findings do not necessarily extend to large language models. Furthermore, we should highlight that the scope of our experiments is constrained by the availability of multilingual data and computational resources. Consequently, our evaluation is limited to two specific datasets and covers only nine languages. While we strive for diversity in our selection, it is important to recognize that broader and more extensive datasets encompassing a wider range of languages could offer additional perspectives and potentially reveal new insights.

Acknowledgments

We thank Debjit Paul, Syrielle Montariol, Angelika Romanou, and Deniz Bayazit for their helpful discussions and feedback on our paper. We also gratefully acknowledge the support of the Swiss National Science Foundation (No. 215390), Innosuisse (PFFS-21-29), the EPFL Science Seed Fund, the EPFL Center for Imaging, Sony Group Corporation, and the Allen Institute for AI.

References


A Dataset Details

This section further elaborates on the datasets that are used in our experiments. Both datasets in this study are translated using Google Translate API to investigate our proposed method’s cross-lingual transfer. Starting from the English dataset, the samples are translated into eight other languages, namely, French (Fr), Farsi (Fa), German (De), Arabic (Ar), Spanish (Es), Chinese (Zh), Russian (Ru), and Italian (It). Below we discuss in more detail each studied dataset.

A.1 RuleTaker Dataset

RuleTaker dataset (Clark et al., 2020b) is a set of five datasets, requiring various depths of inference to answer the questions. Each dataset consists of examples in the form of a triple: (context, statement, answer). The context is composed of a series of facts and rules, while the statement represents the question that needs to be proven and the answer is either “T” (true) if the statement logically follows from the context, or “F” (false) if it does not (false under a closed-world assumption, CWA). All the facts, rules, and question statements are expressed in synthetic English. Essentially, each example represents a self-contained logical theory in linguistic form, with a question asking, “Is it true?” The dataset generation procedure ensures that every question can be answered by a formal reasoner, given the closed-world assumption (CWA).

Each dataset limited by the maximum level of inference needed to validate the facts employed in its corresponding questions. These datasets are categorized based on their depth restrictions (up to depths $D = 0$, $D \leq 1$, $D \leq 2$, $D \leq 3$ and $D \leq 5$ respectively). A depth of $D = 0$ implies that the accurate facts can be readily “proven” by directly looking them up within the given context, without requiring any inference. The fifth dataset, encompasses questions that span up to a depth of 5. This dataset serves as a test to assess the ability to generalize to depths not encountered during training on the other four datasets. In our experiments, we use datasets with depths 0 to 4. Each dataset contains 100k examples randomly split 70/10/20 into train/dev/test partitions.

A.2 LeapOfThought (LoT) Dataset

The primary focus of the LoT dataset (Talmor et al., 2020) revolves around a specific form of inference that integrates implicit taxonomic knowledge (such
As demonstrated in Table 6, given the EXPLICIT evaluation results, the model can still get high reasoning performance even when the entities in the context and statement are not consistent. However, as reasoning performance on IMPLICIT evaluation set drops to (almost) random when noise are injected into the statement entities, we believe that LoT artifacts have less effect on this evaluation setting. Therefore, to evaluate the MultiLM’s reasoning performance, we use the IMPLICIT evaluation set throughout the paper.

As demonstrated in Table 6, given the EXPLICIT evaluation results, the model can still get high reasoning performance even when the entities in the context and statement are not consistent. However, as reasoning performance on IMPLICIT evaluation set drops to (almost) random when noise are injected into the statement entities, we believe that LoT artifacts have less effect on this evaluation setting. Therefore, to evaluate the MultiLM’s reasoning performance, we use the IMPLICIT evaluation set throughout the paper.

B \textbf{Multilingual Reasoning: XLM-R results}

Sections 3.2 and 3.3 discussed the in-language and cross-lingual performance of the mBERT model on monolingual and code-switched data. This section evaluates the XLM-R model on the same evaluation settings as mBERT.

Table 7 demonstrates the average in-language and cross-lingual zero-shot performance of XLM-R

Table 6: This table investigates the artifacts present in the LeapOfThought dataset. We evaluate the model’s performance when either different parts of a given sample are removed (e.g., context-only model) or different noises are injected into the statement (e.g., swapping the subject in the statement with a randomly selected entity). The experiments that involve swapping entities in the statement are performed on the modified version of LoT, as discussed in section A.2.1.
As discussed in section 4.2, our proposed model and the baselines' performances in Tables 4 and 3 are achieved by Bitfit tuning (Zaken et al., 2021). It has been previously observed by Tu et al. (2022) that parameter-efficient fine-tuning (PEFT) has better cross-lingual generalization than full fine-tuning. In our experiments, we also found out that using a PEFT method like Bitfit considerably improves our cross-lingual transfer across different languages.

Table 9 demonstrates the generalization improvement brought by Bitfit over full fine-tune baseline for the RuleTaker dataset, especially in code-switched settings. We observed similar pattern for other RuleTaker depths and the LoT dataset. It is worth noting that using a PEFT method especially helps with transfer to code-switched tasks, which is our main focus in this paper.

C.2 Curriculum Learning

For depths 2 and 3 of RuleTaker dataset, which involves more reasoning hops, we observed that curriculum learning (Bengio et al., 2009) makes the XLM-R training more robust. The curriculum learning is performed by first training the MultiLM for 3 epochs on a subset of dataset that has depth 0 (i.e., no hop is needed for reasoning), and then the training is continued on the full dataset. This technique not only makes the XLM-R training more robust, but also improves the final reasoning per-
formance.

C.3 Hyperparameters

The hyperparameters for all the experiments is provided in Table 10 for both mBERT and XLM-R models. We use the AdamW optimizer with a warmup ratio of 0.1 for all experiments.

D Cross-lingual Query

This section further discusses the methods proposed in section 4.1.

D.1 Structured Attention Dropout

As previously discussed in section 4.1, limiting the cross-lingual attention in the fine-tuning makes this phase more consistent with the pre-training, where the MultiLM mostly deals with monolingual attentions. Table 11 demonstrates that applying dropout on cross-lingual attentions (see Figure 2) considerably improves cross-lingual generalization in code-switched settings. Table 11 results are achieved by a 40% dropout on cross-lingual attentions (i.e., $P_{\text{mask}} = 0.4$).

D.2 Interfering Cross-lingual Query

Inspired by the promising performance of the structured attention dropout, we propose a setting where the query matrix $Q$ also partially handles the cross-lingual attentions, and cross-lingual query $Q_{\text{cross}}$ partially handles monolingual attentions. The only difference between the interfering cross-lingual query and the non-interfering scheme is their respective attention masks, $M_1$ and $M_2$, as illustrated in Figure 3. We also empirically demonstrate in Table 12 that the interfering scheme consistently performs better generalization than the non-interfering one, especially in the code-switched settings. For all the fine-tuning experiments with the interfering cross-lingual query, we use a 70% attention dropout (i.e., $P_{\text{mask}} = 0.7$), meaning that 70% of cross-lingual attentions for query $Q$, and 70% of monolingual attentions for query $Q_{\text{cross}}$ are masked.

E Attention Visualization

As discussed earlier, MultiLMs perform well on in-language, but when they are transferred to other languages (especially code-switched languages) their performance hinders considerably (see Table 2). This section first analyzes the attention pattern of baseline models, both on in-language and cross-lingual evaluation settings. Then, we analyze the attention pattern of our proposed model which incorporates cross-lingual query.

We hypothesize that in order to have a reasonable cross-lingual performance, the cross-lingual samples’ attention pattern should not change significantly compared to the in-language samples. Figure 4 visualizes the attention pattern between tokens in the last (baseline) mBERT layer across all attention heads. The mBERT model is fine-tuned on the mix(en, en-fr) depth-0 of RuleTaker dataset, so the en-fr sample is considered in-language and the en-ar sample is considered a zero-shot transfer. It is worth noting the two samples are semantically the same and only the questions are in different languages. Comparing the two samples’ attention patterns, we can see that the attention pattern considerably changes (especially the strong attention signals getting much weaker when en-ar sample is given as input), which to some extent explains the poor generalization of the baseline models to other code-switched tasks.

In contrast, as demonstrated by Figure 5, the attention pattern of our proposed method, which incorporates cross-lingual query, is much more stable between in-language (i.e., en-fr sample), and the zero-shot transfer (i.e., en-ar sample). We believe that the observed stability in the attention patterns makes our models more language-neutral compared to the baseline, which is also demonstrated by the significant cross-lingual improvements over the baselines in Tables 3 and 4.

F Detailed Cross-lingual Query Results

Tables 3 and 4 demonstrated the average cross-lingual transfer to either monolingual or code-switched settings. This section demonstrates the detailed cross-lingual performance of models with cross-lingual query and the Original and CS-baseline. Tables 13 and 14 present the detailed cross-lingual transfer of mBERT trained on the RuleTaker and LeapOfThought datasets, respectively. Tables 15 and 16 present similar detailed cross-lingual performance of XLM-R model on the RuleTaker and LeapofThought datasets, respectively.
### mBERT

<table>
<thead>
<tr>
<th>Train Method</th>
<th>Dataset</th>
<th>Epoch / Iteration</th>
<th>Batch Size</th>
<th>Learning Rate</th>
<th>Evaluation Metric</th>
<th>Attention Dropout Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full FT</td>
<td>RuleTaker</td>
<td>5</td>
<td>32</td>
<td>1e-5</td>
<td>Accuracy</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>LeapOfThought</td>
<td>10</td>
<td>32</td>
<td>1e-5</td>
<td>Accuracy</td>
<td>N/A</td>
</tr>
<tr>
<td>BitFit</td>
<td>RuleTaker</td>
<td>35</td>
<td>32</td>
<td>4e-4</td>
<td>Accuracy</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>LeapOfThought</td>
<td>30</td>
<td>32</td>
<td>4e-4</td>
<td>Accuracy</td>
<td>0.7</td>
</tr>
<tr>
<td>Pre-training Q</td>
<td>XNLI dataset</td>
<td>500,000</td>
<td>16</td>
<td>2e-5</td>
<td>Perplexity</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### XLM-R

<table>
<thead>
<tr>
<th>Train Method</th>
<th>Dataset</th>
<th>Epoch / Iteration</th>
<th>Batch Size</th>
<th>Learning Rate</th>
<th>Evaluation Metric</th>
<th>Attention Dropout Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full FT</td>
<td>RuleTaker</td>
<td>5</td>
<td>32</td>
<td>5e-6</td>
<td>Accuracy</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>LeapOfThought</td>
<td>10</td>
<td>32</td>
<td>5e-6</td>
<td>Accuracy</td>
<td>N/A</td>
</tr>
<tr>
<td>BitFit</td>
<td>RuleTaker</td>
<td>35</td>
<td>32</td>
<td>4e-4</td>
<td>Accuracy</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>LeapOfThought</td>
<td>30</td>
<td>32</td>
<td>4e-4</td>
<td>Accuracy</td>
<td>0.7</td>
</tr>
<tr>
<td>Pre-training Q</td>
<td>XNLI dataset</td>
<td>500,000</td>
<td>8</td>
<td>2e-6</td>
<td>Perplexity</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 10: Hyperparameters of the pre-training and fine-tuning experiments for mBERT and XLM-RoBERTa models. Learning rate decays linearly from the initial value to zero.

### Table 11: Average cross-lingual transfer of mBERT model when tuned on a mixture of English and English-French (mix(en, en-fr)) RuleTaker dataset (depth-0). The (zero-shot) cross-lingual transfer to code-switched tasks gets considerably better with structured attention dropout (see section 4.1), either in full fine-tune or Bitfit (Zaken et al., 2021) tuning.

<table>
<thead>
<tr>
<th>Training Method</th>
<th>Drop attention</th>
<th>Transfer Setting</th>
<th>Monolingual</th>
<th>en-X</th>
<th>X-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Fine-tune</td>
<td>Yes</td>
<td>90.00</td>
<td>65.57</td>
<td>62.74</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>89.48</td>
<td>62.68</td>
<td>59.53</td>
<td></td>
</tr>
<tr>
<td>Bitfit</td>
<td>Yes</td>
<td>82.98</td>
<td>73.20</td>
<td>68.24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>89.14</td>
<td>65.38</td>
<td>60.81</td>
<td></td>
</tr>
</tbody>
</table>

### Table 12: Average cross-lingual transfer of mBERT model when fine-tuned on a mixture of English and English-French (mix(en, en-fr)) RuleTaker dataset (depth-0). Both models incorporate language-pair-specific cross-lingual query (i.e., Pair $Q_{cross}$) and are trained with Bitfit tuning. The only difference between the two runs is whether an interfered version of the cross-lingual query is used or not. We can observe that the interfered variant consistently outperforms the other variant, in monolingual and code-switched settings.

<table>
<thead>
<tr>
<th>Training Method</th>
<th>Interfering</th>
<th>Transfer Setting</th>
<th>Monolingual</th>
<th>en-X</th>
<th>X-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitfit</td>
<td>Yes</td>
<td>93.65</td>
<td>77.79</td>
<td>68.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>91.96</td>
<td>73.08</td>
<td>66.28</td>
<td></td>
</tr>
</tbody>
</table>
Train Data Method RuleTaker Depth-0 en fr fa de ar es zh ru it en-fr en-it en-es en-zh en-ru en-de en-fa en-ar fr-en de-en ... language pairs or is specific to each language pair, respectively.
Scores are averaged across three different seeds.

**Cross-lingual query is either shared across many language pairs or is specific to each language pair, respectively.**

<table>
<thead>
<tr>
<th>Train Data</th>
<th>RuleTaker Depth-0</th>
<th>RuleTaker Depth-1</th>
<th>RuleTaker Depth-2</th>
<th>RuleTaker Depth-3</th>
<th>RuleTaker Depth-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 61 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-baseline</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-baseline</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-baseline</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CS-baseline</td>
<td>83.71 77.1 79.7 72.9 72.3 76.7 74.8 72.8 78.4 70.5 63.8 75.6 70.5 67.6 70.5 66 72 71.5 70.5 62.7 65 64 65.6 69.1 65.6 63.8 61.5 60 60.5 63.8 61.5 60 60.5 60.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 13:** Cross-lingual transfer of mBERT model on the RuleTaker datasets to either monolingual samples or code-switched language pairs (en-X and X-en). The *original* is the pre-trained model, and the *CS-baseline* is the model that continues pre-training on code-switched data. Shared Qcross and Pair Qcross refer to cases where the cross-lingual query is either shared across many language pairs or specific to each language pair, respectively. Scores are averaged across three different seeds.
Table 14: Cross-lingual transfer of mBERT model on the LeapOfThought dataset to either monolingual samples or code-switched language pairs (en-X and X-en). The original is the pre-trained model, and the CS-baseline is the model that continues pre-training on code-switched data. Shared $Q_{cross}$ and Pair $Q_{cross}$ refer to cases where the cross-lingual query is either shared across many language pairs or is specific to each language pair, respectively. Scores are averaged across three different seeds.

Table 15: Cross-lingual transfer of the XLM-R model on the RuleTaker datasets to either monolingual samples or code-switched language pairs (en-X and X-en). The original is the pre-trained model, and the CS-baseline is the model that continues pre-training on code-switched data. Shared $Q_{cross}$ and Pair $Q_{cross}$ refer to cases where the cross-lingual query is either shared across many language pairs or is specific to each language pair, respectively. Scores are averaged across three different seeds.
Figure 3: Illustration of the attention masks in Section 4.1. In the proposed scheme, two sets of independent query matrices ($Q$ and $Q_{\text{cross}}$) collaborate to compute the attention scores. Matrix $M_1$ enforces the $Q$ matrix to mostly focus on monolingual attentions, and matrix $M_2$ constrains the $Q_{\text{cross}}$ to mostly handle cross-lingual attentions. The difference between masks in the two figures are the structured attention dropout probability being either one (left) or less than one (right). It is worth noting that the first token (e.g., [CLS] in mBERT) is used as a bridge in both $M_1$ and $M_2$, meaning its respective attentions are not masked.

Table 16: Cross-lingual transfer of XLM-R model on the LeapOfThought (Implicit Evaluation Set) dataset to either monolingual samples or code-switched language pairs (en-X and X-en). The original is the pre-trained model, and the CS-baseline is the model that continues pre-training on code-switched data. Shared $Q_{\text{cross}}$ and Pair $Q_{\text{cross}}$ refer to cases where the cross-lingual query is either shared across many language pairs or is specific to each language pair, respectively.

Scores are averaged across three different seeds.
Figure 4: Attention visualization of the baseline mBERT model for in-language (en-fr) and zero-shot transfer (en-ar), both from depth-0 of the RuleTaker dataset. The underlying mBERT model is fine-tuned on the mix(en, en-fr) of the RuleTaker depth-0 dataset. We hypothesize that the poor cross-lingual transfer of baseline models to other code-switched languages partially originates from instability of attention patterns across different languages as depicted in above figures.
Figure 5: Attention visualization of the mBERT model with cross-lingual query for in-language (en-fr) and zero-shot transfer (en-ar), both from depth-0 of the RuleTaker dataset. The underlying mBERT model is fine-tuned on the mix(en, en-fr) of the RuleTaker depth-0 dataset. We can see that attention patterns for our proposed model is more stable between in-language and cross-lingual samples, compared to baseline model in Figure 4.