Enhancing Cross-Lingual Transfer in Low-Resource Languages: Leveraging Parameter-Efficient Adapters for Multilingual Extractive Question Answering

Chibeze Joseph Nwangwu

Bachelor of Science in Computer Science
The University of Bath
2023

Enhancing Cross-Lingual Transfer in Low-Resource Languages: Leveraging Parameter-Efficient Adapters for Multilingual Extractive Question Answering

Submitted by: Chibeze Joseph Nwangwu

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

In recent years, natural language processing (NLP) for low-resource languages has gained significance, driven by the urgency to preserve endangered languages and cater to over 4 billion users who interact using these languages. However, employing large multilingual models in NLP poses obstacles such as resource-intensiveness, complexity, and reduced interpretability. To address these challenges, recent studies have introduced parameter-efficient adapters, with AdapterFusion emerging as a notable solution.

This work expands on previous research by presenting an innovative adapter framework for zero-shot and few-shot cross-lingual transfer, to improve performance in low-resource language scenarios. In this study, we focus on Extractive question-answering tasks. The study aims to assess the adapter framework's performance and scrutinize two main hypotheses regarding its effectiveness. The first hypothesis suggests that AdapterFusion should yield performance gains if at least one task supports the target task. The second hypothesis proposes that, given the same condition, AdapterFusion should enhance performance in cross-lingual settings.

Our findings reveal a noticeable performance improvement using select adapter compositions in both high and low-resource contexts. By investigating these cutting-edge adapter composition techniques, this study contributes to ongoing efforts to overcome the challenges associated with large multilingual models and promote the advancement of NLP applications tailored for low-resource languages.

Contents

List of Figures

List of Tables

Acknowledgements

I extend my sincere gratitude to my family for their constant support and encouragement. I also thank my supervisor, Dr. Harish Tayyar Madabushi, for his invaluable guidance, and my friends for their unwavering support throughout this project.

Chapter 1

Introduction

The central question this research aims to investigate is the potential improvements in natural language understanding tasks like Question answering, particularly focusing on low-resource languages, by leveraging parameter-efficient Adapters (Pfeiffer et al. (2020a); Houlsby et al. (2019)).

Extractive question answering involves identifying and extracting a pertinent text span from a given context to answer a query. In recent years, large neural models capable of achieving state-of-the-art performance in question answering have garnered significant attention with applications ranging from intelligent search engines that perform document searches on vast datasets to sophisticated chatbot models like ChatGPT, BARD, and Bingchat. This surge in popularity is attributed to the advent of large pre-trained language models, which employ the transformer architecture introduced by Vaswani et al. (2017) to learn robust language models for various NLP tasks.

Large pre-trained language models have significantly advanced the field of natural language processing, demonstrating remarkable improvements across various tasks without the need for labelled data (Howard and Ruder (2018)). In addition to this, further advancements have been made with Cross-lingual language modelling to accommodate low-resource languages. This has led to the development of multilingual models, such as mBERT and XLMs (Devlin et al. (2019); CONNEAU and Lample (2019)). These multi-lingual models have gained considerable attention due to their capacity for knowledge transfer across languages in cross-lingual zero-shot learning and language translation settings.

However, recent studies by Artetxe, Ruder and Yogatama (2019); CONNEAU and Lample (2019) have highlighted that these models may not possess adequate capacity to represent all languages, resulting in a trade-off between language coverage and model capacity. This trade-off means that as more languages are included in the model, the understanding of each language decreases. For this reason, large multilingual models under-perform when compared to monolingual models for high-resource languages (Nozza, Bianchi and Hovy (2020); Eisenschlos et al. (2019)). This performance reduction is even more pronounced for low-resource languages (Ogueji et al. (2022)). The obstacles related to model capacity render the objective of incorporating all 7,000 languages into a single-language model exceedingly difficult. Coupled with the high cost of training these models, scaling up becomes a difficult and resource-intensive task. Other limitations of large language models arise when transferring to a task of interest, this process usually involves finetuning all the model weights on the

target task. despite this method gaining state-of-the-art results, it requires fine-tuning all the model weights, which leads to catastrophic forgetting (McCloskey and Cohen (1989)), in the cross-lingual setting, this phenomenon adversely impacts the already scarce low-resource representations in the model (Vu et al. (2022)). To address these issues adapters-tuning, a parameter-efficient model adaptation has been gaining popularity.

Motivated by these challenges. We aim to develop a more efficient approach to cross-lingual learning by leveraging parameter-efficient adapters, particularly in the context of low-resource languages. we focus on the question-answering (QA) task due to its widespread prevalence and dependence on language comprehension and contextual inference. The QA task serves as a valuable benchmark for evaluating multilingual models, as successful performance in low-resource languages can signify that the model is adept at capturing linguistic subtleties, comprehending context, and deducing pertinent information in spite of resource constraints.

In this study, We propose MAD-X Fusion, a cross-lingual learning framework consisting of a two-step training system designed to tackle the curse of multilingualism (CONNEAU and Lample (2019)) and promote knowledge sharing across multiple languages and tasks. **1)** We introduce the key components of our approach, **2)** we train language-agnostic, task-specific adapters using the MAD-X framework. **3)** We integrate the MAD-X framework (Pfeiffer et al. (2020c)) with AdapterFusion (Pfeiffer et al. (2020a)), and introduce four distinct adapter compositions for training Fusion. **4)** We train the Fusion layer on *SQuAD* (Rajpurkar et al. (2016)) and implement few-shot cross-lingual learning on language-specific datasets. **5)** We evaluate all four approaches. **6)** We show that our approach outperforms the baseline model (MAD-X framework) in zero-shot and we report significantly higher performance in few-shot settings.

Chapter 2

Literature, Technology and data Survey

2.1 Introduction

Recent advances in artificial intelligence (AI) and natural language processing (NLP) have enabled the development of state-of-the-art models for a variety of language tasks. While these models have shown impressive performance on high-resource languages such as English, they perform rather poorly on low-resource languages. Ruder (2019) named NLP for low-resource scenarios one of the four biggest open problems in NLP. The term low-resource underlines languages that suffer from a lack of annotated data and are often ignored by the NLP community. Low resource refers to languages that are lacking in training data. An example of this would be Hindi; Hindi is one of the most widely spoken languages in the world, with 545 million speakers. However, in the field of natural language processing, Hindi has historically been considered a "low-resource" language, meaning that there is relatively little data and fewer computational tools available for processing and understanding Hindi text. This is due to several reasons, including the lack of digital resources, lack of funding and language barriers. It is important to identify that low-resource language does consider not only languages but also the specific tasks and domains; for example, take the typically high-resource language English, applied to a specific task or domain could be classed as low-resource, as there may be a scarcity of annotated data for that task/domain in that language.

In this survey, we will focus on the specific challenges associated with LRLs and the methods and processes used to overcome these challenges; we will also examine the available datasets and technologies used to process LRLs and the effectiveness of these solutions. We will provide a comprehensive overview of the current state of the field and will help to identify areas for future research and development.

2.2 Background

What is low-level languages Low-resource languages (LRLs) are languages that have limited resources available for natural language processing (NLP) tasks, such as a small amount of text data, few annotated examples, and a lack of pre-trained models. These limited resources make it difficult to train effective NLP models for LRLs, which can result in poor performance

on tasks such as language translation, text classification, named entity recognition and more.

Why should we care about LRLs? There are several reasons why we care about LRLs. One key reason is the potential economic benefits that can be derived from improving NLP performance on LRLs. As discussed in a study by Magueresse, Carles and Heetderks (2020), many LRLs are spoken in Africa and India, which together account for over 2.5 billion speakers. Improving NLP performance on these languages could thus open up new opportunities for businesses and individuals in these regions.

Another reason to care about LRLs is the ethical motivation to protect endangered languages and cultures. As pointed out in a survey by Hedderich et al. (2020); Magueresse, Carles and Heetderks (2020), many LRLs are spoken by small communities and are at risk of disappearing. Improving NLP performance in these languages can help to preserve them and promote linguistic diversity.

Additionally, there is also a strong open-knowledge motivation. By allowing access to information and knowledge to people who speak LRLs, we can empower communities and promote social inclusion and equal opportunities.

What methods have been used to overcome LRLs challenges? In recent years, the field of natural language processing (NLP) has been faced with the challenge of processing low-resource languages (LRLs), which have limited annotated data available for model training. To address this issue, various methods have been proposed and studied.

One such approach is transfer learning, which has been demonstrated to improve the performance of NLP models on LRLs. This was highlighted in the study by Zoph et al. (2016) where sequential transfer learning was applied to a Neural Machine Translation (NMT) task. Another method that has gained attention is data augmentation, which involves increasing the amount of annotated data available for training. Feng et al. (2021) conducted a survey to investigate the improvements achieved through both rule-based data augmentation Zhang, Zhao and LeCun (2015) and generative data augmentation Liu et al. (2020).

Cross-lingual learning is another avenue that has been explored in overcoming the challenges posed by LRLs. This approach leverages knowledge from high-resource languages to improve the performance of NLP models on LRLs. Examples of cross-lingual pre-trained models include XLM Lample and Conneau (2019). Additionally, model adaptation techniques have also been demonstrated to be effective in addressing the challenges posed by Ko et al. (2021) showed the effectiveness of these techniques in tasks such as machine translation from high-resource to low-resource languages and vice versa.

In this literature review, we focus on examining the current state of transfer learning as a solution to the challenges posed by LRLs.

2.3 Methods and Processes

We will now discuss more methodically the techniques and methods of transfer learning with regard to inductive transfer learning, more specifically the Adapter modules in low-resource scenarios.

The solution to low-resource languages goes beyond creating newer languages-specific datasets that can be used for training. One method of solving the issue with low-resource learning

is transfer learning Nguyen and Chiang (2017); the idea is to have a model trained on a high-resource language and then continue training on a low-resource language by replacing the corpus; there are three types of transfer learning as stipulated by Pan and Yang (2010). A survey on transfer learning, theses are Inductive, Unsupervised and Transductive transfer learning.

What is transfer learning Transfer learning is a method where knowledge learned from a given domain in a given task is used in the learning process of the target domain and task. While data augmentation extends training data, transfer learning reduces the need for this and transfers learned knowledge to a target task or domain; in the aspect of languages, transfer learning can allow the transfer of knowledge from high-resource languages to low-resource languages Zoph et al. (2016); Lample and Conneau (2019); Nguyen and Chiang (2017). Furthermore, studies by Ruder et al. (2019) found that transfer learning methods have significantly improved the state-of-the-art on a wide range of NLP tasks. And Taylor and Stone (2009) found that transfer learning can help improve learning performance in a related but different task. All this is followed by the fact there has been a significant emergence of transfer learning architectures and methods for NLP in recent years, further indicating the relevance of transfer learning in NLP.

2.3.1 Transformers Attention model

The Transformer attention model is a deep learning architecture that was introduced by Vaswani et al. (2017). It has become a cornerstone of many state-of-the-art NLP models due to its efficiency, scalability, and effectiveness in capturing the dependencies between input sequences. The Transformer model is designed to handle long-range dependencies in sequential data and to perform well on tasks such as machine translation, sentiment analysis, and text classification.

The Transformer model is based on the self-attention mechanism, which allows the model to weigh each element's importance relative to every other element in the sequence. This self-attention mechanism is used in place of traditional recurrent neural network (RNN) or convolutional neural network (CNN) structures, which are less well-suited for modelling long-range dependencies.

The architecture of the Transformer model consists of an encoder and a decoder. The encoder takes an input sequence and produces a continuous representation of the input, which is then passed to the decoder. The decoder uses this representation to produce the output sequence. Both the encoder and decoder are comprised of multiple identical layers, each of which consists of two sub-layers: a self-attention mechanism and a fully connected feed-forward network.

The self-attention mechanism is computed by first computing the attention scores between each pair of elements in the input sequence. These attention scores are then used to compute a weighted sum of the input elements, which is then used as input to the feed-forward network. The output of the self-attention mechanism is then passed through a residual connection and layer normalization before being used as input to the feed-forward network.

The fully connected feed-forward network is a standard feed-forward neural network that applies multiple linear transformations to the input. The output of the feed-forward network is then passed through another residual connection and layer normalization before being used as the final output of the layer.

The Transformer model has been shown to be highly effective on a wide range of NLP tasks and has become the architecture of choice for many state-of-the-art models. For example, the transformer-based models BERT Devlin et al. (2019) and GPT-3 Brown et al. (2020) have achieved remarkable performance on various NLP benchmarks and have been widely adopted in industry and academia.

2.3.2 Adapters

Adapters have been gaining significant attention as a novel solution to fine-tune deep learning models for NLP tasks. Adapters are small neural network modules that can be inserted into existing pre-trained models to tailor them to specific tasks. Unlike traditional fine-tuning, adapters allow for the pre-trained models to be adapted to new tasks while retaining their pre-trained knowledge Han, Pang and Wu (2021); Wang et al. (2020a); Houlsby et al. (2019); Wang et al. (2021).

The concept of adapters was first introduced by Houlsby et al. (2019), who showed that adapters could be used to fine-tune pre-trained models for text classification tasks. Their findings demonstrated that adapters outperformed traditional fine-tuning methods and achieved state-of-the-art results on benchmark datasets.

Since then, adapters have been the subject of several studies in NLP. For example, in the work of Pfeiffer et al. (2020a), the authors show that the combination of multiple adapters can result in improved performance on NLU tasks. Several researchers have sought to investigate the benefits of utilizing adapter parameters for transfer learning tasks. Specifically, they have explored the efficiency gains that can be achieved by fine-tuning only the adapter parameters in the model rather than all parameters. Houlsby et al. (2019); Wang et al. (2021); Pfeiffer et al. (2020a) have demonstrated the potential for a substantial reduction in processing time through this approach in their work on parameter-efficient transfer learning.

2.3.3 Different settings of transfer learning

There are three different settings in transfer learning summarised in 2.1.

Inductive transfer learning

As defined by Pan and Yang (2010) as a learning method that aims to improve the target predictive function $f_t(.)$ in the target domain using knowledge from the source domain. This essentially means that we have different tasks in the source and target domains whilst only having labelled data in the target domain. Based on the definition of inductive learning, we see that a few labelled data are required in the target task. this approach can be particularly effective for low-resource transfer learning, where the target task has limited labelled data available for training. By leveraging the pre-trained model, the target model can effectively transfer knowledge from the source task, which can improve its performance on the target task.

In a survey on inductive transfer learning, four different contexts for transfer learning were identified, including transferring knowledge of instances, feature representation, parameters, and relational knowledge Pan and Yang (2010). Examples of methods for each of these contexts include cross-domain adaptation, multi-task learning, knowledge distillation, and transfer learning based on relation networks Jiang and Zhai (2007); Argyriou, Evgeniou and

Pontil (2006); Gao et al. (2008); Mihalkova, Huynh and Mooney (2007). These methods represent the broad range of techniques that can be applied in inductive transfer learning, but in the field of natural language processing (NLP), it is important to consider which processes are most relevant.

A more recent survey Alyafeai, AlShaibani and Ahmad (2020) identifies two primary processes of inductive transfer learning in NLP: sequential learning and multi-task learning. Sequential learning can be further subdivided into fine-tuning, adapter modules, feature-based learning, and zero-shot learning. Fine-tuning Lee, Dernoncourt and Szolovits (2017); Howard and Ruder (2018) involves adapting a pre-trained model to a new task by training it on task-specific data. Adapter modules Houlsby et al. (2019); Stickland and Murray (2019); Pfeiffer et al. (2020a) add small learnable components to a pre-trained model to make it more adaptable to a new task. Feature-based learning Mou et al. (2016) involves training a model to learn features that are specific to a new task. For example, in character, word, and sentence embeddings, models are trained to learn representations based on different levels of language Ling et al. (2015); Mikolov et al. (2013); Pagliardini, Gupta and Jaggi (2017). Finally, zero-shot learning Radford et al. (2019); Brown et al. (2020) allows a model to generalize to new tasks or objects based on knowledge learned from other related tasks or objects without any training examples for the new task or object. In the setting of inductive learning, we also have multi-task learning, which refers to the process of learning multiple tasks in a parallel fashion. There are several studies that have used and explored the capabilities of multi-task learning.

Transductive transfer learning

A term coined by Arnold, Nallapati and Cohen (2007) describes a transfer learning method where the source and the target are the same, whilst the domain can be different, as defined by the Pan and Yang (2010) transductive transfer learning is a transfer learning method that hopes to improve performance of a predictive function $f_t(.)$ in the target domain using knowledge from a source domain and source task, where the source and target domains are different, and the source and target tasks are the same. Transductive transfer learning aims to make predictions for the input data the model has seen during training. Transductive can be divided into two categories domain adaption and Cross-lingual Learning. Domain adaptation refers to the process of adapting to a new data distribution in the target domain, which is particularly useful when the new task has a different distribution or a limited amount of labelled data. In sentiment classification, for example, adapting to a new domain could involve classifying reviews about restaurants instead of hotels. Cross-lingual learning, on the other hand, involves adapting to a different language in the target domain, which is particularly useful when working with low-resource languages. One application of cross-lingual learning is cross-lingual language modelling, which has been studied to assess its impact on low-resource languages Adams et al. (2017). Another more recent study with cross-lingual transfer learning refers to the MAD-X Pfeiffer et al. (2020c) approach where adapters Houlsby et al. (2019) and adapter-fusion Pfeiffer et al. (2020a) has been used to combine multi-task learning, adapter modules and cross-lingual learning processes to improve performance on low-resource language tasks. In this case, we see a combination of the different transfer learning settings (inductive and transductive); this is in fact a common approach in many Machine learning problems. One of which is to use transductive learning to identify similar data points across different domains or tasks and then use inductive transfer learning to transfer knowledge from those similar data points to the new task or domain. For example, in natural language processing, a model could be trained on a large corpus of text in one language using transductive learning to identify

similar patterns in the target language, and then inductive transfer learning could be used to adapt the model to the new language. This is essentially what Pfeiffer et al. (2020c) have achieved in the MAD-X paper.

MAD-X as a solution to Low-resources scenarios: The study identifies that scaling up a multi-lingual model to cover many languages is limited by the curse of multilinguality Conneau et al. (2019). The study proposes a Mad-x framework to resolve this issue; the proposed solution involves adapters, adapter fusion and a new type of adapter, the invertible adapter.

The MAD-x Framework:

The framework is made up of three types of adapters: language, task, and invertible adapters.

Language adapters are a recent adapter method proposed by Pfeiffer et al. (2020a), which are trained on unlabeled data using MLM. The task adapters, on the other hand, are trained with labelled data using the language adapter of the source language. To perform zero-shot learning, the source language adapters (including the language-specific invertible adapters) are simply replaced with the target language adapters. Lastly, the invertible adapters are used to mitigate the gap between multilingual and the target language vocabulary. These adapters are stacked on top of the input embedding layer, and their inverse is placed prior to output embeddings.

Future works include applying MAD-X to different base pre-trained models, using adapters that are suited to languages of a certain property, evaluating additional tasks, and investigating leveraging pre-trained language models for improved transfer to truly low-resource languages.

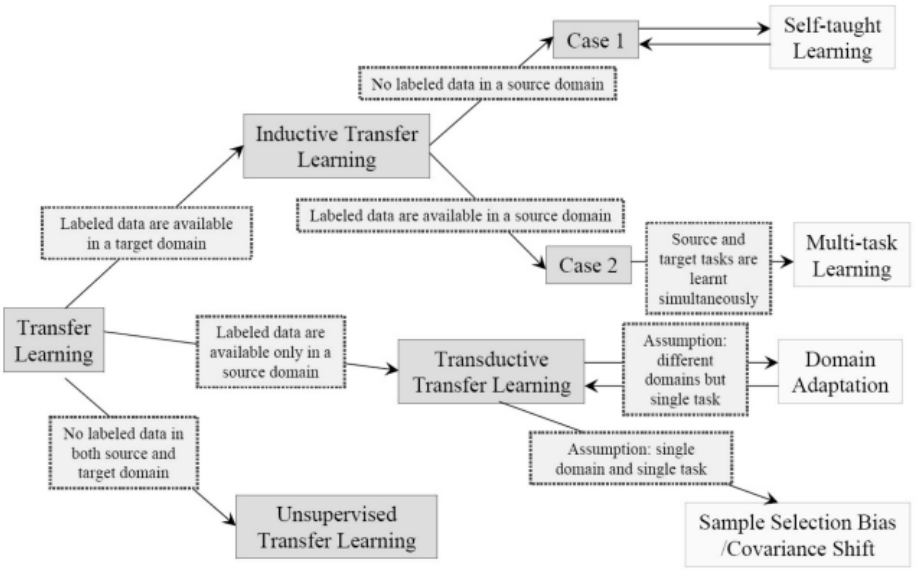

Figure 2.1: This is a summary of the three different settings of transfer learning Pan and Yang (2010)

2.4 Datasets

There are many datasets in NLP across many different domains and tasks. Huggingface hosts an impressive catalogue of publically available datasets, including SQUAD, COPA, IMBD, and more. below is a table of the datasets and the tasks they are curated.

Reference	Dataset/Benchmark	Tasks
Rahimi, Li and Cohn (2019)	WikiANN	NER
Artetxe, Ruder and Yogatama (2019)	XQUAD	QA
Raganato et al. (2020)	XL-WiC	semantic similarity classification
Ponti et al. (2020)	XCOPA	QA, multilingual Common science reasoning
Conneau et al. (2018)	XNLI	NLI
Lewis et al. (2019)	MLQA	QA
Rajpurkar et al. (2016)	SQUAD	QA

Table 2.1: A summary of the different Data-set and the learning tasks they are used for in multilingual NLP.

2.5 Technologies

2.5.1 Adapterhub

AdapterHub Pfeiffer et al. (2020b) is a platform for sharing and deploying pre-trained adapter modules in natural language processing (NLP). One of the key benefits of AdapterHub is that it provides a centralized repository of pre-trained adapter modules for a wide range of tasks, languages, and pre-trained models. This allows researchers and practitioners to easily find and download the adapter modules they need for their specific tasks without the need for extensive retraining or fine-tuning. Additionally, AdapterHub provides a platform for sharing and collaborating on adapter module development, which can help to accelerate progress in the field of NLP. Another benefit of AdapterHub is that it provides a user-friendly interface for deploying adapter modules in NLP applications. This makes it easier for practitioners with limited experience in NLP to use adapter modules in their applications, potentially increasing the adoption of pre-trained language models in the industry and other domains.

2.5.2 Huggingface

Hugging Face Wolf et al. (2019) is an open-source software library for natural language processing (NLP) developed by the Hugging Face team. The library provides state-of-the-art NLP models and a wide range of tools for working with natural language data, including text classification, question-answering, language modelling, and machine translation.

One of the main advantages of Hugging Face is its focus on usability and ease of use. The library is built on top of the popular deep learning framework PyTorch, which allows users to build and train their own NLP models easily. Additionally, Hugging Face provides a large and growing repository of pre-trained models that can be fine-tuned for specific NLP tasks. This enables developers and researchers to quickly and easily leverage the latest advancements in NLP without having to invest the time and resources necessary to train their own models from scratch.

2.6 Conclusion

In conclusion, transfer learning has emerged as a promising solution for addressing the challenges associated with low-resource languages. Adapters models have been shown to be a powerful technique for transferring knowledge across languages and tasks, leading to significant

improvements in model performance.

The MAD-X framework, which employs adapters models, has demonstrated state-of-the-art performance on several downstream tasks, such as named entity recognition, sentiment analysis, and natural language understanding, in low-resource languages. This framework has showcased the power of transfer learning and the effectiveness of recent adapter models in improving the performance of multi-lingual models.

As the demand for natural language processing in low-resource languages continues to grow, transfer learning and adapter models are likely to become increasingly important in addressing this challenge. Future research should investigate how to improve the effectiveness of adapter models, explore new transfer learning techniques, and evaluate their performance on a wider range of low-resource languages and tasks. It is crucial to develop more robust and effective approaches to address the issues surrounding low-resource languages, and it is essential that the natural language processing community continue to prioritize these efforts.

Chapter 3

Backgroud

The traditional approach to performing cross-lingual transfer learning generally involves a two-step process: 1) fine-tuning a pre-trained multilingual model on a specific task on the source language, and 2) evaluating the model's performance on the target language (Hu et al. (2020)). Pre-trained multilingual models, such as mBERT and XLM-R, have demonstrated remarkable success in facilitating this transfer, providing a strong foundation for downstream tasks in various languages (Pires, Schlinger and Garrette (2019); Hu et al. (2020)).

However, despite these advancements, there remain challenges to address, such as the limitations of large multilingual models in terms of capacity and potential performance trade-offs when compared to monolingual models specifically trained for a single language (Artetxe, Ruder and Yogatama (2019)). The MAD-X framework introduced by Pfeiffer et al. (2020c), utilises Adapters to address the capacity issue facing large multilingual models. They introduce **Language Adapters** for learning language-specific transformations and **Invertible Adapters** for expanding the shared vocabulary space allowing for language transfer to languages unseen by the base model in pertaining.

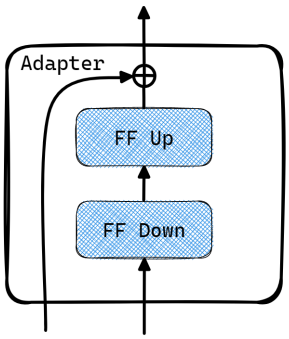

Figure 3.1: Adapter structure: The bottleneck adapter consists of a feedforward down-projection followed by a feedforward up-projection plus the residual weight is incorporated in every layer in the transformer model

3.1 Language Adpaters

Large multilingual models distribute parameters across all languages included in the model, which can result in reduced performance for individual languages compared to monolingual models. To address this issue, Pfeiffer et al. (2020c) propose learning language-specific representations using language adapters. The adapters employed in this approach follow the architecture outlined by Pfeiffer et al. (2020b), and the internal structure is based on the design introduced by Houlsby et al. (2019). These adapters, as illustrated in Figure: 3.1, consist of a feed-forward down-projection layer, with ReLU activation function, followed by a feed-forward up-projection layer and then an addition of the residual connection which is the output of the transformers feed-forward layer. The Language adapters are trained on unlabeled language-specific data from Wikipedia using Masked language modelling (MLM). By integrating these language adapters into the multilingual model, the framework can effectively learn language-specific representations, ultimately enhancing the performance of individual languages within the multilingual model.

3.2 Invertible Adapters

A majority of parameters in multilingual models are allocated to representing token embeddings. For instance, 47% of XLM-R's parameters are dedicated to its embedding matrix. This substantial allocation may result in underperformance for low-resource languages and even poorer outcomes for languages not encompassed within the model's training data. Recent research has attempted to address this issue; one notable example is E-MBERT, proposed by Wang et al. (2020b). E-MBERT extends the M-BERT vocabulary space to incorporate new languages and continues pre-training M-BERT on these languages. Although E-MBERT demonstrates significant improvements for both seen and unseen languages, it may not be the most parameter-efficient solution.

Pfeiffer et al. (2020c) present an alternative method using invertible adapters, which introduce a unique adapter architecture. These adapters address the discrepancy between the multilingual and target vocabularies. They are placed above the embedding layer, while their corresponding inverse adapters precede the output embedding layer. The invertibility feature allows the utilization of the same set of parameters for adapting both input and output representations. To guarantee invertibility, Nonlinear Independent Component Estimation (NICE) is implemented (Dinh, Krueger and Bengio (2014)). NICE facilitates the invertibility of arbitrary non-linear functions via a series of coupling operations. The invertible adapter serves a similar purpose as the language adapter, but it focuses on capturing token-level language-specific transformations. It is trained alongside language adapters using masked language modelling (MLM) on unlabeled data from a specific language. They show that invertible adapters improve performance on many language transfer pairs, especially in the case where the target language is low-resource and unseen during model pretraining.

3.3 Single-Task Adapters

Follows the common adapter training process introduced by Houlsby et al. (2019); Pfeiffer et al. (2020b). These adapters 3.1 are initialised with randomly assigned parameters ϕ_n and are trained on one specific task, like NER or SQUAD. In our approach we train Five single-task

language-agnostic adapters, each of which we think will aid the question-answering task.

In single-task adapters, the model is initially equipped with parameters Θ_0 for each of the N tasks. Furthermore, randomly initialized adapter parameters, denoted by Φ_n, are introduced. During training, only the adapter parameters, Φ_n, are adjusted, while the original parameters, Θ_0, remain unchanged. The objective for each task is to identify the optimal weights for the adapter parameters, which minimize the training loss. This process can be mathematically represented as follows:

$$\Phi_n \leftarrow \arg\min_{\phi}[L_n(D_n; \Theta_0, \Phi)] \tag{3.1}$$

where D_n represents the data, L_n represents the training loss, Φ_n represents the adapter parameters Θ_0 represents the original parameters of the pre-trained model.

3.4 Multi-Task Adpaters

Stickland and Murray (2019) introduced Multi-Task Adapters (MT-A) to concurrently train adapters for N tasks utilizing a multi-task objective. In this approach, the base parameters Θ_0 are fine-tuned in conjunction with the task-specific parameters, Φ_n. The training objective can be formulated as:

$$\Theta \leftarrow \arg\min_{\Theta,\Phi}(\sum_{n=1}^{N} L_n(D_n; \Theta_0, \Phi_n)) \tag{3.2}$$

where $\Theta = \Theta_0^{\{1,...,N\}}, \Phi_1, ..., \Phi_N$.

3.5 Task Adapter Comparison

Based on results observed in Pfeiffer et al. (2020a), we decided to employ single-task adapters as opposed to multi-task adapters as they are more efficient, only training the adapter parameters, highly flexible and produce high-performance results, outperforming multi-task adapters on many GLUE benchmarks.

3.6 MAD-X Framework

After training the source language adapters we add the adapters to the model and train task-specific adapters in the source language stacked on top of the source language adapters as shown in Figure 3.2. During training the language adapter along with the model weights are frozen while only the task adapter weights are trained. With this setting, we expect to produce language-agnostic adapters. To perform zero-shot cross-lingual transfer we replace the source language adapter with the target language adapter. In Figure 3.2 we replace the English language adapter with a Hindi language adapter.

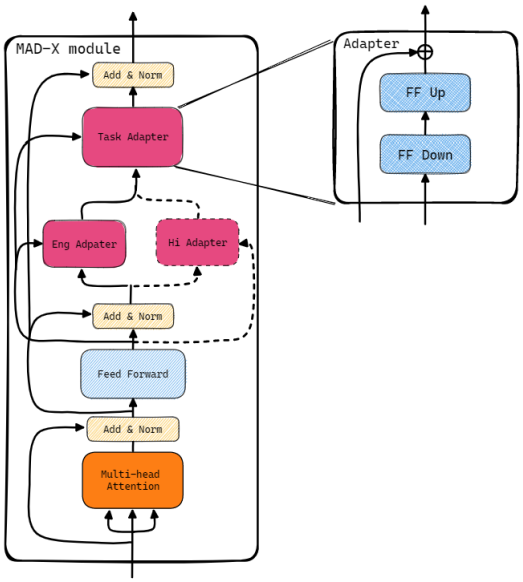

Figure 3.2: The transformer model incorporates the MAD-X framework, with language and task adapters integrated into each Transformer layer. Language adapters undergo masked language modelling (MLM) training while keeping the pre-trained multilingual model static. Task-focused adapters are placed over the source language adapters for downstream task training, like SQUAD (solid lines). In the zero-shot cross-lingual transfer process, target language adapters supplant source language adapters (dotted lines).

3.6.1 MAD-X Analysis

Transfer to unseen data

As referenced in the MAD-X paper cross-lingual transfer is possible to a language that is not seen by the base model (Pfeiffer et al. (2020c)). In the case of our experiments, we only investigate the MAD-X's ability to inject back into the model language-specific knowledge which was lost during pretraining, particularly in the case of low-resource languages. Hence we train and evaluate on low-resource languages with representation in mBert.

Source language

The choice of the source language can have a significant impact on the performance of a task adapter when evaluated on the target language. When the source language is a low-resource language, the performance of the task adapter may be lower than if the source language was a high-resource language (Pfeiffer et al. (2020c)). This is because training high-resource language has more data, hence can provide more information for the task adapter to learn from, resulting in a more robust representation of the task that can be transferred to the target language more effectively. Therefore, it is important to carefully consider the choice of source language when training task adapters for cross-lingual transfer learning. In cross-lingual research, English is commonly used as it meets the requirements owing to the extensive availability of both labelled and unlabeled data on the internet.

Language Agnostic Adapter

By stacking the task adapter on top of the language adapter, it is possible to produce a language-agnostic task representation. This is because the language adapter provides a learned representation of the language-specific information, allowing the task adapter to focus on learning the task-specific representation without incorporating much language-specific information. As a result, the language adapters can be replaced without affecting the performance of the task adapter, allowing for efficient transfer learning to unseen languages.

	Full	Pfeif.	Houl.
RTE (Wang et al., 2018)	66.2	70.8	69.8
MRPC (Dolan and Brockett, 2005)	90.5	89.7	91.5
STS-B (Cer et al., 2017)	88.8	89.0	89.2
CoLA (Warstadt et al., 2019)	59.5	58.9	59.1
SST-2 (Socher et al., 2013)	92.6	92.2	92.8
QNLI (Rajpurkar et al., 2016)	91.3	91.3	91.2
MNLI (Williams et al., 2018)	84.1	84.1	84.1
QQP (Iyer et al., 2017)	91.4	90.5	90.8

Figure 3.3: Mean development scores over 3 runs on the GLUE benchmark using the BERT-Base pre-trained weights. The results are shown for full fine-tuning and for the adapter architectures of Pfeiffer et al. (2020b) and Houlsby et al. (2019), both with a bottleneck size of 48. The scores are presented as F1 for MRPC, Spearman rank correlation for STS-B, and accuracy for the rest. RTE is a combination of datasets. source Pfeiffer et al. (2020b)

3.7 Adapters architecture

The architecture of Housbly et al. and Peffier et al. are similar but are slightly different, with one using two down and up-projection and the other only one at each transformer layer respectively. This means that Houlsby et al. (2019) has more capacity at the cost of training and inference speed whereas Pfeiffer et al. (2020b) through an exhaustive search on adapter position has come up with a more simple architecture that provides similar performance as shown in the Figure:3.3. In our approach, we follow Pfeiffer et al. in using only one adapter module at each transformer layer. From the result data presented in Pfeiffer et al. (2020a) we observe that Pfeiffer et al. architecture outperforms Housbly et al. on GLUE tasks with fusion. In our approach, we hope to leverage this single adapter architecture as it is simple to implement and train whilst maintaining high performance.

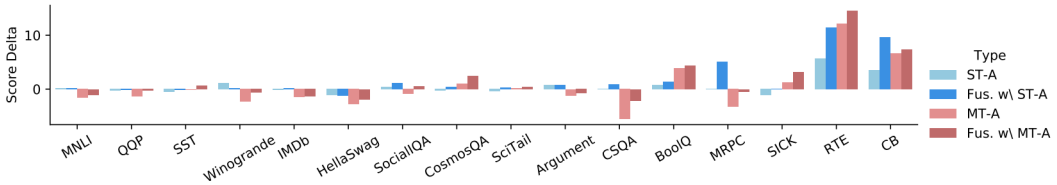

Figure 3.4: Reletive fusion score between multi-task (MT-A) and single-task(ST-A) adapters. where Fus. w ST-A is fusion with single-task adapters Pfeiffer et al. (2020a)

3.8 Training AdapterFusion

Tuning transformer models with Single-Task Adapters helps prevent catastrophic forgetting but does not facilitate information sharing between tasks. To tackle this issue, Pfeiffer et al. (2020a) proposed AdapterFusion, a method that enables knowledge sharing across tasks and has demonstrated intriguing results, as depicted in Figure 3.4. A key motivation for employing AdapterFusion is its performance dependence on training data. In the results shown in Figure 3.4, task-specific datasets are organized by size, with the largest dataset on the left and the smallest on the right. It is worth noting that the performance improvement becomes more pronounced as we move towards the right, due to the dependency on training data. Smaller datasets exhibit enhanced performance with fusion because low-resource task adapters capitalize on the representations of high-resource adapters. We hypothesize that this property can be applied to language-agnostic adapters by training the fusion layer on low-resource languages within a few-shot transfer learning framework.

Fusion Layer

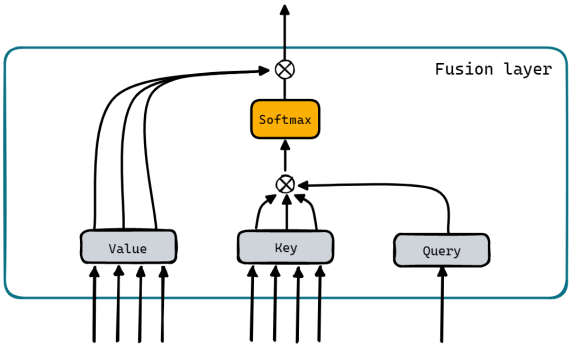

Figure 3.5: Fusion layer in AdapterFusion. The query takes as input the output of the pre-trained transformer weights. Both Key and Value take as input the output of the respective adapters. The dot product of the query with all the keys is passed into a softmax function, which learns to weigh the adapters with respect to the context. These weights are applied to each adapter output (values) using the dot product.

3.8.1 Training

During training, we fix the models' Θ parameter and the N adapters Φ parameters and only train the fusion parameters Ψ. In our experiment, we train on the same dataset twice, first training the target task adapter on dataset D_m and subsequently training the fusion layer on dataset D_x, where $m \subset \{1, ..., x\}$. Furthermore, we extend our investigation to a few-shot learning scenario in which we train the fusion layer on a significantly smaller, low-resource language dataset, D_y, where $y \subset \{1, ..., x\}$.

$$\Psi_m \leftarrow \underset{\Psi}{\operatorname{argmin}} L_m(D_m; \Theta, \Phi_1, \dots, \Phi_N, \Psi) \tag{3.3}$$

Multi-head Attention Mechanism

Drawing inspiration from multi-head attention mechanisms utilized in transformer models (Vaswani et al. (2017)), AdapterFusion employs a similar concept to determine the contextual

activation of each adapter. As depicted in Figure 3.5, the AdapterFusion parameters comprise Key, Query, and Value vectors. The Query vector accepts the output of the pre-trained transformer weights as input, while both Key and Value vectors receive the output of their corresponding adapters. Subsequently, the dot product of the Query vector and all Key vectors is fed into a softmax function, which learns to assign context-dependent weights to the adapters. The fusion layer receives inputs from multiple adapters, each trained on distinct tasks. It then learns a parameterized mixture of the encoded information, effectively combining the diverse knowledge sources for enhanced performance by promoting context-based knowledge sharing.

Chapter 4

Methodology

4.1 Problem Definition

The current state of NLP reveals an underrepresentation of low-resource languages, despite their significant speaker populations. Languages such as Hindi and Arabic, for instance, have limited representation in the ongoing AI revolution.

A key factor contributing to AI's popularity is the development of large, sophisticated chatbot models like ChatGPT. Among ChatGPT's 13 million unique daily users, a significant portion relies on the chatbot for answering questions, which includes non-English questions and contexts. Recent research evaluating ChatGPT's performance in a multilingual learning setting found a *substantial performance gap* between task-specific multilingual models and ChatGPT's generalized model, along with decreased performance in non-English languages (Dac Lai et al. (2023)). Considering that these chatbot models have the largest parameter counts seen thus far and are pre-trained on the entire web, it is reasonable to assume that they have been trained on more non-English data than any other cross-lingual models. Nevertheless, they still exhibit lower performance on task-specific comparisons involving low-resource languages, especially on complex Language comprehension and reasoning tasks.

In this study, we aim to address these challenges in a parameter-efficient manner by utilizing adapter methods on the question-answering task. We initialize and train carefully chosen language-agnostic single-task adapters using the MAD-X framework and combine these adapters with AdapterFusion. We refer to this approach as MAD-X Fusion.

4.2 Base Multilingual Models

In our experiments, we leverage the power of Multilingual BERT (mBERT) (Devlin et al. (2019)), a pre-trained deep learning model that provides sentence representations for 104 languages. mBERT is trained on large amounts of unlabeled text data from multiple languages, allowing it to learn a language-neutral representation that can be used for cross-lingual transfer learning. The data was sourced from Wikipedia and languages that had less data were over-sampled while languages that had large amounts of data were under-sampled. Despite being trained without explicit cross-lingual signals, mBERT has shown good cross-lingual performance on several NLP tasks. However, most evaluations of mBERT have focused on high-resource languages, covering only a third of the languages supported by the model. All

the low-resource languages used in our experiment are represented in mBERT and have shown poor performance as demonstrated by Artetxe, Ruder and Yogatama (2019); CONNEAU and Lample (2019). The adapter methods we use are base model agnostic, meaning that any model can be adapted using these frameworks.

4.3 Data

In this section, we describe the data used in our experiments, as well as the evaluation process for extractive question answering. For training our model, we utilized the SQuAD (Stanford Question Answering Dataset) dataset (Rajpurkar et al. (2016)). SQuAD is a reading comprehension dataset consisting of over 100,000 question-answer pairs, derived from over 500 Wikipedia articles. The dataset tests a model's ability to read a passage of text and then answer questions about it by extracting the relevant spans of text from the passage.

For evaluation and testing, we utilized the datasets outlined in Table 4.1. We characterize a SQuAD-style dataset using the following criteria: A dataset is comprised of context text, posed questions, and corresponding answers. Each answer is a specific text segment and span derived from the associated context. Adopting this format facilitates consistency in both pre-processing and post-processing steps, ensuring a robust assessment of the model's performance in the context of extractive question answering.

4.3.1 Evaluation Metric

The evaluation process for extractive question answering involves measuring the performance of the model on the test datasets by comparing the model's predictions with the ground truth answers. Two common metrics used for this purpose are the **Exact Match (EM)** score and the F1 score. The EM score calculates the percentage of predictions that exactly match the ground truth answers, while the F1 score computes the harmonic mean of precision and recall, considering partial matches between the predicted and ground truth answers.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.1}$$

To ensure a comprehensive evaluation of our model's performance, we performed both single-language and cross-lingual evaluations. In the single-language evaluation, we assessed the model's performance on the test datasets for the source language. For the cross-lingual evaluation, we evaluated the model's ability to generalize its knowledge from one language to another, testing its performance on the non-English datasets without any additional fine-tuning.

4.3.2 Datasets And Languages Adapters.

Considering our computational limitations compared to the team at AdapterHub (Pfeiffer et al. (2020b)), we chose to utilize the open-source language adapters available on the AdapterHub.ml website instead of training our own. This decision imposes certain constraints on the datasets we can evaluate on, as it affects the model's ability to adapt to new unseen languages, i.e., languages that the model has not been exposed to during pre-training.

The suboptimal performance in such scenarios can be attributed to the absence of relevant language-specific invertible adapters. These invertible adapters play a crucial role in expanding

the vocabulary space of the base model, enabling it to properly accommodate unseen languages. Without the specialized invertible adapters, our model's capacity to adjust to these new languages is limited.

Consequently, the number of target languages we can transfer to is restricted by the availability of pre-trained language adapters on the AdapterHub.ml website.

Language adapters for low resource languages

We select our target languages through a sieving process where we filter the languages by their Wikipedia representation, QA dataset availability and availability of language adapters. This is because language adapters were trained on Wikipedia data, where languages with lower representation on Wikipedia are over-sampled, therefore, we filter through a list of low-resource languages with less than 160,000 Wikipedia articles. we higlight the characteristic of the languaes selected in 4.1.

Language	ISO Code	# of wiki articles	In mBERT	Datasets
English	en	6.6M *(high)*	Y	SQuAD2.1
Hindi	hi	156k	Y	XQuAD.hi2.1, MLQA2.1
Bengali	bn	137k	Y	SQuAD_bn, tydiqa_xtreme
Swahili	sw	77k	Y	tydiqa_xtreme
Icelandic	is	57k	Y	Icelandic-qa-NQiI

Table 4.1: Selected languages after filtering 320 languages from Wikipedia.

Language	Dataset	Reference
Bengali	SQuAD_bn	Bhattacharjee et al. (2021)
Icelandic	Icelandic-qa-NQiI	Snæbjarnarson et al. (2021)
Swahili	tydiqa_xtreme	Clark et al. (2020); Ruder et al. (2021)

Table 4.2: language specific Question Answering datasets

4.3.3 Data pre-processing

In the pre-processing stage for a question-answering task for SQuAD-like dataset, the raw data undergoes a series of transformations to prepare it for model training. The process begins by removing unnecessary left whitespace in questions, ensuring better tokenization and preventing issues that may arise during context truncation. Then, the text is tokenized into individual units, such as words or subwords, based on the model's requirements. During tokenization, options for truncation, padding, and stride are configured based on specific flags. The tokenizer generates tokenized versions of the context and question pairs, applying truncation to either the context or question depending on certain conditions. Stride is employed to handle long contexts, creating overlapping segments as required, while padding is applied if the corresponding flag is set.

After tokenization, overflow mappings and offset mappings are extracted to establish relationships between the original examples and the generated features. Offset mappings, in particular, map tokens to their character positions in the original context, facilitating the

accurate computation of start and end positions for the answers. Once the mappings are established, the tokenized examples are labelled as follows:

1. The CLS token index is assigned as the default answer for impossible answers or cases with no given answers.
2. The start and end character positions of the answer in the text are calculated.
3. The start and end token indices of the current span in the text are determined based on sequence IDs.
4. The example is labelled with the CLS index if the answer is outside the span. (i.e. the answer is split across two strides.)
5. If the answer is within the span, the start and end token indices are adjusted to match the answer boundaries.

Upon completion of the pre-processing stage, the tokenized examples, labelled with the appropriate start and end positions, are returned, and effectively prepared for subsequent model training.

4.4 MAD-X Fusion

We introduce MAD-X Fusion a two-step parameter-efficient cross-lingual transfer framework. Our framework utilises the MAD-x framework Pfeiffer et al. (2020c) as a foundation and builds upon this with AdapterFusion Pfeiffer et al. (2020a). Our framework is based on parameter efficient adapters Houlsby et al. (2019); Pfeiffer et al. (2020b), which allows for knowledge transfer by introducing a small number of tasks specific parameters for model adaptation.

4.4.1 Our Approach

We adhere to the following steps: 1) We train a collection of task-specific, language-agnostic adapters that could potentially contribute to the target task (e.g., QA) in the source language. 2a) We combine all the task adapters trained earlier and train our fusion layer on the source language. 2b) As an alternative, we can integrate all the task adapters trained earlier and train our fusion layer on a distinct language. 3) Subsequently, we exchange the source language adapter with the target language adapter, facilitating knowledge transfer. Lastly, 4) we assess our model under both zero-shot and few-shot learning conditions.

By employing zero-shot learning, we seek to examine the model's capacity for generalization and adaptation to new languages without any prior exposure during training. In few-shot learning, the model receives a limited number of samples in the target language, enabling it to refine its knowledge and enhance its performance further.

4.5 Particular Adapters To Aid Question Answering Task

Training the Fusion layer requires a decent amount of computation as the model needs to in parallel perform inference on all the adapters present in the fusion layer and learn some attention context from them. Due to our computational limitations, we were not able to train

our fusion layer on a large number of tasks. To mitigate this we strive to first understand what task may aid QA to do this we study QA and introduce a four-point criteria for evaluating tasks that may aid question answering.

1. Comprehension: Tasks that require a deep understanding of the context and can extract relevant information.
2. Commonsense Reasoning: Tasks like the CommonsenseQA or the Winograd Schema Challenge can help the model develop reasoning skills based on general knowledge and commonsense understanding, which can be useful for answering questions that require such knowledge.
3. Textual Entailment and Natural Language Inference: Tasks like the MultiNLI or SNLI datasets require the model to understand the logical structure and reasoning within a text, which can be beneficial when dealing with complex QA tasks.
4. Entity recognition and linking: Include adapters trained on Named Entity Recognition (NER) or Entity Linking tasks to help the model identify and disambiguate entities in the text. This is often relevant for extractive QA tasks, especially when dealing with questions about specific entities.

The choice of adapters for an extractive QA task should be focused on these points to provide the necessary knowledge to identify and extract relevant information from the given context. Adapters selected must fulfil at least one criterion.

The Tasks we have found to meet some or all of the requirements are:

$$QA, Mnli, QQP, SocialiQA, NER$$

Chapter 5

Implementation

In this chapter, we will explain the different Python libraries used to create our model and the various components involved in the implementation.

5.1 Implementing transformers

5.1.1 HuggingFace

HuggingFace is a widely-used open-source library that provides state-of-the-art models and tools. It offers the Transformers library, which simplifies the use of transformer-based models for various tasks, such as text classification, translation, and question-answering.

5.1.2 Dataset

The HuggingFace Datasets library is used for loading and processing datasets. This library provides a standardized format for handling various datasets, enabling efficient data pre-processing and evaluation. The following code snippet demonstrates loading a dataset using the HuggingFace library:

```
from datasets import load_dataset

dataset = load_dataset('squad')
```

5.1.3 Models

The HuggingFace Transformers library provides a wide range of pre-trained models and architectures. It simplifies the process of fine-tuning these models for specific tasks. The following code snippet shows how to load a pre-trained model using the HuggingFace library:

```
from transformers import AutoModelForQuestionAnswering, AutoTokenizer

model =
    AutoModelForQuestionAnswering.from_pretrained('bert-base-uncased')
tokenizer = AutoTokenizer.from_pretrained('bert-base-uncased')
```

5.2 AdapterHub

AdapterHub is an extension of the HuggingFace Transformers library, specifically designed for training and using adapters. Adapters are lightweight, modular components that can be plugged into pre-trained transformer models, allowing for task-specific fine-tuning without modifying the original model's weights.

5.2.1 Language Adapters

Language adapters are used to adapt a pre-trained model to a specific language. They enable more efficient fine-tuning of models for low-resource languages. The following code snippet demonstrates how to load a language adapter using the AdapterHub library:

```
from transformers import AdapterConfig

language_adapter_name = "adapter_name"
config = AdapterConfig.load("pfeiffer")

model.load_adapter(language_adapter_name, config=config)
```

After loading the language adapter, it can be combined with task-specific adapters using the Adapter Fusion technique, which enhances the model's performance on multilingual tasks.

Chapter 6

Experiments

6.1 Baseline

Our baseline model follows the MAD-X framework, which we have chosen due to its recent success in enhancing cross-lingual transfer between high-resource languages and low-resource languages, both represented and unrepresented in the pre-trained base multilingual model. This achievement is particularly significant in the NLP domain, as it addresses one of the major challenges in the field, as noted by Ruder (2019), which is handling low-resource scenarios and the capacity limitations in multilingual models.

In our approach, we train a SQuAD adapter on an English QA dataset, with the task adapter (SQuAD) stacked on top of an English language adapter. Our aim is to create a language-agnostic adapter for the question-answering task. Subsequently, we perform a zero-shot transfer to each low-resource language listed in Table 4.1. This transfer is achieved by swapping the source language adapter with a target language adapter and evaluating the model using corresponding datasets outlined in Table 4.1.

Baseline Results

Model	en	hi	bn	sw	is	Avg.
MAD-X^{mBERT} -*INV*	81.39	50.72	48.90	57.85	39.83	55.74
MAD-X^{mBERT}	**82.39**	**50.73**	48.90	57.85	39.83	**55.94**
mBERT *FT. squad*	89.12	55.29	56.29	60.99	41.57	60.65

Table 6.1: F1 results for the baseline models, evaluated using the MAD-X framework with and without (-*INV*) invertible adapters. The table also includes mBERT fine-tuned (FT) on the SQuAD dataset, which covers English and Hindi, but none of the other languages in our experiments were present during fine-tuning.

6.2 Experiment setup

6.2.1 Experimental Environment

In our study, we utilized the HEX GPU cluster for our experiments. This cluster comprises 11 nodes with a total of 272 CPU cores and 46 GPUs. The GPUs include 16 NVIDIA RTX 3090s, 8 NVIDIA RTX 3090s, 4 NVIDIA RTX A4000s, and 18 NVIDIA RTX 2080s. Our experiments were conducted on a single NVIDIA GeForce RTX 2080 GPU with 7.9 GB of memory. We employed Docker containers based on PyTorch version 1.13 with Python version 3.10, and CUDA version 11.6.

6.2.2 Task Adapter Training

We train our language-agnostic Task adapters from scratch, using the MAD-X framework. The table below presents the training datasets we employed for this purpose:

Task	HuggingFace Dataset	Reference
NER	WikiANN	Rahimi, Li and Cohn (2019)
QQP	GLUE	Wang et al. (2018)
MNLI	GLUE	Williams, Nangia and Bowman (2017)
SocialIQA	Social_i_qa	Sap et al. (2019)
SQuAD	SQuAD	Rajpurkar et al. (2016)

Table 6.2: Overview of the datasets utilized for training language-agnostic adapters using the MAD-X framework.

In Table: 6.2, we provide details on the tasks, the corresponding HuggingFace datasets, and the associated references for each dataset. The tasks include Named Entity Recognition (NER), Quora Question Pair (QQP), Multi-Genre Natural Language Inference (MNLI), Social Intelligence QA (SocialIQA), and the Stanford Question Answering Dataset (SQuAD). By using the MAD-X framework, we aim to create language-agnostic adapters that can efficiently handle various tasks in a multilingual context.

6.2.3 Fusion Training Compositions

Upon realizing that fusing different classification types might impact the performance of our model on the least represented classification type, we decided to explore various compositions for training Fusion. In Table 6.3, we outline the various tasks along with their corresponding classification types. As mentioned in the AdapterFusion paper, the authors only fuse adapters of similar or the same

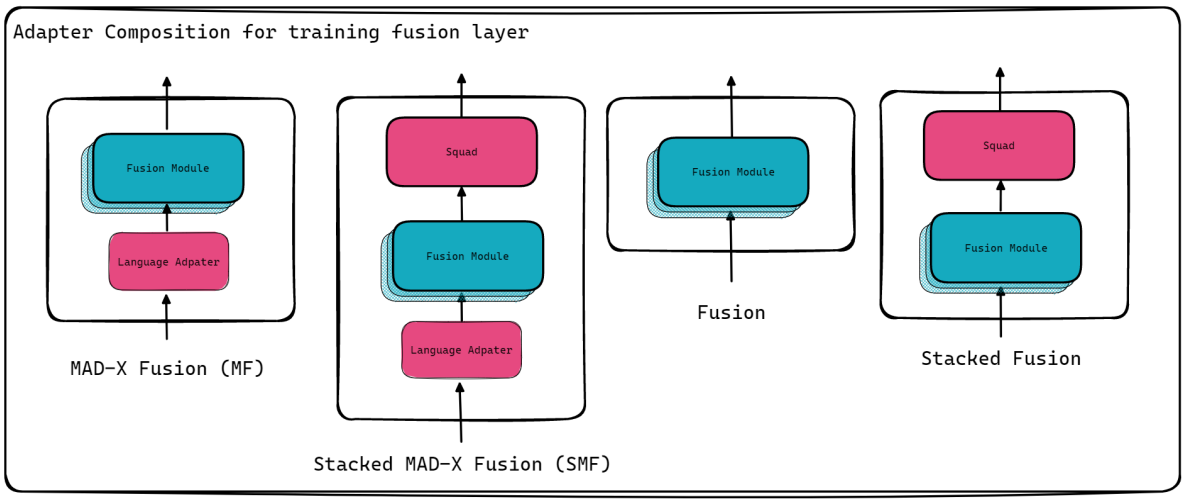

Figure 6.1: Fusion training compositions.

Task	Classification type
QQP	Binary Classification
MNLI	Multi-Class Classification
SocialIQA	Multi-Class Classification
NER	Multi-Class Classification
SQuAD	Span Classification

Table 6.3: Tasks and their classification types

classification types (e.g., label classification); however, in our composition, we have both label classification and span classification types.

Moreover, we sought to evaluate the language-specific representation of language-agnostic adapters in fusion; hence, for some compositions, we do not stack a language adapter under the fusion module. Taking these different compositions into account, we have developed four AdapterFusion training compositions.

6.2.4 MAD-X Fusion (MF)6.1

In this composition, we place the fusion layer on an English language adapter and train the AdapterFusion parameter ψ_m for N tasks, where QA $\in 1, ..., N$. We train and evaluate the model on the SQuAD dataset. In this composition, we fuse different classification types.

6.2.5 Stacked MAD-X Fusion (SMF)6.1

In this configuration, we separate the span classification task by stacking a SQuAD adapter on top and fusing only similar or identical classification types within our fusion layer. By stacking, we aim to capture knowledge specific to the

question-answering task while generalizing across the shared knowledge from the fusion layer. In this setting, we train parameter ψ_m on N tasks, where QA is not in $1, ..., N$. In this composition, we only fuse similar or the same classification types

6.2.6 Fusion and Stacked Fusion (SF)6.1

In this scenario, we experiment with language-agnostic adapters by training AdapterFusion without stacking on a language adapter. We expect this configuration to perform on par with the previously mentioned Fusion compositions, as it utilizes language-agnostic adapters, which help it generalize across languages.

Chapter 7

Results & Analysis

In this chapter, we delve into the hypotheses presented by our research. We evaluate the performance of baseline models against MAD-X Fusion and full-finetuned mBERT models in the context of cross-lingual transfer tasks. Moreover, we conduct an in-depth analysis to comprehend the performance of AdapterFusion models, specifically in relation to the question-answering task in low-resource scenarios.

To further enhance our understanding of the proposed approach, we compare the results generated by different learning algorithms, including MAD-X Fusion and full-finetuned mBERT, as well as assess the impact of few-shot transfer on the performance of these models.

Finally, we synthesize our findings from the experiments and present a comprehensive discussion of the results, highlighting the implications of our research on the broader field of natural language processing.

7.1 Hypothesis

This work aims to investigate a novel learning approach that combines MAD-X and AdapterFusion to further improve performance for the question-answering task in low-resource scenarios. Our underlying research investigates the effectiveness of language-agnostic adapters in improving cross-lingual transfer, specifically by using AdapterFusion. We present three main hypotheses guiding our research.

Hypothesis:

1. We hypothesize that fusing language-agnostic adapters will enhance cross-lingual transfer.

2. We hypothesize that few-shot transfer will yield better results due to the data-dependent performance of AdapterFusion.

3. We hypothesize that decreasing the amount of training data used for Fusion may lead to enhanced performance in zero-shot transfer.

F1 Results							
	Model	en	hi	bn	sw	is	Avg.
Baseline	**MAD-X**^{mBERT} ***-INV***	81.39	50.72	48.90	57.85	39.83	55.74
	MAD-X^{mBERT}	82.39	50.73	48.90	57.85	39.83	55.94
Full FT. *SQuAD*	**mBERT**	**89.12**	55.29	56.29	60.99	41.57	60.65
MAD-X Fusion Zero-Shot	**SMF**^{mBERT}	83.26	50.20	48.89	58.54	34.27	55.03
	SMF^{mBERT} *-INV*	83.26	50.20	48.89	58.54	34.27	55.03
	MF^{mBERT}	81.59	52.20	52.82	**59.72**	34.10	56.09
	MF^{mBERT} *-INV*	81.59	52.20	52.82	**59.72**	34.10	56.09
	SF^{mBERT}	82.16	53.05	52.20	58.42	**37.34**	56.63
	F^{mBERT}	82.60	**53.47**	**52.86**	58.04	34.4	56.87
MAD-X Fusion Few-Shot	**SMF**^{mBERT}	83.26	60.07	**60.41**	75.59	**56.17**	**67.10**
	MF^{mBERT}	81.59	60.13	59.40	**76.69**	55.77	66.72
	SF^{mBERT}	82.16	59.98	58.13	73.58	53.66	65.50
	F^{mBERT}	82.60	**60.36**	57.65	75.68	54.93	66.24

Table 7.1: F1 results for different model variants, including Baseline, Full Fine-Tuning (SQuAD), MAD-X Fusion Zero-Shot, and MAD-X Fusion Few-Shot. The table shows the performance of each model on various languages (en, hi, bn, sw, and is) and their average F1 scores.

7.2 Fine Tuning over MAD-X

We utilize an mBERT model fine-tuned on SQuAD from the HuggingFace repository. Our observations indicate that the performance is indeed lowest for low-resource languages in both baselines (mBERT and MAD-X), which validates the claim that these languages are underrepresented in large multilingual models. The MAD-X model demonstrates inferior performance across all languages compared to the fully fine-tuned model, exhibiting an average drop of 4.71 points across all languages and the most significant performance decline in English.

We anticipate that this level of performance is a result of only evaluating on low-resource languages that have been seen by the base model. In the case of MAD-X, we expect better performance when evaluated on languages not seen by the base model; under these conditions, MAD-X outperforms the fully fine-tuned mBERT. Furthermore, we assess MAD-X both with and without the invertible adapters, observing no significant difference in performance between these two models. These outcomes suggest that mBERT possesses a more robust language

representation than the representation achievable through language adapters for languages seen by the base model.

7.3 MAD-X Fusion

Mad-X fusion aims to improve performance on the target task by transferring task-specific knowledge from a set of N task adapters.

7.3.1 Zero-shot

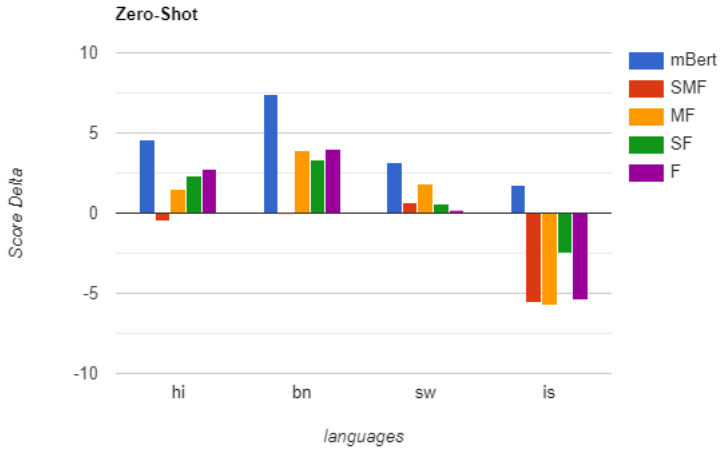

Figure 7.1: Relative performance difference of the different adapter compositions and the fully fine-tuned mBERT model compared to MAD-X (Baseline). MAD-X Fusion improves over most of the languages in zero-shot transfer, and FFT mBERT improves on all languages in the zero-shot setting.

MAD-X Fusion Against MAD-X

Our observations reveal that MAD-X Fusion enhances performance for most languages compared to MAD-X, with the exception of Icelandic, the lowest-resource language in our study. Fusion emerges as the best-performing adapter composition for zero-shot transfer to both Hindi and Bengali. However, this trend does not hold for lower-resource languages such as Swahili, where we notice negligible improvement across all compositions.

The lowest performance scores are found when transferring to Icelandic, indicating that MAD-X Fusion models might be unsuitable for zero-shot transfer in low-resource settings, especially when transferring from high-resource tasks. This limitation can be attributed to AdapterFusion's reliance on training size, as larger

training data tends to result in diminished performance gains. In our experiments, we train the fusion layer on 100,000 question-answering pairs.

Interestingly, the **SF** composition demonstrates the most stable performance during zero-shot transfer to Icelandic, experiencing a modest performance drop of only 7% compared to other compositions, which decline by over 14%. Overall, **F** attains the highest average performance across all languages, implying that, in a zero-shot setting, task sharing is more prevalent among higher-resource languages.

MAD-X Fusion Against Full FT. SQuAD

The fully fine-tuned mBERT model surpasses all the MAD-X Fusion compositions and the MAD-X baseline in terms of performance for all languages in zero-shot transfer. We notice a relatively consistent performance difference between MAD-X Fusion and the fully fine-tuned mBERT model across all languages. The most significant performance increase is observed in Bengali, while the least improvement is seen in Icelandic. These findings align with the results where MAD-X underperforms on high/mid-high resource languages already represented in the base model Pfeiffer et al. (2020c).

7.3.2 Few-shot

In this experiment, we extend our approach by training the fusion layer on language-specific Question Answering datasets, limiting the training sample size to 5,000, which constitutes 5% of the SQuAD training size. As illustrated in Table 7.2, most languages have less training data than the imposed cap.

Language	Train Size	Size Compared to SQuAD 88.7k
Hindi	4.9k	5.5%
Bengali	5.0k	5.6%
Swahili	2.7k	3.0%
Icelandic	4.6K	5.2%

Table 7.2: Few-shot train sample size for each selected language.

MAD-X Fusion Against MAD-X and Full FT. mBERT

The MAD-X Fusion models exhibit remarkable performance improvements in the few-shot transfer setting, particularly for low-resource languages. SMF^{mBERT} outperforms other compositions across most languages, achieving the highest F1 scores for English, Bengali, and Icelandic, and the best overall average F1 score

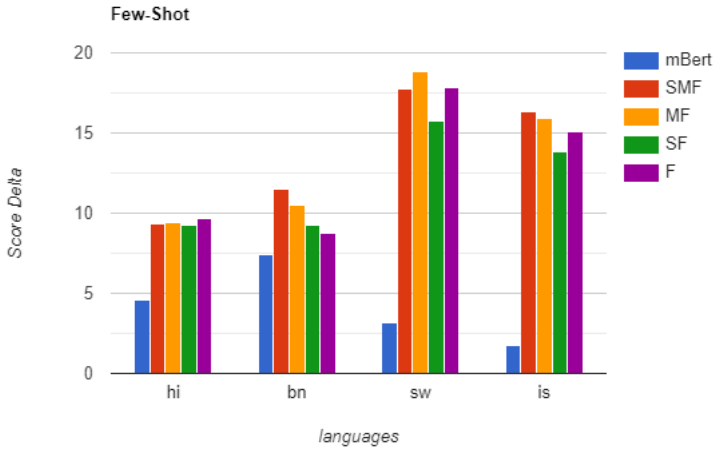

Figure 7.2: Relative performance differences between various adapter compositions and the fully fine-tuned mBERT model compared to MAD-X (Baseline). MAD-X Fusion demonstrates significant improvement in few-shot transfer for all languages.

of 67.10%. MF^{mBERT} attains the highest F1 score for Swahili, while SMF^{mBERT} also achieves the top result for Icelandic.

These results(7.2) showcase a remarkable 32% performance increase on Swahili compared to the baseline model. Notably, the improvements are more pronounced for lower-resource languages such as Swahili and Icelandic. This can be attributed to AdapterFusion's ability to promote knowledge sharing across lower-resource tasks. Interestingly, there is an inverse correlation between the size of the few-shot training sample and our models' performance, with Swahili, the lowest dataset size, boasting the highest performance increase.

The MAD-X Fusion (MF) composition consistently yields some of the highest results across all languages. In comparison, the Stacked MAD-X Fusion (SMF) performs the best on average. As illustrated in Table 7.1, in the few-shot setting, SMF achieves the highest average F1 score of 67.10%, outperforming other compositions. These results demonstrate the effectiveness of MAD-X Fusion models in few-shot transfer settings, particularly for low-resource languages, highlighting their potential for tackling multilingual tasks in real-world applications.

7.4 Hypothesis analysis

7.4.1 Hypothesis:

We hypothesize that fusing language-agnostic adapters will enhance cross-lingual transfer.

7.4.2 Validation

To test our hypothesis, we develop MAD-X Fusion and compare it with MAD-X in zero-shot settings. We observe improvements across the majority of languages, with the exception of Icelandic. This suggests that the fusion of language-agnostic adapters can indeed enhance cross-lingual transfer performance by promoting knowledge sharing across adapters that have robust task representations with abstracted language-specific information. Consequently, this validation demonstrates the potential of MAD-X Fusion in improving cross-lingual transfer for a diverse set of languages, further emphasizing the importance of incorporating language-agnostic adapters in the model design.

7.4.3 Hopthesis:

We hypothesize that few-shot transfer will yield better results due to the data-dependent performance of AdapterFusion, particularly for low-resource languages like Swahili and Icelandic.

Upon analyzing the activation of AdapterFusion during the zero-shot experiments, we observe limited activation from task adapters other than SQuAD.

7.4.4 Validation

Considering the data-dependent nature of AdapterFusion, we decide to train the fusion layer in a few-shot setting with significantly fewer QA data. In doing so, we observe substantial improvements in performance compared to both the baseline and full fine-tuning approaches. We assume this performance increase is directly related to the size of the training set, shown in Table 7.2, as well as the language representation in mBERT, where low-resource languages, such as Swahili and Icelandic, tend to perform better on MAD-X Fusion. Swahili, a low-resource language with the smallest few-shot sample size, demonstrates the largest performance increase among all languages, particularly when utilizing the MAD-X Fusion (MF) adapter composition. Similarly, Icelandic, another low-resource language, also benefits from few-shot transfer with AdapterFusion.

This validation supports our hypothesis that few-shot transfer with AdapterFusion

can lead to improved results, especially for low-resource languages like Swahili and Icelandic.

7.4.5 Hypothesis:

We hypothesize that reducing the amount of training data used for Fusion may lead to improved performance in zero-shot transfer.

7.4.6 Investigation

We train a new MF AdapterFusion composition on only 5K SQuAD data points and evaluate it on both English and Icelandic. In the case of single language evaluation, where we train MF Fusion on English and evaluate on English, we observe a performance decrease when compared to training on 88.7K SQuAD data points. In the case of zero-shot transfer, we observe a small performance increase when compared to training on 88.7K SQuAD, as shown in Table: 7.3.

Language	$MF^{english}$ F1 Score	F1 Score Δ (Baseline)
English	81.20	-1.19
Icelandic	34.67	-5.16

Table 7.3: F1 scores for the reduced SQuAD training data.

Despite observing a minor improvement in the zero-shot transfer setting, the overall performance remains significantly lower than desired. This outcome may suggest that the size of the training data does not contribute as much to the improvements seen in the few-shot transfer setting as initially hypothesized. Further investigation is necessary for a comprehensive understanding of this phenomenon. However, based on the current findings, we cannot confirm the hypothesis that reducing the amount of training data used for Fusion leads to enhanced performance in zero-shot transfer.

7.5 Summary

In summary, the results demonstrate the effectiveness of the MAD-X Fusion models, particularly in few-shot transfer settings for low-resource languages. The Stacked MAD-X Fusion (SMF) model performs the best on average, indicating its potential for tackling multilingual tasks in real-world applications.

Chapter 8

Conclusion and Future Works

In this study, we have presented the MAD-X Fusion framework, an innovative adapter-based approach for zero-shot and few-shot cross-lingual transfer, with a particular focus on improving performance in low-resource language scenarios. Through the integration of the MAD-X framework and AdapterFusion, we introduced distinct adapter compositions that facilitated knowledge sharing across languages and tasks, resulting in substantial performance improvements in both high- and low-resource contexts.

Our findings revealed the potential of the MAD-X Fusion framework to achieve enhanced performance in few-shot learning scenarios, even with limited training data. These results demonstrate the effectiveness of few-shot learning as a powerful approach for addressing data scarcity challenges in low-resource languages and promoting more inclusive NLP research. By fostering a more equitable NLP landscape, few-shot learning techniques contribute to language preservation efforts and cater to a more diverse user base.

Moreover, the success of the MAD-X Fusion framework in the question-answering task underlines the importance of continued research and development in this area. By exploring novel techniques and methodologies, the NLP research community can unlock new possibilities for efficient language model adaptation and further improve the performance of NLP models in low-resource settings.

In conclusion, this study has showcased the promise of adapter-based approaches, such as the MAD-X Fusion framework, for advancing NLP applications in low-resource languages. By harnessing the power of few-shot learning and promoting knowledge sharing across multiple languages and tasks, we contribute to ongoing efforts to overcome the challenges associated with large multilingual models and pave the way for a more inclusive and effective NLP research landscape.

Future works

8.1 Future Work

Several potential avenues for future research emerge from our findings. These include:

1. **Fusion of language adapters:** One possible direction is to explore the fusion of language adapters by training related task adapters. This approach may result in better adaptation to specific languages and tasks, further enhancing the model's performance.
2. **Investigating AdapterFusion with alternative attention mechanisms:** Another area of interest is to examine how AdapterFusion performs when combined with different attention mechanisms. This could provide insights into the compatibility and effectiveness of various attention mechanisms in improving cross-lingual transfer and generalization capabilities.
3. **Evaluating various AdapterFusion combinations for different target tasks:** Further research could focus on assessing the performance of different AdapterFusion compositions with respect to various target tasks. This may involve applying Fusion to multiple tasks across multiple languages and evaluating the fusion attention activations to identify the most relevant tasks for a given target task. The findings can then be used to create a framework or lookup table that maps a given target task to the most relevant list of task adapters based on task properties and language characteristics.
4. **Expanding the application of MAD-X Fusion:** It could be beneficial to investigate the effects of fusing more task adapters when applying MAD-X Fusion to evaluate other NLP tasks. This may provide a more comprehensive understanding of the relationship between the number of fused task adapters and performance improvements in cross-lingual and multilingual settings.

By pursuing these research directions, we aim to refine and extend our understanding of AdapterFusion and its potential in enhancing the performance of natural language processing models, particularly in low-resource language scenarios.

word count 9821

Bibliography

Adams, O., Makarucha, A., Neubig, G., Bird, S. and Cohn, T., 2017. Cross-lingual word embeddings for low-resource language modeling. *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 1, long papers*. pp.937–947.

Alyafeai, Z., AlShaibani, M.S. and Ahmad, I., 2020. A survey on transfer learning in natural language processing. *arxiv preprint arxiv:2007.04239*.

Argyriou, A., Evgeniou, T. and Pontil, M., 2006. Multi-task feature learning. *Advances in neural information processing systems*, 19.

Arnold, A., Nallapati, R. and Cohen, W.W., 2007. A comparative study of methods for transductive transfer learning. *Seventh ieee international conference on data mining workshops (icdmw 2007)*. IEEE, pp.77–82.

Artetxe, M., Ruder, S. and Yogatama, D., 2019. On the cross-lingual transferability of monolingual representations. *Corr*, abs/1910.11856. 1910.11856.

Bhattacharjee, A., Hasan, T., Samin, K., Islam, M.S., Rahman, M.S., Iqbal, A. and Shahriyar, R., 2021. Banglabert: Combating embedding barrier in multilingual models for low-resource language understanding. 2101.00204.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., 2020. Language models are few-shot learners. *Arxiv*, abs/2005.14165.

Clark, J.H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V. and Palomaki, J., 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the association for computational linguistics*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán,

F., Grave, E., Ott, M., Zettlemoyer, L. and Stoyanov, V., 2019. Unsupervised cross-lingual representation learning at scale. *arxiv preprint arxiv:1911.02116*.

CONNEAU, A. and Lample, G., 2019. Cross-lingual language model pretraining [Online]. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, eds. *Advances in neural information processing systems*. Curran Associates, Inc., vol. 32. Available from: `https://proceedings.neurips.cc/paper_files/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf`.

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H. and Stoyanov, V., 2018. Xnli: Evaluating cross-lingual sentence representations. *arxiv preprint arxiv:1809.05053*.

Dac Lai, V., Ngo, N.T., Pouran Ben Veyseh, A., Man, H., Dernoncourt, F., Bui, T. and Nguyen, T.H., 2023. ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning. *arxiv e-prints* [Online], arXiv:2304.05613. 2304.05613, Available from: `https://doi.org/10.48550/arXiv.2304.05613`.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Arxiv*, abs/1810.04805.

Dinh, L., Krueger, D. and Bengio, Y., 2014. Nice: Non-linear independent components estimation. *Corr*, abs/1410.8516.

Eisenschlos, J., Ruder, S., Czapla, P., Kadras, M., Gugger, S. and Howard, J., 2019. MultiFiT: Efficient multi-lingual language model fine-tuning [Online]. *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*. Hong Kong, China: Association for Computational Linguistics, pp.5702–5707. Available from: `https://doi.org/10.18653/v1/D19-1572`.

Feng, S.Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T. and Hovy, E.H., 2021. A survey of data augmentation approaches for nlp. *Arxiv*, abs/2105.03075.

Gao, J., Fan, W., Jiang, J. and Han, J., 2008. Knowledge transfer via multiple model local structure mapping. *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining*. pp.283–291.

Han, W., Pang, B. and Wu, Y.N., 2021. Robust transfer learning with pretrained language models through adapters. *Arxiv*, abs/2108.02340.

Hedderich, M.A., Lange, L., Adel, H., Strötgen, J. and Klakow, D., 2020. A survey on recent approaches for natural language processing in low-resource scenarios. *arxiv preprint arxiv:2010.12309*.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., Laroussilhe, Q. de, Gesmundo, A., Attariyan, M. and Gelly, S., 2019. Parameter-efficient transfer learning for nlp. *International conference on machine learning*.

Howard, J. and Ruder, S., 2018. Universal language model fine-tuning for text classification. *arxiv preprint arxiv:1801.06146*.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O. and Johnson, M., 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *Arxiv*, abs/2003.11080.

Jiang, J. and Zhai, C., 2007. Instance weighting for domain adaptation in nlp. *Findings*. ACL.

Ko, W.J., El-Kishky, A., Renduchintala, A., Chaudhary, V., Goyal, N., Guzmán, F., Fung, P., Koehn, P. and Diab, M.T., 2021. Adapting high-resource nmt models to translate low-resource related languages without parallel data. *Arxiv*, abs/2105.15071.

Lample, G. and Conneau, A., 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems (neurips)*.

Lee, J.Y., Dernoncourt, F. and Szolovits, P., 2017. Transfer learning for named-entity recognition with neural networks. *arxiv preprint arxiv:1705.06273*.

Lewis, P., Oguz, B., Rinott, R., Riedel, S. and Schwenk, H., 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arxiv preprint arxiv:1910.07475*, arXiv: 1910.07475.

Ling, W., Dyer, C., Black, A.W., Trancoso, I., Astudillo, R.F., Amir, S., Marujo, L. and Luís, T., 2015. Finding function in form: Compositional character models for open vocabulary word representation. *Conference on empirical methods in natural language processing*.

Liu, R., Xu, G., Jia, C., Ma, W., Wang, L. and Vosoughi, S., 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation [Online]. *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*. Online: Association for Computational

Linguistics, pp.9031–9041. Available from: `https://doi.org/10.18653/v1/2020.emnlp-main.726`.

Magueresse, A., Carles, V. and Heetderks, E., 2020. Low-resource languages: A review of past work and future challenges. *Arxiv*, abs/2006.07264.

McCloskey, M. and Cohen, N.J., 1989. Catastrophic interference in connectionist networks: The sequential learning problem [Online]. In: G.H. Bower, ed. *Psychology of learning and motivation*. Academic Press, *Psychology of Learning and Motivation*, vol. 24, pp.109–165. Available from: `https://doi.org/https://doi.org/10.1016/S0079-7421(08)60536-8`.

Mihalkova, L., Huynh, T. and Mooney, R.J., 2007. Mapping and revising markov logic networks for transfer learning. *Aaai*. vol. 7, pp.608–614.

Mikolov, T., Chen, K., Corrado, G.S. and Dean, J., 2013. Efficient estimation of word representations in vector space. *International conference on learning representations*.

Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L. and Jin, Z., 2016. How transferable are neural networks in nlp applications? *arxiv preprint arxiv:1603.06111*.

Nguyen, T.Q. and Chiang, D., 2017. Transfer learning across low-resource, related languages for neural machine translation [Online]. *Proceedings of the eighth international joint conference on natural language processing (volume 2: Short papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp.296–301. Available from: `https://aclanthology.org/I17-2050`.

Nozza, D., Bianchi, F. and Hovy, D., 2020. What the [mask]? making sense of language-specific bert models. *Arxiv*, abs/2003.02912.

Ogueji, K., Ahia, O., Onilude, G., Gehrmann, S., Hooker, S. and Kreutzer, J., 2022. Intriguing properties of compression on multilingual models. *Conference on empirical methods in natural language processing*.

Pagliardini, M., Gupta, P. and Jaggi, M., 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *North american chapter of the association for computational linguistics*.

Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. *Ieee transactions on knowledge and data engineering*, 22(10), pp.1345–1359.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K. and Gurevych, I., 2020a.

Adapterfusion: Non-destructive task composition for transfer learning. *Arxiv*, abs/2005.00247.

Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulic, I., Ruder, S., Cho, K. and Gurevych, I., 2020b. Adapterhub: A framework for adapting transformers. *Conference on empirical methods in natural language processing*.

Pfeiffer, J., Vulic, I., Gurevych, I. and Ruder, S., 2020c. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *Conference on empirical methods in natural language processing*.

Pires, T., Schlinger, E. and Garrette, D., 2019. How multilingual is multilingual BERT? [Online]. *Proceedings of the 57th annual meeting of the association for computational linguistics*. Florence, Italy: Association for Computational Linguistics, pp.4996–5001. Available from: `https://doi.org/10.18653/v1/P19-1493`.

Ponti, E.M., s, G.G., Majewska, O., Liu, Q., Vuli'c, I. and Korhonen, A., 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. *arxiv preprint* [Online]. Available from: `https://ducdauge.github.io/files/xcopa.pdf`.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. et al., 2019. Language models are unsupervised multitask learners. *Openai blog*, 1(8), p.9.

Raganato, A., Pasini, T., Camacho-Collados, J. and Pilehvar, M.T., 2020. Xl-wic: A multilingual benchmark for evaluating semantic contextualization. *arxiv preprint arxiv:2010.06478*.

Rahimi, A., Li, Y. and Cohn, T., 2019. Massively multilingual transfer for NER [Online]. *Proceedings of the 57th annual meeting of the association for computational linguistics*. Florence, Italy: Association for Computational Linguistics, pp.151–164. Available from: `https://www.aclweb.org/anthology/P19-1015`.

Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arxiv e-prints*, arXiv:1606.05250. `1606.05250`.

Ruder, S., 2019. The 4 biggest open problems in nlp. Https://www.ruder.io/4-biggest-open-problems-in-nlp/. [Accessed 2023-01-26].

Ruder, S., Constant, N., Botha, J., Siddhant, A., Firat, O., Fu, J., Liu, P., Hu, J., Garrette, D., Neubig, G. and Johnson, M., 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation [Online]. *Proceedings*

of the 2021 conference on empirical methods in natural language processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp.10215–10245. Available from: `https://doi.org/10.18653/v1/2021.emnlp-main.802`.

Ruder, S., Peters, M.E., Swayamdipta, S. and Wolf, T., 2019. Transfer learning in natural language processing. *North american chapter of the association for computational linguistics*.

Sap, M., Rashkin, H., Chen, D., Bras, R.L. and Choi, Y., 2019. Social iqa: Commonsense reasoning about social interactions. *Conference on empirical methods in natural language processing*.

Snæbjarnarson, V., Einarsson, B.T.T., Auðunardóttir, I.I., Sæmundsson, U.I., Bjarnadóttir, H., Gunnarsson, H.V. and Einarsson, H., 2021. NQiI - natural questions in icelandic - v1.0 [Online]. CLARIN-IS. Available from: `http://hdl.handle.net/20.500.12537/143`.

Stickland, A.C. and Murray, I., 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. *International conference on machine learning*.

Taylor, M.E. and Stone, P., 2009. Transfer learning for reinforcement learning domains: A survey. *J. mach. learn. res.*, 10, pp.1633–1685.

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., 2017. Attention is all you need. *Arxiv*, abs/1706.03762.

Vu, T., Barua, A., Lester, B., Cer, D.M., Iyyer, M. and Constant, N., 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. *Conference on empirical methods in natural language processing*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S.R., 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Arxiv*, abs/1804.07461.

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Ji, J., Cao, G., Jiang, D. and Zhou, M., 2020a. K-adapter: Infusing knowledge into pre-trained models with adapters. *Findings*.

Wang, X., Tsvetkov, Y., Ruder, S. and Neubig, G., 2021. Efficient test time adapter ensembling for low-resource language varieties. *Conference on empirical methods in natural language processing*.

Wang, Z., K, K., Mayhew, S. and Roth, D., 2020b. Extending multilingual BERT to low-resource languages [Online]. *Findings of the association for computational linguistics: Emnlp 2020*. Online: Association for Computational Linguistics, pp.2649–2656. Available from: `https://doi.org/10.18653/v1/2020.findings-emnlp.240`.

Williams, A., Nangia, N. and Bowman, S.R., 2017. A broad-coverage challenge corpus for sentence understanding through inference. *North american chapter of the association for computational linguistics*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Brew, J., 2019. Huggingface's transformers: State-of-the-art natural language processing. *Arxiv*, abs/1910.03771.

Zhang, X., Zhao, J.J. and LeCun, Y., 2015. Character-level convolutional networks for text classification. *Arxiv*, abs/1509.01626.

Zoph, B., Yuret, D., May, J. and Knight, K., 2016. Transfer learning for low-resource neural machine translation. *Conference on empirical methods in natural language processing*.

Appendix A

Model Diagrams

Model	# Parameter Trained
mBERT	176M
MAD-X Fusion^{mBert}	21M
MAD-X^{mBert}	8M

Table A.1: Number of parameters trained per model.

Figure A.1: MAD-X Framework

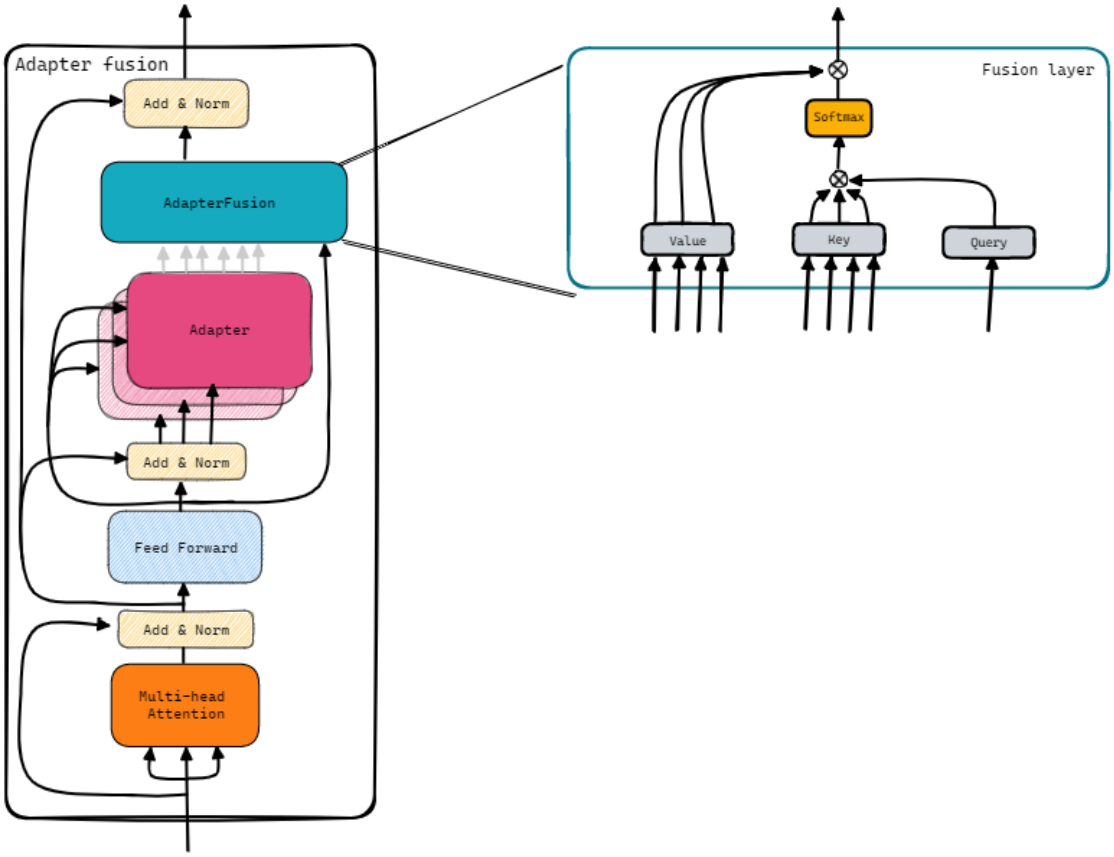

Figure A.2: AdapterFusion Method

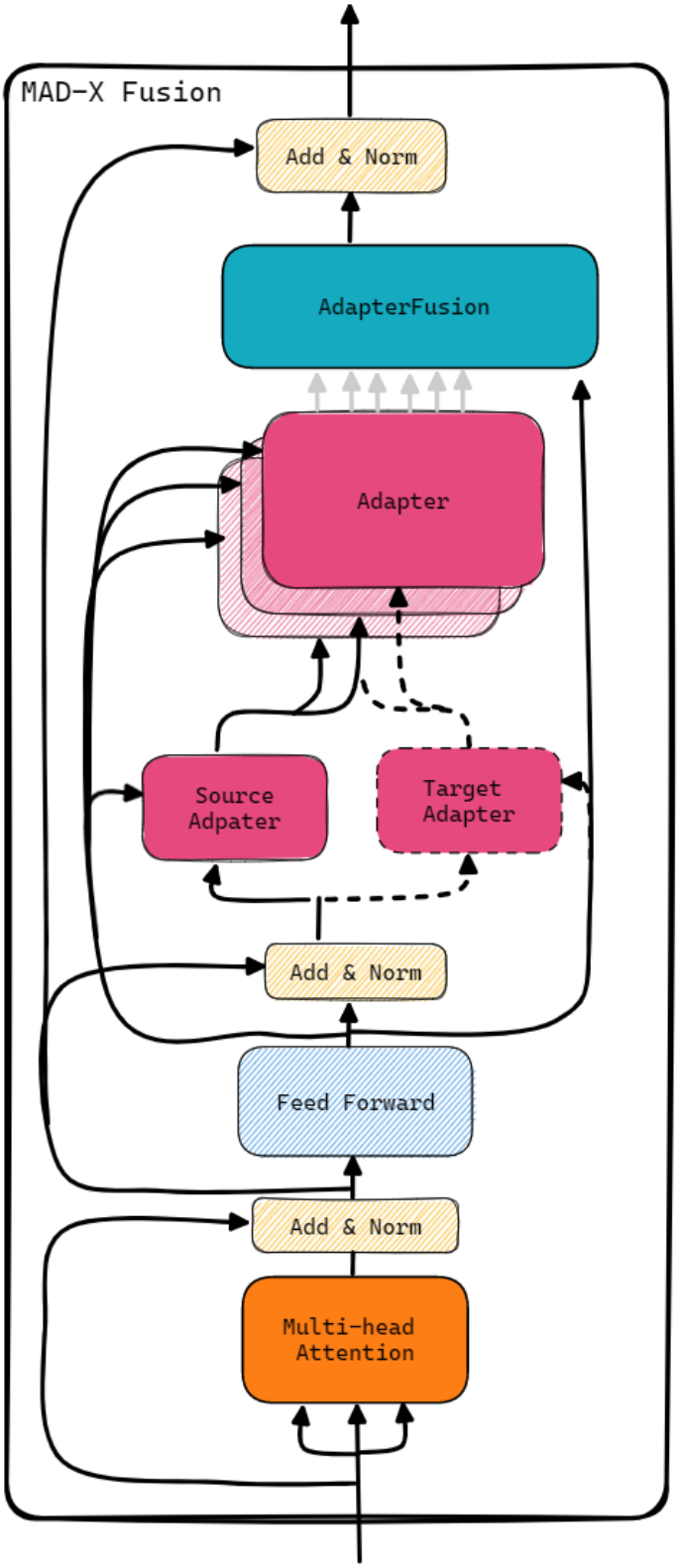

Figure A.3: MAD-X Fusion

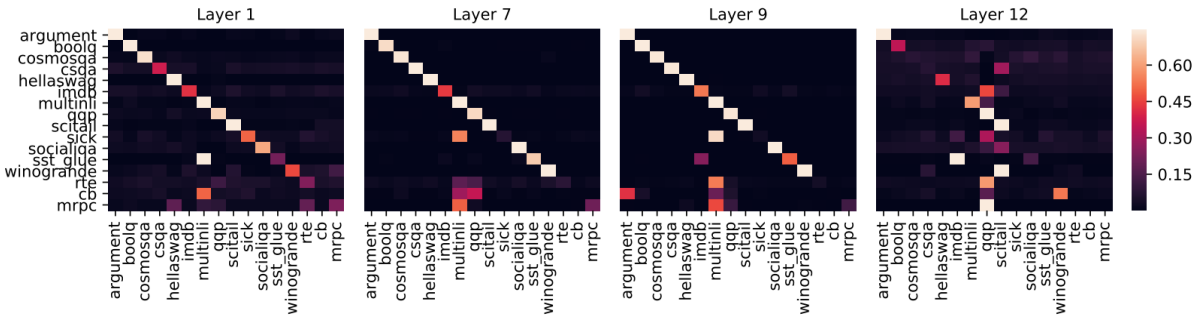

Figure A.4: The activations of AdapterFusion for pretrained ST-Adapters are shown in a matrix where rows represent the target task m and columns represent adapters n. A high softmax activation for $\Phi_{n,l}$ indicates that the information from adapter n is useful for task m. For our analysis, we calculate the average softmax activation for each adapter $\Phi_{n,l}$, where $n \in \{1, \ldots, N\}$, over all instances in the development set within the same layer l. Pfeiffer et al. (2020a)