International Journal on Cybernetics & Informatics (IJCI) Vol.13, No.2, April 2024

Ara-CANINE: Character-based Pre-trained Language Model for Arabic Language Understanding

Abdulelah Alkesaiberi[1⋆], Ali Alkhathlan[1⋆⋆], and Ahmed Abdelali[2⋆⋆⋆]

[1]Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia
[2]National Center for AI, SDAIA, Riyadh, Saudi Arabia

Abstract. Recent advancements in the field of natural language processing have markedly enhanced the capability of machines to comprehend human language. However, as language models progress, they require continuous architectural enhancements and different approaches to text processing. One significant challenge stems from the rich diversity of languages, each characterized by its distinctive grammar resulting in a decreased accuracy of language models for specific languages, especially for low-resource languages. This limitation is exacerbated by the reliance of existing NLP models on rigid tokenization methods, rendering them susceptible to issues with previously unseen or infrequent words. Additionally, models based on word and subword tokenization are vulnerable to minor typographical errors, whether they occur naturally or result from adversarial misspellings. To address these challenges, this paper presents the utilization of a recently proposed free-tokenization method, such as Cannine, to enhance the comprehension of natural language. Specifically, we employ this method to develop an Arabic-free tokenization language model. In this research, we will precisely evaluate our model's performance across a range of eight tasks using Arabic Language Understanding Evaluation (ALUE) benchmark. Furthermore, we will conduct a comparative analysis, pitting our free-tokenization model against existing Arabic language models that rely on sub-word tokenization. By making our pre-training and fine-tuning models accessible to the Arabic NLP community, we aim to facilitate the replication of our experiments and contribute to the advancement of Arabic language processing capabilities. To further support reproducibility and open-source collaboration, the complete source code and model checkpoints will be made publicly available on our Huggingface[1]. In conclusion, the results of our study will demonstrate that the free-tokenization approach exhibits comparable performance to established Arabic language models that utilize sub-word tokenization techniques. Notably, in certain tasks, our model surpasses the performance of some of these existing models. This evidence underscores the efficacy of free-tokenization in processing the Arabic language, particularly in specific linguistic contexts.

Keywords: NLP, Free-Tokenization, Large Language Model, Arabic Language

1 Introduction

Natural Language Processing (NLP) is a sub-field of computer science, particularly artificial intelligence (AI), that deals with natural (human) language [1]. It has met with great success in many problems and applications such as Neural Machine Translation for Machine Translation [2], and GPT-3 for Text Generation [3]. NLP has reached significant milestones starting from natural language models to pre-trained models including machine and deep learning-based models. Researchers in [4, 5] proposed a solution called "attention", whose goal is to get as much contextual information as it can from the encoder. In doing so, the encoder sends data involving all the hidden states to the decoder rather than just the last hidden state [6]. Hereby, the decoder is still the same except for the

[1]`https://huggingface.co/csabdulelah/Ara-CANINE`

Bibhu Dash et al: NLPAI, IOTBC -2024
pp. 45-60, 2024. IJCI – 2024
DOI:10.5121/ijci.2024.130204

additional attention layer, which means the input to the decoder will be changed. In 2017, everything changed in the NLP world after Transformer emerged [7], changing the way computers deal with natural languages. It achieved state-of-art results in NLP tasks and most of today's NLP model architectures are built on Transformer architectures. It introduced "self-attention," which is a new form of attention that helps Transformer pass input sequences in parallel providing faster performance than RNN. In addition, it captures the relationships between all words in a sentence[8]. The key components of a transformer are the encoder, decoder, positional embedding, and self-attention [9].

BERT, which stands for Bidirectional Encoder Representations from Transformers, was proposed in 2018 [10]. BERT is based on transformer architecture except that BERT authors used only the encoder part and excluded the decoder part. GPT3 is also a widely used transformer-based model with 175 billion parameters published in 2020 [3]. Although these models have achieved state-of-art results in NLP tasks, for example, according to [10] BERT improves GLUE score by 7.7%.[10]. One of the most spoken languages globally by approximately 370 million speakers is Arabic [11]. However, given that Arabic is a complex language with multiple dialects, grammar, and letter shapes, it is critical for Latin-based NLP models to understand Arabic text accurately. Therefore, researchers have developed natural language models designed specifically for understanding the Arabic language. For instance, AraBERT [12] is one of the common NLP models used in the Arabic community. It is based on BERT architecture and uses SentencePiece [13] sub-word tokenization algorithm. Later, authors in [14] proposed two models based on WordPiece subword tokenization [15]: ARBERT which was pre-trained on 6 different data sources, and MARBERT which was pre-trained on 1 Billion tweets. Both ARBERT and MARBERT models outperformed AraBERT, the best-performing Arabic model at the time [12].

Nevertheless, state-of-the-art NLP models rely on distinct rigid subword tokenization techniques, which restrict their ability to generalize and adapt to new settings.[16]. On the other hand, free-tokenization or character-based language models have gained popularity recently due to their potential for several tasks, including machine translation and question-answering [16]. The ability of these models to extract morphological information from characters is a major factor in their effectiveness. The key concept of free-tokenization is to operate on raw characters without any explicit tokenization [16, 17]. Hence, this research proposes a character-level Arabic Language model. Constructing a pre-trained character-level Arabic Language model is expected to perform comparably to subword models. As proof, in [18] authors created an Arabic language model based on BERT architecture called JABER. They also developed another new version called Char-JABER, where they inject the character information along with the subword tokens. Based on the results, JABER outperformed all other existing Arabic language models achieving an average score of 73.7% of the ALUE benchmark with 8 tasks. Additionally, Char-JABER improved the accuracy on the ALUE benchmark and outperformed the JABER model with an increase of 1.6% in the average ALUE score. In addition to this advantage, character-level (free-tokenization) models have the advantage of reducing the engineering efforts of input preprocessing[17].

Our contributions may be succinctly summarized as follows:

- Answer the question of how Arabic free-tokenization models perform against Arabic sub-word tokenization models
- Know how free-tokenization language models perform on verities of Arabic language
- Making pre-training and fine-tuning models available to the Arabic NLP community (We will enable the public replication of our experiment).

The rest of the paper consists of five sections. Section 2 describes the Related works. Section 3 presents the Methodology and experiments. Section 4 discusses the evaluation and results.

2 Related Works

2.1 Overview of Tokenization in NLP

Tokenization is a foundational step in natural language processing (NLP) where text is segmented into smaller units called tokens[19]. Three primary forms of tokenization are employed in NLP: word, subword, and character tokenization.

- **Word Tokenization:** This is the most basic form where sentences are split into individual words. One application that uses word tokenization is Word2Vec architecture which is considered a word tokenizer [20]. Word2Vec captures the contextual word relations so that words with similar contexts have similar embeddings. One of the main disadvantages of word tokenization is Vocabulary (OOV) words which are the issues of new words encountered during the test phase. Transformer-XL is one of the models that use word tokenization in which the tokens are split based on space and punctuation [21].
- **Subword Tokenization:** The main reason for using subword tokenization is to mitigate the out-of-vocabulary (OOV) issue where the model is exposed to unseen data. Techniques such as Byte Pair Encoding (BPE) have been pivotal in this arena[22, 23]. Some of pioneering models like BERT, DistilBERT, ALBERT, XLNet, and RoBERTa have utilized subword tokenization techniques.
- **Character Tokenization:** Character tokenization or free tokenization model is a hot topic in NLP nowadays. The main idea is that the model does not need any explicit segmentation. Instead, the input is the raw text that is converted into character byte-level vectors. Each token consists of a single character which represents a small-sized vocabulary based on the number of characters of the language. Hence, this reduces the processing time. Clark et al. [17] feeds to the model raw character and each character is turned into its Unicode code point.

2.2 Transformer

Making a huge impact on shaping the present of the NLP field, Transformer emerged in 2017. Nowadays, most of the language models are built on the Transformer architecture. The input of this model is either word-level or subword-level tokens, with each technique coming with its pros and cons. Word tokenization has faced problems in dealing with rare data. Hence, Subword tokenization has the advantage of handling rare or unseen data. It enables models to process tokens it has never encountered before by breaking them down into known subwords. Transformer emerged in 2017 making a major impact in the NLP field [7]. Transformer architectures achieved state-of-art results in NLP tasks and most of the current NLP models are developed based on transformer architectures. The key components of a transformer are the encoder, decoder, positional embeddings, and self-attention. As mentioned previously, input strings need to be turned into vectors using word embeddings. A problem that faces traditional word embedding is that changing the position of the word changes the context of word meaning. Here the transformer proves useful with positional embeddings. Positional embedding is another vector that represents the position of the word which is added to the word vector to produce the input of the

encoder. After the input is fed into the encoder, the first layer in the encoder is self-attention, which answers the question of how a word is relevant to other words in the same sentence[9]. Figure 6 shows the Transformer model architecture.

BERT BERT is a stack of encoders of transformers, which is firstly pre-trained on text to understand the language and then fine-tuned on a wide range of language tasks. In this study, the pre-training process involved training on two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). BooksCorpus (800M words) and English Wikipedia (2,500M words) data were used for pre-training. BERT outperforms the previous models in different tasks [24]. In BERT [10], authors used WordPiece, which was first mentioned in 2012 [15] and also introduced by Google in 2016 [2], containing 30,000 tokens. The input of BERT consists of three embeddings: (1) the sum of token embeddings using WordPiece [15],(2) segmentation embeddings, which is sentence number that is encoded into a vector, and (3) position embeddings, which is the position of the token (word) in that sentence [25]. The main concept in BPE is to iteratively replace the most common pair of bytes in a sequence with a single unused byte. In Sennrich et al. [22], the concept of character-level was adopted to combine character sequences instead of the frequently occurring pairs of bytes.

Arabic Language Models Arabic is a morphologically rich language, but also lacks resources compared to English language. We will discover some Arabic language pre-trained models that are based on BERT architecture and make use of WordPiece and BPE sub-word tokenization [18]

- **ARBert and MARBert** In [14], the authors contributed by pre-training BERT on Arabic Dataset. They proposed the ARBert model which was pre-trained on 68GB of MSA text data for 42 epochs, and the MARBert model, pre-trained on 128GB of dialects data for 36 epochs.
- **AraBert** In this research [12], the authors present the AraBERT model, a contribution to the field of NLP tailored specifically for the Arabic language. AraBERT has been training on an extensive dataset, approximately 27 gigabytes (GB) of MSA text data. The training was for 27 epochs.
- **CAMeLBERT-Mix** While the authors in this study proposed multiple models that trained on different types of data they proposed their largest model which is CAMeLBERT-Mix which trained on 167GB of MSA and Dilects data for only 2 epochs [26].

2.3 Free Tokenization Models

In recent years, there has been a surge of interest in character-based language models. These models have shown promising performance in a variety of tasks such as machine translation and question answering [16]. The success of these models is mainly due to their ability to catch characters' morphological information. This section provides a review of the existing literature on character-based language models and discusses the benefits and challenges of this approach.

Santos and Zadrozny[27] stated that word representation models normally ignore morphological characteristics and shape of words and Choe et al.[28] underlined that character-level processing to understand the morphology of the word. Accordingly, Santos and Zadrozny[27] proposed a deep neural network that learns a character-level representation of words with words and associates them with usual word representations. They developed state-of-the-art POS taggers for two languages: English, with 97.32% accuracy on the Penn Treebank WSJ corpus; and Portuguese, with an accuracy of 97.47%.

Furthermore, the ability of decoders to generate one character each time is arguably an important topic for research. Hence, Chung et al. [29] addressed this issue by conducting experiments in four languages. The results show that the model performed well on rich-morphological languages, which indicates that it is possible for decoders to translate at the character-level.

A more comprehensive description of the model is provided by Jason Lee [30]. The study was built on the previous research [29] that proposed a fully character-level Neural Machine Translation (NMT) model. The model accepts a sequence of characters as an input in one language and outputs a sequence of characters in another language without any explicit segmentation. The authors also proved that multiple languages could share a single character-level encoder without increasing the model size or parameters when building multilingual translation systems.

A great number of authors in the literature discussed the fact that the majority of recent research has applied transformer architecture [7] and BERT architecture [10] for building character-level-based models. However, it is noteworthy to mention that applying character-level processing on the Transformer increases the inputs by 4x which results in a slower model [17], but this challenge was overcome using Convolution Neural Network (CNN)[31, 32].

In 2020, Ma et al. [33] proposed the character-aware pre-trained language method, CharBERT, to enhance existing models. It combined character representations and sub-word representations using a novel heterogeneous interaction module The proposed model was evaluated on 8 benchmarks and outperformed BERT and RoBERTa baselines. The study also proposed a new pre-training task called NLM (Noisy LM) which trains the model on noisy data to increase the robustness of the model.

Moreover, El Boukkouri et al. [34] proposed the CharacterBERT model which is based on BERT [10] excluding the WordPiece system. Instead, it uses Character-CNN which produces word-level contextual representations by consulting characters. The CharacterBERT model was evaluated on multiple specialized medical tasks where the model outperformed BERT without the WordPiece vocabulary. Additionally, they proved that general-domain WordPiece vocabularies are not suitable for specialized domain applications.

[17], authors proposed CANINE, the first pre-trained tokenization-free deep encoder. It is an encoder for large languages with a deep transformer stack. The inputs of the model are a sequence of Unicode characters. A model that performs no tokenization on the input avoids the lossy information bottleneck associated with most pre-processing. Thereby, CANINE provides an efficient architecture to enable tokenization-free modeling by directly encoding long sequences of characters with a speed comparable to vanilla BERT.

Furthermore, Ghaddar et al. [18], the authors investigated the usefulness of injecting the characters at the input layer. They proposed three Arabic BERT-based models namely JABER, Char-JABER, and SABER, where the latter two are enhanced versions of JABER model. Besides being pre-trained on high-quality filtered data, each word in Char-JABER is represented by a character-level vector using a multilayer CNN encoder. These vectors are then added to the BERT input representation vector to acquire the final representation. Based on the results, Char-JABER outperforms JABER by 1.6% in the average ALUE score. It is noteworthy to mention that on ALUE's diagnostic data (DIAG), Char-JABER outperforms SABER by 0.5%.

A recent study conducted by Tay et al.[16] in 2021 introduced a gradient-based subword tokenization module (GBST) that automatically learns subword representations from characters. Additionally, they proposed CHARFORMER which is a transformer-based model

with an integrated GBST layer. CHARFORME operates at character-level increasing the processing speed by 28-100% in comparison to that of Transformer's.

Overall, transformer-based character-based language models have shown great results in a variety of tasks[16, 17], especially when trained on special domain task [34]. However, there are still many challenges and critical questions that need to be addressed. In particular, whether the character-level version of a model will outperform the current-version model on rich morphological languages such as the Arabic language.

3 Methodology: Pre-training

In this study, we endeavor to develop an Arabic language model that employs a free-tokenization approach, aimed at evaluating the comparative effectiveness of character-based models against subword tokenization methods. We introduce Ara-CANINE, a model inspired by and based upon the Shiba model [35], which itself is an implemented version of the CANINE model, an acronym for Character Architecture with No tokenization In Neural Encoders [17]. The fundamental principle of CANINE is the utilization of raw characters as input, hence, obviating the need for any explicit tokenization step [17]. A distinctive feature of Ara-CANINE, setting it apart from existing models such as Char-JABER has its exclusive reliance on raw character input. In contrast, Char-JABER integrates character-level representations in conjunction with sub-tokens representations [18]. This differentiation underscores the unique approach of Ara-CANINE in processing the Arabic language, exploring the potential of character-level encoding in capturing linguistic nuances.

3.1 Pre-Training Setup

We pre-trained on the QADI dialects tweets dataset [36] which included 18 Arabic dialects. We trained for 40 epochs (610K steps) with batch size 32 and gradient accumulation steps of 8 which resulted in 256 batch size for each GPU device. Using the Adam optimizer with beta1 value of 0.9 and beta2 of 0.98 with a linearly decayed learning rate of 0.0001 where 2.5% of the steps are used for warm-up. We used a sequence length 2048 which resulted in 512 down-sampled positions in its core deep transformer stack. We pre-trained on 3 A100 GPUs 40GB each on a different node for approximately 25 days. Furthermore, in the context of distributed learning, the utilization of the DeepSpeed library [37], an open-source optimization library specifically for deep learning in PyTorch was employed. The library is architecturally engineered to facilitate the training of extensively distributed models by enhancing parallelism on pre-existing computational hardware, concomitantly minimizing computational power and memory usage. It is meticulously designed to execute training with minimized latency and maximized throughput, ensuring efficient model development[37].

3.2 DataSet

In the pre-training phase of our study, we utilized the QADI dialects tweets dataset [36], which comprises a substantial corpus of tweets. This dataset, automatically collated from Twitter with the language tag 'ar', encompasses a country-level dialectal tweet corpus featuring approximately 540K tweets from 2525 distinct users, covering 18 different Arabic dialects. Due to the constraints posed by our computational resources, we elected to randomly select a subset of 40GB of text from the entire dataset for our analysis. Additionally, 2% of the data, amounting to a substantial portion, was reserved for validation

purposes. This resulted in a total of 11,749,714 training examples and 234,994 validation examples. The decision to focus on a dialects dataset was strategically aligned with the requirements of the ALUE benchmark, which predominantly consists of tasks related to dialects. Therefore, this approach was deemed most suitable for aligning our pre-training phase with the anticipated tasks of the ALUE benchmark.

Data Preprocessing In the initial phase of our methodology, we implemented a streamlined preprocessing approach. This was achieved by utilizing the AraBert Preprocess class [12], a tool specifically designed for the preparation of text data in the context of Arabic language processing. The functionalities of this tool include the removal of emojis, tashkeel (diacritical marks), and tatweel (character elongation). Additionally, it is programmed to replace URLs, email addresses, and user mentions with designated placeholders (i.e., 'URL', 'USER', and 'EMAIL') and to eliminate HTML line breaks and markup. The incorporation of this preprocessing phase is pivotal in enhancing the quality of the text data, thereby laying a solid foundation for the subsequent stages of our study. It ensures that the data is cleaned and standardized, which is crucial for the effectiveness and accuracy of the model's performance in later phases.

Model Pre-training Model pre-training constituted the pivotal phase and epitomized our contribution within the domain of Arabic Natural Language Processing (NLP). Throughout this phase, the model was trained on 40GB tweets text from 18 different countries to acquire a comprehensive understanding of Arabic language. The process took a significant amount of time due to the size of the data, and the computing resources we had. We encountered several challenges during this phase, mainly related to the limited availability of Graphic Processing Units (GPUs). Initially, we had access to only one GPU, which made it difficult to train all the data efficiently. To address this, we divided the data into four chunks, each containing 10GB of text; we then attempted to conduct training using three GPUs when we granted it. However, we still faced issues with GPU memory until we utilized the DeepSpeed optimization library, which eased the situation significantly. The core concept that DeepSpeed relies on is ZeRO "Zero Redundancy Optimizer" which is a method that reduces memory redundancies by dividing the optimizer, gradient, and parameters instead of duplicating them, to make efficient use of all available memory. The models were trained, employing two principal tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). During this phase, a dialects Arabic data set derived from Twitter was introduced to the model, aimed at validating the hypothesis that free-tokenization models exhibit comparable performance on rich morphological languages, such as Arabic. The pre-training was done on the MLM task by randomly masking 15% of span characters rather than a single character – the size of span used is 2 and a span random character rather than a single character, according to the authors of [35] who experimented with single and span characters and random-span characters result better than random single characters.

Model Fine-tuning In the fine-tuning phase, we will teach the model how to perform a specific task. Mimicking how humans learn, the model will learn specific NLP tasks separately after understanding the language. We will fine-tune our pre-trained model with labeled data from the ALUE benchmark [38]. ALUE is a set of datasets for multi-task learning that contains 8 tasks. Each task has a different dataset and different evaluation metrics. The first task is FID Irony Detection Task where the model detects ironic tweets. Secondly, MDD (Dialect Detection) contains 25 labels (dialects), and the model detects

to which label a given sentence belongs. The third task is the MQ2Q task which is about question semantic similarity, in which the model checks if two questions in this task have the same answer and meaning. Dataset number 4 OSACT4 contains 2 tasks where the model detects offensive language and hate speech. The fifth dataset is SemEval-2018 which contains multiple subtasks. ALUE used 2 tasks which are the Emotion Classification task (SEC) and the Sentiment Intensity Regression task (SVREG). The first one is to classify the label of the tweet while the second one is a regression task to predict the positivity or negativity of a tweet between 0-1, where 0 is the most negative and 1 is the most positive. The final dataset is XNLI Cross-lingual Sentence Representations which contains pairs of sentences that express a hypothesis and premise, and the model is supposed to classify the relationship between each pair[38].

Task	Train	Dev	Test	Text Type	Task Type
SEC	2.3k	600	1.5k	DIAL	Single-sentence classification - 11 labels
MDD	42k	5.2k	5.2K	DIAL	Single-sentence classification - 26 labels
FID	4k	-	1k	DIAL	Single-sentence classification - 2 labels
MQ2Q	12k	-	3.7	MSA	Sentence-pair classification - 2 labels
XNLI	5k	-	2.5k	MSA	Sentence-pair classification - 3 labels
OHSD	7k	1k	2k	DIAL	Single-sentence classification - 2 labels
SVREG	900	100	700	DIAL	Single-sentence regression - (0-1)
OOLD	7k	1k	2k	DIAL	Single-sentence classification - 2 labels

Table 1. Statistics of train, dev, and test sets of tasks in ALUE benchmark. Modern Standard Arabic (MSA) and Arabic dialects (DIAL) are the two Arabic text types involved in the tasks. Task type describes each task in terms of the inputs and the associated labels

4 Evaluation

In this study, we conducted a comprehensive evaluation of Ara-CANINE, our proposed model for Arabic language processing, using the Arabic Language Understanding Evaluation (ALUE) benchmark [38]. The ALUE benchmark is a widely recognized standard in the field, consisting of eight diverse tasks that test various aspects of language understanding. These tasks include both one and multi-label classification challenges, as well as a single regression task. Specifically, the tasks cover areas such as sentiment analysis, text categorization, and others, offering a robust test of Ara-CANINE's capabilities in handling the complexities of Arabic language

Building upon this foundation, it was crucial to ensure a fair and comprehensive evaluation by comparing Ara-CANINE with several existing Arabic language models previously assessed using the ALUE benchmark. This comparison included critical aspects like the size of the training datasets. Providing this context not only establishes a baseline for comparison but also situates Ara-CANINE within the landscape of Arabic natural language processing.

Model	Arabic-BERT	AraBERT	CaMELBERT	ARBERT	MARBERT	Ara-CANINE
#Params	110M	135M	108M	163M	163M	159M
Vocab Size	32k	64k	30	100k	100k	-
Tokenizer	WordPiece	WordPiece	WordPiece	WordPiece	WordPiece	Free-Tokenization
Data Size	95GB	27GB	167GB	61GB	128GB	40GB
#Epochs	27	27	2	42	36	40

Table 2. Comparative Overview of Different Arabic Language Models. This table presents a side-by-side comparison of Ara-CANINE with other prominent Arabic language models, namely Arabic-BERT, AraBERT, CaMELBERT, ARBERT, and MARBERT. Key comparative metrics include the number of parameters (#Params), vocabulary size (Vocab Size), tokenizer type, the size of training data (Data Size), and the number of training epochs (#Epochs). [Statistics adapted from [18]]

The fine-tuning of Ara-CANINE on the ALUE benchmark revealed distinct performance characteristics. Specifically, as detailed in the following Table 3. Ara-CANINE demonstrates superior performance in tasks involving Twitter data and dialects. This improvement is attributed to the model being specifically trained on datasets rich in dialectical content and social media text. Conversely, Ara-CANINE shows less proficiency in tasks focused on Modern Standard Arabic (MSA), underperforming in comparison to tasks more aligned with its training data. For a comprehensive evaluation, we compared Ara-CANINE's performance with two other models that were trained on datasets of similar size. These comparisons are crucial to understanding the specific strengths and limitations of Ara-CANINE in handling various facets of the Arabic language

	AraBERT	**ARBERT**	**Ara-CANINE**	Δ
MQ2Q	89.2	89.3	82.3	-6.9
MDD	58.9	61.2	57.9	-1.0
SVREG	56.3	66.8	57.0	+0.7
SEC	24.5	30.3	21.2	-3.3
FID	85.5	85.4	82.0	-3.4
OOLD	88.9	89.5	91.0	+0.5
XNLI	67.4	70.7	54.8	-12.6
OHSDI	76.8	78.2	78.9	+1.1

Table 3. Performance Comparison of Arabic Language Models Across Various ALUE Tasks. The table displays a comparative analysis of AraBERT, ARBERT, and Ara-CANINE on eight distinct tasks, identified by their respective acronyms (e.g., MQ2Q, MDD, SVREG). Each task's performance scores are presented, along with a column 'Difference' showing the performance deviation of Ara-CANINE from the other models. Positive values indicate an improvement, while negative values denote a decrease in performance. This comprehensive comparison highlights the strengths and weaknesses of each model in specific areas, providing a nuanced understanding of their capabilities in processing the Arabic language.

Further expanding on this comparative analysis, Table 4 presents Ara-CANINE's performance alongside other models, some trained on significantly larger datasets. Despite the disparity in training data size, Ara-CANINE showed promising results, outperforming several larger models in specific tasks, such as the OOLD task. This not only underscores the efficiency of Ara-CANINE's design but also highlights its potential to achieve high performance with comparatively limited data resources. These results demonstrate the robustness and adaptability of Ara-CANINE in handling diverse and complex language processing tasks.

In conclusion, the comprehensive evaluation of Ara-CANINE using the ALUE benchmark has demonstrated its notable capabilities in Arabic language processing. Despite cer-

tain limitations in tasks involving Modern Standard Arabic, Ara-CANINE's performance in tasks related to dialectical content and social media text is particularly commendable. These findings not only validate the effectiveness of Ara-CANINE in specific contexts but also shed light on potential areas for further improvement and optimization. Looking forward, these insights pave the way for future enhancements to the model, potentially extending its applicability and efficiency in even broader aspects of Arabic natural language processing. This study thus serves as a foundational step towards advancing the field of free-tokenization language models.

	Arabic-BERT	**AraBERT**	**CaMELBERT**	**ARBERT**	**MARBERT**	**Ara-CANINE**
MQ2Q	85.7	89.2	89.4	89.3	83.3	82.3
MDD	59.7	58.9	61.3	61.2	61.9	57.9
SVREG	55.1	56.3	69.5	66.8	75.9	57.0
SEC	25.1	24.5	30.3	30.3	36.0	21.2
FID	82.2	85.5	85.5	85.4	85.3	82.0
OOLD	89.5	88.9	90.3	89.5	92.1	91.0
XNLI	61.0	67.4	56.1	70.7	64.3	54.8
OHSDI	78.7	76.8	80.6	78.2	78.9	79.6

Table 4. Performance Comparison of Ara-CANINE and Other Arabic Language Models on Various ALUE Tasks. This table illustrates a detailed comparison of the performance scores of different models, including Arabic-BERT, AraBERT, CaMELBERT, ARBERT, MARBERT, and Ara-CANINE, across a series of tasks (e.g., MQ2Q, MDD, SVREG). The bold values indicate the scores of Ara-CANINE, providing a direct comparison against other models.

5 Limitations

In this study several limitations are noteworthy. The reliance on the QADI dialects tweets dataset, primarily sourced from Twitter, may introduce a bias towards social media language styles, potentially not representing the full spectrum of Arabic used in other contexts. The study's dependency on the Aziz Supercomputer for computational resources, while beneficial for processing, limits reproducibility in less resource-intensive environments, posing challenges for broader applicability. The Ara-CANINE model, despite its innovative approach, showed limitations in handling Modern Standard Arabic, raising concerns about its generalizability across the diverse Arabic language spectrum. Training constraints, including a limited dataset size and number of training epochs, might have restricted the model's learning depth and overall performance. Furthermore, the evaluation relying solely on the ALUE benchmark may not fully capture the model's capabilities. Addressing these limitations in future work could involve diversifying data sources, enhancing computational resources, expanding evaluation benchmarks, and further innovating the model architecture to better cater to the complexities of Arabic language processing.

6 Conclusion

In this paper, we provide detailed steps to pre-train a new character-based Arabic model with no need for explicit tokenization. Our intensive experiments show that our model achieves comparable results with the existing subword tokenization model for the ALUE benchmark. Additionally, for some tasks, it outperforms some of the existing models, for example, in the SVREG task, it outperforms the Arabic-Bert baseline and AraBert.

In the OOLD task, it outperforms all other models except MARBERT. Lastly, in the OHSDI task, it outperforms all other models except CaMELBERT-Mix. Furthermore, we intend to pre-train models with different pre-training setups and mixed Arabic MSA and dialects datasets. We are committed to supporting the wider research community through open access to our work. To this end, we will promptly publish the source code and model checkpoints of our study in a dedicated GitHub repository as soon as possible after the publication of our findings. This initiative aims to ensure transparency, enable peer verification, and foster further research and development.

7 Acknowledgments

All experiments were carried out on the Aziz Supercomputer; therefore, we would like to thank the management of the High-Performance Computing (HPC) Center at King Abdulaziz University.

Bibliography

[1] Hannes Max Hapke Hobson Lane, Cole Howard. *Natural Language Processing in Action Understanding, analyzing, and generating text with Python Version 10*. Manning, 2018.

[2] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL `https://arxiv.org/abs/1609.08144`.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL `https://arxiv.org/abs/1409.0473`.

[5] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015. URL `https://arxiv.org/abs/1508.04025`.

[6] Jay Alammar. Visualizing a neural machine translation model (mechanics of seq2seq models with attention). 2018. URL `https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/`.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[8] Carolin Becker, Nico Hahn, Bailan He, Marianna Plesaik, Victoria Szabo, Xiao-Yin To, Rui Yang, and Joshua Wagner. *Modern Approaches in Natural Language Processing*. LMU Munich, Munich, 2020.

[9] Jay Alammar. The illustrated transformer. 2018. URL `http://jalammar.github.io/illustrated-transformer/`.

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

[11] Federico Blank. Most spoken languages in the world in 2023. 2023. URL `https://lingua.edu/the-most-spoken-languages-in-the-world/`.

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

[13] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018. URL https://arxiv.org/abs/1808.06226.

[14] M Abdul-Mageed, A Elmadany, and EMB Nagoudi. Arbert & marbert: Deep bidirectional transformers for arabic. arxiv 2020. *arXiv preprint arXiv:2101.01785*, 2020.

[15] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012. doi: 10.1109/ICASSP.2012.6289079.

[16] Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*, 2021.

[17] Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *arXiv preprint arXiv:2103.06874*, 2021.

[18] 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 Philippe Langlais. Revisiting pre-trained language models and their evaluation for arabic natural language understanding, 2022. URL https://arxiv.org/abs/2205.10687.

[19] aravindpai. What is tokenization in nlp? 2020. URL shorturl.at/pOSTW.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[21] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019. URL https://arxiv.org/abs/1901.02860.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2015. URL https://arxiv.org/abs/1508.07909.

[23] Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, feb 1994. ISSN 0898-9788.

[24] Jay Alammar. The illustrated bert, elmo, and co. 2018. URL https://jalammar.github.io/illustrated-bert/.

[25] CodeEmporium. BWorld Robot Control Software. https://www.youtube.com/watch?v=xI0HHN5XKDo, 2020. [article; accessed 2022].

[26] Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. The interplay of variant, size, and task type in arabic pre-trained language models. In Nizar Habash, Houda Bouamor, Hazem M. Hajj, Walid Magdy, Wajdi Zaghouani, Fethi Bougares, Nadi Tomeh, Ibrahim Abu Farha, and Samia Touileb, editors, *Proceedings of the Sixth Arabic Natural Language Processing Workshop, WANLP 2021, Kyiv, Ukraine (Virtual), April 9, 2021*, pages 92–104. Association for Computational Linguistics, 2021. URL https://www.aclweb.org/anthology/2021.wanlp-1.10/.

[27] Cicero Dos Santos and Bianca Zadrozny. Learning character-level representations for part-of-speech tagging. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1818–1826, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/santos14.html.

[28] Dokook Choe, Rami Al-Rfou, Mandy Guo, Heeyoung Lee, and Noah Constant. Bridging the gap for tokenizer-free language models, 2019. URL https://arxiv.org/abs/1908.10322.

[29] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation, 2016. URL https://arxiv.org/abs/1603.06147.

[30] Thomas Hofmann Jason Lee, Kyunghyun Cho. Fully character-level neural machine translation without explicit segmentation. 2017. URL http://arxiv.org/abs/1610.03017.

[31] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[33] Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. Charbert: Character-aware pre-trained language model. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.4. URL https://doi.org/10.18653%2Fv1%2F2020.coling-main.4.

[34] Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. 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 (article), dec 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.609. URL https://www.aclweb.org/anthology/2020.coling-main.609.

[35] Joshua Tanner and Masato Hagiwara. Shiba: Japanese canine model. https://github.com/octanove/shiba, 2021.

[36] Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual), April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wanlp-1.1.

[37] Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale, 2022.

[38] Haitham Seelawi, Ibraheem Tuffaha, Mahmoud Gzawi, Wael Farhan, Bashar Talafha, Riham Badawi, Zyad Sober, Oday Al-Dweik, Abed Alhakim Freihat, and Hussein Al-Natsheh. Alue: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184, Kyiv, Ukraine (Virtual), April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.wanlp-1.18.

Authors

Abdulelah Alkesaiberi currently pursuing a Master's degree in Computer Science at King Abdulaziz University, with a keen research interest in Natural Language Processing

(NLP) and Arabic language modeling. His academic journey in this field is driven by a passion for exploring the intricacies of language and the application of advanced computational techniques to enhance language understanding and processing. His research primarily focuses on the development and optimization of models for Arabic language processing, leveraging the latest advancements in artificial intelligence and machine learning. As a part of my Master's thesis, he's delving into the challenges of Arabic dialect processing and the implementation of effective strategies for improved language models. His work aims to contribute to the growing body of knowledge in NLP, specifically in the context of Arabic language, and to develop solutions that address the unique linguistic characteristics of Arabic and its dialects

Dr. Ali Alkhathlan serving as an Assistant Professor at King Abdulaziz University, specializes in Arabic Natural Language Processing, applying cutting-edge Artificial Intelligence techniques. He obtained his Ph.D. from the University of Colorado at Colorado Springs, with a research focus on Arabic Word Sense Disambiguation, and holds a Master's degree in Computer Science from the University of Colorado Denver. His recent research includes the classification of Arabic text, particularly in identifying abusive content and stance detection. Currently, he is exploring the use of Large Language Models to advance the computational understanding and processing of Arabic text.

Dr. Ahmed Abdelali is a Senior Research Scientist at SDAIA (Saudi Data and Artificial Intelligence Authority). His research interest focuses on natural language processing namely, areas of machine translation, information retrieval, and extraction with emphasis on applications related to Arabic language and its dialects. Dr. Abdelali received his MS and Ph.D. in Computer Science from the New Mexico Institute of Mining and Technology. He worked as a Senior Software Engineer at Qatar Computing Research Institute, Arabic Language Technologies research group before joining SDAIA. Dr. Abdelali has published and co-authored several research papers and articles in various peer-reviewed conferences and journals.