electronics

MDPI

Article

A Benchmark Evaluation of Multilingual Large Language Models for Arabic Cross-Lingual Named-Entity Recognition

Mashael Al-Duwais *, Hend Al-Khalifa and Abdulmalik Al-Salman

College of Computer and Information Sciences, King Saud University, P.O. Box 2614, Riyadh 13312, Saudi Arabia; hendk@ksu.edu.sa (H.A.-K.); salman@ksu.edu.sa (A.A.-S.)
* Correspondence: malduwais@ksu.edu.sa

Abstract: Multilingual large language models (MLLMs) have demonstrated remarkable performance across a wide range of cross-lingual Natural Language Processing (NLP) tasks. The emergence of MLLMs made it possible to achieve knowledge transfer from high-resource to low-resource languages. Several MLLMs have been released for cross-lingual transfer tasks. However, no systematic evaluation comparing all models for Arabic cross-lingual Named-Entity Recognition (NER) is available. This paper presents a benchmark evaluation to empirically investigate the performance of the state-of-the-art multilingual large language models for Arabic cross-lingual NER. Furthermore, we investigated the performance of different MLLMs adaptation methods to better model the Arabic language. An error analysis of the different adaptation methods is presented. Our experimental results indicate that GigaBERT outperforms other models for Arabic cross-lingual NER, while language-adaptive pre-training (LAPT) proves to be the most effective adaptation method across all datasets. Our findings highlight the importance of incorporating language-specific knowledge to enhance the performance in distant language pairs like English and Arabic.

Keywords: Arabic NER; cross-lingual transfer learning; language adaptation; multilingual large language model; named-entity recognition

check for updates

Citation: Al-Duwais, M.; Al-Khalifa, H.; Al-Salman, A. A Benchmark Evaluation of Multilingual Large Language Models for Arabic Cross-Lingual Named-Entity Recognition. *Electronics* **2024**, *13*, 3574. https://doi.org/10.3390/electronics13173574

Academic Editors: Long Yang and Noura Al Moubayed

Received: 18 August 2024
Revised: 2 September 2024
Accepted: 6 September 2024
Published: 9 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Named-Entity Recognition (NER) is the task of identifying and classifying named entities into predefined categories. NER is a key component in text processing and information extraction. NER-supervised models require sufficient annotated data for downstream tasks. However, manually annotating a large quantity of data for every language can be extremely costly. A viable solution to the annotated data scarcity problem is cross-lingual transfer.

Cross-lingual transfer is a form of transfer learning that is concerned with maximizing the performance of the target language. Cross-lingual transfer allows leveraging annotated data from resource-rich languages to improve performance on low-resource languages to overcome the lack of labeled data in the target languages. Cross-lingual transfer learning is like domain adaptation; the domains, in this case, are different languages.

Multilingual large language models (MLLMs) have recently become a default paradigm for cross-lingual transfer of natural language processing models. By training language models on multiple languages, the models can learn general features that are applicable across languages. This helps in improving the model's ability to generalize to new, unseen, or limited languages. MLLMs fine-tuned in one source language can be directly used for inference on a different target language for specific tasks, without requiring any labeled data in that target language (i.e., zero-shot approach). Overall, utilizing cross-lingual transfer can address labeled data scarcity, enhance model generalization, optimize resource utilization, and improve the performance across different languages.

Recently, several MLLMs have been released, like mBERT [1], XLM-RoBERTa [2], and mT5 [3]. Existing works on cross-lingual NER have mainly focused on similar languages,

e.g., transferring from English to Latin languages: German or Spanish [4,5]. However, there is a lack of comprehensive study for Arabic cross-lingual NER that compares the performances of recent state-of-the-art multilingual models on the different datasets to understand their limitations.

Arabic is a rich morphological language with different morphology, syntax, and structure from English. Arabic usually uses a Verb–Subject–Object (VSO) word order, while English uses the Subject–Verb–Object (SVO) word order. Moreover, there is no lexical overlap between English and Arabic, which makes cross-lingual NLP tasks challenging. Transferring between distant languages like English and Arabic poses specific challenges. Piers et al. [6] studied the cross-lingual performance of multilingual BERT (mBERT) for 16 languages, including Arabic, on different tasks and showed that transfer works best for typologically similar languages. The multilingual representation of mBERT failed to learn systematic transformations to accommodate a target language with different word order.

In this study, we thereby examine the following research questions:

- RQ-1: What are the performances of the existing state-of-the-art multilingual language models on Arabic cross-lingual NER? Which model performs better and on which dataset?
- RQ-2: How can we optimally adapt multilingual models to improve the cross-lingual zero-shot performance? And which adaptation methods work best for Arabic NER?

To answer the above questions, we perform the following main contributions:

- Present a systematic benchmark evaluation of the existing state-of-the-art multilingual large language models (MLLMs) to evaluate their effectiveness in cross-lingual NER transfer for the Arabic language.
- Compare the performance of language adaptation methods for Arabic, including language-adaptive pre-training (LAPT) [7], incorporating parallel data into the pre-training [8] and injecting adapters instead of full fine-tuning [9,10].
- Assess the role of language-specific preprocessing by injecting a layer of Arabic-specific morphological tokenizer [11] into the multilingual language models and analyze its influence on the cross-lingual transfer performance.
- Perform a detailed error analysis across different models to identify and understand the specific challenges and limitations of each approach.

In this study, we choose English as the source language, as it is the highest resource language available for most NLP tasks and has been used as a source of transfer in popular multilingual benchmarks like XTREME [12] and XGLUE [13]. We conducted our study on different NER datasets: the CoNLL2003 English dataset [14], the ANERCorp Arabic dataset [15], the CLEANANERCorp Arabic dataset [16], and the multilingual NER dataset WikiANN [17,18]. To the best of our knowledge, we are the first to conduct in-depth experiments and analyses of multilingual language models for English–Arabic cross-lingual NER task transfer.

The rest of this paper is organized as follows: Section 2 presents background information about the challenges of Arabic NER, the concept of cross-lingual transfer, multilingual language models and their language adaptation methods, the NER datasets, and highlights the related works of cross-lingual NER. Section 3 elaborates on experimental setups, including the methodology used in the research. Section 4 analyzes the experimental results. Section 5 presents the error analysis of the results. Finally, the conclusion and future work are drawn in Section 6.

2. Background

This section introduces background information about the challenges of Arabic NER, the concept of cross-lingual transfer learning, the multilingual language models, language models adaptation methods, NER datasets, and the related work.

2.1. Challenges of Arabic NER

Arabic belongs to the Afro-Asiatic language family and is the most widely spoken Semitic language, with around 400 million native speakers [19]. Arabic imposes many challenges for NLP tasks in general and for NER in particular [20,21]. These challenges include the following:

1. Absence of capital letters: The English language has a special orthogonal feature for named entities, which is capital letters distinguishing them from other words. The lack of capitalization in Arabic increases the ambiguity both in the recognition and categorization of the named entities. For example, the word ("Apple") in English may refer to the fruit name or the company name. This ambiguity could be resolved by the existence of a capital letter. However, in Arabic, the word ("أحمد") may refer to a person's name or a verb, but there are no orthogonal differences between them. Therefore, analyzing the context of the sentence helps distinguish the named entity from other words.
2. Morphological complexity: Arabic is a highly inflectional language with rich morphology. Named entities can have various prefixes, suffixes, and infixes that change their forms. For example, the Arabic named entity ("البريطانيين") ("British") is attached with ("ال") (definite article) and ("ين") plural suffix to form a noun phrase ("the British"). In order to recognize and categorize such entities, a language-specific morphological tokenizer should be utilized to remove the suffix and prefix, which are anticipated highly in Arabic NER.
3. Named-entity ambiguity: Some names can also be common nouns, verbs, or adjectives, leading to ambiguities. For example, ("حمد") ("Hamad") is a person's name but also comes as a verb. Arabic names are sometimes indistinguishable from verbs, which might cause ambiguity. Likewise, Arabic names that are derived from adjectives are usually ambiguous, which causes a significant challenge for some Arabic NLP applications. For example, the name ("كريم"), which means ("generous"), can be a person's name or an adjective.
4. Lack of standardized spelling: There is often no consistent way to spell names, particularly when transliterating from other languages. This inconsistency causes difficulties in identifying and normalizing named entities. For example, the transliterating of ("Google") into Arabic produces variant forms such as "قوقل", "جوجل", or "غوغل". The reason behind this variation is that Arabic has a greater number of pronunciation sounds compared to English, which might potentially result in a higher number of variations in named entities, leading to ambiguity or errors.
5. Absence of diacritics: The Arabic language uses diacritics or short vowels to encode phonetic information. Diacritics are crucial in resolving ambiguity in named entities, as they help to distinguish between words of similar spelling but different meanings. However, modern Arabic text is un-diacritized, which causes ambiguity in NER systems. For example, the word ("هيّا") is a verb that means ("Let's go"), while the same spelling but without diacritics could refer to a person's name ("هيا"), in English ("Haya").

2.2. Cross-Lingual Transfer Learning

Supervised learning requires a sufficient number of labeled data for every new setting, whether for a new task, domain, or language. In contrast, transfer learning (TL) is a machine-learning approach where a model developed for a particular task, domain, or language is reused as the starting point for a model on another task, domain, or language. This approach leverages the knowledge gained from the first task to improve learning efficiency and performance on the second task. Those can significantly reduce training time and improve model performance, making it a powerful tool in the field of machine learning. Transfer learning has been categorized into two subcategories [22], depending

on whether the source and the target tasks are equal or not: Transudative and Inductive transfer learning. Transudative is transferring between the same tasks, while inductive is transferring between different tasks. If two tasks differ and they are learned in sequential order, it is called Sequential Learning. If the two tasks differ and are learned simultaneously, it is called Multi-task Learning. Conversely, if two tasks are the same and have different domains, this is called Domain Adaptation, and if they have different languages, this is called Cross-lingual transfer learning. Figure 1 shows the taxonomy of transfer learning.

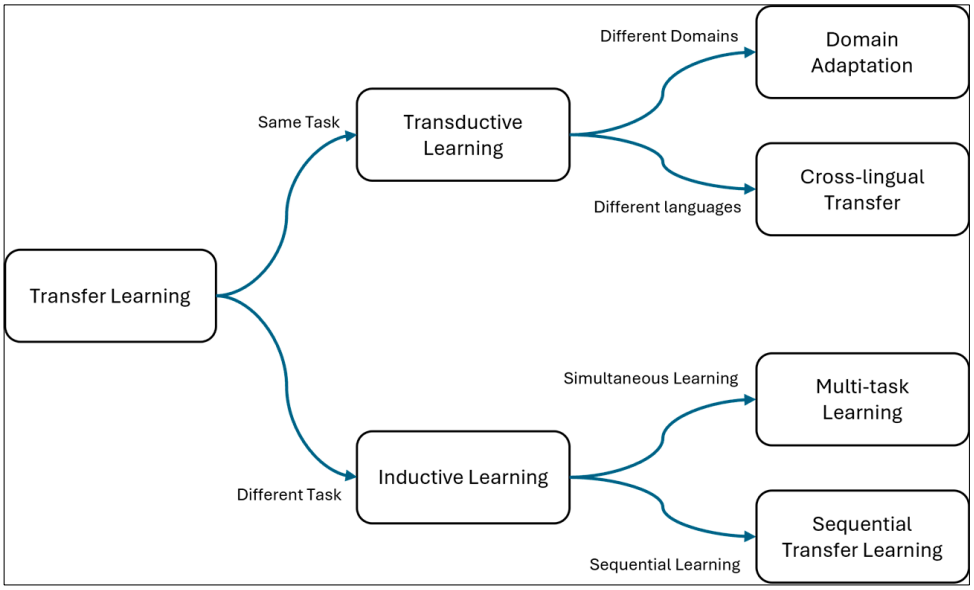


Figure 1. A taxonomy for transfer learning for NLP. Modified from [20].

Recently, with the emergence of multilingual pre-trained language models [1–3], fine-tuning multilingual language models have become the default paradigm for cross-lingual transfer. In cross-lingual transfer, we fine-tune a pre-trained multilingual language model for a downstream task using labeled data in language A for a specific task. Then, we test the fine-tuned model on language B to evaluate the cross-lingual transfer performance without seeing any labeled data from the target language B. This is known as zero-shot cross-lingual transfer. This enables the transfer of NLP tasks from high-resource languages to low-resource ones. MLLMs have demonstrated their ability for zero-shot cross-lingual transfer in previous works [6,23,24].

2.3. Multilingual Language Models

Pre-trained multilingual language models have demonstrated significant success across a wide range of cross-lingual NLP tasks in various languages. However, selecting the most suitable model for a specific language and task is challenging due to differences in model design, training objectives, training data size, tokenization, and vocabulary size. Despite fine-tuning being the common approach to choosing the best-performing pre-trained model for a specific task, we have selected the most widely used MLLMs (As of August 2024, XLM-R base and large has 17.5 and 10.5 million downloads, respectively, per month on the HuggingFace Hub; mBERT has 3.3 million downloads; and GigaBERT has 114 downloads, which is the highest English–Arabic bilingual model downloaded.) that cover English and Arabic for our investigation:

- mBERT (Multilingual BERT) [1] is a multilingual version of BERT trained on Wikipedia in 104 languages, with a masked language model and next-sentence prediction objectives.
- XLM-R (Cross-lingual Language Model-RoBERTa) [2] is a RoBERTa [25] version of XLM [26], which functions without using a translation language model in pre-training. It is trained on 100 languages, using a multilingual corpus from

Common Craw, and has become the new state of the art on cross-lingual tasks. It utilizes the SentencePiece tokenizer, which allows it to handle languages without clear whitespace delimiters effectively.

- XLM-V [27] follows the same approach as XLM-R, with an increased vocabulary capacity, a 1 M token vocabulary.
- GigaBERT [28] is a bilingual English–Arabic BERT model that outperforms other multilingual models in zero-shot transfer from English to Arabic. GigaBERT is trained on the Arabic and English Gigaword corpora, Arabic and English Wikipedia, and the OSCAR corpus.

Table 1 presents a high-level comparison of the multilingual language models.

Table 1. Multilingual pre-trained language models. The number of tokens and vocabulary for each language are taken from [27].

Model	#Param	Dataset	#Lang	#Tokens (all/en/ar)	#Lang	Vocabulary (all/en/ar)
mBERT	110 M	Wikipedia	104	21.9 B/2.5 B/153 M	WordPiece	110 k/53 k/5 k
XLM-R bases	270 M	CommonCrawl	100	295 B/55.6 B/2.9 B	SentencePiece	250 k/80 k/14 k
XLM-R large	550 M	CommonCrawl	100	295 B/55.6 B/2.9 B	SentencePiece	250 k/80 k/14 k
XLM-V	270 M	CommonCrawl	100	295 B/55.6 B/2.9 B	SentencePiece	1 M/280 k/174 k *
GigaBERT-v4	125 M	Gigaword + Wikipedia + Oscar	2	10.4 B/6.1 B/4.3 B	WordPiece	50 k/21 k/26 k

* per language vocabulary in XLM-V refers to a single cluster, not a single language.

2.4. Language Models Adaptation Methods

To specialize a pre-train language model to a given target language, different methods have been widely used:

2.4.1. Language-Adaptive Pre-Training (LAPT)

Similar to domain adaptation, the general-purpose pre-trained language model is adapted to a specialized domain [29,30]. LAPT, or target-language adaptation, is an approach to specialize a multilingual pre-trained language model for a target language. LAPT is conducted by fine-tuning a pre-trained multilingual language model via the masked language model (MLM) on unlabeled data of the target language. This helps the model to adjust its weights to better understand the context and suit the language-specific features. The approach proved to lead to improvements on downstream tasks on that target language [7,31–33]. Figure 2 shows the process of language-adaptive pre-training.

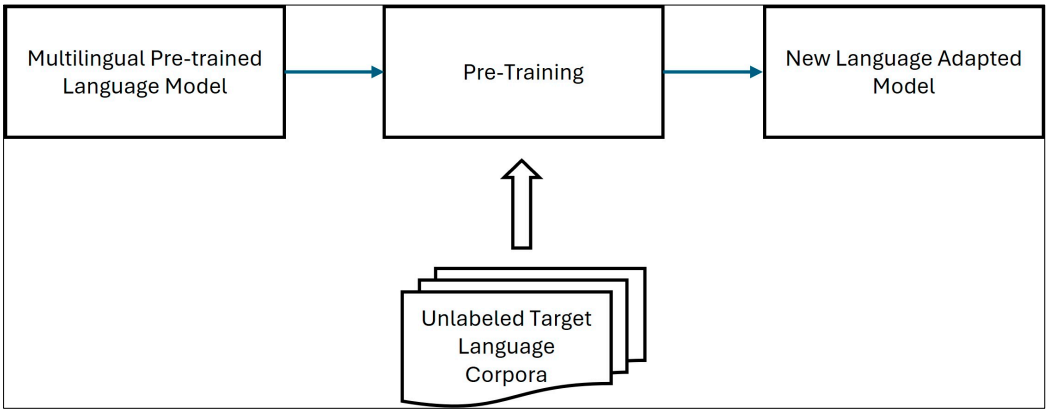


Figure 2. Language-adaptive pre-training (LAPT).

2.4.2. Adapters

Fine-tuning large pre-trained models is an effective transfer-learning approach, but it still is parameter-inefficient. Full fine-tuning involves updating all the parameters of the pre-trained language model, which is time-consuming and computationally expensive for large language models. To address this challenge, Parameter-Efficient Fine-Tuning (PEFT) has emerged as an efficient approach to decrease the computational cost of full fine-tuning. The adapter [9,34,35] is one of the PEFT methods used to specialize a pre-trained encoder in new domains or languages. Adapters are compact and extensible modules injected at every layer of the transformer. These modules are trainable parameters per task or language, while the original model parameters are frozen.

The bottleneck adapters proposed by [9,10] introduced bottleneck feed-forward layers in each layer of a transformer model. The bottleneck adapter consists of a down-projection $D \in R^{d \times h}$, where h is the hidden size of the transformer model, and d is the dimension of the adapter, followed by a *ReLU* activation and an up-projection $U \in R^{d \times h}$ at every layer l:

$$A(h_l, r_l) = U_l(ReLU(D(h_l))) + r_l \quad (1)$$

where h_l and r_l are the transformer-hidden state and the residual at layer l, respectively. The residual connection r_l is the output of the transformer's feed-forward layer, whereas h_l is the output of the subsequent layer normalization.

Adapters have different configurations depending on the location of the adapter in the transformer block. Ref. [9] places adapter layers after both the multi-head attention and feed-forward block, while [10] places an adapter layer only after the feed-forward block in each transformer layer. Ref. [36] introduces parallel adapters, which place adapter layers parallel to the original transformer layers.

The MAD-X [10] proposes language adapters to learn language-specific transformations. Language adapters are bottleneck adapters with an extra invertible adapter layer after the language-model-embedding layer. Language adapters are trained for each language, including English, using masked language modeling (MLM) objectives on monolingual corpora. Then, a language adapter can be stacked before a task adapter for training on a downstream task. For zero-shot cross-lingual transfer, the English adapter is switched with the target language adapter, and the model is subsequently evaluated. MAD-X outperforms the state-of-the-art models in cross-lingual transfer across a representative set of typologically diverse languages on named-entity recognition.

2.4.3. Language-Specific Preprocessing

Languages have varying linguistic properties, morphologies, and syntaxes and require special handling for tokenization, stemming, and normalization due to those differences. Sub-word tokenization methods commonly used in pre-trained language models are sub-optimal at handling morphologically rich languages and can produce poor segmentations for languages with a rich morphology. Adding a language-specific preprocessing layer that is morphology-aware can make the tokenization algorithm more robust to deal with inaccurate segmentations. Monolingual language models [37,38] use a language-specific tokenizer to better represent the word and improve the performance of transformer models. However, multilingual pre-trained language models like mBERT and XLM-R use the same tokenizer for all languages during pre-training and do not consider language-specific features that harm the cross-lingual transfer performance [39].

The Arabic language encodes a large amount of information in its morphology and word structure due to various attachable clitics. Previous works [40,41] have defined four classes of clitics: conjunction proclitic (cnj+), particle proclitic (prt+), definite article (art+), and pronominal enclitic (+pro). The sentence below shows the cliticization in a strict order to a word base:

$$[\text{cnj+ [prt+ [art+ Word_Base +pro]]}] \quad (2)$$

Table 2 shows the different classes of clitics.

Table 2. Types of clitics attached to Arabic words with examples.

Type	Category	Example	Transliteration	Meaning
art+	Definite article	ال+	al+	the
cnj+	Conjunction proclitic	و+	w+	and
		ف+	f+	then
prt+	Particle proclitic	ل+	l+	to/for
		ب+	b+	by/with
		ك+	k+	as
+pro	Pronominal enclitics	هم+	+hm	their/them (male)
		ها+	+ha	her
		هما+	+hma	their/them (for two)
		هن+	+hn	their/them (female)

Arabic morphological tokenizers like MADAMIRA [42], Farasa [43], and CAMeL Tools [11] use different tokenization schemas to segment the words into the base form. Several orthographic and morphological adjustment rules are applied to the word to change it back to the base form according to each scheme. Table 3 shows some of the existing tokenization schemas [41] from the literature, ranging from coarse to fine schemas. The sub-word tokenization schemes break the words into smaller units. They can significantly reduce the vocabulary size, as it includes common sub-word units rather than all possible word forms. This helps the model generalize better, handle unseen named entities more effectively, and reduce the Out-of-Vocabulary (OOV) word rate.

Table 3. Arabic tokenization schemes.

Schema	Formula	Definition
D0	word	No Tokenization
D1	cnj + word	Separates the conjunction proclitic
D2	cnj + prt + word	D1 + Separates prepositional clitics and particles
ATB	cnj + prt + word + pro	Separates all clitics except the definite article
D3	cnj + prt + art + word + pro	Separates all clitics, including the definite article and the pronominal enclitics

The CAMeL tool's morphological tokenizer implemented three tokenization schemes discussed in the literature: the Penn Arabic Treebank ATB [44], D3 [45], and Buckwalter tag tokenization BW [46]. Table 4 presents an example of a sentence processed using CAMeL Tool's morphological tokenizer. We will investigate the performance of incorporating all the three schemes of CAMeL morphological tokenizers: ATB, D3, and WB, to assess their effect on the cross-lingual ability of multilingual language models for the NER task.

Table 4. A sentence tokenized using CAMeL tool's morphological tokenizer.

Schema	Example
D0	['عدد' , 'السكان' , 'البريطانيين' , 'بلندن']
ATB	['عدد' , 'السكان' , 'البريطانيين' , 'ب+_لندن']
D3	['عدد' , 'ال+_سكان' , 'ال+_بريطانيين' , 'ب+_لندن']
WB	['عدد' , 'ال+_سكان' , 'ال+_بريطاني_+ين' , 'ب+_لندن']

2.5. NER Datasets

This section briefly presents the NER datasets used to fine-tune multilingual language models and evaluate the models considered in this benchmark:

- CoNLL-2002/2003 [14,47] is a named-entity recognition dataset released as a part of the CoNLL-2003 shared task. It covers four languages (English, German, Dutch, Spanish) from the news domain annotated with four entity tags: LOC (location), PER (person), and ORG (organization), MISC (miscellaneous). The corpus consists of Reuters news stories from August 1996 to August 1997. MUC-6 annotation scheme and IOB tagging are followed. Words tagged with nine classes: B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC, and O, which denotes other words that are not named entities.
- ANERCorp [15] is one of the earliest and most frequently used Arabic NER datasets. It contains 150,286 tokens and 32,114 named entities from 316 articles that were selected from different newspapers. Following CoNLL2003, the corpus was annotated with four types of named entities: Person, Location, Organization, and Miscellaneous.
- CLEANANERCorp [16] is an enhanced version of ANERCorp where 6.3% of tagging errors have been corrected and cleaned.
- WikiANN [17] is a multilingual named-entity recognition dataset that is automatically annotated and covers 282 languages. It consists of Wikipedia articles that have been annotated with three tags: LOC (location), PER (person), and ORG (organization) tags in the IOB2 format. The dataset is used by XTREME [12] benchmark for cross-lingual NER evaluation. We used the version in [18], which has a balanced train, development, and test splits and supports 176 of the 282 languages from the original WikiANN corpus.

We considered the CoNLL2003 English NER dataset as the source language and ANERCorp Arabic news NER dataset as the target language. We also evaluated the models with the new corrected version of CLEANANERCorp to obtain a more accurate evaluation of the models. In order to make a fair comparison with the current multilingual benchmarks, we experimented with the multilingual NER dataset WikiANN. Table 5 presents a general overview of the studied NER datasets.

Table 5. NER Datasets.

Dataset	Languages	#Tags	Source
CoNLL-2003	English	4 tags	Reuters
CoNLL-2002	Spanish, Dutch	4 tags	Reuters
ANERcorp	Arabic	4 tags	News
CLEANANERcorp	Arabic	4 tags	News
WikiANN	176	3 tags	Wikipedia

2.6. Related Work

Early cross-lingual NER approaches aim to automatically generate labeled data for a target language using annotation projection techniques [48–52]. In annotation projection, annotations from a resource-rich language (like English) are projected onto a low-resource language through parallel corpora or word alignment techniques. A machine translation system is first used to translate gold-labeled source text into the target language. Then, word alignment algorithms [53,54] are applied to project the labels from the source to the target language to obtain labeled target language data. The result is an automatically generated dataset in the target language that can be used to train a sequence-labeling model.

Later methods make use of cross-lingual word embeddings [55] that learn aligned word vectors [56] from different types of cross-lingual supervision. First, a word embedding for each language is learned independently. After that, monolingual embeddings can be aligned using cross-lingual supervision in the form of a bilingual dictionary [57–60] or parallel sentences [61,62]. Moreover, unsupervised learning methods have been used to eliminate the requirement for cross-lingual supervision [63–65]. Furthermore, cross-lingual alignment has been effectively applied to contextual word embeddings [66–69].

With the introduction of multilingual pre-trained language models, fine-tuning multilingual language models has become the default paradigm for cross-lingual transfer. Prior studies [6,23] have evaluated the MLLMs capabilities across different NLP tasks

and languages and found that vocabulary overlap is one of the important factors as the cross-lingual performance is better between typologically similar languages. To help standardize the evaluation of MLLMs, several benchmarks have been proposed: XGLUE [13], XTREME [12], and XTREME-R [70].

To improve the cross-lingual performance of a specific language, different methods for adapting pre-trained models to a target language have been used. The simplest of these methods is to perform additional pre-training on unlabeled data in the target language [7,10,33,71], which is inspired by domain adaptation [30,72].

Another direction is to perform additional pre-training to specifically train modular adapter layers for specific tasks or languages [9,10]. Adapters are layers with a small number of parameters injected into models to help transfer learning [34]. According to [28], task-specific adapters are shown to be more effective than standard fine-tuning. In [8], invertible adapters and the MAD-X framework are introduced, which use language and task adapters for cross-lingual transfer. The process involves freezing model weights and training invertible and language adapters for each language, including English, using masked language modeling (MLM). These English-specific adapters are then combined with a task adapter to learn from labeled English data. For zero-shot transfer, the invertible and language adapters are swapped with those trained on the target language, and the model is evaluated. Moreover, instead of adapting the MLLMs to one target language, BAD-X [73] proposed bilingual adapters to adapt the MLLMs to a language pair to improve transfer performance for a particular transfer direction.

Other adaptation techniques have been focused on improving the representation of the target language by extending the model's vocabulary [31,32,74] or altering the tokenization schemes [37,39]. KinyaBERT [37] integrates a Kinyarwanda morphological analyzer into transformer architecture to learn morphology-aware input representations. Similarly, for Arabic language models, HULMonA [75] and AraBERT [38] language models utilize Arabic morphological tokenizer MADAMIRA [42] and Farasa [43], respectively, to preprocess the training dataset and separate words' prefixes and suffixes. After that, the segmented pre-training datasets are fed into the language model tokenizers to produce the sub-word vocabulary.

3. Experimental Setup

In this section, we describe our methodology, our evaluation settings, the datasets used for pre-training, evaluation metrics, the hyperparameter and fine-tuning setup, and finally, the computational cost.

3.1. Methodology

Our goal is to establish a performance baseline on English–Arabic cross-lingual NER. We evaluated the performance of fine-tuning multilingual language models and investigated the approaches discussed in Section 2.4 to improve cross-lingual performance. During the evaluation, we considered English as the source language and Arabic as the target language. We adopted the zero-shot transfer setting for evaluation, where the models are only fine-tuned on English training data and evaluated on Arabic testing data. To achieve our goal, we conducted the following experiments:

- Perform a thorough evaluation of the state-of-the-art multilingual large language models in zero-shot cross-lingual setting for the named-entity recognition task in Arabic and benchmark it against in-language models using different datasets.
- Investigate few-shot learning as a mechanism for improving zero-shot transfer on the target language and provide insight into sample efficiency and potential of the models with limited target language data.
- Apply language-adaptive pre-training (LAPT) to further specialize XLM-R using an Arabic monolingual dataset [76]. XLM-R is trained for additional epochs on monolingual Arabic data through masked language modeling (MLM), and the performance is evaluated in a zero-shot cross-lingual setting.

- Explore the effect of incorporating parallel data into the pre-training [8] using English–Arabic parallel data [77]. The performance is evaluated for a zero-shot cross-lingual setting.
- Investigate the effect of injecting adapters instead of full fine-tuning to adapt the XLM-R for Arabic using language adapter [10] and bilingual adapter BAD-X [73]. The language adapter is trained with a monolingual Arabic dataset and injected into the XLM-R model. The bilingual adapter is trained with EN–AR parallel corpora and added to the XLM-R model. The performance is evaluated in a zero-shot cross-lingual setting.
- Explore the effect of adding a layer of Arabic-specific morphological tokenizers before XLM-R tokenization using different tokenization schemes: the Penn Arabic Treebank (ATB) schema, D3 schema, and Buckwalter (BW) schema. The model performance is evaluated in a zero-shot cross-lingual setting.
- Conduct an error analysis on the different models to determine their limitations and strengths.

3.2. Evaluation Settings

We compare three cross-lingual transfer settings based on the usage of source and target language data: in-language, zero-shot, and few-shot settings. In an in-language setting, the model is fine-tuned and evaluated on the target language data. In a zero-shot setting, the model is fine-tuned on source language training data and evaluated on target language data. In few-shot settings, the model is fine-tuned on a limited number of examples (few) of target language training data.

3.3. Pre-Training Datasets

To perform language-adaptive pre-training, we obtained the monolingual texts from the ANAD news article dataset [76]. The dataset comprises over 500,000 articles from 12 Arabic news websites collected over one year from 1 January 2021 to 31 December 2021. For the bilingual dataset, we obtained the parallel dataset from [77], a 10-million-word hand-crafted Arabic–English parallel dataset.

3.4. Evaluation Metrics

We evaluated the performance of NER models by measuring the micro F1 score based on three random initializations. A named entity is correct only if it is an exact match of the corresponding entity in the data. The standard deviation of F1 scores is reported to measure the variability and reliability of the model's performance.

3.5. Hyperparameter and Fine-Tuning Setup

For NER fine-tuning, we select the best hyperparameters of each model by searching a combination of batch size, learning rate, and the number of fine-tuning epochs with the following range: learning rate $\{2 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$; batch size {8, 16, 32}; number of epochs {3, 5, 10}. Following the XTREME benchmark, all hyper-parameter tuning is conducted on English validation data. Fine-tuning time ranges from 0.2 to 0.6 hours for each model. Table 6 shows the hyperparameters selected for NER fine-tuning.

Table 6. Hyperparameters used for fine-tuning each language model for NER Task.

	mBERT	**XLM-R Base**	**XLM-R Large**	**XLM-V**	**GigaBERT**
Batch size	8	16	16	16	8
Learning rate	2×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}	2×10^{-5}
Epochs	5	5	5	5	5

Our goal is to evaluate multilingual language models and analyze state-of-the-art language adaptation approaches for the Arabic language. The following describes our implementation of these methods.

Following [71], we focused on the base version of XLM-R as our baseline language model. We considered two settings based on continued pre-training. In the first setting, (+LAPT), continue training XLM-R with an MLM objective on an ANAD monolingual dataset for 5 k steps. The training took about 1.26 days (30 h), and the hyperparameters used are the following: learning rate = 5×10^{-5}, weight decay = 0.01, warmup ratio = 0.1, epochs = 5, batch size = 8. In the second setting (+Parallel), we followed the method used in [8] and incorporated a parallel dataset in the training for 5 k steps and a total of 15 h. For this setting, we alternated between batches consisting of sentences from the source language and batches from the target language.

For the adapters, we followed the MAD-X framework [10], using language, invertible, and task adapters. This is denoted as (+Adapter). To train the task adapters, we used language and invertible adapters for the source language from AdapterHub [35]. We trained a task adapter for NER using the CoNLL 2003 English training set. Then, we trained a language adapter for Arabic on the ANAD dataset with a masked-language-modeling objective for 6.4 k steps. The training took about 5.4 h.

A language adapter is a bottleneck adapter with a reduction factor = 2. We also considered another setting, the same as the BAD-X framework [73], where we trained a bilingual adapter for EN–AR using the parallel dataset [77]. This is denoted as (+BiAdapter). The bilingual adapter was trained for 6 k steps and took about 6 h. The hyperparameters used to train the language and bilingual adapter are as follows: learning rate = 1×10^{-4}, weight decay = 1×10^{-3}, warmup ratio = 0.05, epochs = 10, and batch size = 8. For NER fine-tuning, we stacked a task adapter on the top of the trained language adapter and fine-tuned the task adapter with NER-labeled data with the following hyperparameters: learning rate = 1×10^{-4}, weight decay = 1×10^{-3}, warmup ratio = 0.05, epochs = 10, and batch size = 8.

For the language-specific preprocessing, denoted as (+MorphTOK), we added a layer of morphological tokenizers using CAMel Tools [11] before the language model sub-word tokenizer. We experiment with ATB, D3, and WB tokenization schemes. The Arabic test set passes first to the morphological tokenizer before the multilingual tokenizer.

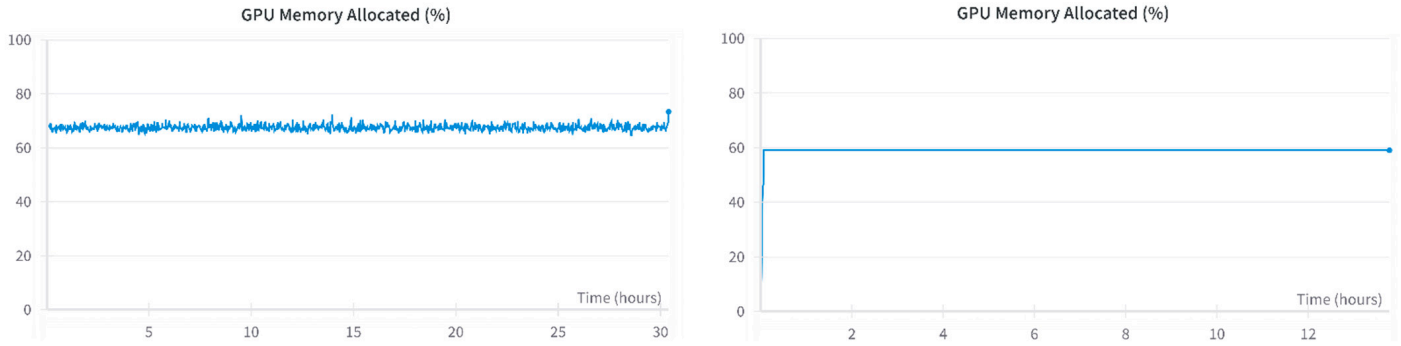
3.6. Computational Cost

All the experiments were implemented with the PyTorch framework and run using Paperspace Gradient ("https://www.paperspace.com/" (accessed on 1 July 2024)), a cloud-based machine learning platform that offers GPU-powered virtual machines. We used an A100 NVIDIA 80 GB GPU machine instance with 90 G RAM, and 12 CPUs from Paperspace cloud provider which is located in New York, USA.

Table 7 shows the configuration of the adapter model. We can see that the number of parameters in the language adapter is less than 3% compared to the number of parameters in the full model. Only language adapter parameters are updated in adapter tuning, while the rest of the model parameters remain frozen. In contrast, the LAPT method updates the full model parameters during training. This makes the computational costs of LAPT much higher than adapter training. Our experiments took about 30 h of training, using 90 GB of RAM, 12 CPUs, and an 80 GB GPU machine LAPT. The adapter took around 6 h. Figure 3 shows the GPU memory allocated for LAPT and adapter training.

Table 7. Adapter model configuration.

Name	Architecture	#Param	%Param
ner_adapter	bottleneck	7,091,712	2.551%
lang_adapter	bottleneck	7,387,776	2.657%
Full Model		278,043,648	100%



(**a**) (**b**)

Figure 3. GPU memory allocated for (**a**) LAPT and (**b**) adapter training.

4. Results and Discussion

In this section, we report our experimental results.

4.1. Zero-Shot Cross-Lingual Transfer of MLLMs on CoNLL/ANERcorp/CLEANANERcorp Datasets

This section reports the zero-shot cross-lingual evaluation results of fine-tuning multilingual language models on CoNLL2003, CoNLL2002, ANERCorp, and CLEANANERCorp datasets. The models were fine-tuned on the English subset of the CoNLL2003 and evaluated on the other datasets (English, Spanish, Dutch, and Arabic). Table 8 and Figure 4 show the results of the experiments. Overall results show that XLM-R large is the best-performing zero-shot transfer model for Latin languages (Spanish and Dutch). However, for Arabic cross-lingual NER, XLM-V and GigaBERT outperform XLM-R large on the two tested datasets. GigaBERT shows a strong cross-lingual performance for Arabic datasets and outperforms XLM-R large by at least 7 points and 10 points in ANERCorp and CLEANANERCorp, respectively. The performance of GigaBERT on CLEANANERCorp Arabic is comparable to the performance of XLM-R in Latin languages. This strong cross-lingual ability is attributed to the increase in the vocabulary size and Arabic training data size in XLM-V and GigaBERT.

Table 8. Average F1 score (±Standard Deviation) of zero-shot cross-lingual NER of MLLMs Fine-tuned using CoNLL2003 English data with in-language as the baseline (EN: English, ES: Spanish, NL: Dutch, AR: Arabic). The best result for each target dataset is bolded.

Model	EN	ES	NL	ANERCorp (AR)	CLEANANERCorp (CLEAN_AR)
Cross-lingual zero-shot transfer (models are trained on English data)					
mBERT	0.899 ± 0.005	0.715 ± 0.038	0.742 ± 0.013	0.457 ± 0.011	0.484 ± 0.002
XLM-R base	0.901 ± 0.002	0.750 ± 0.006	0.752 ± 0.008	0.505 ± 0.007	0.604 ± 0.008
XLM-R large	**0.919** ± 0.003	**0.775** ± 0.009	**0.768** ± 0.010	0.531 ± 0.029	0.633 ± 0.005
XLM-V	0.911 ± 0.001	0.770 ± 0.009	0.741 ± 0.004	0.539 ± 0.001	0.646 ± 0.008
GigaBERT	0.904 ± 0.003	-	-	**0.608** ± 0.006	**0.736** ± 0.006
In-language models (models are trained on the target language training data)					
mBERT	-	0.857 ± 0.007	0.881 ± 0.016	0.733 ± 0.004	0.794 ± 0.007
XLM-R base	-	0.853 ± 0.002	0.899 ± 0.002	0.770 ± 0.003	0.826 ± 0.010

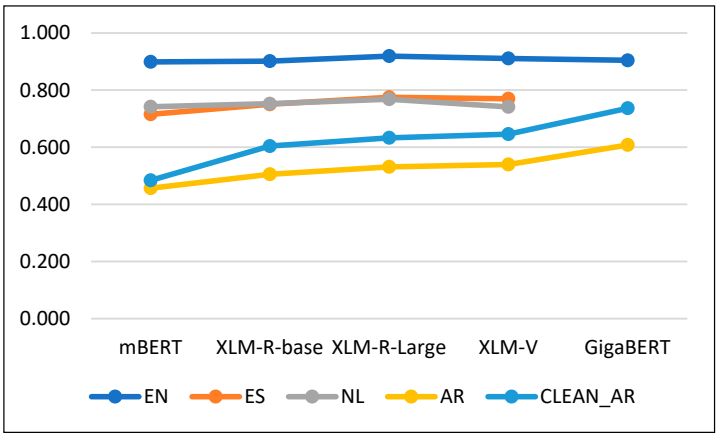


Figure 4. Zero-shot cross-lingual fine-tuning on CoNLL2003.

4.2. Zero-Shot Cross-Lingual Transfer of MLLMs on the WikiANN Dataset

This section reports the zero-shot cross-lingual evaluation results of fine-tuning MLLMs on WikiANN datasets. The models were fine-tuned on the English subset of the WikiANN and evaluated on the same CoNLL languages (English, Spanish, Dutch, and Arabic). Table 9 and Figure 5 show the results of the experiments. For Spanish, mBERT has achieved the best result, while for Dutch XLM-R large, it still achieved the best result. For Arabic, XLM-R large outperforms GigaBERT and XLM-V in contrast to the previous finding in Section 4.1.

Table 9. Average F1 score (±standard deviation) of zero-shot cross-lingual NER of MLLMs Fine-tuned using WikiANN English data with in-language as the baseline (EN: English, ES: Spanish, NL: Dutch, AR: Arabic). The best result for each target dataset is bolded.

Model	EN	ES	NL	AR
Cross-lingual zero-shot transfer (models are trained on English data)				
mBERT	**0.832 ± 0.004**	**0.754 ± 0.004**	0.804 ± 0.001	0.404 ± 0.019
XLM-R base	0.811 ± 0.015	0.751 ± 0.017	0.786 ± 0.001	0.384 ± 0.003
XLM-R large	0.826 ± 0.016	0.746 ± 0.051	**0.808 ± 0.024**	**0.435 ± 0.019**
XLM-V	**0.832 ± 0.002**	0.737 ± 0.015	0.802 ± 0.003	0.428 ± 0.034
GigaBERT	0.798 ± 0.016	-	-	0.432 ± 0.035
In-language models (models are trained on the target language training data)				
mBERT	-	0.912 ± 0.004	0.902 ± 0.002	0.865 ± 0.008
XLM-R base	-	0.893 ± 0.005	0.887 ± 0.008	0.859 ± 0.009

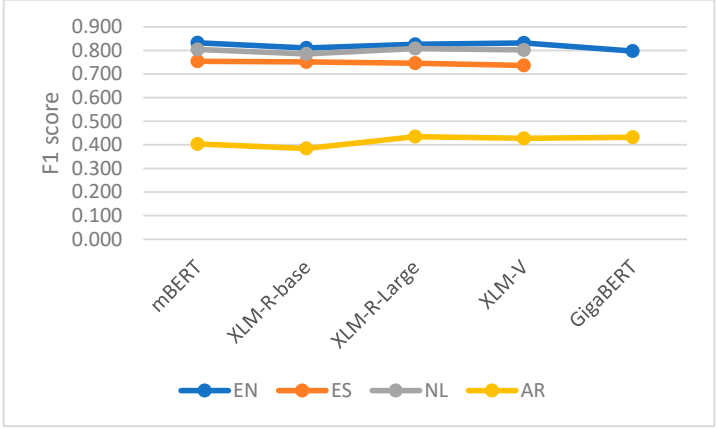


Figure 5. Zero-shot cross-lingual performance on WikiANN dataset.

4.3. Few-Shot Learning

In this section, we investigate the few-shot transfer settings using a varying number of target language examples. Our baseline models are mBERT and XLM-R base models. We experiment with the CLEANANERcorp dataset. Following [78], we continue the fine-tuning process by feeding k additional training examples that are randomly chosen from target language data. Table 10 and Figure 6 show the overall results. We can observe that adding a few target language examples consistently outperforms the zero-shot performance and reduces the cross-lingual transfer gap between zero-shot and in-language settings, especially for mBERT. The model can reach a performance comparable to the supervised setting with only 1000 target language examples.

Table 10. Results of the few-shot experiments with varying numbers of target language examples k. For each k, we report the F1 score and the difference (Δ) with respect to the zero-shot setting.

		k = 10		**k = 50**		**k = 100**		**k = 500**		**k = 1000**	
Model	k = 0	F1 score	Δ	F1 score	Δ	F1 score	Δ	F1 score	Δ	F1 score	Δ
mBERT	0.484	0.501	0.017	0.560	0.076	0.631	0.147	0.674	0.190	0.733	0.249
XLM-R	0.604	0.629	0.025	0.669	0.065	0.681	0.077	0.740	0.136	0.777	0.173

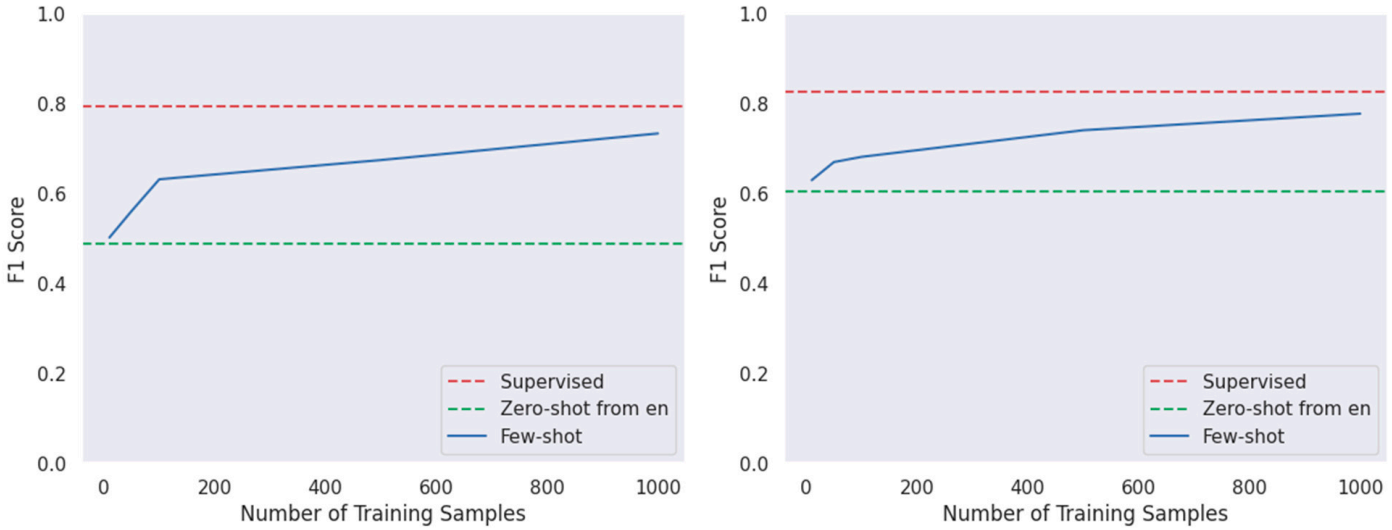


Figure 6. Few-shot transfer performance of (**a**) mBERT and (**b**) XLM-R models with varying the number of target language examples.

4.4. Language Adaptation Techniques

This section reports the evaluation results of adapting the XLM-R base model to Arabic using the language adaptation techniques discussed in Section 2.3. The models were evaluated on three different datasets. Table 11 and Figure 7 show the overall results of the experiments. Interestingly, the simplest approach of continuing pre-training on the target language continues to boost the results on all the datasets. LAPT has increased the F1 score by 4.75%, 6.5%, and 13.28% on ANERCorp, CLEANANERCorp, and WikiANN datasets, respectively.

Table 11. Average F1 Score (±Standard Deviation) of cross-lingual NER of language adaptation techniques on XLM-R base model. All results above the base model are bolded.

Model/Dataset	CoNLL2003/ANERCorp	CoNLL2003/CLEANANERCorp	WikiANN (en/ar)
XLM-R base	0.505 ± 0.007	0.600 ± 0.008	0.384 ± 0.003
+LAPT	**0.529** ± 0.015	**0.639** ± 0.015	**0.435** ± 0.021
+Parallel	**0.520** ± 0.003	0.589 ± 0.006	**0.423** ± 0.023
+Adapter	0.499 ± 0.008	0.595 ± 0.007	0.368 ± 0.028
+ BiAdapter	0.472 ± 0.009	0.570 ± 0.001	0.368 ± 0.028
+MorphTOK (ATB)	**0.555** ± 0.008	**0.659** ± 0.005	0.313 ± 0.015
+MorphTOK (D3)	0.495 ± 0.003	**0.620** ± 0.004	0.183 ± 0.014
+MorphTOK (BW)	0.471 ± 0.008	**0.620** ± 0.004	0.169 ± 0.041
+LAPT + MorphTOK (ATB)	**0.567** ± 0.004	**0.668** ± 0.001	0.343 ± 0.010

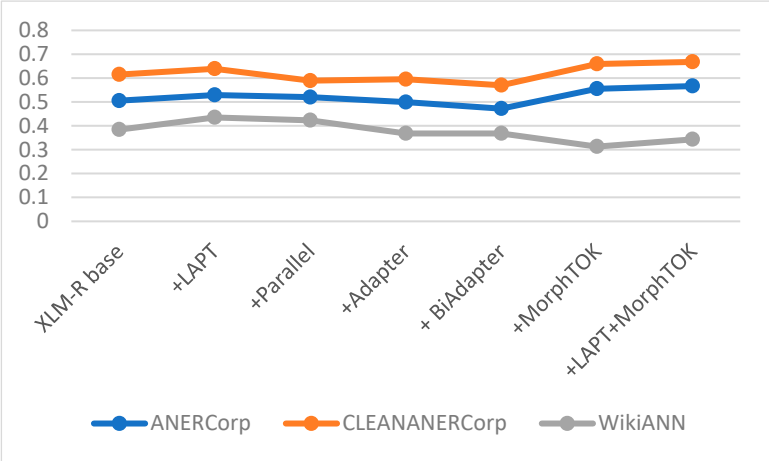


Figure 7. Performance of different adaptation methods on datasets.

Likewise, the performance of incorporating the EN–AR parallel dataset into XLM-R pre-training has increased the F1 score, except for the CLEANANERCorp dataset, where the baseline model performs slightly better. In contrast, the performance of both language adapter and bilingual adapter are lower compared to the symbolic baseline of full fine-tuning.

Furthermore, incorporating the ATB morphological tokenizer improves the accuracy of the XLM-R model by up to 9% when compared with the D3 and BW schemes in ANERcorp and CLEANANERcorp datasets. This indicates that the fine-grain segmentations of words into smaller units can lead to discrepancies, making it harder for the model to learn and recognize entity boundaries, thus reducing the F1 score. In contrast, incorporating morphological tokenizers leads to a lower F1 score for the WikiANN dataset. The varying effect of the morphological tokenizer on the NER accuracy across the dataset can be attributed to the characteristics of each dataset. ANERcorp, a gold-standard dataset, undergoes manual annotation with careful consideration of Arabic morphology, while WikiANN, a silver-standard dataset, automatically transfers annotations from English using Wikipedia hyperlinks [17]. Moreover, the WikiANN dataset exploits Wikipedia markups to segment each token into its stemming form and affixes in order to determine the name boundary and type. This automatic process may not align with the results of the Arabic morphological tokenizer, particularly when it comes to entity boundaries and annotation types.

Finally, combining the two methods, LAPT and ATB morphological tokenizer, yields the best results for the ANERCorp and CLEANANERCorp datasets. The F1 score has increased over the un-adapted model by (12.27% and 11.3%) on ANERCorp and CLEANANERCorp, respectively.

5. Errors Analysis

In this section, we performed further analysis to investigate the language adaptation approaches implemented and identified the patterns of success. We provided both quantitative and qualitative error analysis in order to understand the strengths and limitations of each approach. Our investigation focused on four models: un-adapted XLM-R base model as a baseline, the XLM-R model with language-adaptive pre-training (+LAPT), XLM-R with language adapter (+Adapter), and XLM-R with ATB morphological tokenizer (+MorphTOK). To avoid annotation errors, we inspected the result of the corrected version of the CLEANANERCorp dataset.

5.1. Per-Tag Performance

Figures 8 and 9 show the classification report and confusion matrix for each model. We can observe that ORG has the lowest F1 score in (a) XLM-R base, which indicates that detecting organization entities poses a challenge to the baseline model. The language adaptation performed improved the ORG performance in (b) XLM-R + LAPT and (d) XLM-R + Morphology models by 11% and 22%, respectively. The LOC tag was improved by 4.5% and 15%, and the MISC tag was improved by 12.9% and 11%, respectively, in models (b) and (d) over the baseline. In Figure 8, the diagonal lines of the confusion matrix represent the correct predictions, while the off-diagonal represents the misclassified entities. From the plot, we can see that matrix (a) tends to confuse the B-PER, I-PER, O, and B-LOC with the I-MISC and the B-PER with the B-LOC entities the most. The later confusion between the B-PER with B-LOC was almost resolved in matrices (b) and (d). However, the models still struggle to predict the I-MISC tag correctly. Matrix (b) shows a significant improvement in correctly identifying non-entity tokens, reducing the confusion with named-entity tags. It also shows improvements in identifying the beginning of person names. In matrix (c), the addition of the adapter seems to decrease the accuracy for some labels (B-PER dropped from 0.62 to 0.49). However, for labels like I-LOC and B-MISC, the performance improves slightly (I-LOC increased from 0.69 to 0.79). In matrix (d), with the morphological tokenizer, the overall accuracy improved for O (from 0.76 to 0.87) and for B-PER (from 0.62 to 0.74). The confusion between (B-PER with B-LOC) was reduced, and the frequency of misclassification into the O category was generally reduced for I-ORG. Overall, adding a morphological adapter has generally improved the model's ability to differentiate between entities and non-entities and enhanced the overall precision in recognizing the beginnings of names and miscellaneous entities. This suggests that incorporating linguistic features like morphology can significantly boost NER performance in a multilingual setting.

	precision	recall	f1-score	support
LOC	0.63	0.70	0.66	681
MISC	0.53	0.55	0.54	835
ORG	0.39	0.54	0.45	591
PER	0.66	0.73	0.70	901
micro avg	0.56	0.64	0.59	3008
macro avg	0.55	0.63	0.59	3008
weighted avg	0.57	0.64	0.60	3008

(**a**)

	precision	recall	f1-score	support
LOC	0.62	0.77	0.69	681
MISC	0.59	0.62	0.61	835
ORG	0.49	0.50	0.50	591
PER	0.70	0.72	0.71	901
micro avg	0.61	0.66	0.64	3008
macro avg	0.60	0.65	0.62	3008
weighted avg	0.61	0.66	0.63	3008

(**b**)

	precision	recall	f1-score	support
LOC	0.61	0.68	0.64	681
MISC	0.46	0.53	0.49	835
ORG	0.36	0.58	0.45	591
PER	0.65	0.69	0.67	901
micro avg	0.51	0.62	0.56	3008
macro avg	0.52	0.62	0.56	3008
weighted avg	0.53	0.62	0.57	3008

(**c**)

	precision	recall	f1-score	support
LOC	0.69	0.84	0.76	681
MISC	0.62	0.58	0.60	865
ORG	0.52	0.59	0.55	618
PER	0.72	0.71	0.71	936
micro avg	0.65	0.68	0.66	3100
macro avg	0.64	0.68	0.66	3100
weighted avg	0.65	0.68	0.66	3100

(**d**)

Figure 8. Classification reports for the different adaption methods implemented. (**a**) XLM-R base model; (**b**) XLM-R + LAPT; (**c**) XLM-R +Adapter; (**d**) XLM-R + morphology tokenizer.

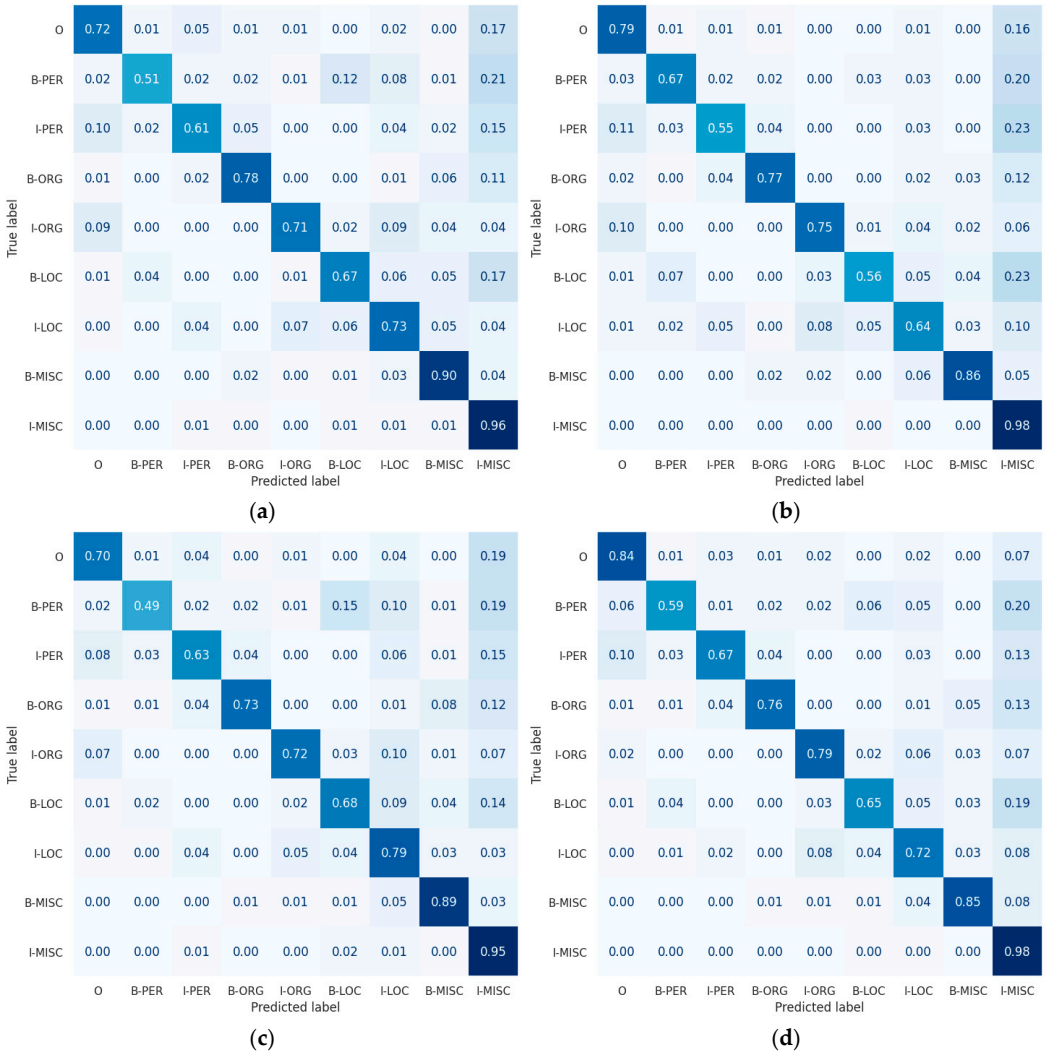


Figure 9. Confusion matrixes for the different adaption methods implemented. (**a**) XLM-R base model; (**b**) XLM-R + LAPT; (**c**) XLM-R +Adapter; (**d**) XLM-R + morphology tokenizer.

5.2. Error Distribution

To investigate the four models further, we manually inspected the first 100 sentences from each model output and classified the error types produced. Following [79], we identify five types of errors:

1. Type-1: False Positive: An entity is predicted by the NER model but is not annotated in the hand-labeled text.
2. Type-2: False Negative: A hand-labeled entity is not predicted by the model.
3. Type-3: Wrong label, Right span: A hand-labeled entity and a predicted one have the same spans but different tags.
4. Type-4: Wrong label, overlapping span: A hand-labeled entity and a predicted one have overlapping spans but different tags.
5. Type-5: Right label, overlapping span: A hand-labeled entity and a predicted one have overlapping spans and the same tags.

Figure 10 shows examples of the five errors from the CLEANANERCorp dataset. Table 12 shows the distribution of the errors in all the models tested. Figures 11 and 12 show a visualization of the statistics. By calculating the distribution of error types, we observed that for all assessed models, at least 28% of the errors are recognized as type-5 errors. Moreover, we observed that better NER models generate more type-5 errors.

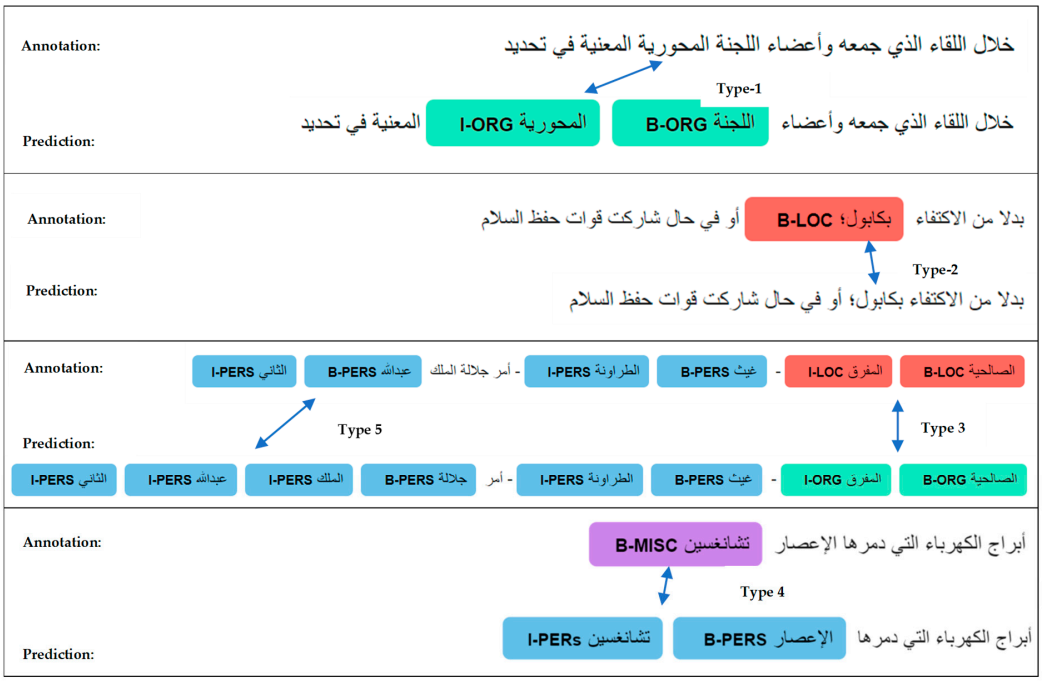


Figure 10. Examples of errors produced by XLM-R base model on CLEANANERCorp dataset.

Table 12. Percentage of errors produced by each model.

	Type-1	Type-2	Type-3	Type-4	Type-5
XLM-R base	36%	11%	16%	9%	28%
[+LAPT]	17%	15%	15%	6%	47%
[+Adapter]	37%	10%	13%	11%	30%
[+TOK]	26%	17%	20%	7%	31%

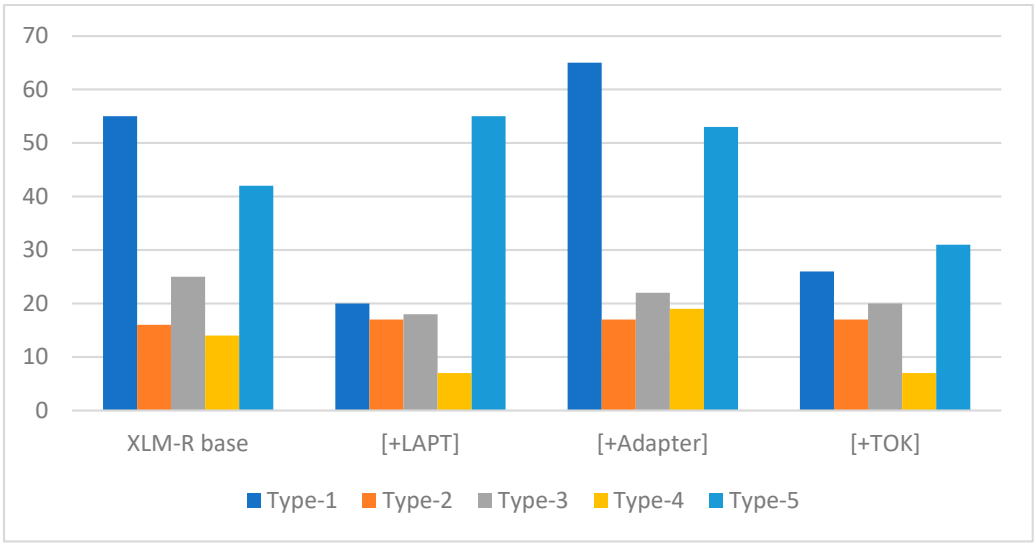


Figure 11. Distribution of errors for each model.

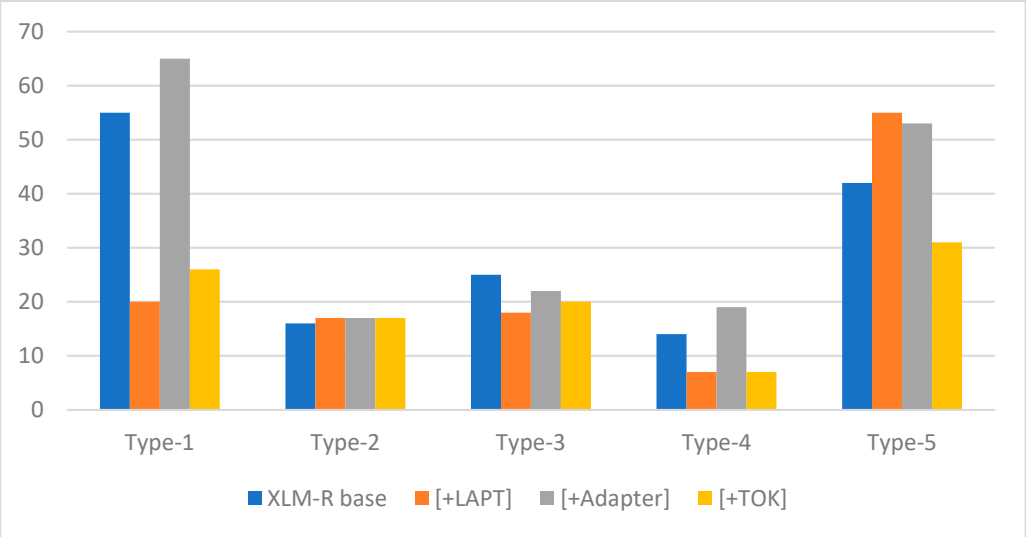


Figure 12. Distribution of errors for each error type.

5.3. Analysis of Error Types across Models

This section analyzed the error types produced by each model. Figure 10 shows that a significant proportion of the errors belong to type-1 and type-5 in all the models. So, we should investigate the errors produced by the XLM-R base model in comparison with other models.

Type-1 error is the highest-occurring error type in the XLM-R base model. The error occurs when the model is over-predicting non-named entities as named entities. An inspection into type-1 errors in the XLM-R base model shows that almost 30% represent the start of a sentence where ambiguity occurs [16]. For example, in the sentence ("كيف ساءت الأمور إلى هذا الحد وبهذه السرعة؟ :الأحمر، لنا أن نتساءل"), the word ("الأحمر") has been mislabeled as B-PER by the model while annotated as O.

Type-5 error is regarding the boundary of the named entity. Arabic named entities usually comprise prefix with the name. For example, the Arabic entity "صحيفة الجارديان" refers to the newspaper name, whereas in English, the name is "The Guardian" without the word "Newspaper". The grammatical difference between English and Arabic makes it difficult to accurately predict the correct boundary under the zero-shot learning setting. Another cause for type-5 errors is the Arabic conjunction letters connected to the named entities. For example, in the sentence ("يمكن أن يحدث للعراق"), the word ("للعراق") should be labeled as LOC. However, the XLM-R tokenizer has tokenized the word as ("_للع","راق") and partially labeled as LOC, which causes a type-5 error. Adding an ATB morphological tokenizer mostly resolved this issue by breaking the word down into more efficient representations before processing with the XLM tokenizer. The word in the example has been tokenized as ("_ل","العراق") by the ATB tokenizer.

Type-3 errors occur frequently in the XLM-R base model when dealing with entities referring to press agencies. An inspection into type-3 errors shows that 20% of the errors were produced because of the ambiguity between a press agency name and a location name. As the dataset is from the news domain, most articles contain the name of the press agency, which the model has mislabeled. A press agency name can entail the meaning of both organization and location. For example, the names (e.g., "الشرق الأوسط، الجزيرة، الفرنسية") that refer to press agencies have been mislabeled as LOC, which should have been labeled as ORG.

The language-adaptive pre-training in LAPT helped the model to acquire more knowledge about the linguistic characteristics, syntax, and semantics of the Arabic language,

thus helping to enhance the model's ability to accurately identify and classify named entities. Furthermore, adding the ATB morphological tokenizer can better handle the word variations by accurately segmenting words into meaningful sub-words, improving the model's understanding of the language and hence improving the model predictions. Consequently, all error types have been reduced in the two models compared to the base model, except for type-5 errors.

6. Conclusions and Future Work

We contribute to the literature by providing a comparative study on the recent multilingual large language models for Arabic cross-lingual named-entity recognition. Further, we present a comparison of language adaptation approaches with detailed error analysis. Based on the experimental results, we found that the zero-shot cross-lingual ability of a bilingual language model for a given language pair like GigaBERT outperforms the general-purpose multilingual language models like mBERT and XLM-R. This could be attributed to the increased resources allocated for each language, including training data size and vocabulary, which results in reducing out-of-vocabulary words. For language adaptation techniques, language-adaptive pre-training (LAPT) consistently leads to better performance for all the datasets. This demonstrates the effectiveness of specialized adaptive pre-training techniques for other morphologically rich and low-resource languages. Experimental results showed that incorporating language-specific knowledge is essential for improving cross-lingual NER, especially for heterogeneous language pairs. This might provide a novel insight into further cross-lingual NER research.

Several avenues for future work can be explored to advance further the field of Arabic cross-lingual named-entity recognition (NER). These include exploring other adaptation and pre-training techniques, such as multi-task learning, vocabulary expansion, and parameter-efficient transfer; investigating a wider range of k values; and developing strategies for optimizing few-shot adaptation. Furthermore, inspecting the integration of external knowledge sources, such as knowledge graphs, lexicons, ontologies, and domain-specific embeddings, to enrich the contextual understanding of the language models is a promising method to improve NER accuracy for ambiguous entities. To ensure the generalization of our findings, we encourage future research to evaluate the adaptation methods on a wider range of datasets and domains and consider a more comprehensive evaluation framework that considers additional measures, such as the entity boundary.

Author Contributions: Conceptualization, M.A.-D. and H.A.-K.; methodology, M.A.-D. and H.A.-K.; software, M.A.-D.; validation, M.A.-D., H.A.-K., and A.A.-S.; formal analysis, M.A.-D. and H.A.-K.; investigation, M.A.-D. and H.A.-K.; resources, M.A.-D.; data curation, M.A.-D.; writing—original draft preparation, M.A.-D.; writing—review and editing, H.A.-K. and A.A.-S.; visualization, M.A.-D.; supervision, H.A.-K. and A.A.-S.; project administration, H.A.-K. and A.A.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original CoNLL2003 dataset presented in the study is openly available at https://www.clips.uantwerpen.be/conll2003/ner/ (accessed on 15 January 2024). The CoNLL2002 dataset presented in the study is openly available at https://www.clips.uantwerpen.be/conll2002/ner/ (accessed on 15 January 2024). The original ANERcorp dataset presented in the study is available upon request at https://camel.abudhabi.nyu.edu/anercorp/ (accessed on 15 January 2024). The CLEANANERCorp dataset presented in the study is openly available at https://github.com/iwan-rg/CLEANANERCorp (accessed on 15 February 2024). The WikiANN dataset presented in the study is openly available at https://huggingface.co/datasets/unimelb-nlp/wikiann (accessed on 15 January 2024).

Acknowledgments: The authors would like to thank King Saud University and the College of Computer and Information Sciences.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
2. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 8440–8451.
3. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 483–498.
4. Wu, Q.; Lin, Z.; Karlsson, B.; Lou, J.-G.; Huang, B. Single-/Multi-Source Cross-Lingual NER via Teacher-Student Learning on Unlabeled Data in Target Language. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 6505–6514.
5. García-Ferrero, I.; Agerri, R.; Rigau, G. Model and Data Transfer for Cross-Lingual Sequence Labelling in Zero-Resource Settings. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 6403–6416.
6. Pires, T.; Schlinger, E.; Garrette, D. How Multilingual is Multilingual BERT? In Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4996–5001.
7. Chau, E.C.; Lin, L.H.; Smith, N.A. Parsing with Multilingual BERT, a Small Corpus, and a Small Treebank. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 1324–1334.
8. Reid, M.; Artetxe, M. On the Role of Parallel Data in Cross-lingual Transfer Learning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, ON, Canada, 9–14 July 2023; pp. 5999–6006.
9. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; Laroussilhe, Q.D.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-Efficient Transfer Learning for NLP. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2790–2799.
10. Pfeiffer, J.; Vulić, I.; Gurevych, I.; Ruder, S. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 7654–7673.
11. Obeid, O.; Zalmout, N.; Khalifa, S.; Taji, D.; Oudah, M.; Alhafni, B.; Inoue, G.; Eryani, F.; Erdmann, A.; Habash, N. CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing. In Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, 11–16 May 2020; pp. 7022–7032.
12. Hu, J.; Ruder, S.; Siddhant, A.; Neubig, G.; Firat, O.; Johnson, M. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation. In Proceedings of the 37th International Conference on Machine Learning, PMLR, Virtual Event, 13–18 July 2020; pp. 4411–4421.
13. Liang, Y.; Duan, N.; Gong, Y.; Wu, N.; Guo, F.; Qi, W.; Gong, M.; Shou, L.; Jiang, D.; Cao, G.; et al. XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 6008–6018.
14. Sang, E.F.T.K.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition 2003. In Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in Cooperation with HLT-NAACL 2003, Edmonton, AB, Canada, 31 May–1 June 2003.
15. Benajiba, Y.; Rosso, P.; BenedíRuiz, J.M. ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy. In Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2007, Mexico City, Mexico, 18–24 February 2007; pp. 143–153.
16. AlDuwais, M.; Al-Khalifa, H.; AlSalman, A. CLEANANERCorp: Identifying and Correcting Incorrect Labels in the ANERcorp Dataset. In Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation, LREC-COLING 2024, Torino, Italy, 20–25 May 2024; pp. 13–19.
17. Pan, X.; Zhang, B.; May, J.; Nothman, J.; Knight, K.; Ji, H. Cross-lingual Name Tagging and Linking for 282 Languages. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1946–1958.
18. Rahimi, A.; Li, Y.; Cohn, T. Massively Multilingual Transfer for NER. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; pp. 151–164.
19. Hackett, J. Semitic Languages. In *Encyclopedia of Language & Linguistics*, 2nd ed.; Brown, K., Ed.; Elsevier: Oxford, UK, 2006; pp. 229–235; ISBN 978-0-08-044854-1.
20. Shaalan, K.; Siddiqui, S.; Alkhatib, M.; Abdel Monem, A. Challenges in Arabic Natural Language Processing. In *Computational Linguistics, Speech and Image Processing for Arabic Language*; Series on Language Processing, Pattern Recognition, and Intelligent Systems; World Scientific: Singapore, 2017; Volume 4, pp. 59–83; ISBN 978-981-322-938-9.

21. Mohit, B. Named Entity Recognition. In *Natural Language Processing of Semitic Languages*; Zitouni, I., Ed.; Theory and Applications of Natural Language Processing; Springer: Berlin/Heidelberg, Germany, 2014; pp. 221–245; ISBN 978-3-642-45357-1.
22. Ruder, S. Neural Transfer Learning for Natural Language Processing. Ph.D. Thesis, NUI Galway, Galway, Ireland, 2019.
23. Wu, S.; Dredze, M. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 833–844.
24. Karthikeyan, K.; Wang, Z.; Mayhew, S.; Roth, D. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In Proceedings of the International Conference on Learning Representations, Scottsdale, AZ, USA, 6–9 May 2019.
25. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
26. Conneau, A.; Lample, G. Cross-Lingual Language Model Pretraining. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019); Vancouver, BC, Canada, 8–14 December 2019.
27. Liang, D.; Gonen, H.; Mao, Y.; Hou, R.; Goyal, N.; Ghazvininejad, M.; Zettlemoyer, L.; Khabsa, M. XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, 6–10 December 2023; pp. 13142–13152.
28. Lan, W.; Chen, Y.; Xu, W.; Ritter, A. An Empirical Study of Pre-trained Transformers for Arabic Information Extraction. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 4727–4734.
29. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018; pp. 328–339.
30. Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 8342–8360.
31. Chau, E.C.; Smith, N.A. Specializing Multilingual Language Models: An Empirical Study. In Proceedings of the 1st Workshop on Multilingual Representation Learning (MRL 2021), Punta Cana, Dominican Republic, 11 November 2021; pp. 51–61.
32. Wang, Z.; Karthikeyan, K.; Mayhew, S.; Roth, D. Extending Multilingual BERT to Low-Resource Languages. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 2649–2656.
33. Muller, B.; Anastasopoulos, A.; Sagot, B.; Seddah, D. When Being Unseen from mBERT Is just the Beginning: Handling New Languages with Multilingual Language Models. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 448–462.
34. Rebuffi, S.-A.; Bilen, H.; Vedaldi, A. Learning Multiple Visual Domains with Residual Adapters. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 506–516.
35. Pfeiffer, J.; Rücklé, A.; Poth, C.; Kamath, A.; Vulić, I.; Ruder, S.; Cho, K.; Gurevych, I. AdapterHub: A Framework for Adapting Transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020, Online, 16–20 November 2020; pp. 46–54.
36. He, R.; Liu, L.; Ye, H.; Tan, Q.; Ding, B.; Cheng, L.; Low, J.; Bing, L.; Si, L. On the Effectiveness of Adapter-Based Tuning for Pretrained Language Model Adaptation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, Virtual Event, 1–6 August 2021; pp. 2208–2222.
37. Nzeyimana, A.; Niyongabo Rubungo, A. KinyaBERT: A Morphology-Aware Kinyarwanda Language Model. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 5347–5363.
38. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 11–16 May 2020; pp. 9–15.
39. Wang, X.; Ruder, S.; Neubig, G. Multi-View Subword Regularization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6–11 June 2021; pp. 473–482.
40. Alotaiby, F.; Foda, S.; Alkharashi, I. Clitics in Arabic Language: A Statistical Study. In Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, PACLIC 24, Tohoku University, Sendai, Japan, 4–7 November 2010; pp. 595–601.
41. El Kholy, A.; Habash, N. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC), Valletta, Malta, 17–23 May 2010.
42. Pasha, A.; Al-Badrashiny, M.; Diab, M.; El Kholy, A.; Eskander, R.; Habash, N.; Pooleery, M.; Rambow, O.; Roth, R. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, 26–31 May 2014; pp. 1094–1101.

43. Abdelali, A.; Darwish, K.; Durrani, N.; Mubarak, H. Farasa: A Fast and Furious Segmenter for Arabic. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016, San Diego, CA, USA, 12–17 June 2016; pp. 11–16.
44. Maamouri, M.; Bies, A.; Buckwalter, T.; Mekki, W. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools, Cairo, Egypt, 23–24 September 2004.
45. Habash, N.; Sadat, F. Arabic Preprocessing Schemes for Statistical Machine Translation. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, New York, NY, USA, 4–9 June 2006; pp. 49–52.
46. Khalifa, S.; Zalmout, N.; Habash, N. YAMAMA: Yet Another Multi-Dialect Arabic Morphological Analyzer. In Proceedings of the 26th International Conference on Computational Linguistics, COLING 2016, Osaka, Japan, 11–16 December 2016; pp. 223–227.
47. Tjong Kim Sang, E.F. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In Proceedings of the 6th Conference on Natural Language Learning, CoNLL 2002, Held in cooperation with COLING 2002, Taipei, Taiwan, 26–30 August 2002.
48. Jain, A.; Paranjape, B.; Lipton, Z.C. Entity Projection via Machine Translation for Cross-Lingual NER. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 1083–1092.
49. Fei, H.; Zhang, M.; Ji, D. Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 7014–7026.
50. Ehrmann, M.; Turchi, M.; Steinberger, R. Building a Multilingual Named Entity-Annotated Corpus Using Annotation Projection. In Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, Hissar, Bulgaria, 12–14 September 2011; pp. 118–124.
51. Ni, J.; Dinu, G.; Florian, R. Weakly Supervised Cross-Lingual Named Entity Recognition via Effective Annotation and Representation Projection. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1470–1480.
52. Fu, R.; Qin, B.; Liu, T. Generating Chinese Named Entity Data from a Parallel Corpus. In Proceedings of the Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, 8–13 November 2011; pp. 264–272.
53. Dyer, C.; Chahuneau, V.; Smith, N.A. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In Proceedings of the Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Atlanta, GA, USA, 9–14 June 2013; pp. 644–648.
54. Dou, Z.-Y.; Neubig, G. Word Alignment by Fine-Tuning Embeddings on Parallel Corpora. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, 19–23 April 2021; pp. 2112–2128.
55. Ruder, S.; Vulić, I.; Søgaard, A. A Survey Of Cross-Lingual Word Embedding Models. *J. Artif. Intell. Res.* **2019**, *65*, 569–631. [CrossRef]
56. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
57. Mikolov, T.; Le, Q.V.; Sutskever, I. Exploiting Similarities among Languages for Machine Translation. *arXiv* **2013**, arXiv:1309.4168.
58. Faruqui, M.; Dyer, C. Improving Vector Space Word Representations Using Multilingual Correlation. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, Gothenburg, Sweden, 26–30 April 2014; pp. 462–471.
59. Gouws, S.; Søgaard, A. Simple Task-Specific Bilingual Word Embeddings. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2015, Denver, CO, USA, 31 May–5 June 2015; pp. 1386–1390.
60. Artetxe, M.; Labaka, G.; Agirre, E. Learning Principled Bilingual Mappings of Word Embeddings While Preserving Monolingual Invariance. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, TX, USA, 1–4 November 2016; pp. 2289–2294.
61. Hermann, K.M.; Blunsom, P. Multilingual Distributed Representations without Word Alignment. *arXiv* **2014**, arXiv:1312.6173.
62. Gouws, S.; Bengio, Y.; Corrado, G. BilBOWA: Fast Bilingual Distributed Representations without Word Alignments. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015; pp. 748–756.
63. Zhang, M.; Liu, Y.; Luan, H.; Sun, M. Adversarial Training for Unsupervised Bilingual Lexicon Induction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1959–1970.
64. Lample, G.; Conneau, A.; Ranzato, M.; Denoyer, L.; Jégou, H. Word Translation without Parallel Data. *arXiv* **2018**, arXiv:1710.04087.
65. Zhou, C.; Ma, X.; Wang, D.; Neubig, G. Density Matching for Bilingual Word Embedding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 1588–1598.

66. Schuster, T.; Ram, O.; Barzilay, R.; Globerson, A. Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-Shot Dependency Parsing. *arXiv* **2019**, arXiv:1902.09492. [CrossRef]
67. Wang, Y.; Che, W.; Guo, J.; Liu, Y.; Liu, T. Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 5721–5727.
68. Cao, S.; Kitaev, N.; Klein, D. Multilingual Alignment of Contextual Word Representations. *arXiv* **2020**, arXiv:2002.03518.
69. Aldarmaki, H.; Diab, M. Context-Aware Cross-Lingual Mapping. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 3906–3911.
70. Ruder, S.; Constant, N.; Botha, J.; Siddhant, A.; Firat, O.; Fu, J.; Liu, P.; Hu, J.; Garrette, D.; Neubig, G.; et al. XTREME-R: Towards More Challenging and Nuanced Multilingual Evaluation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 7–11 November 2021; pp. 10215–10245.
71. Ebrahimi, A.; Kann, K. How to Adapt Your Pretrained Multilingual Model to 1600 Languages. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1–6 August 2021; pp. 4555–4567.
72. Han, X.; Eisenstein, J. Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019; pp. 4238–4248.
73. Parović, M.; Glavaš, G.; Vulić, I.; Korhonen, A. BAD-X: Bilingual Adapters Improve Zero-Shot Cross-Lingual Transfer. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, USA, 10–15 July 2022; pp. 1791–1799.
74. Zhang, R.; Gangi Reddy, R.; Sultan, M.A.; Castelli, V.; Ferritto, A.; Florian, R.; Sarioglu Kayi, E.; Roukos, S.; Sil, A.; Ward, T. Multi-Stage Pre-training for Low-Resource Domain Adaptation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 5461–5468.
75. ElJundi, O.; Antoun, W.; El Droubi, N.; Hajj, H.; El-Hajj, W.; Shaban, K. hULMonA: The Universal Language Model in Arabic. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, WANLP 2019, Florence, Italy, 28 July–2 August 2019; pp. 68–77.
76. Altamimi, M.; Alayba, A.M. ANAD: Arabic News Article Dataset. *Data Brief* **2023**, *50*, 109460. [CrossRef]
77. Alotaibi, H.M. Arabic-English Parallel Corpus: A New Resource for Translation Training and Language Teaching. *Arab World Engl. J.* **2017**, *8*, 319–337. [CrossRef]
78. Lauscher, A.; Ravishankar, V.; Vulić, I.; Glavaš, G. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16–20 November 2020; pp. 4483–4499.
79. Nejadgholi, I.; Fraser, K.C.; de Bruijn, B. Extensive Error Analysis and a Learning-Based Evaluation of Medical Entity Recognition Systems to Approximate User Experience. In Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing, BioNLP 2020, Online, 9 July 2020; pp. 177–186.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.