Revisiting Pre-trained Language Models and their Evaluation for Arabic Natural Language Processing

Abbas Ghaddar[1,*] **Yimeng Wu**[1,*] **Sunyam Bagga**[1] **Ahmad Rashid**[1] **Khalil Bibi**[1]
Mehdi Rezagholizadeh[1] **Chao Xing**[1] **Yasheng Wang**[1] **Duan Xinyu**[2]
Zhefeng Wang[2] **Baoxing Huai**[2] **Xin Jiang**[1] **Qun Liu**[1] **and Philippe Langlais**[3]
[1] Huawei Technologies Co., Ltd.
[2] Huawei Cloud Computing Technologies Co., Ltd
[3] RALI/DIRO, Université de Montréal, Canada
{abbas.ghaddar,yimeng.wu,sunyam.bagga}@huawei.com
{ahmad.rashid,khalil.bibi,mehdi.rezagholizadeh}@huawei.com
{xingchao.ml,wangyasheng,duanxinyu}@huawei.com
{wangzhefeng,huaibaoxing,jiang.xin,qun.liu}@huawei.com
felipe@iro.umontreal.ca

Abstract

There is a growing body of work in recent years to develop pre-trained language models (PLMs) for the Arabic language. This work addresses two major problems in existing Arabic PLMs that limit the progress of the Arabic NLU and NLG fields. First, existing Arabic PLMs are not well-explored and their pre-training can be improved significantly using a more methodical approach. Second, there is a lack of systematic and reproducible evaluation of these models in the literature. We revisit both the pre-training and evaluation of Arabic PLMs. In terms of pre-training, we explore the impact of the quality of the pretraining data, the size of the model and the incorporation of character-level information to Arabic PLMs. As a result, we release three new Arabic BERT-style models (JABER, Char-JABER, and SABER), and two T5-style models (AT5S and AT5B). In terms of evaluation, we conduct a comprehensive empirical study to systematically evaluate the performance of existing state-of-the-art models on ALUE, a leaderboard-powered benchmark for Arabic NLU tasks, and on a subset of Arabic generative tasks. We show that our models significantly outperform existing Arabic PLMs and achieve a new state-of-the-art performance on both discriminative and generative tasks.

1 Introduction

Pre-trained language models (PLMs) such as BERT (Devlin et al., 2018), GPT (Radford et al., 2018), and T5 (Raffel et al., 2019) have become the default standard architectures for modern natural language understanding (NLU) systems in both academic (Kalyan et al., 2021; Min et al., 2021) and industrial settings (Chakravarti et al., 2020; Tunstall et al., 2022; Li et al., 2021). On the evaluation side, the community has widely adopted the leaderboard paradigm[1] as a reliable and fair tool to track the progress on various NLP tasks (Mehri et al., 2020; Wang et al., 2018, 2019).

Recent years have seen tremendous efforts to develop language-specific PLMs (Le et al., 2020; Chan et al., 2020; Canete et al., 2020; Ulčar and Robnik-Šikonja, 2020) and leaderboards (Xu et al., 2020, 2021; Shavrina et al., 2020; Wilie et al., 2020) for languages other than English. These language-specific PLMs have proven to be more accurate than multilingual ones in monolingual evaluation settings (Martin et al., 2019; Wei et al., 2019; Safaya et al., 2020). Moreover, creating high-quality human-curated benchmarks is considered to be of utmost importance for reliable evaluation (DeYoung et al., 2020; Kiela et al., 2021).

For some high-resource languages like Chinese, the community has been able to be on par with English NLU in terms of developing PLMs (Sun et al., 2019, 2020, 2021; Zeng et al., 2021) and evaluating them on publicly available leaderboards (Xu et al., 2020). However, we find that the NLP community is unfortunately lagging behind for other languages like Arabic. Despite the wide availability of Arabic PLMs (Abdul-Mageed et al., 2021; Antoun et al., 2020; Nagoudi et al., 2022; Inoue et al., 2021) and datasets (Zeroual et al., 2019; El-Khair, 2016; Nagoudi et al., 2020), there are two major issues that constrain the progress of Arabic NLU field.

First, we observe that the latest techniques for improving pre-training (Brown et al., 2020; Clark et al., 2022; Di Liello et al., 2021) are under-

[*] Equal contribution

[1] We use the same definition of leaderboard as Ethayarajh and Jurafsky (2020).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3135 - 3151
December 7-11, 2022

explored in the context of Arabic PLMs. In this work, we investigate three ways to improve on the existing Arabic PLMs: quality of the pre-training data, size of the model, and morphology. We propose JABER, a BERT-base model pre-trained on high-quality filtered data, that significantly outperforms the best Arabic PLM baseline by 1.5% on ALUE (Seelawi et al., 2021), a newly proposed benchmark with a leaderboard for sequence classification Arabic tasks.[2] We also explore two other variants of JABER and report further gains in performance: (i) Char-JABER which exploits character-level information and (ii) SABER which involves a BERT-large model.

Second, there is a lack of systematic and reproducible evaluation. As a matter of fact, most of the existing work on Arabic PLMs does not follow the recommended evaluation protocols (Pineau, 2020; Chen et al., 2022) which include extensive hyperparameter tuning, performing multiple runs on the development set, and reporting performance on hidden test sets. To address this issue, we systematically compare five popular BERT-based Arabic PLMs by carefully assessing their performance on the ALUE leaderboard. We find that the performance ranking of models drastically changes when measured on dev sets as compared to the leaderboard test sets, thereby calling for caution when comparing models without a leaderboard setting.

Furthermore, we extend our work to T5 encoder-decoder models and Arabic generative tasks. We pre-train two T5 small and base models for Arabic: AT5S and AT5B. AT5B achieves state-of-the-art results on several generative tasks (Naous et al., 2020; Ladhak et al., 2020) by outperforming the recently proposed AraT5-base model (Nagoudi et al., 2022) both on automatic and human evaluations. We further observe that T5-based Arabic PLMs perform worse than the BERT-based models on the ALUE benchmark which is in contrast to the powerful performance of T5-models on English language tasks (Raffel et al., 2019). We conclude with a set of suggestions and directions to explore for pushing progress forward in the Arabic NLU community.

2 Related Work

There have been several efforts to improve on the pre-training paradigm by scaling up the model size (Lepikhin et al., 2021; Brown et al., 2020) and data size (Liu et al., 2019), exploring new pre-training tasks (Di Liello et al., 2021; Panda et al., 2021) and model architectures (Lan et al., 2019; Voita et al., 2019), and support for long input sequences (Choromanski et al., 2021; Beltagy et al., 2020). In this work, we use the original setting of BERT (Devlin et al., 2018) and T5 (Raffel et al., 2019) models to pre-train our Arabic encoder-only and encoder-decoder models respectively. The broader goal is to be fairly and directly comparable with other existing Arabic PLMs discussed below.

Table 1 shows the configuration used by popular publicly available Arabic BERT models as well as those of JABER and SABER. AraBERT (Antoun et al., 2020) and Arabic-BERT (Safaya et al., 2020) were amongst the first to pre-train 12-layer BERT-base models specifically for Arabic. Abdul-Mageed et al. (2021) proposed two BERT-based models: ARBERT which is tailored for Modern Standard Arabic (MSA) NLU tasks and MARBERT dedicated to tasks that include Arabic dialects (especially tweets). ARBERT and MARBERT are pre-trained on 61GB and 128GB of MSA and tweets data respectively. Inoue et al. (2021) go one step further and pre-train a single BERT-base model called CAMeLBERT, on 167GB of MSA, dialect and classic Arabic data. The major difference between JABER and these existing Arabic PLMs is that JABER is pre-trained on a high-quality and strictly filtered dataset (115GB out of 514GB).

A wide range of methods have been proposed lately to enrich PLMs with character-level information, as it has been shown to be beneficial for morphologically rich languages like Arabic (Kim et al., 2016; Gerz et al., 2018; Clark et al., 2022). Ma et al. (2020) proposed Noisy Language Modeling, a new unsupervised pre-training objective for learning character representations. Pinter et al. (2021) proposed their XRayEmb method that involves adding character-level information to existing PLMs without the need for pretraining them from scratch. CharacterBERT (El Boukkouri et al., 2020) uses a character-CNN module to learn representations for entire words by consulting the characters of each token, thus avoiding to recourse to word-pieces (Wu et al., 2016). Our character-enhanced BERT-base model, Char-JABER, uses a simple and efficient method to inject character-level representations alongside the sub-tokens representations only at the input layer of BERT, with minimal additional parameters and no computational overhead.

[2]The Arabic equivalent of GLUE (Wang et al., 2018).

Model	Arabic-BERT	AraBERT	CAMeLBERT	ARBERT	MARBERT	JABER	SABER
#Params (w/o emb)	110M (85M)	135M (85M)	108M (85M)	163M (85M)	163M (85M)	135M (85M)	369M (307M)
Vocab Size	32k	64k	30k	100k	100k	64k	64k
Tokenizer	WordPiece	WordPiece	WordPiece	WordPiece	WordPiece	BBPE	BBPE
Normalization	✘	✓	✓	✘	✘	✓	✓
Data Filtering	✘	✘	✘	✘	✘	✓	✓
Textual Data Size	95GB	27GB	167GB	61GB	128GB	115GB	115GB
Duplication Factor	3	10	10	-	-	3	3
Training epochs	27	27	2	42	36	15	5

Table 1: Configuration of publicly available Arabic BERT models and our JABER and SABER models. AraBERT and MARBERT did not provide their data duplication factor. Char-JABER has the same characteristics as JABER.

Recent efforts have also been made to develop benchmarks for Arabic NLU tasks. Abdul-Mageed et al. (2021) proposed the ARLUE benchmark which is a collection of 42 discriminative classification tasks. Nagoudi et al. (2022) proposed the ARGEN benchmark which consists of 19 datasets for generative tasks. However, both benchmarks have certain limitations which make it challenging to meaningfully evaluate Arabic PLMs. For many tasks, the authors use their own train-dev-test splits which are not made publicly available, as of May 10, 2022. In addition, the access to some datasets is not available free of cost. Furthermore, none of the tasks include privately-held test data which is important to ensure that a benchmark is used fairly (Wang et al., 2018). Therefore, we adopt the ALUE benchmark (Seelawi et al., 2021) for evaluating our models on classification tasks because this benchmark has a public leaderboard and includes privately-held test sets for many tasks. For evaluating our Arabic T5 models, we select a subset of generative tasks from the ARGEN benchmark whose results are freely reproducible (see Section 4.1).

3 Pre-training

3.1 Data Collection and Processing

We collect our pre-training corpus from the following four sources:

Common Crawl (CC): We use 10 shards of Common Crawl[3] data from March to December 2020. After removing non-Arabic text, this dataset is 444GB in size. Additionally, we use the monthly shard of CC from November 2018 provided by the OSCAR project (Suárez et al., 2019). We download the unshuffled version (31GB) from HuggingFace Datasets (Lhoest et al., 2021).

NEWS: We use the links provided in the Open Source International Arabic News Corpus (Zeroual et al., 2019) to collect 21GB of textual data from 19 popular Arabic news websites.

EL-KHAIR: We use the 1.5 billion words Arabic Corpus (El-Khair, 2016) which is a collection of newspaper articles published by 10 Arabic news sources between 2002-2014.

WIKI: We use the Arabic Wikipedia dump[4] from June 2021 and extract the text of articles using WikiExtractor (Attardi, 2015).

Recent studies have highlighted the importance of cleaning up raw pre-training data for achieving better performance on downstream tasks (Raffel et al., 2019; Brown et al., 2020). We developed a set of heuristics for cleaning our Arabic corpora that is able to filter out gibberish, noisy and duplicated texts (see Appendix A.1).

Source	Original	Clean	
CC	475GB	87GB	(18%)
NEWS	21GB	14GB	(67%)
EL-KHAIR	16GB	13GB	(82%)
WIKI	1.6GB	1GB	(63%)
Total	514GB	115GB	(22%)

Table 2: Size of our pre-training corpus before and after applying the data cleaning methods. Parentheses indicate the proportion of the remaining data.

Table 2 shows the size of our pre-training corpora before and after data pre-processing. The final pre-training dataset represents only 22% of the original corpus and is 115GB in size. Although our approach seemingly filters out a large proportion of the dataset, our corpus size is comparable with other models such as Arabic-BERT

[3] https://commoncrawl.org

[4] https://dumps.wikimedia.org/

Task	|Train|	|Dev|	|Test|	Metric	|Classes|	Domain	Lang	Seq. Len.
Single-Sentence Classification								
MDD	42k	5k	**5k**	F1-macro	26	Travel	DIAL	7±3.7
OOLD	7k	1k	1k	F1-macro	2	Tweet	DIAL	21±13.3
OHSD	7k	1k	1k	F1-macro	2	Tweet	DIAL	21±13.3
FID	4k	-	**1k**	F1-macro	2	Tweet	DIAL	23±11.7
Sentence-Pair Classification								
MQ2Q	12k	-	4k	F1-macro	2	Web	MSA	13±2.9
XNLI	5k	-	**3k**	Accuracy	3	Misc	MSA	27±9.6
Multi-label Classification								
SEC	2k	600	1k	Jaccard	11	Tweet	DIAL	18±7.8
Regression								
SVREG	1k	138	1k	Pearson	1	Tweet	DIAL	18±7.9

Table 3: Task descriptions and statistics of the ALUE benchmark. Test sets in bold use labels that are publicly available. The average sequence length and standard deviations are computed based on the word count of the tokenized text of the training set.

(95GB) and MARBERT (128GB). Moreover, as we will discuss in Section 4, our models are able to significantly outperform other models that used light pre-processing (Safaya et al., 2020; Abdul-Mageed et al., 2021). We also utilise the Arabic text-normalization procedure of AraBERT[5] which involves removing emojis, tashkeel, tatweel, and HTML markup (Antoun et al., 2020).

3.2 Our Models

We pre-train both BERT- and T5-style models. JABER and SABER stand for Junior (12-layer) and Senior (24-layer) Arabic BERT models respectively. They follow the default configuration of BERT-base and BERT-large (Devlin et al., 2018) respectively. We also enhance JABER with character-level representations at the input layer, which we refer to as the Char-JABER model.

For Char-JABER, each word is represented as a sequence of characters, and we use a m-layer CNN encoder (Chiu and Nichols, 2016; Lee et al., 2018) to obtain a continuous vector of character-level representation for each word. The final input representation is obtained by adding those vectors to the original BERT input representations (token, segment, and position). Note that all sub-tokens of the same word share the same character-level representation of that word.

AT5B and AT5S use the same encoder-decoder architecture and configuration of T5-base and T5-small (Raffel et al., 2019) respectively. AT5B is directly comparable with AraT5-base (Nagoudi et al., 2022), the state-of-the-art model for Arabic generative tasks. The configurations and implementation details of our models are listed in Appendix A.2 and A.3.

4 Experimental Protocol

4.1 Datasets

We evaluate all models on the newly proposed ALUE benchmark (Seelawi et al., 2021). ALUE is a collection of eight Arabic NLU tasks: four single-sentence, two sentence-pair, and one multi-label classification task, as well as one regression task. Five of the eight ALUE tasks are sourced from Twitter data whereas six tasks involve dialectal Arabic. We refer the readers to Seelawi et al. (2021) for a detailed description of each task.

The final score is computed as the unweighted average over those tasks. ALUE is powered by a leaderboard[6] with privately-held test sets and we present a brief summary of the ALUE tasks in Table 3.

We note three potential limitations with the ALUE benchmark: (1) the size of training data and average sequence lengths across tasks are smaller when compared with GLUE (Wang et al., 2018), (2) the test set labels are public for three tasks: FID, XNLI, and MDD, and (3) development sets are not available for three tasks: FID, XNLI and MQ2Q.

Following Seelawi et al. (2021), we use the available test set as the development set for FID and

[5] https://github.com/aub-mind/arabert/blob/master/preprocess.py

[6] https://www.alue.org/leaderboard

XNLI. In order to create the development set for MQ2Q, we use a simple approach: (i) we convert the development set of another task called QQP[7] from English to Arabic using an online translation service, (ii) we pick a random sample of 2k positive and 2k negative instances. We only pick sentence pairs that do not contain any English letters to create a high-quality development set.

Unfortunately, there is no equivalent of the ALUE benchmark for Arabic generative tasks which has a leaderboard with a fixed train/dev/test split and privately-held test set. Therefore, we evaluate encoder-decoder models on a selected set of generative tasks from the ARGEN benchmark (Nagoudi et al., 2022): Text Summarization (TS), Question Generation (QG), and Question Answering (QA), where the latter is treated as a sequence-to-sequence generative task. In addition, we experiment on a single-turn dialogue task using the Empathetic Dialogue (EMD) dataset (Naous et al., 2021). See Appendix B.1 for a detailed description of the datasets, splits, and evaluation metrics.

4.2 Baselines

On one hand, we compare our JABER, Char-JABER and SABER models with the popular Arabic PLMs: Arabic-BERT (Safaya et al., 2020), AraBERT (Antoun et al., 2020), CAMeLBERT (Inoue et al., 2021), ARBERT and MARBERT (Abdul-Mageed et al., 2021). On the other hand, we evaluate our AT5S and AT5B models against the recently proposed AraT5-base (Nagoudi et al., 2022) and AraB2B (Naous et al., 2021) models. The latter is an encoder-decoder model initialized with the weights of AraBERT. CAMeLBERT and AraT5-base refer to CAMeLBERT-MIX and AraT5 models in (Inoue et al., 2021) and (Nagoudi et al., 2022) respectively. These models were pretrained on a mix of MSA and tweets (the largest possible corpus) and achieve the best overall performance in their respective papers.

4.3 Implementation Details

In order to ensure a fair comparison amongst existing models, we define a systematic evaluation protocol following the recommendations of Pineau et al. (2021). The following four-step approach is applied to every model (including the baselines) for each of the ALUE and generative tasks: **(1)** We conduct extensive hyperparameter-search experiments (e.g. 60 for BERT models) to find the best combination of batch size, learning rate, and dropout rate; **(2)** We use the best found hyperparameter-setting to perform 5 runs with different random seeds; **(3)** We report the average and the standard deviation on the development set; **(4)** We use the best-performing models of the development set experiments for the ALUE leaderboard submissions, as well as for reporting the test-set scores of the encoder-decoder models.

For BERT-style models, we use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate decay. Fixing the number of epochs to 30, we perform grid search with multiple runs to find the best hyperparameters: learning rate from {7e-6, 2e-5, 5e-5}, batch-size from {8, 16, 32, 64, 128}, hidden dropout from {0.1, 0.2, 0.3, 0.4}. For encoder-decoder models, we use the Adafactor (Shazeer and Stern, 2018) with inverse square root decay and pick a learning rate from {1e-3, 1e-4, 1e-5}. The fine-tuning code is based on the PyTorch (Paszke et al., 2019) version of the Transformers library (Wolf et al., 2020). We run all experiments on a single NVIDIA Tesla V100 GPU. The best hyperparameters for the generative and ALUE tasks can be found in Table 12 and Table 13 respectively (Appendix B).

5 Results of BERT-Style Models

5.1 ALUE Dev

The performance of all BERT-based models including the baselines on the development set of ALUE tasks is presented in Table 4. We report the average and standard deviation of 5 runs with different random seeds. We observe that the variance in performances of the multiple runs is low and is approximately the same on average for all BERT-base models, with the exception of OHSD where all models exhibit higher variance. Interestingly, Char-JABER and SABER report a lower variance across the five runs when compared to the BERT-base models.

It can be seen that Arabic-BERT and AraBERT have comparable performances (average score of 72.4% and 72.6% respectively). This could be due to the similar size of training data used by both models: Arabic-BERT was pre-trained on 95GB of text data that was duplicated 3 times (285GB), while AraBERT was pre-trained on 27GB dupli-

[7] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question

Model	MQ2Q*	MDD	SVREG	SEC	FID	OOLD	XNLI	OHSD	Avg.
Baselines									
CAMeLBERT	68.9±1.1	62.9±0.1	86.7±0.1	45.4±0.5	84.9±0.6	91.3±0.4	55.7±1.2	81.1±0.7	72.1±0.6
Arabic-BERT	73.3±0.6	61.9±0.2	83.6±0.8	42.4±0.4	83.9±0.6	88.8±0.5	66.0±0.6	79.3±1.0	72.4±0.6
AraBERT	73.5±0.5	61.1±0.3	82.3±0.9	42.2±0.6	85.2±0.2	89.7±0.4	67.2±0.4	79.9±1.8	72.6±0.6
MARBERT	69.1±0.9	63.2±0.3	88.0±0.4	47.6±0.9	84.7±0.4	91.8±0.3	63.3±0.7	83.8±1.4	73.9±0.7
ARBERT	74.7±0.1	62.5±0.2	83.5±0.6	43.9±0.6	85.3±0.3	90.5±0.5	70.8±0.5	81.9±2.0	74.1±0.6
Ours									
JABER	75.1±0.3	65.7±0.3	87.4±0.7	46.8±0.8	84.8±0.3	92.2±0.5	72.4±0.7	85.0±1.6	76.2±0.7
Char-JABER	76.8±0.2	67.3±0.2	87.5±0.3	47.8±0.4	85.7±0.2	93.3±0.1	72.7±0.3	86.4±0.5	77.2±0.3
SABER	**77.7±0.4**	**67.4±0.2**	**89.3±0.3**	**49.0±0.5**	**86.1±0.3**	**93.4±0.4**	**75.9±0.3**	**88.9±0.3**	**78.5±0.3**

Table 4: DEV performances and standard deviations over 5 runs on the ALUE benchmark. Bold entries describe the best results among all models, while underlined entries show best results among BERT-base models. * indicates that the results are on our own MQ2Q dev set.

cated 10 times (270GB). While CAMeLBERT outperforms the other baseline models on certain tasks, it achieves the lowest average score of 72.1. This is due to its poor performance on MQ2Q (68.9) and XNLI (55.7), both of which are sentence-pair classification tasks and involve MSA data.

ARBERT achieves the highest average score of 74.1% closely followed by MARBERT (73.9%). MARBERT was pre-trained on a large corpus of Arabic tweets and we observe that it performs well on tasks that involve tweet-data. The opposite holds true for ARBERT.

Our JABER model significantly outperforms the best existing baseline model (ARBERT) by 2.3% on the average ALUE score. While MARBERT performs marginally better on the SVREG and SEC tasks, JABER significantly outperforms MARBERT on all other tasks, particularly the MSA tasks – XNLI and MQ2Q – where it achieves gains of +9.1% and +6.0% respectively.

We see further improvements when the JABER model is enhanced with character representations at the input level. Char-JABER performs better than JABER on all ALUE tasks resulting in a one point jump in the average ALUE score. Moreover, it can be seen that Char-JABER outperforms MARBERT on all tasks (except on SVREG) that involve tweets and dialect data, despite not being pre-trained on tweet corpora.

Character-level information can be crucial for morphologically rich languages like Arabic, where many less frequent dialect words share the same root and meaning as more frequent MSA words. We integrate this information in an unsupervised manner into both pretraining and fine-tuning stages. We do so without adding any computational overhead and without requiring massive amounts of Twitter data (unlike MARBERT) which can be difficult to obtain for a large section of the research community.

As expected, our large SABER model outperforms all the BERT-base models on all ALUE tasks (including MARBERT on SVREG), achieving a 2.3% and 1.3% improvements on ALUE average over JABER and Char-JABER respectively. In our study, it seems that increasing the model capacity is more important than adding character level information for modelling low frequent dialect words. Nevertheless, combining both techniques may further improve performance, which we leave for future work.

5.2 ALUE Test

Table 5 shows the test performance of all BERT-based models on the ALUE leaderboard. The top two rows correspond to the baselines provided by the ALUE authors and the values are directly taken from the leaderboard. The middle and the bottom sections display the performances of our competitors' baselines and our own models respectively. We keep the baseline results private[8] since we are not the owners of these models. Figure 1 in Appendix includes a screenshot of the leaderboard from June 2022.

Interestingly, we observe that our private submission of the Arabic-BERT model achieves an average ALUE score of 69.3% which is 2.2 percentage points higher than the one available on the ALUE leaderboard. This can directly be attributed to our extensive fine-tuning protocol (described

[8]We contacted the owners of the ALUE leaderboard to submit the other baseline models in private mode.

Model	MQ2Q	MDD	SVREG	SEC	FID	OOLD	XNLI	OHSD	Avg.	DIAG
ALUE Baselines										
mBERT	83.2	61.3	33.9	14.0	81.6	80.3	63.1	70.5	61.0	19.0
Arabic-BERT	85.7	59.7	55.1	25.1	82.2	89.5	61.0	78.7	67.1	19.6
Our Private Submissions of Baselines										
AraBERT	89.2	58.9	56.3	24.5	85.5	88.9	67.4	76.8	68.4	23.5
Arabic-BERT	89.7	59.7	58.0	26.5	84.3	89.1	67.0	80.1	69.3	19.0
CAMeLBERT	89.4	61.3	69.5	30.3	85.5	90.3	56.1	80.6	70.4	11.8
ARBERT	89.3	61.2	66.8	30.3	85.4	89.5	70.7	78.2	71.4	24.3
MARBERT	83.3	61.9	75.9	36.0	85.3	92.1	64.3	78.9	72.2	12.3
Ours										
JABER	93.1	64.1	70.9	31.7	85.3	91.4	73.4	79.6	73.7	24.4
Char-JABER	92.0	66.1	74.5	34.7	86.0	92.3	73.1	83.5	75.3	**26.7**
SABER	**93.3**	**66.5**	**79.2**	**38.8**	**86.5**	**93.4**	**76.3**	**84.1**	**77.3**	26.2

Table 5: Leaderboard test results (as of 24/06/2022) of experiments on ALUE tasks and their diagnostic dataset (DIAG). Bold entries describe the best results among all models, while underlined entries show best results among BERT-base models.

in Section 4.3). Specifically, the proper tuning of the hyperparameters for our version of the Arabic-BERT model resulted in an overall improvement.

Surprisingly, we also observe that the relative ranks of the baseline models have changed drastically as compared to the dev set (Table 4). CAMeLBERT had the lowest average ALUE score of 72.1% on the dev set, but it now outperforms AraBERT and Arabic-BERT on the leaderboard test-set. Similarly, MARBERT outperforms ARBERT by 0.8% on the leaderboard while being 0.3% behind on the dev set. This happens despite our extensive hyperparameter tuning protocol and the fact that we perform multiple runs. This observation underscores the importance of having separate privately-held test sets to determine the actual state-of-the-art rankings for Arabic PLMs.

We observe that our models consistently rank at the top for both ALUE dev and test sets. JABER outperforms all other existing Arabic language models achieving an average score of 73.7%. Char-JABER outperforms JABER with a 1.6% increase in the average ALUE score. SABER expectedly further boosts the average score by 2% compared to JABER, achieving the new state-of-the-art score of 77.3% on the ALUE benchmark.

It is interesting to note that Char-JABER is able to outperform the much larger SABER model (by 0.5%) on ALUE's diagnostic data (DIAG), a dataset which is designed to capture the complex linguistic phenomena of Arabic (Seelawi et al., 2021). Moreover, it performs better than JABER on all the ALUE tasks (except MQ2Q and XNLI). Therefore, we argue that augmenting language models with character information is a worthy pursuit for Arabic NLU.

6 Results of Encoder-Decoder Models

Table 6 shows the performance of our T5 models (AT5S and AT5B) and AraT5-base (Nagoudi et al., 2022) on the development split of the ALUE tasks. Expectedly, the smaller variant AT5S achieves a lower average score. The performance of our AT5B model is very similar to that of AraT5-base with both models slightly outperforming each other on four tasks each.

Task name	AT5S	AT5B	AraT5-base
MQ2Q⋆	73.0±0.1	**73.7±0.1**	70.5±2.1
OOLD	88.4±0.2	90.0±0.4	**90.5±0.4**
OHSD	81.0±1.8	**81.2±2.1**	78.3±1.4
SVREG	75.6±1.6	78.1±2.4	**80.8±1.3**
SEC	41.3±0.5	43.8±0.7	**44.0±0.6**
FID	82.1±0.6	**83.1±0.5**	82.3±0.4
XNLI	67.9±0.3	72.2±0.4	**72.5±1.5**
MDD	63.1±0.3	**64.7±0.2**	63.6±0.2
Avg	71.5±0.7	**73.3±0.9**	73.0±1.0

Table 6: ALUE scores of Arabic T5-style models on the development set. Results on our own MQ2Q dev set are marked by a ⋆.

Moreover, comparing Table 4 and Table 6, we observe that T5-style Arabic PLMs perform worse than the BERT-based models on the same ALUE

benchmark. This is in contrast to the powerful performance of T5-models on English language tasks (Raffel et al., 2019). This observation requires further investigations, and therefore we did not submit our T5 models to the ALUE leaderboard.

Model	Dev		Test	
	EM	F1	EM	F1
AT5S	36.8±0.4	57.5±0.3	29.2	65.1
AT5B	**40.8±0.7**	**61.6±1.1**	**31.6**	**67.2**
AraT5-base	40.2±0.4	61.4±0.8	31.2	65.7
AraB2B	27.3±2.5	47.9±1.6	22.7	54.0

Table 7: F1-score and Exact Match (EM) of T5-style models on the Question Answering task.

Model	QG		EMD	
	Dev	Test	Dev	Test
AT5S	7.8±0.4	15.6	2.1±0.1	1.9
AT5B	**8.1±0.1**	**17.0**	**2.3±0.1**	**2.0**
AraT5-base	6.7±0.1	13.5	2.0±0.0	1.8
AraB2B	4.7±0.3	11.7	2.0±0.0	1.8

Table 8: BLEU score of T5-style models on the Question Generation and Empathetic Dialogue tasks.

	Rouge1	Rouge2	RougeL
WikiLingua Dev			
AT5S	24.3±1.3	9.5±0.6	21.6±1.0
AT5B	**26.1±2.8**	**10.5±1.6**	**23.2±2.5**
AraT5-base	25.0±0.2	10.0±0.0	22.4±0.2
WikiLingua Test			
AT5S	25.2	9.9	22.4
AT5B	**27.8**	**11.5**	**24.8**
AraT5-base	25.1	10.2	22.5
EASC Test			
AT5S	11.3	2.7	10.1
AT5B	**12.6**	**3.5**	**11.3**
AraT5-base	10.7	2.7	9.3

Table 9: T5-style models' performances on the Text Summarization task.

In order to perform a more meaningful evaluation, we also evaluate the Arabic T5 models on four other tasks: Empathetic Dialogue (EMD), Text Summarization (TS), Question Answering (QA) and Question Generation (QG). We present the performances of all T5-based models on QA in Table 7, on QG and EMD in Table 8 and on TS in Table 9. Note that we do not experiment with AraB2B on TS as BERT model is constrained by a maximum input length of 512.

Our AT5B model significantly outperforms AraT5-base on Question Generation and WikiLingua summarization tasks by 3.5 points and 2.7 points respectively. On the remaining QA and EMD tasks, the performance of the two models is similar with our AT5B model performing marginally better. Moreover, we observe in Table 8 that even our smaller AT5S model is able to outperform the bigger AraT5-base on QG and EMD tasks while achieving comparable scores on TS and QA tasks. This can be very useful for the community for operating in a low latency setting.

Finally, we observe from Table 7 and Table 8 that the performance of AraB2B model is worse than all other T5-based models. We believe that the BERT2BERT approach for Arabic response generation adopted in (Naous et al., 2021) is not well suited for such generation tasks, and it is preferable to pre-train the model from scratch compared to initializing the encoder-decoder architecture with pre-trained weights.

		Acceptable	Best
QG	AT5B	**68%±10**	**56%±12**
	AraT5-base	37%±11	19%±14
	AraB2B	40%±12	25%±02
EMD	AT5B	**53%±08**	**50%±07**
	AraT5-base	50%±12	37%±10
	AraB2B	27%±04	13%±05
TS	AT5B	**74%±08**	**66%±05**
	AraT5-base	61%±12	34%±04

Table 10: Human evaluation performances on 3 generative tasks.

The ideal way to measure performance on language generation tasks is to ask humans to evaluate the models' outputs (Sai et al., 2022). Thus, we evaluate our T5-based and AraB2B models on the three generation tasks of QG, EMD and TS using human annotators. Each of the three models' outputs was evaluated by four annotators. We perform both absolute and relative comparison of the three models. Specifically, we asked the annotators to label a hundred outputs from each model and each task for two scenarios: (1) Acceptable: each model output is labeled for whether it is acceptable (not strictly perfect) to the annotator or not, and (2) Best: where the annotator must pick exactly one best output out of the three ones. In order to

mitigate annotation biases, we randomly shuffle, anonymize and rename the three models' outputs.

The results of our human evaluation study for both Acceptable and Best scenarios are presented in Table 10. First, we assert that the reported values are reliable as the standard deviation is low (approximately 10%) across all tasks. Second, we observe that the scores obtained in the human evaluation study are much higher than what the corresponding BLEU and ROUGE scores reported in Table 8 would suggest. On EMD, for example, our AT5B model achieves a score of 53% for the Acceptable scenario as compared to the previously reported BLEU score of 2.0. This is possible because the dialogue generated by the model could be conveying the same tone and emotion as the reference dialogue which led the annotators to mark it as Acceptable, despite having a low n-gram overlap with the reference dialogue.

Finally, we can conclude from Table 10 that our AT5B model was preferred by the annotators for both scenarios on each of the three tasks. The improvement over AraT5-base is considerably large for QG and TS tasks as compared to the empathetic dialogues task. On EMD, we observe that only a fraction of all of the models' responses are considered acceptable by the annotators. However, even in that scenario, the annotators pick our AT5B model as the best-performing one since it is able to produce the most syntactically correct and coherent responses. One reason for the overall low performance on these tasks is the quality of the datasets available for Arabic NLP: the data is not originally in Arabic and the datasets were created via automatic translation from English datasets. Therefore, in order to make meaningful progress in Arabic NLP, we argue that the community needs to curate high-quality datasets dedicatedly for Arabic.

7 Conclusion

In this work, we revisit the pre-training and evaluation of Arabic PLMs. We introduced five new Arabic language models using both BERT- and T5-style pre-training schemes. Our models outperform all existing Arabic models on the generative tasks as well as on the ALUE benchmark, with SABER setting a new state-of-the-art on ALUE.

In order to accelerate the progress in Arabic NLU, we advocate for the creation of more *leaderboard-based* benchmarks with privately-held evaluation sets that cover a wide array of tasks. Moreover, we strongly recommend researchers follow a systematic approach similar to the one we propose when evaluating Arabic models, with extensive hyperparameter tuning and multiple runs with different random seeds.

In the future, our research will mainly focus on scaling up Arabic PLMs to tens (and hundreds) of billions of parameters in an energy-efficient manner (Du et al., 2021; Chowdhery et al., 2022) as well as scaling up with high-quality pre-training data (Hoffmann et al., 2022). Having met all the other conditions in the Reproducibility Checklist (Pineau, 2020), we make our source code and models freely available at https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/JABER-PyTorch.

Limitations

While we evaluated our models on a diverse set of classification and generative tasks, there are several NLP tasks that were not accounted for in our study. It would be worthwhile to explore other tasks such as named-entity recognition (Benajiba and Rosso, 2007) or coreference resolution (Pradhan et al., 2012). Also, there are other Arabic PLMs (Talafha et al., 2020; Lan et al., 2020) that were not used in our evaluation study. Those models have been reported to underperform the PLMs we have considered as baselines in our study. However, there is a small chance that including them might change the performance ranking in our evaluation.

As the focus of this study is on overall benchmark performances, we did not assess the robustness of our models on out-of-domain datasets. Finally, our study lacks a qualitative exploration of the datasets and models' error analyses, which we leave for future work. In particular, we wish to explore the impressive performance of Char-JABER on ALUE's diagnostic data.

Acknowledgments

We thank Mindspore,[9] a new deep learning computing framework, for the partial support of this work. We are also thankful to the anonymous reviewers for their insightful comments.

[9] https://www.mindspore.cn/

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Giuseppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Yassine Benajiba and Paolo Rosso. 2007. Anersys 2.0: Conquering the ner task for the arabic language by combining the maximum entropy with pos-tag information. In *IICAI*, pages 1814–1823.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

José Canete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. *PML4DC at ICLR*, 2020.

Rishav Chakravarti, Anthony Ferritto, Bhavani Iyer, Lin Pan, Radu Florian, Salim Roukos, and Avi Sil. 2020. Towards building a robust industry-scale question answering system. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 90–101, Online. International Committee on Computational Linguistics.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. *arXiv preprint arXiv:2010.10906*.

Yanran Chen, Jonas Belouadi, and Steffen Eger. 2022. Reproducibility issues for bert-based evaluation metrics. *arXiv preprint arXiv:2204.00004*.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Trans. Assoc. Comput. Linguistics*, 4:357–370.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Trans. Assoc. Comput. Linguistics*, 10:73–91.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4443–4458. Association for Computational Linguistics.

Luca Di Liello, Matteo Gabburo, and Alessandro Moschitti. 2021. Efficient pre-training objectives for transformers.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2021. Glam: Efficient scaling of language models with mixture-of-experts.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2010. Using mechanical turk to create a corpus of arabic summaries.

Ibrahim Abu El-Khair. 2016. 1.5 billion words Arabic Corpus. *arXiv preprint arXiv:1611.04033*.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of nlp leaderboards. *arXiv preprint arXiv:2009.13888*.

Daniela Gerz, Ivan Vulic, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the*

2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 316–327. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop, WANLP 2021, Kyiv, Ukraine (Virtual), April 9, 2021*, pages 92–104. Association for Computational Linguistics.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. AMMUS: A survey of transformer-based pretrained models in natural language processing.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in nlp. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124.

Yoon Kim, Yacine Jernite, David A. Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749. AAAI Press.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. 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)*, pages 4727–4734.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490.

Chanhee Lee, Young-Bum Kim, Dongyub Lee, and Heui-Seok Lim. 2018. Character-level feature extraction with densely connected networks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3228–3239.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, et al. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184.

Guangjun Li, Xianzhi Wang, and Minxi Li. 2021. A review of recent trends and industry prospects for artificial intelligence technologies. In *2021 8th International Conference on Behavioral and Social Computing (BESC)*, pages 1–7. IEEE.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 39–50, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

S. Mehri, M. Eric, and D. Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *ArXiv*, abs/2009.13570.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Tariq Alhindi. 2020. Machine generation and detection of Arabic manipulated and fake news. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 69–84, Barcelona, Spain (Online). Association for Computational Linguistics.

Tarek Naous, Wissam Antoun, Reem Mahmoud, and Hazem Hajj. 2021. Empathetic BERT2BERT conversational model: Learning Arabic language generation with little data. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 164–172, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Tarek Naous, Christian Hokayem, and Hazem Hajj. 2020. Empathy-driven Arabic conversational chatbot. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 58–68, Barcelona, Spain (Online). Association for Computational Linguistics.

Subhadarshi Panda, Anjali Agrawal, Jeewon Ha, and Benjamin Bloch. 2021. Shuffled-token detection for refining pre-trained RoBERTa. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 88–93, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Joelle Pineau. 2020. Machine learning reproducibility checklist v2.0. `https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf`.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research: a report from the neurips 2019 reproducibility program. *Journal of Machine Learning Research*, 22.

Yuval Pinter, Amanda Stent, Mark Dredze, and Jacob Eisenstein. 2021. Learning to look inside: Augmenting token-based encoders with character-level information.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. 2021. Alue: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184.

Alexander Sergeev and Mike Del Balso. 2018. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799*.

Tatiana Shavrina, Alena Fenogenova, Anton Emelyanov, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. Russiansuperglue: A russian language understanding evaluation benchmark. *arXiv preprint arXiv:2010.15925*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8968–8975.

Bashar Talafha, Mohammad Ali, Muhy Eddin Za'ter, Haitham Seelawi, Ibraheem Tuffaha, Mostafa Samir, Wael Farhan, and Hussein Al-Natsheh. 2020. Multi-dialect arabic bert for country-level dialect identification. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 111–118.

Lewis Tunstall, Leandro von Werra, and Thomas Wolf. 2022. *Natural Language Processing with Transformers*. " O'Reilly Media, Inc.".

Matej Ulčar and Marko Robnik-Šikonja. 2020. Finest bert and crosloengual bert: less is more in multilingual models. *arXiv preprint arXiv:2006.07890*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Junqiu Wei, Qun Liu, Yinpeng Guo, and Xin Jiang. 2021. Training multilingual pre-trained language model with byte-level subwords. *arXiv preprint arXiv:2101.09469*.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, X. Li, Zhi Yuan Lim, S. Soleman, R. Mahendra, Pascale Fung, Syafri Bahar, and A. Purwarianti. 2020. Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*.

Liang Xu, Xiaojing Lu, Chenyang Yuan, Xuanwei Zhang, Hu Yuan, Huilin Xu, Guoao Wei, Xiang Pan, and Hai Hu. 2021. Fewclue: A chinese few-shot learning evaluation benchmark. *arXiv preprint arXiv:2107.07498*.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*.

Imad Zeroual, Dirk Goldhahn, Thomas Eckart, and Abdelhak Lakhouaja. 2019. Osian: Open source international arabic news corpus-preparation and integration into the clarin-infrastructure. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 175–182.

A Pretraining Details

A.1 Filtering Heuristics

1. Remove sentences with HTML or Javascript code (Raffel et al., 2019).
2. Remove sentences if it has less than 70% Arabic characters.
3. Remove sentences with less than 8 words.
4. Remove sentences with more than 3 successive punctuation marks (excluding dot).
5. Remove documents with less than 64 words.
6. Remove long spans of non-Arabic text (mostly English) inside a sentence. We observe that most of these sentences were gibberish/noisy text and not related to the original content.
7. Represent each sentence by the concatenation of the first and last 3 words. We only consider words that did not include any digits and were longer than 3 characters. Then, we deduplicate the corpus by only keeping the first occurrence of sentences with the same key.
8. Discard a document if more than 30% of its sentences are removed in our filtering pipeline.

A.2 BERT-style Models

For tokenization, we use the byte-level Byte Pair Encoding (BBPE) (Wei et al., 2021) training method which considers the text as a byte sequence. This method improves the learning of the representations of rare words and eliminates the out-of-vocabulary problem. We use a vocabulary size of 64K which is comparable to that of AraBERT, twice the size of Arabic-BERT and CAMeLBERT, and 36% smaller than ARBERT and MARBERT. Our JABER and SABER models use the same architecture as that of BERT-base and BERT-large (Devlin et al., 2018) respectively. The former is a stack of 12 Transformer-encoder layers (768 hidden units) while the latter consists of 24 Transformer-encoder layers (1024 hidden units).

For Char-JABER, we first randomly initialize a character embedding lookup table with a vocab size of 160 characters (induced from the pre-training corpus) and a 768 hidden size. Each word is split into a sequence of characters with a maximum character sequence length of 10. We use two 1-D CNN layers with each layer having a filter size of 348 and a sliding window of size 3. Note that we apply a maxpooling layer with a window size of 5 after the first CNN layer. After the second CNN layer, a linear layer is used to map the final representation to the 768 hidden size. Although this architecture adds an additional 700K parameters to Char-JABER, this has a negligible computational overhead on JABER.

Following Devlin et al. (2018), we pre-train our models on two unsupervised tasks: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). Specifically for MLM, we use whole word masking with a probability of 15%. The original tokens are replaced 80% of the time with the special [MASK] token, 10% of the time by a random token, and remains unchanged 10% of the time. We choose a duplication factor of 3: each input sequence generates 3 random sets of masked tokens.

We pre-train our JABER and SABER models on 16 servers for 15 and 5 epochs respectively. Each server constitutes 8 NVIDIA Tesla V100 GPUs with 32GB of memory. The distributed training is achieved through Horovod (Sergeev and Del Balso, 2018) with full precision. We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate decay setting the initial learning rate to 1e-4 with 10,000 warm-up steps. We train with a maximum sequence length of 128, and set the per-GPU batch size to 64 for JABER and 32 for SABER. It takes approximately 16 and 32 hours to conclude one epoch for JABER and SABER respectively. Finally, the pre-training setting of our Char-JABER model is identical to that of JABER with the exception of using a smaller initial learning rate of 5e-5.

A.3 Encoder-Decoder Models

Our text-to-text Transformer models AT5B and AT5S use the same encoder-decoder architecture as T5-base and T5-small (Raffel et al., 2019) respectively. The encoder and decoder components of T5-base have similar configuration as that of BERT-base (12 layers) while T5-small is a smaller model with only 6 layers and 8-headed attention. We use the self-supervised denoising objective (Raffel et al., 2019) to pre-train our models. Specifically, 15% of tokens are randomly dropped-out from the input and all consecutive spans of such tokens are replaced by a single sentinel token. The

expected output is a sequence of these dropped-out tokens separated by the corresponding sentinel token. We train our T5-style models using the same vocabulary and pre-training corpus as that of our BERT-style models.

The models are pre-trained on 64 GPU clusters for 200k steps. The pre-training code is based on the PyTorch (Paszke et al., 2019) version of the Transformers library (Wolf et al., 2020). The distributed training is achieved by PyTorch's native distributed training capabilities. We use the Adafactor optimizer (Shazeer and Stern, 2018) with an initial learning rate of 1 and inverse square-root decay until the end of pre-training.

For both AT5S and AT5B, the maximum sequence length is set to 512 for the encoder and 114 for the decoder. We use a per-GPU batch size of 56 and 16 for AT5S and AT5B respectively (the maximum batch size that can fit on a single GPU). It is important to note that most of our implementation choices (learning rate, optimizer, etc.) are adopted from Raffel et al. (2019) and Nagoudi et al. (2022). We only differ from AraT5 through the use of a different pre-training corpus and vocabulary.

B Fine-tuning Details

B.1 Generative Tasks Datasets

While there is no equivalent for ALUE for generative tasks, Nagoudi et al. (2022) recently introduced the ARGEN benchmark for Arabic natural language generation composed of 7 tasks and 19 datasets. Besides the lack of a public leaderboard and private test sets, there are certain issues with this benchmark. Some datasets are not available publicly (e.g. ARGEN$_{NTG}$), and in some cases, the exact data split is not made public (e.g. ARGEN$_{TS}$). Therefore, we only consider three ARGEN tasks in our evaluation: Question Generation (QG), Question Answering (QA), and Text Summarization (TS). We did not include any tasks that involved non-Arabic text (e.g. translation) since we restrict the scope of this work to a monolingual setting.

Furthermore, we also evaluate our models on the EMpathetic Dialogues (EMD) dataset (Naous et al., 2020), which is an Arabic conversational dataset of empathetic conversations curated by translating its English counterpart (Rashkin et al., 2019). Table 11 shows the number of instances in the train/dev/test splits for each dataset. The data collection process and evaluation metrics are adopted from (Nagoudi et al., 2022; Naous et al., 2021). Specifically, we use ROUGE (Lin, 2004) to evaluate models on the TS task and BLEU (Papineni et al., 2002) for QA, QG and EMD tasks.

Task	|Train|	|Dev|	|Test|
TS	23.4k	2.9k	2.9k/153
EMD	19.5k	2.8K	2.5k
QG/QA	101.6k	517	11.6k

Table 11: Train/Dev/test sizes of the datasets used to evaluate encoder-decoder models. Note that the test set for TS consists of 2.9K articles from WikiLingua (Ladhak et al., 2020) and 153 articles from Essex Arabic Summaries Corpus (EASC) (El-Haj et al., 2010).

We adopt the generative format for QA where the input is a pair of passage and question text, and the model is expected to generate the answer. Following Nagoudi et al. (2022), we use the same dataset for QG as QA except that the input now is a pair of passage and answer text, and the model must generate the corresponding question. Note that for the summarization task, Nagoudi et al. (2022) did not publish the exact split they used on WikiLingua (Ladhak et al., 2020). We create our own splits by first randomly shuffling the dataset (with seed = 42) and then splitting with the same proportions of 80% train, 10% dev and 10% test. We will make the code publicly available to reproduce our splits and empirical results.

It is important to mention that the performance scores obtained from our re-implementations on TS and EMD tasks are significantly lower than the original scores reported in (Nagoudi et al., 2022) and (Naous et al., 2021). This is due to errors in the original implementations. For TS, we found a major error in the calculation of the ROUGE score as the ROUGE tool used by the authors was incompatible with Arabic. For EMD, we found the original BLEU scores to be inflated as the authors compute it on segmented text and not at the word-level (after de-segmentation).

Rank	Name	Model	Details	Score	MQ2Q	MDD	SVREG	SEC	FID	OOLD	XNLI	OHSD	DIAG	Public
1	Huawei Noah's Ark Lab MTL	SABER		77.3	93.3	66.5	79.2	38.8	86.5	93.4	76.3	84.1	26.2	
2	Huawei Noah's Ark Lab MTL	Char-JABER		75.3	92.0	66.1	74.5	34.7	86.0	92.3	73.1	83.5	26.7	
3	Huawei Noah's Ark Lab MTL	JABER		73.7	93.1	64.1	70.9	31.7	85.3	91.4	73.4	79.6	24.4	
4	Huawei Noah's Ark Lab MTL	MARBERT		72.2	83.3	61.9	75.9	36.0	85.3	92.1	64.3	78.9	12.3	
5	Huawei Noah's Ark Lab MTL	ARBERT		71.4	89.3	61.2	66.8	30.3	85.4	89.5	70.7	78.2	24.3	
6	Huawei Noah's Ark Lab MTL	CAMeLBERT-MIX		70.4	89.4	61.3	69.5	30.3	85.5	90.3	56.1	80.6	11.8	
7	Huawei Noah's Ark Lab MTL	ARABIC-BERT		69.3	89.7	59.7	58.0	26.5	84.3	89.1	67.0	80.1	19.0	
8	Huawei Noah's Ark Lab MTL	ARABERTv0.1-base		68.4	89.2	58.9	56.3	24.5	85.5	88.9	67.4	76.8	23.5	

Figure 1: Screenshot of ALUE leaderboard as by 13/10/2022. Red buttons indicate our private submission baselines which are not visible to the public.

Model	**QA**	**QG**	**EMD**	**TS**
	AraB2B			
batch size	16	16	32	-
hidden dropout	0.1	0.2	0.1	-
learning rate	1e-03	1e-03	1e-03	-
	AraT5-base			
batch size	32	8	8	4
hidden dropout	0.2	0.2	0.1	0.1
learning rate	1e-03	1e-03	1e-03	1e-03
	AT5S			
batch size	32	16	32	4
hidden dropout	0.1	0.2	0.1	0.1
learning rate	1e-03	1e-03	1e-03	1e-03
	AT5B			
batch size	16	32	16	4
hidden dropout	0.1	0.1	0.1	0.2
learning rate	1e-03	1e-03	1e-03	1e-03

Table 12: Best hyperparameters for Arabic encoder-decoder models on the generative tasks.

Model	MQ2Q	MDD	SVREG	SEC	FID	OOLD	XNLI	OHSD
				Arabic-BERT				
batch size	64	16	16	16	32	32	64	16
hidden dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
learning rate	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05
				AraBERT				
batch size	128	32	8	8	8	32	32	16
hidden dropout	0.1	0.1	0.2	0.1	0.1	0.1	0.3	0.1
learning rate	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05
				CAMeLBERT				
batch size	16	8	8	32	8	128	32	8
hidden dropout	0.2	0.2	0.2	0.1	0.2	0.1	0.1	0.1
learning rate	5e-05	2e-05	2e-05	5e-05	2e-05	2e-05	2e-05	2e-05
				ARBERT				
batch size	64	16	32	8	32	128	32	32
hidden dropout	0.1	0.1	0.3	0.3	0.1	0.1	0.1	0.3
learning rate	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	7e-06
				MARBERT				
batch size	64	64	16	8	64	64	64	64
hidden dropout	0.3	0.2	0.1	0.3	0.1	0.2	0.2	0.1
learning rate	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05
				JABER				
batch size	64	32	8	16	32	128	16	32
hidden dropout	0.3	0.2	0.1	0.1	0.1	0.2	0.1	0.3
learning rate	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	2e-05	7e-06
				Char-JABER				
batch size	64	32	32	16	8	32	64	16
hidden dropout	0.1	0.2	0.1	0.2	0.2	0.2	0.2	0.1
learning rate	7e-06	2e-05	2e-05	2e-05	2e-05	7e-06	2e-05	7e-06
				SABER				
batch size	32	32	8	8	32	32	32	32
hidden dropout	0.1	0.1	0.2	0.2	0.3	0.2	0.2	0.1
learning rate	7e-06	2e-05	7e-06	2e-05	2e-05	7e-06	7e-06	7e-06
				AT5S				
batch size	16	32	8	16	16	16	8	32
hidden dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
learning rate	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03
				AT5B				
batch size	8	16	16	16	8	16	8	64
hidden dropout	0.2	0.2	0.1	0.1	0.1	0.1	0.1	0.1
learning rate	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03
				AraT5-base				
batch size	64	64	16	64	32	64	32	8
hidden dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
learning rate	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03	1e-03

Table 13: Best Hyperparameters for Arabic BERT-based and T5-based models on all ALUE tasks.