AraMUS: Pushing the Limits of Data and Model Scale for Arabic Natural Language Processing

Asaad Alghamdi[1],[*] Xinyu Duan[2],[*] Wei Jiang[2] Zhenhai Wang[2] Yimeng Wu[3]
Qingrong Xia[2] Zhefeng Wang[2] Yi Zheng[2] Mehdi Rezagholizadeh[3] Baoxing Huai[2]
Peilun Cheng[1] Abbas Ghaddar[3]

[1] AI Cognitive Team, Tonomus
[2] Huawei Cloud Computing Technologies Co., Ltd.
[3] Huawei Technologies Co., Ltd.

{asaad.alghamdi,eddie.chengpeilun}@neom.com
{duanxinyu,jiangwei160,wangzhenhai1,yimeng.wu,xiaqingrong,wangzhefeng,
zhengyi29,mehdi.rezagholizadeh,huaibaoxing,abbas.ghaddar}@huawei.com

Abstract

Developing monolingual large Pre-trained Language Models (PLMs) is shown to be very successful in handling different tasks in Natural Language Processing (NLP). In this work, we present AraMUS, the largest Arabic PLM with 11B parameters trained on 529GB of high-quality Arabic textual data. AraMUS achieves state-of-the-art performances on a diverse set of Arabic classification and generative tasks. Moreover, AraMUS shows impressive few-shot learning abilities compared with the best existing Arabic PLMs.

1 Introduction

Scaling-up Pre-trained Language Models (PLMs) has led to astonishing performance gains on a vast variety of Natural Language Processing (NLP) tasks (Du et al., 2021; Zoph et al., 2022; Smith et al., 2022). It has also opened new perspectives for studying the opportunities and limitations of large PLMs (Raffel et al., 2019; Dale, 2021; Bommasani et al., 2021), as well as their social and ethical impacts (Bender et al., 2021; Weidinger et al., 2021; Tamkin et al., 2021; Rae et al., 2021a; Susnjak, 2022).

Although for some languages such as English and Chinese, several PLMs with even more than hundred billions of parameters have been developed (Rae et al., 2021b; Chowdhery et al., 2022; Zeng et al., 2021; Sun et al., 2021), little or no progress has been made on this direction for many other languages including Arabic.[1] While there have recently been few attempts to develop multi-billion parameters Arabic PLMs (Nagoudi et al., 2022a; Antoun et al., 2021b; Lakim et al., 2022), still, their performances and abilities have not been well investigated. The largest well-studied Arabic PLM has no more than 370M parameters (Nagoudi et al., 2022b; Ghaddar et al., 2022).

In this work, we introduce AraMUS, an 11B parameter encoder-decoder T5 (Raffel et al., 2019) model, which is pre-trained on 529GB of high-quality Arabic text (filtered out of 8.8TB). To the best of our knowledge, AraMUS is the largest Arabic PLM in terms of pre-training data and model size. Furthermore, it is the first time a multi-billion parameter Arabic PLM is *systematically* evaluated, against the existing state-of-the-art models, on a diversified set of discriminative and generative task models. More precisely, AraMUS achieves new state-of-the-art performances of 79.8% on the ALUE (Seelawi et al., 2021) benchmark, which is a collection of 8 discriminative tasks. In addition, it significantly outperforms the best available encoder-decoder models on multiple generative tasks. Finally, AraMUS shows remarkable abilities to maintain its performance under few-shot settings.

2 Related Work

Recently, there has been a growing body of the literature on very large-scale English PLMs by thoroughly studying different aspects of their scaling. These efforts can be summarized into scaling their pre-training data (Hoffmann et al., 2022) and model size (Dale, 2021; Rae et al., 2021b; Smith et al., 2022), designing efficient architectures (Zoph et al., 2022; Chowdhery et al., 2022) and pre-training objectives (Bajaj et al., 2022; Tay et al., 2022), democratizing their access (Zhang et al., 2022), and making them useful in real-world applications (Ouyang et al., 2022; Qu et al., 2023). Besides English, there have been multiple attempts to develop multilingual (Scao et al., 2022), as well

[*] Equal contribution

[1] Arabic is among top 10 most popular languages in the world with 420M native speakers, and more than 25 popular dialects (Guellil et al., 2021).

Findings of the Association for Computational Linguistics: ACL 2023, pages 2883–2894
July 9-14, 2023 ©2023 Association for Computational Linguistics

as non-Anglocentric (Zeng et al., 2021; Sun et al., 2021; Shin et al., 2022) multi-billion PLMs.

Unfortunately, the development of Arabic PLMs does not follow the same pace as that of English. The earliest released Arabic PLMs (Antoun et al., 2020; Safaya et al., 2020) were based on the BERT-*base* (as well as *-large*) architecture (Devlin et al., 2018) and pre-trained on less than 100GB of unfiltered data. Successive works tried to improve Arabic BERT-base models performance by scaling up the pre-training data up to 197GB and 167GB of unfiltered Arabic text for MARBERT (Abdul-Mageed et al., 2021) and CAMeLBERT (Inoue et al., 2021) respectively. In addition, other works focused on developing Arabic PLMs to support other architectures like AraElectra (Antoun et al., 2021a), AraGPT (Antoun et al., 2021b), AraT5 (Nagoudi et al., 2022b), and AraBART (Eddine et al., 2022) which are equivalent to English ELECTRA (Clark et al., 2020), GPT (Radford et al., 2018), T5 (Raffel et al., 2019), and BART (Lewis et al., 2019) respectively.

Recently, Ghaddar et al. (2022) developed state-of-the-art Arabic BERT (JABER and SABER) and T5 models (AT5S and AT5B) by improving the pre-training data quantitatively and qualitatively. More precisely, they pre-trained Arabic BERT-base/large and T5-small/base models on 115GB of high-quality Arabic text data (filtered out of 514GB). AraGPT-Mega (Antoun et al., 2021b), Jasmine (Nagoudi et al., 2022a), NOOR (Lakim et al., 2022) are the only existing multi-billion Arabic PLMs. These are decoder-only GPT models with 1.5B, 6.7B, and 10B parameters respectively. However, these aforementioned works suffer from the absent (e.g. in AraGPT, NOOR) or limited (e.g. Jasmine) comprehensive evaluation on NLP end-tasks. Moreover, some of these models (such as NOOR and Jasmine) are not publicly available for custom evaluations.[2] Evaluation is a key factor for understanding the strengths and limitations of these models, without which the progress of the Arabic NLP field is hindered.

3 AraMUS

3.1 Pre-training Data

We mainly leverage all (up to July 2022) of the 90 Common Crawl [3] monthly web scrapes in order to collect massive amount of Arabic textual data. This is significantly larger compared to JABER (Ghaddar et al., 2022), NOOR (Lakim et al., 2022), and Jasmine (Nagoudi et al., 2022a), which use 10, 21, and 71 monthly CC shards, respectively. Then, we apply aggressive noise filtering and deduplication, which give rise to 529GB of high-quality Arabic text data. Nagoudi et al. (2022a) introduced the closest comparable pre-training corpus size to us with 413GB (22% smaller than ours) of Arabic text data. Our data mainly differs in using 2.5 times more CC data, while they used 3.8 times more dialect data than ours. We refer the reader to Appendix A.1 for technical details regarding the pre-training data collection.

3.2 Model and Implementation

AraMUS follows the same encoder-decoder architecture and configuration as T5-xxl (Raffel et al., 2019) model with 64k vocabulary size. We choose encoder-decoder T5 architecture because it was found to deliver a good balance between the performance of the discriminative and generative tasks (Raffel et al., 2019; Tay et al., 2022), compared to encoder-only BERT (discriminative tasks focused) and decoder-only GPT (Radford et al., 2019) (generative tasks focused). AraMUS has 11B parameters in total, which makes it the largest existing Arabic T5 model. It was pre-trained using 128 NVIDIA A100 GPUs for 2 months. Technical details regarding implementation and hyper-parameters used for pre-training are listed in Appendix A.2.

3.3 Evaluation Protocol

We assess AraMUS by performing extensive fine-tuning experiments on a diverse set of NLP tasks. On one side, we experiment on 8 tasks from the well-established ALUE benchmark (Seelawi et al., 2021), which includes one regression (SVREG), one multi-label classification (SEC), 4 single-sentence (MDD, FID, OOLD, and OHSD) and 2 sentence-pair (MQ2Q and XNLI) classification tasks. On the generative tasks side, we evaluate on Question Answering (QA), Question Generation (QG), and Text Summarization (TS).

We compare AraMUS with state-of-the-art Arabic PLMs in the literature, including ARBERT, MARBERT, JABER (BERT-base), SABER, ALM-1.0 (BERT-large), AT5B and AraT5-base (T5-base). The experimental protocol is designed to ensure the diversity of the tasks, and the public availability of models. Most importantly, we make sure that

[2] We refer the reader to Appendix B.2 for detailed positioning of AraMUS against each of these three models.

[3] https://commoncrawl.org

Model	**#Params**	**MQ2Q**	**MDD**	**SVREG**	**SEC**	**FID**	**OOLD**	**XNLI**	**OHSD**	**Avg.**
					BERT-models					
ARBERT	163M	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
MARBERT	163M	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
JABER	135M	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
SABER	369M	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
					T5-models					
AT5B	296M	73.7±0.1	64.7±0.2	78.1±2.4	43.8±0.7	83.1±0.5	90.0±0.4	72.2±0.4	81.2±2.1	73.3±0.9
AraT5-base	289M	70.5±2.1	63.6±0.2	80.8±1.3	44.0±0.6	82.3±0.4	90.5±0.4	72.5±1.5	78.3±1.4	73.0±1.0
AraMUS	11B	**80.7±0.1**	**68.0±0.2**	**89.8±0.3**	**49.6±0.7**	**86.6±0.4**	**93.8±0.4**	**82.9±0.2**	88.2±1.0	**79.9±0.2**

Table 1: Dev set performances and standard deviations over 5 runs on the ALUE benchmark.

Model	**#Params**	**MQ2Q**	**MDD**	**SVREG**	**SEC**	**FID**	**OOLD**	**XNLI**	**OHSD**	**Avg.**	**DIAG**
JABER	135M	93.1	64.1	70.9	31.7	85.3	91.4	73.4	79.6	73.7	24.4
ALM-1.0	350M	94.5	65.1	70.1	35.3	86.0	91.7	77.7	85.7	75.8	30.2
SABER	369M	93.3	66.5	79.2	38.8	86.5	93.4	76.3	84.1	77.3	26.2
AraT5-base	282M	91.3	63.8	65.9	30.5	82.3	88.8	68.2	77.9	71.1	15.4
AraMUS	11B	**95.2**	**67.5**	**80.4**	**41.6**	**87.2**	**95.5**	**83.2**	**87.4**	**79.8**	**42.0**

Table 2: Results of top-ranked models on the ALUE leaderboard.

datasets are of high quality, open-sourced, and supported by a well-established evaluation protocol. Our goal is to have a fair comparison between models, as well as the credibility and reproducibility of the results. A detailed description of fine-tuning datasets, evaluation metrics, baselines, and implementation details are available in Appendix B.

3.4 Results

Table 1 shows the dev set results of the eight ALUE tasks with their average scores and standard deviations of 5 runs. The baseline results are directly brought from (Ghaddar et al., 2022) and they are directly comparable with AraMUS since we follow the same evaluation protocol. Table 2 shows the test set performances of the state-of-the-art models on the ALUE leaderboard.

As we expect, AraMUS outperforms all other baseline models on both dev and test sets and achieves a new state-of-the-art performances on ALUE. While our average ALUE result is 1.4% better than the best baseline, SABER, the latter outperforms AraMUS on the OHSD dataset. On the other hand, AraMUS significantly outperforms SABER by 2.5% on average and 3.3% on OHSD when comparing results on the leaderboard test. Interestingly, this is roughly a similar performance gap (2.1%) on the English GLUE (Wang et al., 2018) between the English T5-xxl (Raffel et al., 2019) (11B parameters) and the well-trained English Roberta-large (Liu et al., 2019) model.

Moreover, we observe a huge gap of 13.8% between AraMUS and SABER on the ALUE diagnostic set. DIAG was specifically designed to evaluate models' abilities to capture complex linguistic phenomena in Arabic (Seelawi et al., 2021). These observations clearly indicate that scaling the model with more data and parameters greatly improves the robustness and generalization abilities of Arabic PLMs. It is worth mentioning that our results are in contrast with previous observations reported in (Nagoudi et al., 2022b; Ghaddar et al., 2022) that encoder-decoder T5 architecture Arabic models (e.g. AraT5-base and AT5B) significantly underperform BERT models on discriminative tasks. Our results suggest that, for Arabic, encoder-decoder models require more data and parameters to catch up with encoder-only models on discriminative tasks.

	Dev		**Test**	
Model	**EM**	**F1**	**EM**	**F1**
AraT5-base	40.2±0.4	61.4±0.8	31.2	65.7
AT5B	40.8±0.7	61.6±1.1	31.6	67.2
AraMUS	**49.8±1.1**	**69.1±0.9**	**35.3**	**72.3**

Table 3: F1-score and Exact Match (EM) scores of T5-style models on the Question Answering (QA) task.

We further validate the performance of AraMUS by conducting an extensive set of experiments on the ALUE benchmark under few-shot setting. Figure 1 shows AraMUS and the best publicly avail-

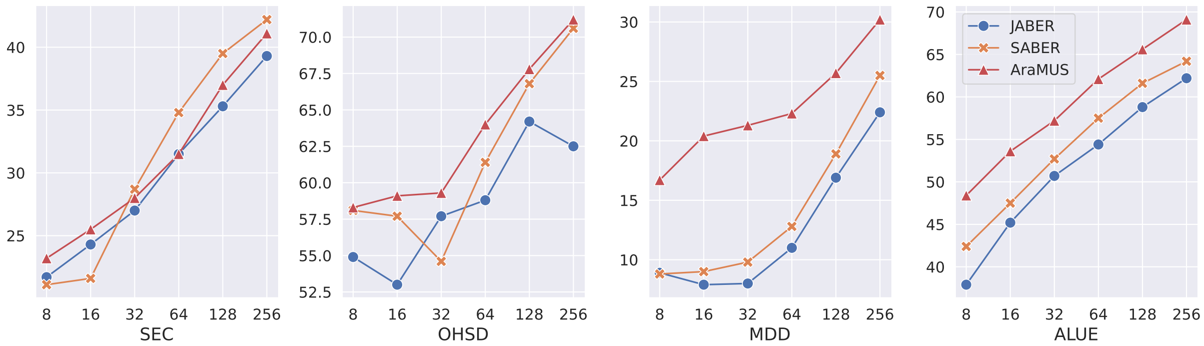

Figure 1: Models performance on the dev set of 3 ALUE tasks and the ALUE average score in the few-shot setting.

able Arabic PLMs (JABER and SABER) performances on 3 representative ALUE tasks (see the full results in Table 7 of Appendix C) and the average ALUE score. The 3 selected tasks are: SEC because it shows specific results; OHSD since with FID and OOLD they show similar result patterns, and MDD as a representative of trends observed for tasks MQ2Q, SVREG, and XNLI.

	Rouge1	Rouge2	RougeL
	WikiLingua Dev		
AraT5-base	25.0±0.2	10.0±0.0	22.4±0.2
AT5B	26.1±2.8	10.5±1.6	23.2±2.5
AraMUS	**30.5±0.1**	**13.2±0.1**	**26.9±0.1**
	WikiLingua Test		
AraT5-base	25.1	10.2	22.5
AT5B	27.8	11.5	24.8
AraMUS	**30.9**	**13.5**	**27.1**
	EASC Test		
AraT5-base	10.7	2.7	9.3
AT5B	12.6	3.5	11.3
AraMUS	**16.1**	**6.7**	**13.3**

Table 4: T5-style models' performances on the Text Summarization task.

First, we notice that exceptionally on SEC, AraMUS performs on par with JABER and underperforms SABER on many data points. We think that this is because the text-to-text approach is not effective for multi-label classification tasks under a few-shot setting. Second, we observe that AraMUS has a marginal gain compared to the best baseline (SABER) on some tasks like OHSD, e.g. 0.2%, 1.0% and 6.0% on 8, 128, and 256 examples respectively. As for the remaining 4 tasks (represented by MDD), we observe that AraMUS significantly outperforms both baselines by a large margin. Overall, AraMUS shows a consistent performance gain between 4% to 6% when averaging the results on the 8 ALUE tasks compared to SABER.

Model	**Dev**	**Test**
AraT5-base	6.7±0.1	13.5
AT5B	8.1±0.1	17.0
AraMUS	**8.6±0.1**	**17.4**

Table 5: Question Generation dev and test sets BLEU score of T5-style models.

Finally, we assess the text generation abilities of AraMUS by experimenting on 3 generative tasks in Table 3, 4 and 5. Overall, the observations are consistent with the results obtained on ALUE, AraMUS reports the highest scores on all tasks and across all metrics. More precisely, AraMUS significantly outperforms AT5B, the state-of-the-art Arabic T5-base model, by 7.5% and 5.1% on QA F1 score dev and test sets respectively. Similarly, AraMUS has a gain of 4.4%, 4.1%, and 3.5% on TS dev, test, and EASC test rouge1 score respectively. However, gains are not always significant on generative tasks, as we observe a smaller margin of improvement of 0.5% and 0.4% and against the best baseline on QG dev and test sets respectively.

4 Conclusion

In this paper, we introduced AraMUS which is not only the largest Arabic PLM in terms of pretraining data and model size, but also the first multi-billion Arabic PLM to be extensively evaluated on a wide range of NLP tasks. Since our work gives clues on the benefits and limitations of scaling up data and model sizes, we hope that it will pave the way for the Arabic NLP community to focus on problems that are beyond the reach of PLM scaling.

Limitations

While our model shows state-of-the-art results on many discriminative and generative tasks, we can think of the following main caveats of our work. First, the number of generative tasks that we evaluate on is relatively small especially when consider that AraMUS is text-to-text encoder-decoder model. This is mainly because of the rarity of Arabic generative datasets that are at the same time well-established and open-source. Second, it would be important to study how end-tasks performances is impacted when ablating the model size (e.g. 1-6 billion parameters models), pretraining data quantity or/and quality.

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.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021a. AraELECTRA: Pre-training text discriminators for Arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021b. Aragpt2: Pre-trained transformer for arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207.

Payal Bajaj, Chenyan Xiong, Guolin Ke, Xiaodong Liu, Di He, Saurabh Tiwary, Tie-Yan Liu, Paul Bennett, Xia Song, and Jianfeng Gao. 2022. Metro: Efficient denoising pretraining of large scale autoencoding language models with model generated signals. *arXiv preprint arXiv:2204.06644*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

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*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Robert Dale. 2021. Gpt-3: What's it good for? *Natural Language Engineering*, 27(1).

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*.

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.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. Arabart: a pretrained arabic sequence-to-sequence model for abstractive summarization. *arXiv preprint arXiv:2203.10945*.

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*.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2022. Orca: A challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*.

Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Duan Xinyu, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Phillippe Langlais. 2022. Revisiting Pre-trained Language Models and their Evaluation for Arabic Natural Language Understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3135–3151, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2021. Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*, 33(5):497–507.

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.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

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.

Imad Lakim, Ebtesam Almazrouei, Ibrahim Abualhaol, Merouane Debbah, and Julien Launay. 2022. A holistic assessment of the carbon footprint of noor, a very large arabic language model. In *Proceedings of BigScience Episode\# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 84–94.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *In International Conference on Learning Representations*.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2022a. Jasmine: Arabic gpt models for few-shot learning. *arXiv preprint arXiv:2212.10755*.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022b. 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.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends. *arXiv preprint arXiv:2302.03512*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

J Rae, G Irving, and L Weidinger. 2021a. Language modelling at scale: Gopher, ethical considerations, and retrieval. *DeepMind Blog*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021b. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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*.

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.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

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.

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

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

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*.

Teo Susnjak. 2022. Chatgpt: The end of online exam integrity? *arXiv preprint arXiv:2212.09292*.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv preprint arXiv:2102.02503*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

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.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. 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*, pages 483–498.

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.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*.

A Pretraining

A.1 Data Collection

Our pre-training corpus is mainly sourced from the publicly available web scrapes of the Common Crawl (CC) project. We downloaded 90 shards of CC monthly data ranging from May 2013 (the earliest available) up to July 2022. Also, we use an *in-house collection* of 47GB of Arabic dialect textual data (DIALECT) in order to enhance the awareness of our model to Arabic dialects (Abdul-Mageed et al., 2021). In addition, we include high-quality news corpora such as NEWS (Zeroual et al., 2019) and El-KHAIR (El-Khair, 2016) which are commonly used in previous Arabic PLM works (Safaya et al., 2020; Antoun et al., 2020; Nagoudi et al., 2022b; Ghaddar et al., 2022). Finally, we use 28GB of *in-house* Arabic data curated from different text genres like literature, books, and Wikipedia.

Source	Original	Clean	Filtering %
CC	8.7TB	439GB	95%
DIALECT	-	47GB	-
NEWS	21GB	14GB	34%
EL-KHEIR	16GB	13GB	19%
Others	28GB	16GB	45%
Total	8.8TB	529GB	94%

Table 6: Size of the pre-training corpora before (Original) and after (Clean) applying data filtering and deduplication heuristics.

As it has been shown to be crucial for English (Raffel et al., 2019), multilingual (Xue et al., 2021), and Arabic (Ghaddar et al., 2022) PLM end-tasks performance, we aggressively filter and deduplicate the collected data using the heuristics described in (Ghaddar et al., 2022). Table 6 shows data sizes before and after applying the heuristics. While we discard 95% of CC data, it is still considered, along with DIALECT, to form more than 90% of our 529GB final pre-training corpus.

A.2 Implementation Details

We use the SentencePiece (Kudo and Richardson, 2018) tokenizer in order to process text into subtokens. We train the tokenizer from scratch on our pre-training corpus by setting the vocabulary size to 64k, a value which is used commonly by previous Arabic PLMs (Antoun et al., 2020; Ghaddar et al., 2022; Nagoudi et al., 2022a).

Following (Raffel et al., 2019), we pre-train AraMUS on the *Replace corrupted spans* tasks with a random token probability of 15%. The pre-training code is based on the PyTorch (Paszke et al., 2019) version of the Megatron-LM library (Shoeybi et al., 2019). AraMUS is pre-trained on 16 sever, each occupied with 8 NVIDIA A100 GPUs with 80GB memory. Model and data parallel sizes are set to 4 and 32 respectively. The total batch size is 4096, which is based on the max batch size which can fit on a single GPU (32). To speed up the pre-training, we use mixed-precision training (Micikevicius et al., 2018), except when calculating attention softmax and when reducing gradients. We use the Adafactor optimizer (Shazeer and Stern, 2018) with an initial learning rate of 0.005, 10k warm-up steps with the inverse square-root scheduler.

B Finetuning

B.1 Datasets and Evaluation

ALUE (Seelawi et al., 2021) is a well-established benchmark that consists of a collection of eight Arabic NLU tasks. Although its datasets are relatively small compared to the one of the English GLUE (Wang et al., 2018) benchmark, but it is supported by a public leaderboard with hidden test sets which ensures a fair comparison between models. Following (Seelawi et al., 2021), we report Pearson correlation on SVREG, Jaccard on SEC, and accuracy on XNLI, and use the F1 score otherwise. We also report the unweighted average sum over the 8 tasks.

As for generative tasks, we follow (Ghaddar et al., 2022) by considering 3 tasks for evaluation, as their datasets are fully open source. We use Wikilingua (Ladhak et al., 2020) and EASC (El-Haj et al., 2010) for TS, and the set of datasets used in (Abdul-Mageed et al., 2021; Nagoudi et al., 2022b) for QA and QG. We follow (Ghaddar et al., 2022) for splitting the data into train/dev/test, and report Rouge scores (Lin, 2004) on TS, BLEU (Papineni et al., 2002) on QG, and Exact Match (EM) and F1 score on QA. Therefore, AraMUS results can be directly comparable with the baselines reported by (Ghaddar et al., 2022).

B.2 Baseline

We compared AraMUS with the state-of-the-art Arabic PLMs that have been evaluated on publicly available datasets, these include:

- **ARBERT and MARBERT** are respectively MSA and Arabic Dialect BERT-base (Devlin et al., 2018) models provided by (Abdul-Mageed et al., 2021).
- **JABER and SABER** are respectively BERT-base and BERT-large models provided by (Ghaddar et al., 2022).

- **ALM-1.0** [4] is a recently published Arabic BERT-large model.
- **AraT5-base and AT5B** are Arabic T5-base (Raffel et al., 2019) models provided by (Nagoudi et al., 2022b) and (Ghaddar et al., 2022) respectively.

It is worth mentioning that it was not possible to compare AraMUS with its counterpart multi-billion Arabic GPT models because:

B.2.1 NOOR

NOOR (Lakim et al., 2022) is the largest existing Arabic PLM with 10B parameters. In their work, the authors didn't make their model publicly available neither reported their results on public datasets.

B.2.2 AraGPT-Mega

AraGPT-Mega (Antoun et al., 2021b) has 1.5B parameters and is publicly available for download. However, we tried to run *in-house* experiments with this model but it didn't perform well on many tasks. Most likely because it was only pre-trained on 27GB of Arabic text, which is considered small compared to the model size. Therefore, we preferred not to report weak results for this model.

B.2.3 Jasmine

Jasmine (Nagoudi et al., 2022a) is an *in-progress* project that aims to develop and evaluate a set of Arabic GPT models up to 13B parameters. This in-progress work was released at the time of writing our paper. The authors mentioned that the 13B model is still at early pre-training stage, while the 6.7B version is only pre-trained for 143k steps. Therefore, their *fully pre-trained* Jasmine has 2.7B parameters only. This model is evaluated, in a few shot setting only, on a set of discriminative and generative tasks on the ARLUE (Abdul-Mageed et al., 2021) and ARGEN (Nagoudi et al., 2022b) benchmarks respectively. However, many of the datasets in ARLUE and ARGEN have not been publicized yet (Elmadany et al., 2022; Ghaddar et al., 2022). In addition, the authors didn't open source their model weights nor shared their code to replicate their dataset splits.

B.3 Implementation Details

We used early stopping based on the performance of the dev sets during our extensive hyper-parameter search. We search the learning rate from the set of {5e-5, 1e-4, 2e-4, 1e-3}, batch size from {8, 16, 32, 64}, the learning rate scheduler from {constant, cosine}, and the dropout rate from {0.1, 0.15, 0.2, 0.3}, and fixed the epoch number to a maximum of 120 for all the experiments. Each fine-tuning experiment uses 4 NVIDIA A100 GPUs, with the model parallel size set to 4.

After finding the best hyper-parameters, we ran all the experiments 5 times and reported the average score on the dev sets [5], in order to validate the credibility of our results. For each ALUE task, we selected the best-performing model among the 5 runs and used it for the ALUE leaderboard test submission, and we computed the scores on generative tasks datasets.

We simulate a few-shot setting on the ALUE tasks by randomly sampling a subset of {8, 16, 32, 64, 128, 256} examples of the training data. When the number of classes is more than the number of samples (e.g. MDD and SEC with 8 examples) we randomly add one example for each missing class in order to ensure that each class has a represented data point. All models are identically fine-tuned, and we report the average and standard deviation of 5 randomly selected folds.

C Few-Shot Results

[4] `https://github.com/FlagAI-Open/FlagAI/tree/master/examples/ALM`

[5] We use the MQ2Q dev set curated by (Ghaddar et al., 2022) to make our results compatible with their baselines.

Model	**MQ2Q***	**MDD**	**SVREG**	**SEC**	**FID**	**OOLD**	**XNLI**	**OHSD**	**Avg.**
8 Examples									
JABER	50.0±15.8	8.9±1.8	18.8±17.5	21.7±0.2	56.7±13.5	56.5±7.9	35.7±2.3	54.9±5.9	37.9±8.1
SABER	53.5±6.9	8.8±1.4	34.2±09.6	21.1±0.8	63.0±11.2	65.3±12.6	35.5±1.9	58.1±7.3	42.4±6.5
AraMUS	**60.2±3.7**	**16.7±1.8**	**54.5±8.7**	**23.2±3.5**	**69.0±2.8**	**69.5±1.6**	**35.8±1.1**	**58.3±7.7**	**48.4±3.9**
16 Examples									
JABER	56.2±14.5	7.9±1.1	45.2±16.1	24.3±3.0	69.9±5.6	68.0±12.5	37.0±3.4	53.0±5.7	45.2±7.7
SABER	54.6±8.2	9.0±2.1	47.7±16.7	21.6±1.9	73.0±2.8	80.3±7.9	35.8±2.3	57.7±8.0	47.5±6.2
AraMUS	**61.4±4.7**	**20.4±1.9**	**66.6±5.6**	**25.5±4.8**	**74.3±1.2**	**82.3±1.7**	**39.1±4.9**	**59.1±7.5**	**53.6±4.0**
32 Examples									
JABER	66.9±3.3	8.0±1.8	63.7±11.7	27.0±3.3	72.1±3.9	71.7±5.9	38.7±2.9	57.7±7.7	50.7±5.1
SABER	63.3±6.6	9.8±2.3	72.3±9.3	28.7±4.1	74.5±1.4	81.2±9.3	37.4±1.4	54.6±7.2	52.7±5.2
AraMUS	**69.2±4.3**	**21.3±1.1**	**74.5±3.6**	28.0±5.0	**74.8±2.6**	**85.5±2.1**	**45.3±3.7**	**59.3±6.9**	**57.2±3.7**
64 Examples									
JABER	68.6±3.5	11.0±1.9	72.6±7.8	31.5±1.5	73.7±0.8	77.0±2.7	42.4±2.2	58.8±8.4	54.4±3.6
SABER	67.8±2.8	12.8±1.9	79.6±3.3	34.8±1.7	77.2±1.4	87.0±2.1	39.6±4.2	61.4±7.4	57.5±3.1
AraMUS	**74.8±1.8**	**22.3±1.0**	**81.8±3.7**	31.5±1.8	**77.7±0.7**	**89.6±1.5**	**55.5±3.6**	**64.0±8.7**	**62.2±2.8**
128 Examples									
JABER	70.0±1.5	16.9±0.6	80.5±1.3	35.3±1.8	76.4±1.1	82.4±2.8	44.6±1.0	64.2±4.0	58.8±1.8
SABER	72.1±0.9	18.9±2.0	83.6±2.0	39.5±2.8	78.3±1.3	88.7±1.4	44.8±4.0	66.8±4.0	61.6±2.3
AraMUS	**77.5±1.1**	**25.7±1.7**	**84.1±0.9**	37.0±1.4	**78.6±0.5**	**90.4±0.9**	**63.6±1.5**	**67.8±4.1**	**65.6±1.5**
256 Examples									
JABER	72.7±1.0	22.4±0.6	83.7±0.7	39.3±0.8	79.0±1.1	84.9±1.0	53.1±2.2	62.5±6.2	62.2±1.7
SABER	72.8±1.7	25.5±1.9	85.0±1.3	42.2±0.5	79.8±1.2	89.6±0.7	48.0±13.5	70.6±1.3	64.2±2.8
AraMUS	**78.1±1.2**	**30.2±0.8**	**86.3±1.3**	41.1±0.7	**80.8±1.7**	**92.3±0.9**	**72.6±0.7**	**71.2±3.4**	**69.1±1.3**

Table 7: Dev ALUE performances across training set sizes. Underline figures indicates extra samples where added to ensure that each class is represented at least by one data point.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
5

☒ A2. Did you discuss any potential risks of your work?
There is no risks of our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

3.4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
A.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3.4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
A.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.