Exploring Arabic Pre-Trained Language Models for Arabic Abstractive Text Summarization

Dhuha Alqahtani * , Maha Al-Yahya §
Department of Information Technology, College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
* 441203741@student.ksu.edu.sa , §malyahya@ksu.edu.sa

2023 Tenth International Conference on Social Networks Analysis, Management and Security (SNAMS) | 979-8-3503-1890-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/SNAMS60348.2023.10375464

Abstract— **Abstractive text summarization (ABS) is the task of providing a concise and meaningful summary for a given text. It is accomplished by first understanding the text and then rephrasing it in a shorter form, highlighting the main points of the original text. ABS is useful in applications such as news aggregators, article summarizers, legal-case summary production, business-meeting summarization, and social-media summarization. There are many types of ABS such as headline summaries, highlight summaries and full summaries. Compared to research on ABS in other languages, study of its use for the Arabic language is still limited. Moreover, very few works explore the use of pretrained transformer-based models for Arabic language ABS. This study investigates the effectiveness of pretrained transformer-based models for the downstream task of Arabic ABS. We present several experiments to fine-tune pretrained transformer model AraBART for Arabic text summarization. The used datasets are AHS dataset for headline summaries, WikiLingua dataset for highlight summaries, and XL-Sum dataset for full summaries. We report on the performance of the models using the ROUGE, BLEU and BERTScore metrics for ABS evaluation. We compare the results of the models with the state of the art for Arabic ABS. The best performance obtained with our experiments is the AraBART model fine-tuned on the AHS dataset to generate headline summaries, with the following results: ROUGE-1=55, ROUGE-2=40.15, ROUGE-L=54.55, BLEU=56.26, and BERTScore=88.06.**

Keywords*— *Arabic language, Abstractive text summarization, Headline Summary, Highlight Summary, Full summary, Pre-trained language model, Transformers, AraBART.

I. INTRODUCTION

Automatic text summarization (ATS) is an important task in the field of NLP research. It is a technique used to extract, and possibly rephrase, the important parts of a large piece of text, thus generating a concise and summarized version. Automatic text summarization can be used in many fields and for various purposes. For example, it can be used in academia to summarize research articles and highlight important findings. It can also be used in healthcare to provide summaries for specific patient cases and patient histories. Moreover, it can also be used to summarize news articles or provide headlines.

In general, ATS systems are generally classified into three classes: (1) extractive summarization, (2) abstractive summarization, and (3) hybrid summarization. Extractive summarization works by choosing the most important sentences and concatenating them to formulate the summary. It is simple and straightforward, but has no textual coherence, which means it lacks fluency and flow [1] [2]. By contrast, abstractive summarization works on understanding the semantics of the sentences while extracting the important information and then builds the summary, possibly using different wordings. Hybrid summarization is a combination of both, extractive and abstractive.

Research on abstractive summarization approaches is less common than research on extractive approaches [3], as it is more complex and difficult to generate, and uses advanced NLP techniques. ATS systems can also be categorized based on the number of documents used per summary; it can be either single-document summarization or multi-document summarization. Single-document summarization aims to summarize one document, whereas multi-document aims to produce a summary for several documents. This study focuses on single-document summarization since this is the most commonly required type.

Looking specifically at Abstractive Text Summarization (ABS), it is usually classified based on the length of the generated summary into: headline, sentence-level, highlights, or full summary [4]. Headline summary aims to capture the main idea of the text in a headline that consists of a few words or phrases. Sentence-level summary aims to provide a summary of the text by extracting the most important sentences or words. Highlights summary aims to capture key details from the text while covering the significant aspects of the content. The full summary aims to provide a comprehensive overview of the entire text.

Recently, there has been a large volume of published research in the field of ABS for English and other languages [5]. However, for Arabic language, there is a clear lack of studies [3]. This is due to the complexity of Arabic's morphology, structure, and syntax [2]. Moreover, studies on Arabic summarization are mainly limited to extractive summarization approaches, rather than abstractive approaches [6]. Abstractive approaches are needed because ABS deals with understanding the semantics of the text [3], however, they are challenging, which has limited the research in this field. Therefore, in this study we aim to fill this gap and investigate new approaches to Arabic ABS using pre-trained transformer-based models [7].

This study focuses on the automated generation of headline summary, highlights summary, and full summary using current state of the art pretrained transformer models. The transformer is a deep learning model that is based on the self-attention mechanism, presented in [8], which was initially used for the task of translation. The Bidirectional Auto-Regressive Transformers (BART) model employs a standard transformer-based neural machine translation architecture. Two main research questions are addressed in this study:

1. To what extent can pre-trained transformer models provide satisfactory performance for the task of Arabic ABS?
2. What impact does the generated summary length has on the performance of the models generated for the Arabic ABS?

The remainder of the paper is organized as follows: Section II presents relevant work in the area of abstractive text summarization for the Arabic language. Section III presents our research methodology, including model selected, datasets, and evaluation metrics. Section IV presents the experimental setup and details of the experiments conducted. Section V presents the findings and discusses the results obtained. Finally, Section VI presents our conclusions and recommendations for future work on Arabic ABS.

II. Related Work

Research in the field of ABS text summarization has recently witnessed rapid growth for many languages. Although the number of studies on Arabic is not comparable to studies for English and other languages, there has been significant advancement in the field. We first review studies that address abstractive summarization using hybrid approaches combining abstractive and extractive summarization. Next, we review studies based on deep neural networks and sequence to sequence methods, and finally studies which use transformers.

Regarding abstractive approaches to Arabic text summarization, some studies use a hybrid approach such as studies presented in [9], [10], and [11]. The method described in [9] applies as a first step of extractive summarization using Rhetorical Structure Theory (RST) and sentence scoring, and then uses a rule-based abstractive approach. An abstractive summary is produced by using reduction rules which remove unnecessary clauses such as words, clauses, and sub-sentences from the extractive summary. The data for training and testing the approach is collected in-house by the authors. A total of 150 documents were collected from six Arabic newspapers from different countries. The authors report results on producing a summary of size 31% of the original input text. The performance achieved is as follows: 75 for ROUGE-1, 65 for ROUGE-2, and 71 for ROUGE-L. The quality of the generated summaries was also reviewed and evaluated by 2 Arabic language experts and achieved a score of 4.53/5, which indicates excellent level of quality of the generated summaries. The high values for the ROUGE may be due to using extractive summarization before the abstractive step.

As for studies using deep learning and neural models for Arabic ABS, the studies are limited [7]. The study presented in [12] uses a dataset which is built by the authors, the Arabic Headline Summary (AHS). It consists of 300K entries and is collected from Arabic Articles of Mawwdoo3 website (https://mawdoo3.com). The introduction of the article on the website is considered as the document to be summarized and the title is considered as the headline summary. The method the authors used is an abstractive approach that uses a deep learning method based on a neural network by using a sequence-to-sequence encoder-decoder model framework. It achieves a score of 44.23 for ROUGE-1 F-measure, which is the highest when compared by the authors to other similar studies. Another study using deep learning is presented in [13] using a seq2seq encoder-decoder models with attention. Different models are tested which are Corner-Stone model, Pointer-Generator model, Scheduled-Sampling model, and Policy-Gradient model. Models are trained and evaluated on Arabic and English datasets including CNN-Daily Mail dataset of English language, an Arabic news articles, and Saudi newspapers articles. The results for Arabic data by using advanced cleaning with Corner-Stone model is 60.79 of ROUGE-1, 41.28 of ROUGE-2, 50.08 of ROUGE-L, and 46.60 of BLEU. Using the Scheduled-Sampling model the results are: 52.97 of ROUGE-1, 36.68 of ROUGE-2, 45.82 of ROUGE-L, and 44.40 of BLEU. But when reviewed by humans, the Scheduled-Sampling produces better quality sentences than the Corner-Stone method. The results are good, and the model achieves high scores in ROUGE.

Another study using deep learning integrates extractive and abstractive summarization is presented in [10]. The authors propose a hybrid summarizer of an ensemble learning with soft voting of the following classifiers: SVM, Random Forest, and Logistic Regression for the extractive summaries. For abstractive summarization the authors use a bidirectional encoder-decoder LSTM model. The extractive model extracts and selects the statistical and semantic features to improve the performance of the abstractive model. The datasets used for this study are the EASC [14], and Abu El-Khair datasets [15], Abu El-Khair dataset is collected from 10 Arabic newspapers and consist of 5M articles, each with a length of 100 words. The authors report a performance of 54.3 for ROUGE-1 F-measure and 53.7 for ROUGE-2 F-measure. The results are excellent as compared to other studies in the area.

The study presented [6] demonstrates an application of a deep neural network (DNN) approach using a decoder-encoder sequence-to-sequence model. The study aims to compare three techniques of DNN, namely Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM). It also presents a comparison between two words embedding techniques, skip-gram and continuous bag of words (CBOW) word2Vec. The architecture which obtained the best results is a model composed of three layers of BiLSTM in the encoder. Also, the results show that using skip-gram word2Vec is better than CBOW word2Vec. To improve the performance, the authors use AraBERT pre-processing for dataset pre-processing phase. The datasets used for this study are the AHS [12] and the Arabic Mogalad Ndeef (AMN) [13]. The AMN is a dataset of size 265K collected from news articles and the accompanying summaries. For the AHS dataset, the authors report the best performance as follows: For the AHS dataset, ROUGE-1 F-measure is 51.49, 12.27 for ROUGE-2, 34.37 for ROUGE-L, and a BLEU score of 41. For

the AMN dataset, the best performance as follows: 44.2 for ROUGE-1, 18.35 for ROUGE-2, 32.46 for ROUGE-L and a BLEU score of 41.

The recent development of pretrained transformer models that can be fine-tuned for downstream tasks have encouraged studies for Arabic ABS. One of the earliest work reported using transformer fine-tuning for Arabic ABS uses a transformer model for extractive and abstractive summarization [7]. By fine-tuning a pre-trained BERT model for English and the multilingual BERT(M-BERT) for Arabic text summarization. The training is applied on the CNN/Daily-Mail news dataset for English language and on the KALIMAT dataset [16] for Arabic. The highest performance achieved for English is: 30.35 for ROUGE-1, 11.33 for ROUGE-2, and 25.3 for ROUGE-L. The highest performance achieved for Arabic is: 12.21 for ROUGE-1,4.36 for ROUGE-2, and 12.19 for ROUGE-L. The results for Arabic are very low as compared to other studies. A hybrid approach [11] uses a transformer model fine-tuned on the EASC dataset [14] to learn Arabic ATS. In this study, the AraBERT transformer model [17] is used for extractive summarization, and the output is then used to fine tune an mT5 transformer model [18] for abstractive summarization. For extractive summarization, the authors report a performance of ROUGE-L: with a precision of 53, a recall of 55, and an F-measure of 49. As for the abstractive summarization, reported results are the human evaluation which yielded a total score of 3/5 measuring the quality of the generated summary. The results are good and using human evaluation 3 out of 5 adds to the reliability of the results.

Recently, a new pre-trained Arabic transformer model has been published [1]. The authors released a sequence-to-sequence pre-trained model for the Arabic language based on the BART Transformer model [19], they named it AraBART [20]. The model was finetuned for the abstractive summarization task. The authors used several well-known datasets for ATS tasks: Gigaword [21] and XL-Sum dataset [22]. AraBART is compared with other models: BERT2BERT model, mBART25, and mT5 model. The results obtained by using AraBART model outperforms the other models. The authors report the best performance as follows: 42.4 for ROUGE-1, 28.8 for ROUGE-2, 40.3 for ROUGE-L, and a BERTScore of 69.8. The results are promising, and using the BERTScore is very suitable for abstractive summarization, better than ROUGE, since it reflects the semantic similarity between sentences.

The AraBART transformer model was also used as a part of the experiments in this review study [2]. The review compares it with other models (mT5, PEGASUS-XSum, PEGASUS-Large, mBART-Large) by using various datasets: AHS [23], WikiHow Dataset[24], Arabic News Articles (ANA)[13]. From the results of the study, the PEGASUS-Large transformer using the WikiHow dataset outperforms the other models including AraBART, it achieves 94.62 for ROUGE-1, 88.72 for ROUGE-2 and 94.58 for ROUGE-L. The authors report that the achieved results are high as compared to others because of the nature of the model. The study in [25] uses two different architectures, RNN-based and transformer-based by using different pre-trained language models, including mBERT, AraBERT, AraGPT2, and AraT5 for Arabic abstractive summarization. The authors built an Arabic summarization dataset of 84,764 text-summary pairs, SumArabic dataset. They collect the data from two newspapers:

1. Emaratalyoum(https://www.emaratalyoum.com/)
2. Aalmamlakatv (https://www.almamlakatv.com/)

In their work, they report that the transformer-based architecture outperforms the RNN-based architecture. They found that AraT5 outperforms the other used models, indicating that an encoder-decoder pre-trained transformer is more suitable for summarizing Arabic text. The evaluation metrics and their corresponding values are as follows: (ROUGE-1 F-measure: 49.06, ROUGE-2 F-measure: 30.81, ROUGE-L F-measure: 46.87). However, the study evaluation is on news articles, and its performance may vary for other types of texts.

Regarding datasets, Arabic ABS Studies in the literature have used various datasets to train and test text summarization models (Table I). Regarding evaluation metrics used for ABS, the popular metrics are as follows: The ROUGE metric, based on precision, recall and F-measure, with all its variants such as ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-L-SUM, The BLEU metric, and the BERTScore metric. Formulas for the metrics can be found in Section III of this paper.

TABLE I. ARABIC TEXT SUMMARIZATION DATASETS

Dataset Name	Dataset size	Document (Single/ Multi)	Summarization type
AHS [23]	300,000	Single	Abstractive
ANA [13]	265,000	Single	Abstractive
EASC [14]	153	Single	Extractive
Abu El-khair [15]	5 million	Single	Extractive
AMN [13]	265,000	Single	Abstractive
KALIMAT [16]	20,291 single 2,057 multi	Single + multi	Extractive
Gigaword [21]	2,716,995	multi	Abstractive
XL-Sum [22]	46,897	Single	Abstractive
WikiLingua [24]	29,229	Single	Abstractive

To summarize our review of related work, we can observe that for ABS, the studies are limited. Moreover, transformer-based models are the state of the art for English and many other languages, however, for Arabic, a handful of studies [1] [2][25] have investigated the use of pre-trained language models for the downstream task of ABS. With regard to standard datasets, the most popular datasets for Arabic are based on corpora of news articles, looking at other kinds of datasets, as we will investigate, will provide more insights into the performance of ABS. A drawback of the available datasets is that they are mainly designed for extractive summarization studies [1]. Therefore, this study will target the gap in the research literature by investigating new transformer models for Arabic ABS.

III. METHODOLOGY

We approach abstractive text summarization by fine-tuning the AraBART transformer [1] for the downstream task of ABS. AraBART stands for Arabic BART. The AraBART language model is based on the BART architecture [19]; it has 139 million parameters and 6 layers for both encoder and decoder. An

additional layer is added in AraBART for normalization as proposed in mBART [26].

To answer the first research question, we first find the optimal parameter values for finetuning AraBART. Next, we test the model on the best parameters using the AHS dataset [23], and evaluate the results on three different sizes of the AHS dataset (small -3K, medium-30K, large-300K) to find the best model. To answer our second research question, and investigate the impact of summary length on performance, we finetune AraBART on three different kinds of summary lengths using three different datasets AHS [23] (headline summary), Wikilingua [24] (highlight summary), and XL-Sum [22] (full summary) described below, and we evaluate the results.

The Arabic Headline Summary (AHS) [23] is a relatively new dataset, published in 2020, that differs in type of text from other datasets traditionally used by researchers. The traditional datasets used in studies in the literature are usually news articles. However, the AHS dataset is a collection of short informative articles on diverse topics collected from the Arabic mawdoo3 website (https://mawdoo3.com). It is suitable for single-document summarization and for abstractive summarization. It also includes headline summarization. It consists of approximately 300,000 entries, and each entry includes a text segment that is extracted from the article's introduction section, as well as an associated headline which is the article title (the abstractive summary). The average length of an input text in the dataset is 83.1 tokens, and the average length of a headline is 3.3 tokens. The maximum length of an input text is 1334 tokens, and the maximum length of a summary is 28 tokens.

The Wikilingua dataset [24] is an abstractive summarization dataset; it consists of 770000 pairs of text and summary for 18 languages. The Arabic part contains 29,229 entries which is scraped from Wikihow (https://www.wikihow.com) that provides guides on how to do anything as a set of numbered instructions. The summary is human written as points, so it can be used for highlight summary experiments. The average length of the text is 337 tokens, and the average length of the summary is 29 tokens. While the maximum length of the text is 1835 tokens and the maximum length of the summary is 195 tokens.

XL-Sum dataset [22] is one of the most popular datasets used in summarization tasks, it is a multilingual dataset, and the English version is named X-Sum. XL-Sum is suitable for abstractive summarization with single document, where the summaries are human generated. It is collected from the BBC news website, and consists of 44 languages, with a size of 1 million pairs of texts and their summaries. The size of the Arabic language set is 46897 pairs [22]. It contains text, full summary, and titles; we use the text and full summary. It is publicly available in JSON format. The average length of the text is 429 tokens, and the average length of the summary is 25 tokens. The maximum length of the text is 8149 tokens, and the maximum length of the summary is 236 tokens.

For evaluation of summarization tasks, the F-measure is computed using the ROUGE score (Recall Oriented Understudy for Gisting Evaluation) [27], a popular metric for evaluating abstractive summarization systems. The ROUGE score compares the system-produced summary with the reference summary (human created) to determine its quality. There are several variants of the ROUGE score including ROUGE-1, ROUGE-2, and ROUGE-L, which are used to evaluate the performance of summarization systems. ROUGE-1 is used to measure the overlap in unigram, whereas ROUGE-2 is for bi-grams, and ROUGE-N is for n-grams. ROUGE-L is based on the longest common subsequence (LCS) which is the longest sequence which is common among two sentences. In this study, we use ROUGE-1, ROUGE-2, and ROUGE-L, computed using F-score. The equations below show how these metrics are computed:

$$ROUGE-N = \frac{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count(gram_n)} \text{ [27]}$$

$$Precision = \frac{number\ of\ overlapping\ words\ between\ both\ summaries}{total\ words\ in\ reference\ summary} \text{ [12]}$$

$$Recall = \frac{number\ of\ overlapping\ words\ between\ both\ summaries}{total\ words\ in\ generated\ summary} \text{ [12]}$$

$$F-measure = 2 * \frac{Precision*Recall}{Precision+Recall} \text{ [12]}$$

We also use the BLEU score (BiLingual Evaluation Understudy)[28] to measure the model's performance. The BLEU score is the reverse of the ROUGE score. It is mostly used in evaluating machine translation tasks; however, it has also been reported to be used in evaluating summarization tasks. The BLEU score is based on a calculation of the matched n-grams in the candidate sentence that is found in the reference sentence. The equation is shown below, where BP is the brevity penalty, *pn* is n-gram precisions, N is the length of n-grams, and *wn* is the positive weights.

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n \log p_n!) \quad \text{[28]}$$

Since the ROUGE and BLEU scores are mainly used for extractive text summarization and do not take into account implicit semantics (the use of different words for the same meaning), we use another score to evaluate our model, the BERTScore [29]. The BERTScore is based on using BERT contextual embeddings. The BERTScore computes the level of similarity (and not the matching) of tokens in both candidate and reference sentences. This is similar to human judgments, which makes the BERTScore a suitable score for our task. The equation is shown below. FBERT, RBERT, and PBERT are BERTScore's F-measure, recall, and precision, respectively. More details about the metrics can be found in [29].

$$F_{BERT} = 2 * \frac{P_{BERT}*R_{BERT}}{P_{BERT}+R_{BERT}} \text{ [29]}$$

IV. Experiments

All experiments have been conducted in the Google Colab environment using Python. To answer the first research question, several experiments were conducted varying batch size, number of epochs, and learning rate to find the optimal for finetuning ArabART for ABS. Then, to investigate the impact of dataset size on model performance for ABS, we conduct experiments using different dataset sizes. In each experiment, AraBART was finetuned with three sizes of dataset: small (3,000), medium (30,000), and large (300,000—full dataset) using AHS dataset. All parameters were set up to the default for the model, however, we initially experimented on a sample of 2,000 texts from the dataset (2K) with several different values for batch size, number of epochs, and learning rate. The results are shown in Tables III, IV, and V. For the maximum input

sequence length, we set a value of 512 tokens, and for the output sequence length we set a value of 10 tokens. These values were chosen based on the statistics of the dataset. It is important to note that with 4 epochs for the large dataset, it took almost 12 hours using a Google Colab pro-plus account. For the medium dataset, it took 3.5 hours. For the small dataset, it took half an hour to complete the experiment and fine-tune the model. We named the best model after fine-tuning the *AraHeadline model*, which was finetuned on the AHS full dataset by using AraBART vanilla version as presented in [1].

Finally, to answer our second research question and explore the impact of summary length on performance, we finetune AraBART for three kinds of summaries: (1) headline summary, (2) highlight summary and (3) full summary type. The experiments were conducted using the three datasets: AHS (headline summary), WikiLingua (highlight summary), and XL-Sum (full summary). We finetuned AraBART using 3K entries from each dataset.

In all experiments, the dataset split is 70% for training, 15% for evaluation, and 15% for testing. The best models for all three kinds of summaries were obtained using the following parameters:

- Headline best model, *AraHeadline*, was obtained using a batch size of 16, a learning rate of 5e-5, and a total of 4 epochs.
- Highlight summary best model was obtained using a batch size of 4, a learning rate of 5e-5, and a total of 4 epochs.
- Full summary best model best model was obtained using a batch size of 1 (to avoid out of memory errors), a learning rate of 5e-5, and a total of 4 epochs.

V. Results and Discussion

From the experiments, we observed that the best batch size was 16 (Table II); the best values for the number of epochs were 4 and 8 (Table III); and the best learning rate was 5.00E-05 (Table IV). These values were used for fine-tuning *AraHeadline*. Regarding the number of epochs, we used the minimum that showed good performance, which was 4 epochs.

TABLE II. Model Performance on (2K) with respect to the batch size in terms of Rouge-1 as R1, Rouge-2 as R2, Rouge-L as RL, and BertScore as BS.

Batch size	R1	R2	RL	BLEU	BS
2	49.4%	**34.12%**	49.4%	54.34%	86.43%
4	47.48%	32.64%	47.35%	54.61%	86.27%
8	48.87%	32.75%	48.5%	53.37%	86.22%
16	**50.02%**	33.99%	**49.82%**	53.52%	**86.44%**
32	45.52%	28.53%	44.76%	**55.71%**	85.41%

TABLE III. Model Performance on 2K with respect to number of epochs in terms of Rouge-1 as R1, Rouge-2 as R2, Rouge-L as RL, and BertScore as BS.

No of Epochs	R1	R2	RL	BLEU	BS
1	47.67%	31.32%	47.67%	59.62%	86.85%
2	49.06%	33.63%	48.55%	57.25%	86.48%
3	49.61%	34.33%	49.06%	55.78%	86.64%
4	**50.18%**	**37.64%**	**50.08%**	**55.8%**	**86.93%**
5	48.28%	33.08%	48%	55.69%	86.41%
6	49.99%	34.09%	49.31%	55.04%	86.8%
7	48.74%	32.66%	48.66%	56.35%	86.78%
8	**51.63%**	**35.32%**	**51.37%**	**56.51%**	**87.38%**
9	48.78%	34.35%	48.78%	53.48%	86.15%

TABLE IV. Model Performance on (2K) with respect to the learning rate in terms of Rouge-1 as R1, Rouge-2 as R2, Rouge-L as RL, and BertScore as BS.

Learning rate	R1	R2	RL	BLEU	BS
1.00E-05	44.73%	30.09%	44.6%	58.77%	85.76%
2.00E-05	45.65%	29.47%	45.5%	54.87%	85.55%
5.00E-05	***47.67%***	***31.32%***	***47.67%***	***59.62%***	***86.85%***
5.00E-06	45.11%	30%	45%	56.82%	85.8%

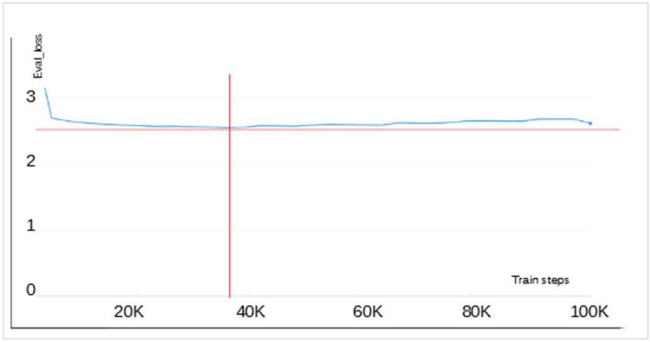

Fig. 1. Loss function values tracking with respect to the number of train steps (during training with 8 epochs on the large dataset size)

Fig. 1 shows the loss function during the training of the *AraHeadline* large model with 8 epochs. The loss function is determined in the training arguments as the metric used to choose the best model. The intersection point in the chart shows the minimum value of the loss function when using the large (300,000) dataset. The lowest value was in step 38,000 which was in epoch 3 specifically in "epoch": 2.95. From the chart shown, we can also observe that there is no need for 100,000 steps (8 epochs) because the best model was obtained in step 38,000. In the experiment with a medium dataset (30,000), the best model was obtained in epoch 4. So, we observe that using 4 epochs in our experiments is the most suitable, since it is the smaller number of epochs that achieve better results by having less loss function.

To compare the performance of the model in terms of the size of the dataset, we ran several experiments on different dataset sizes. Table V shows the results obtained. The best model was the one fine-tuned using the complete dataset. This

means that dataset size is an important factor in fine-tuning for the downstream task of ABS. This finding is in support of the findings presented in the literature [30].

TABLE V. COMPARING PERFORMANCE USING DIFFERENT DATASET SIZES IN TERMS OF ROUGE-1 AS R1, ROUGE-2 AS R2, ROUGE-L AS RL, AND BERTSCORE AS BS.

Dataset size	R1	R2	RL	BLEU	BS
Small 3K	45.84%	31.87%	45.65%	54.82%	85.42%
Medium 30K	51.11%	36.0%	50.77%	55.89%	86.98%
Large 300K	***55***%	***40.15***%	***54.55***%	***56.26***%	***88.06***%

Table VI shows the results of *AraHeadline* compared to other similar Arabic ABS models reported in the literature. The studies presented in [12] and [2] use the AHS dataset, the same dataset used to build *AraHeadline*. However, [12] used a different approach, a seq2seq deep-learning neural-based approach. Meanwhile, the study presented in [2] uses the AraBART model, the same base model used to build *AraHeadline*. It is noticeable that the *AraHeadline* model outperforms the others with regard to the F-measure ROUGE-1, ROUGE-2, and ROUGE-L.

We compare the performance of the *AraHeadline* model with the AraBART baseline model, with the same dataset split of AHS by using the evaluate built-in method. The results are presented in Table VII. *AraHeadline* improves upon the AraBART baseline model with regard to all metrics.

To study the impact of summary length on the performance of the fine-tuned ABS model, we set up three experiments on three summarization dataset types: XL-Sum for full summary, WikiLingua for highlight summary, and AHS for headline summary. From Table VIII we can see that with shorter summary length, the performance increases. However, the results show that XL-Sum (full summary) is better than WikiLingua (highlight), this might be due to the fact that the XL_SUM dataset is one of the datasets used to pretrain the base model, AraBART [1]. In addition, AraBART is pre-trained and designed for Arabic language.

TABLE VI. COMPARING ARAHEADLINE TO OTHER MODELS IN THE LITERATURE IN TERMS OF ROUGE-1 AS R1, ROUGE-2 AS R2, AND ROUGE-L AS RL.

Model	R1	R2	RL
AraHeadline	**55**%	**40.15**%	**54.55**%
[12]	44.23%	NA	NA
[2]	34.74%	17.50%	34.08%

TABLE VII. PERFORMANCE OF ARAHEADLINE WITH ARABART BASELINE IN TERMS OF ROUGE-1 AS R1, ROUGE-2 AS R2, ROUGE-L AS RL, AND BERTSCORE AS BS.

Model	R1	R2	RL	BLEU	BS
AraBART baseline	17.17%	7%	16.42%	14.23%	67.36%
AraHeadline	**55**%	**40.15**%	**54.55**%	**56.26**%	**88.06**%

TABLE VIII. RESULTS OF FINETUNING ARABART ON HEADLINE, HIGHLIGHT, AND FULL SUMMARY DATASETS IN TERMS OF ROUGE-1 AS R1, ROUGE-2 AS R2, ROUGE-L AS RL, AND BERTSCORE AS BS.

Summary Type	R1	R2	RL	BLEU	BS
Headline	**45.84%**	**31.87%**	**45.65%**	**54.82%**	**85.42%**
Highlight	24.06%	10.06%	23.81%	32.27%	78.13%
Full	31.77%	18.81%	29.63%	28.21%	78.84%

VI. CONCLUSIONS AND FUTURE WORK

In this study, we investigated the effectiveness of pretrained transformer models in addressing ABS for the Arabic language. The best fine-tuned model called, *AraHeadline,* as it produces a headline summary. The used dataset is the AHS dataset [23], we also conduct experiments with three sizes: small, medium, and large to study the effect of dataset size on finetuning for ABS. We observed that increasing the dataset size increases the performance of the fine-tuned model as a direct correlation relationship. Moreover, we conducted several experiments to study the impact of the length of generated summary (headline, highlight and full summary) on model performance, we find that the headline summary achieves the best results.

The performance of the system is measured using ROUGE F-measure, BLEU, and BERTScore metrics. AraHeadline gave the following results: ROUGE-1=55, ROUGE-2=40.15, ROUGE-L=54.55, BLEU=56.26, and BERTScore=88.06. These encouraging results from the experiments and the final *AraHeadline* model show that transformer-based models can be fine-tuned for the downstream task of Arabic abstractive headline summarization with satisfactory performance. Our results indicate that using transformer-based models for the improvement of Arabic summarization systems is promising and will yield results close to human performance.

For future work, we plan to conduct more experiments to explore other transformers such as mT5 [31] and AraT5 [32]. We also plan to investigate summarization for other types of text, such as Arabic scientific text, and to determine whether *AraHeadline* can be used to summarize different parts of scientific texts and combine these short summaries into a meaningful coherent full summary.

REFERENCES

[1] M. K. Eddine, N. Tomeh, N. Habash, J. L. Roux, and M. Vazirgiannis, "AraBART: a Pretrained Arabic Sequence-to-Sequence Model for Abstractive Summarization." arXiv, Mar. 21, 2022. doi: 10.48550/arXiv.2203.10945.

[2] H. Chouikhi and M. Alsuhaibani, "Deep Transformer Language Models for Arabic Text Summarization: A Comparison Study," *Applied Sciences*, vol. 12, no. 23, Art. no. 23, Jan. 2022, doi: 10.3390/app122311944.

[3] L. M. Al Qassem, D. Wang, Z. Al Mahmoud, H. Barada, A. Al-Rubaie, and N. I. Almoosa, "Automatic Arabic Summarization: A survey of methodologies and systems," *Procedia Computer Science*, vol. 117, pp. 10–18, Jan. 2017, doi: 10.1016/j.procs.2017.10.088.

[4] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, p. 113679, Mar. 2021, doi: 10.1016/j.eswa.2020.113679.

[5] N. Ibrahim Altmami and M. El Bachir Menai, "Automatic summarization of scientific articles: A survey," *Journal of King Saud*

University - Computer and Information Sciences, vol. 34, no. 4, pp. 1011–1028, Apr. 2022, doi: 10.1016/j.jksuci.2020.04.020.

[6] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, "Abstractive Arabic Text Summarization Based on Deep Learning," *Computational Intelligence and Neuroscience*, vol. 2022, p. e1566890, Jan. 2022, doi: 10.1155/2022/1566890.

[7] K. N. Elmadani, M. Elgezouli, and A. Showk, "BERT Fine-tuning For Arabic Text Summarization," *arXiv:2004.14135 [cs]*, Mar. 2020, Accessed: Mar. 15, 2021. [Online]. Available: http://arxiv.org/abs/2004.14135

[8] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Feb. 19, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[9] A. M. Azmi and N. I. Altmami, "An abstractive Arabic text summarizer with user controlled granularity," *Information Processing & Management*, vol. 54, no. 6, pp. 903–921, Nov. 2018, doi: 10.1016/j.ipm.2018.06.002.

[10] A. Fadel and G. Esmer, "A Hybrid Long Arabic Text Summarization System Based on Integrated Approach Between Abstractive and Extractive," Apr. 2020, pp. 109–114. doi: 10.1145/3397125.3397129.

[11] A. Reda *et al.*, "A Hybrid Arabic Text Summarization Approach based on Transformers," in *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, May 2022, pp. 56–62. doi: 10.1109/MIUCC55081.2022.9781694.

[12] M. Al-Maleh and S. Desouki, "Arabic text summarization using deep learning approach," *J Big Data*, vol. 7, no. 1, p. 109, Dec. 2020, doi: 10.1186/s40537-020-00386-7.

[13] A. M. Zaki, M. I. Khalil, and H. M. Abbas, "Deep Architectures for Abstractive Text Summarization in Multiple Languages," in *2019 14th International Conference on Computer Engineering and Systems (ICCES)*, Dec. 2019, pp. 22–27. doi: 10.1109/ICCES48960.2019.9068171.

[14] M. El-Haj, U. Kruschwitz, and C. Fox, "Using mechanical turk to create a corpus of arabic summaries," 2010.

[15] I. A. El-khair, "1.5 billion words Arabic Corpus." arXiv, Nov. 12, 2016. doi: 10.48550/arXiv.1611.04033.

[16] M. El-Haj and R. Koulali, "KALIMAT a Multipurpose Arabic Corpus," p. 4.

[17] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, Marseille, France: European Language Resource Association, May 2020, pp. 9–15. Accessed: Feb. 25, 2023. [Online]. Available: https://aclanthology.org/2020.osact-1.2

[18] L. Xue *et al.*, "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer," presented at the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 483–498.

[19] M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.

[20] "moussaKam/AraBART · Hugging Face." https://huggingface.co/moussaKam/AraBART (accessed Feb. 19, 2023).

[21] Parker, Robert, Graff, David, Chen, Ke, Kong, Junbo, and Maeda, Kazuaki, "Arabic Gigaword Fifth Edition." Linguistic Data Consortium, p. 3286401 KB, Oct. 21, 2011. doi: 10.35111/P02G-RW14.

[22] T. Hasan *et al.*, "XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, Aug. 2021, pp. 4693–4703. doi: 10.18653/v1/2021.findings-acl.413.

[23] M. Al-Maleh, "An Arabic Dataset Used for Article Headline Generation Using Deep Learning Techniques," Jan. 2020, Accessed: Mar. 15, 2021. [Online]. Available: https://osf.io/btcnd/

[24] F. Ladhak, E. Durmus, C. Cardie, and K. McKeown, "WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 4034–4048. doi: 10.18653/v1/2020.findings-emnlp.360.

[25] M. Bani-Almarjeh and M.-B. Kurdy, "Arabic abstractive text summarization using RNN-based and transformer-based architectures," *Information Processing & Management*, vol. 60, no. 2, p. 103227, Mar. 2023, doi: 10.1016/j.ipm.2022.103227.

[26] Y. Liu *et al.*, "Multilingual Denoising Pre-training for Neural Machine Translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, Nov. 2020, doi: 10.1162/tacl_a_00343.

[27] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. Accessed: Sep. 21, 2023. [Online]. Available: https://aclanthology.org/W04-1013

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, in ACL '02. USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. doi: 10.3115/1073083.1073135.

[29] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," presented at the International Conference on Learning Representations, Mar. 2020. Accessed: Feb. 18, 2023. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[30] Z. Zhao and P. Chen, "To Adapt or to Fine-tune: A Case Study on Abstractive Summarization," in *Proceedings of the 21st Chinese National Conference on Computational Linguistics*, Nanchang, China: Chinese Information Processing Society of China, Oct. 2022, pp. 824–835. Accessed: Feb. 19, 2023. [Online]. Available: https://aclanthology.org/2022.ccl-1.73

[31] L. Xue *et al.*, "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*, Online: Association for Computational Linguistics, Jun. 2021, pp. 483–498. doi: 10.18653/v1/2021.naacl-main.41.

[32] E. M. B. Nagoudi, A. Elmadany, and M. Abdul-Mageed, "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)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 628–647. doi: 10.18653/v1/2022.acl-long.47.