applied sciences

MDPI

Article

Deep Transformer Language Models for Arabic Text Summarization: A Comparison Study

Hasna Chouikhi [1] **and Mohammed Alsuhaibani** [2,*]

[1] LIMTIC Laboratory, UTM University, Tunis 1068, Tunisia
[2] Department of Computer Science, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia
* Correspondence: m.suhibani@qu.edu.sa

Abstract: Large text documents are sometimes challenging to understand and time-consuming to extract vital information from. These issues are addressed by automatic text summarizing techniques, which condense lengthy texts while preserving their key information. Thus, the development of automatic summarization systems capable of fulfilling the ever-increasing demands of textual data becomes of utmost importance. It is even more vital with complex natural languages. This study explores five State-Of-The-Art (SOTA) Arabic deep Transformer-based Language Models (TLMs) in the task of text summarization by adapting various text summarization datasets dedicated to Arabic. A comparison against deep learning and machine learning-based baseline models has also been conducted. Experimental results reveal the superiority of TLMs, specifically the PEAGASUS family, against the baseline approaches, with an average F1-score of 90% on several benchmark datasets.

Keywords: automatic text summerization (ATS); transformer language models (TLMs); Arabic ATS

Citation: Chouikhi, H.; Alsuhaibani, M. Deep Transformer Language Models for Arabic Text Summarization: A Comparison Study. *Appl. Sci.* **2022**, *12*, 11944. https://doi.org/10.3390/app122311944

Academic Editors: Habib Hamam, Ateeq Ur Rehman and Mohamed Tahar Ben Othman

Received: 3 November 2022
Accepted: 15 November 2022
Published: 23 November 2022

1. Introduction

Automatic Text Summarization (ATS) is the process of extracting and generating a coherent, fluent and meaningful summary by covering the most important information of a given text [1] and is one of the fastest growing fields in Artificial Intelligence (AI), Machine Learning (ML) and Natural Language Processing (NLP). ATS is exponentially growing nowadays due to the vast amount of textual data that arises on a daily basis on the internet, such as the exponentially growing usage of social networks, online newspapers, and user reviews in online stores, to name a few. Alongside such rich sources of textual data, there are also essential textual data available in electronic books and novels, legal and biomedical documents, and scientific papers, amongst many others. In fact, and as an instance of the significant increase in today's internet data, 90% of the data on the internet has been created in the last couple of years [2]. Moreover, more than two billion websites are currently active and hosted somewhere on the internet.

Manually summarizing a text is a costly process in terms of time, cost and effort. Therefore, ATS is considered one of the essential fields in AI, ML and NLP. ATS automatically generates a summary (and reduces the size) of any text. ATS systems were developed as a time-saving method to address the issue of having to read lengthy texts on the same subject in order to understand the main point [3]. In comparison to hiring a qualified human summary, it also costs less. Hence, the need for ATS systems has arisen, which encourages researchers and scientific communities to conduct various research in the field [4,5]. Search engine snippets that are produced after a document is searched and news websites that produce condensed news in the form of headlines to help with browsing are a few examples of applications for ATS [6]. The summarization of clinical and biomedical texts is a further application, in addition to lawsuit abstractions [4].

The methods for ATS are broadly categorized into *extractive*, *abstractive*, or *hybrid* [7]. Some assessment methods call for *extracting* the text's most crucial passages (usually

sentences). Typically, either explicitly or implicitly, the length of the final summary is determined. Therefore, an *extractive* algorithm can, for example, select 10 to 15 essential sentences from a document that contains around 50 phrases [8]. *Abstractive* summarization functions as well as humans. The algorithm reads the text, determines what it says, and then uses word combinations to describe the material. Theoretically, this approach might offer a superior, more condensed memory. In fact, this is challenging since it calls for both correct application of the content and knowledge of it at the level of an educated human reader [9].

In reality, most of the available ATS systems are mainly proposed to summarize texts written in English, with relatively little work being completed in other natural languages. There are fewer attempts on Arabic ATS, despite the fact that Arabic is among the top five most spoken languages in the world, with more than 20 nations using it as their official language and more than 400 million native and non-native speakers [10]. This is owing to the difficulty of the structure, syntactic and morphology of Arabic, as well as the compression ratio seen when summarizing numerous texts as opposed to a single document.

Extractive summarization methods are the common approaches among the timid attempts for Arabic ATS. Such extractive methods produce factual, comprehensible summaries, but they often lack flow and are overly verbose [11]. In order to solve this issue, abstractive models are flexible in their word selection and turn to generalization and paraphrasing in order to produce more fluid and cohesive descriptions. For Arabic abstractive models, which is the main focus of this paper, the architecture of dominant choice is sequence-to-sequence (seq2seq) [12]. For example, Al-Maleh and Desouki [13] use the pointer-generator network [14]. Similarly, Wazery et al. [15] suggest a more general RNN-based approach.

Most recently, and with the development of Transformer Language Models (TLMs) such as Bidirectional Encoder Representations from Transformers (BERT) [16], Bidirectional and Auto-Regressive Transformers (BART) [17], XLNet [18], Robustly Optimized BERT (RoBERTa) [19], Generative Pre-trained Transformer (GPT-3) [20], and Text-To-Text Transfer Transformer (T5) [21], NLP has experienced unprecedented advancements. TLMs can be described as pre-trained contextual language models with multilayer bidirectional self-attention mechanisms. For transformer encoders, pre-training and fine-tuning are the two key processes.

State-of-the-art results for a wide range of NLP tasks, including abstractive ATS [22], are being witnessed nowadays thanks to TLMs [16,19,23,24].

Taking advantage of the breakthrough of TLMs, the literature has seen recent attempts at developing TLMs-based abstractive ATS either as multilingual systems functioning on various natural languages or specifically proposed as monolingual (e.g., Arabic). For example, Kamal Eddine et al. [11] presented AraBART, the first Arabic model based on BART, where the encoder, as well as the decoder, are end-to-end pre-trained. Similarly, Kahla et al. [25] have used pre-trained language models such as multilingual BERT, AraBERT, and multilingual BART by fine-tuning a variety of neural abstractive ATS systems for Arabic.

However, the literature is still lacking a comprehensive comparison among Arabic ATS, which we aim to address in this paper. In particular, the contribution of this work is four-fold:

- A thorough comparison study among all existing abstractive TLMs-based Arabic and Arabic-supported multilingual ATS systems with various evaluation metrics.
- Utilizing various existing diverse Arabic datasets for abstractive ATS, including Arabic Headline Summaries (AHS) [13] and Arabic News Articles (ANA) [26], to conduct a thorough comparison.
- Empirically studying the impact of fine-tuning the TLMs for Arabic ATS on the resulting output summary.
- Empirically studying the performance of TLMs and deep-learning-based Arabic ATS systems.

The remaining part of the paper proceeds as follows: The related work is presented in Section 2, the text summarization methodology is covered in Section 3, and the experiments and results are presented in Sections 4 and 5, respectively. Section 6 discusses the findings. Finally, in Section 7, we give our conclusions and some recommendations for the future.

2. Background and Related Work

As early as the late 1950s, ATS attracted scientific communities to conduct research on text summarization [1]. At the time, there was a particular focus on generating abstracts of technical documentation. Years later, the literature witnessed a kind of decline in the interest in the area of ATS until the renaissance of AI and its technologies.

The early approaches of ATS mainly utilized statistical models to solely select, copy and paste the essential part of the original text [4]. For example, Edmundson [27] proposed a method that adopts statistical techniques. Such statistical methods principally use information about the frequency and distribution of words to calculate the relative significance. The text summary is then produced using the sentences with the most significance. However, such early approaches were not able to generate abstractive text summarization due to the lack of understanding of the original text. As such, there was a need for more intelligent systems that were able to understand and analyze the semantics of the natural languages to address the various challenges of using the early statistical-based approaches.

As was previously mentioned, there are two basic categories into which the ATS techniques can be broadly divided: *extractive* and *abstractive*. Early research on ATS was essentially focusing on extractive methods. However, most recently, more focus has been shifted toward abstractive approaches. Given the aim of this paper, which is a comparative study of abstractive Arabic ATS, the related work discussed in this section will be limited to the abstractive related work.

Abstractive ATS systems require a deeper understanding and analysis of the original text [28]. Abstractive ATS systems focus on generating a summary after understanding the main ideas in the original text without using the same sentences. Such abstractive approaches use NLP methods to create the summary text without copying sentences from the input (original) text. The abstractive ATS approaches are generally categorized into three main categories, structure-based, semantic-based and deep learning-based approaches [29]. The structure-based approaches use pre-defined structures such as graphs and ontologies. Whereas the semantic-based methods mainly focus on using the natural language generation systems and text semantic representation to generate the summary.

Deep learning-based approaches use deep neural networks to build ATS systems, which tend to report encouraging results in the ATS systems. Precisely, the sequence-to-sequence learning (seq2seq) model has shown impressive results in abstractive ATS with the English language [30]. For such approaches, Recurrent Neural Network (RNN) [31] with an attention encoder–decoder is utilized. For example, Hou et al. [30] proposed a seq2seq model for ATS with various phases such as the conversion of the dataset data to plain texts, storing the original text (news articles) and the summaries separately, word segmentation to process the data, and representing the words with pre-trained vectors. The experiments were conducted with a Chinese public dataset made available by NLPCC2017 shared task3 (http://tcci.ccf.org.cn/conference/2017/taskdata.php, accessed date 2 November 2022). The dataset consists of 2K texts without matching summaries for testing and around 40K document-summary pairs for training. The reported results were 0.34, 0.21 and 0.30 on ROUGE-1, ROUGE-2 and ROUGE-L, respectively. Later, such steps are utilized for training the model with bidirectional and unidirectional Long Short-Term Memory (LSTM) for the encoder and decoder, respectively. Chen et al. [32] have also proposed a method using the attention mechanism. Bidirectional gated recurrent units' architecture has been utilized in the proposed method to perform the encoding and decoding tasks. Additionally, Gu et al. [33] have added a copying mechanism to the neural model's encoder–decoder to aid in the sequences learning. In this proposed approach, the copying mechanism was used to determine which portion of the input sequence should be attached to the

appropriate location in the output sequence. The proposed approach was then evaluated on the recently released LCSTS [34] dataset, a sizable dataset for short ATS, and reported a slight improvement over models without copying mechanism with an average of 2–4% in ROUGE scores.

Following the direction of using attention mechanisms in ATS systems, Vaswani et al. [35] proposed the novel and currently well-known architecture "transformers". Such architecture was, independently of using sequence recurrence or convolution, able to determine the input and output representations. It is also known for its efficiency in terms of training time and performance as compared to standard deep learning approaches. Most recently, due to the BERT breakthrough, pre-trained TLMs have gained a great deal of popularity in the fields of AI, ML and NLP, achieving state-of-the-art results in a variety of tasks, including ATS in general, and abstractive Arabic ATS in particular [11].

Several review and survey articles have been proposed recently summarizing the efforts on Arabic ATS. For example, Elsaid et al. [9] provide an overview of the recent research concerning the Arabic language with a particular focus on deep learning ATS approaches, as well as an explanation of the general architecture, advantages, and disadvantages of Arabic ATS approaches. Some light was also shed on two initial extractive BERT-based approaches for Arabic ATS, particularly the Elmadani et al. [36] and Abu Nada et al. [37] proposals using a multipurpose Arabic dataset (KALIMAT [38]) with slightly more than 20K articles associated with their extractive summaries.

Nevertheless, as of yet, there are no comprehensive comparison studies among all existing deep TLMs-based Arabic ATSs that obtain SOTA results on various dedicated datasets. Hence, the goal of this paper is to address this gap.

3. Text Summarization Methodology

Text summarization is the act of separating long distributions into sensible passages or sentences. The technique extricates basic information while also guaranteeing that the section's sense is saved. This abbreviates the time it takes to understand long materials, such as insightful articles, without ignoring basic data. The most widely recognized approach to encouraging a brief, solid, and natural summary of a lengthier text report, including highlighting the text's essential centers are known as text summarization.

Text summarization presents a few issues, counting content distinctive confirmation, interpretation, frame time, and an examination of the subsequent summary. Perceiving significant expressions in the record and taking advantage of them to uncover applicable information to add to the synopsis are fundamental positions in an extraction-based summarization. As highlighted earlier, there are a few crucial text summarization types, as shown in Figure 1. In this study, we will focus on the abstractive text summarization for the Arabic language with a single document input. Particularly, the sole focus will be on the TLMs-based approaches.

Abstractive ATS approaches are classified as structure, semantic, discourse structure and deep learning-based techniques. They require more examination of the input source text and are mostly founded on understanding the semantics of a given article, restructuring sentences at the word-level, and lastly, producing abstracts with fewer and more clear words [39]. Summary generation can produce new sentences instead of just replicating sentences from the source record [40]. Vaswani et al. [35] recently shifted the direction and introduced a new deep learning-based model. The model is called a transformer and it makes use of several methods and mechanisms.

A transformer model is a neural network that learns the setting and, consequently, importance by following connections in successive information very much like the words in this sentence. Transformer models apply a propelling arrangement of numerical methods, called consideration or self-consideration, to distinguish unpretentious ways to be sure far-off information components in a series influence and rely upon one another. Transformers [35] are among the most modern and one of the most remarkable classes of models designed to date.

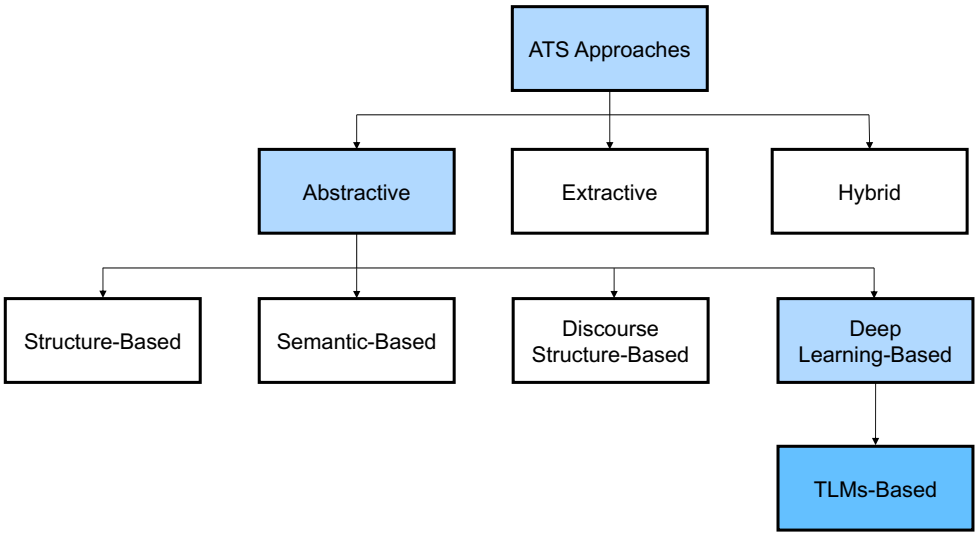


Figure 1. ATS approaches and their connected methods.

They are driving a rush of advances in AI, ML and NLP, and some have been named transformer AI or transformer NLP. Encoder and decoder layers are part of the transformer model, and one is coupled to the other through layers of the feed-forward network and multi-head attention. The cosine and sine functions, which produce positional encoding, assist the model and recall the order and position of words. Self-attention is a method used by the encoder and decoder layer's multi-head attention layer (see Figure 2).

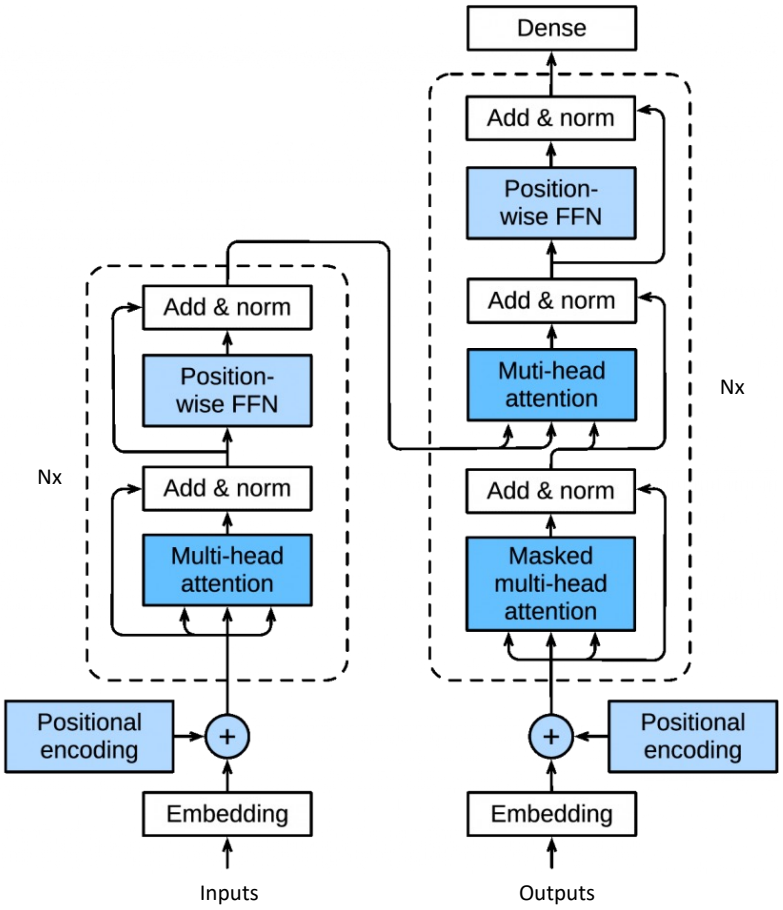


Figure 2. Architecture of transformers models [35].

From transformers-based models, the revolution of TLMs has emerged. For example, a TLM that is based on encoders and is learned in both directions, Bidirectional Encoder Representations from Transformers (BERT) [16], was introduced by Google AI. The BERT model's inputs are encoded using a specific format that consists of three pieces: wordpiece tokenization embeddings, segment embeddings, and position embeddings. It should be noted that all sequences now start with the special "CLS" token.

Typically employed for classification tasks, this token can be seen as the representation of the whole input sequence. Additionally, each sentence ends with the unique separator symbol "SEP". There are various versions of BERT for different languages, such as French camemBERT [41], ArabicBERT [42], AraBERT [43] and CAMeLBERT [44]. Likewise, Radford et al. [45] presented the Generative Pre-training Transformer (GPT) model. A total of 12 decoders are utilized to construct the input embeddings. Byte Pair Encoding (BPE), an information pressure calculation appropriate for word division that takes into mind encoding rare and out-of-vocabulary (OOV) terms, is used to encode the data successions. This is fundamental since transformers (in contrast to RNNs) consider every one of the data tokens immediately and hence, have no idea of the request for the tokens. This model's unidirectional nature is one of its limitations because the model was only designed to predict the next word from the current word, not the other way around. Hence, it was later enhanced with GPT-2 [46] and GPT-3 [20].

The primary commitment of TLMs was to pre-train one general TLM and fine-tune it straightforwardly for different tasks. For instance, without making significant task-specific architecture modifications, the pre-trained BERT model can be improved with just one additional output layer to produce cutting-edge models for a variety of applications, including ATS. In particular, we just insert the task-specific inputs and outputs (see Figure 2) into BERT and fine-tune all the parameters from beginning to end for each task (for the ATS task in our case). Consequently, several pre-trained models were proposed and were fine-tuned and implemented mainly for ATS tasks in different natural languages, including Arabic, to give fairly good summaries, such as multilingual Bidirectional and Auto-Regressive Transformers (mBART) [47], Pre-training with Extracted Gap-sentences for Abstractive Summarization (PEGASUS) [48], and mT5 [49], are the targeted models in our study and will be discussed in further detail in the experiment part (Section 4.4). The overall methodology for TLMs-based ATS systems is summarized in Figure 3, which is also the methodology we followed in this comparative study.

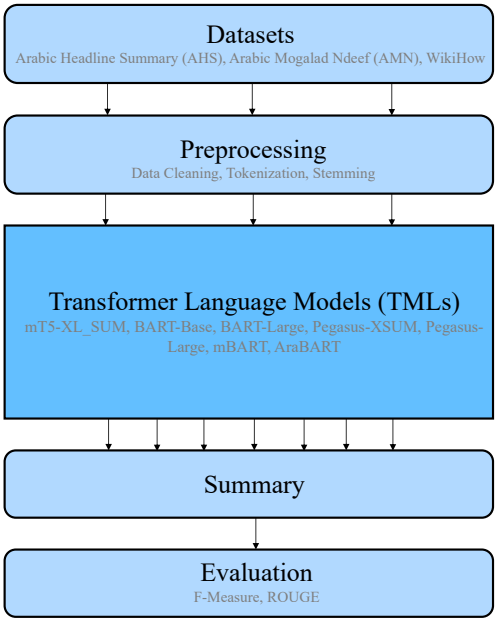


Figure 3. TLMs-based ATS systems general architecture.

4. Experiments

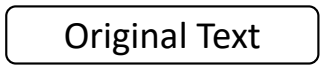
In this experiments section, we will initially present the datasets utilized during this comparative study. We then shed some light on the used TLMs. Next, the experimental setup is presented. We then introduce the various evaluation metrics that were adopted to evaluate the performance of the TLMs-based Arabic ATS.

4.1. Arabic ATS Datasets

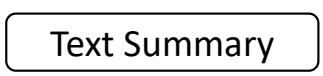
To evaluate the Arabic TLMs-based ATS models, we have utilized and conducted experiments on the three publicly available Arabic abstractive text summarization datasets.

- *Arabic Headline Summary (AHS)* [13]. It is utilized for the abstractive summary of a single document. The news on the Mawdoo3 website served as the source for this dataset [15]. There are 300k texts in it. Opening sentences (introduction paragraph) were regarded as the original text, and their titles serve as the summary.
- *WikiHow Dataset* [50]. It includes 770,000 WikiHow articles and summary pairs in 18 different languages. It also contains a summary of one abstractive document and 29,229 Arabic newswire texts.
- *Arabic News Articles (ANA)* [26]. A combination of multiple Arabic datasets from different news articles, Arabic News and Saudi Newspapers, formed this large ANA dataset with 265k news articles. Each article in this dataset has one summary.

Following [13], several preprocessing steps have been applied to the above-detailed datasets such as eliminating any diacritical marks, repetitions, and extraneous spaces as well as taking out unusual entries, such as poems. A sample of the original text and target output instance from the ANA dataset is shown in Figure 4.



إن أهم أسباب ضغط العين هو قضاء وقت طويل أمام الأجهزة الإلكترونية مثل الكمبيوتر والتلفاز ، وهذا ما تفرضه ثقافة اليوم . ولهذا يحث الكثير على إيجاد بدائل مثل قراءة الكتب الورقية بدلا من القراءة على الأجهزة اللوحية لتقليل الآثار الضارة التي عادة ما تكون نتيجة لقضاء الكثير من من الوقت أمام الشاشة . ولا يمكن تجنبه ولكن نحاول أن نوجد الحلول الجزئية مثل خفض مستوى الشاشة أو استخدام شاشة فيها ما يمنع التوهج لتقليل الضغط على العين . أما من يعاني من إجهاد العين المستمر الذي يسبب الإزعاج في الحياة اليومية ، فمن المهم استشارة الطبيب . حيث أن هناك احتمالية لوجود مشكلة في الرؤية . والوقاية دائماً هي خير من العلاج ، لذا من المهم زيارة الطبيب لتفادي تفاقم المشكلة والحصول على الدواء اللازم إن لزم الأمر . من المهم كذلك محاولة مساعدة النفس على الاسترخاء بإغلاق العينين حين منع النفس من النوم . مع محاولة تكرار هذا الأمر مرات من ثلاثة إلى خمسة في اليوم لبضع ثواني .



تقليل وقت التعرض للشاشات . الذهاب إلى الطبيب . إغلاق العينين للاسترخاء .

Figure 4. An example of a similar sample of Arabic text summary extracted from the ANA [26] dataset.

4.2. Used Transformer Language Models (TLMs)

There are a number of pre-trained TLM models that are proposed, fine-tuned, improved upon, and put into use primarily for ATS tasks in several natural languages, including Arabic. Next, we discuss the models that are adopted and considered in this comparative study.

- **mBART**: Following BART, mBART [47] is constructed using a seq2seq model with denoising as a pre-training objective. It models architecture that combines an encoder and a decoder using a typical seq2seq. The pre-training assignment incorporates a new approach where text ranges are exchanged with a single mask token and modifying the starting sentences order randomly. The autoregressive BART decoder is controlled for developing sequential NLP tasks such as text summarization. The denoising pre-training objective is strongly tied to the fact that the data are taken from the input but altered. As a result, the encoder's input is the input sequence embedding, and the decoder's output is produced autoregressively. BART only pre-trained for English, but mBART thoroughly investigated the impacts of pre-training on many sets of languages (e.g., Japanese, French, German, and Arabic). It utilized a common sequence-to-sequence Transformer design with 12 layers of encoders and 12 layers of decoders on 16 heads (corresponding to around 680 M parameters). The training was stabilized by adding a layer-normalization layer on top of the encoder and decoder.
- **mT5**: Transfer learning is the principle underpinning the mT5 [51] model, which is an extended version of T5. The original model was initially trained using transfer learning on a task with a lot of text before being fine-tuned on a downstream task to help the model develop general-purpose abilities and knowledge that can be used for tasks such as summarizing. T5 employed a sequence-to-sequence creation technique that produces an autoregressive output from the decoder after feeding it the encoded input through cross-attention layers. T5 only pre-trained for English; however, mT5 came to carefully examine the effects of pre-training on various natural languages, including Arabic.
- **PEGASUS**: A sequence-to-sequence model, PEGASUS [48] separates out important lines from the input text and compiles them as independent outputs. Additionally, selecting only pertinent sentences works better than selecting sentences at random. As it is analogous to the work of reading the complete document and producing a summary, this style is chosen and preferred for abstractive summarizing.
- **AraBART**: The architecture of AraBART [11], which has 768 hidden dimensions and 6 encoder and 6 decoder layers, is based on that of BART Base. AraBART has 139 M parameters in total. To stabilize training, it has a normalization layer on top of the encoder and the decoder. Sentencepiece is used by AraBART to construct its vocabulary. A randomly chosen subset of the pre-training corpus, measuring 20 GB in size, was used to train the sentencepiece model. The size of the vocabulary is 50 K tokens.

4.3. Experimental Setup

During this comparative study, the overall architecture is shown in Figure 3. As input, every dataset is saved in a CSV file format post applying the aforementioned preprocessing steps to each dataset. Afterward, the tokenization step is applied to obtain the special token. Every token is an input for any selected transformers models (encoder/decoder model). Regarding the output, it is going to be a generated summary. Furthermore, AdamW is used as an optimizer, and the maximum length of the summary is fixed at 150.

Regularly, data pre-processing is the beginning step applied to the input sentence, highlighting changing the information into a steady and standardized structure. It covers various tasks and cycles that change by information module and application. We apply the accompanying pre-processing steps:

- Tokenization, to separate the info texts into tokens.
- The report has been then harmed by supplanting ranges of text with the "MASK" token.
- Frame every token to an index in light of the pre-trained models lexicon.

The experimental settings of the compared TLMs-based Arabic ATS in this comparative study are shown in Table 1. To compare these models, we used the Transformer library of HuggingFace (https://huggingface.co/docs/transformers/index, accessed date 20 September 2022). We truncated each input document to 200 tokens and at most 12 tokens for each generated summary. We used beam search (num-beam = 4). The batch size was set to 6. All of our experiments were run using NVIDIA GeForce MX150.

Table 1. Comparison of TLM settings.

Models	Layers	Parameters	Vocab Size	Epochs	Batch Size
mT5	6	600 M	250 K	3	6
PEGASUS-XSum	16	568 M	96 K	3	6
PEGASUS-Large	16	568 M	96 K	3	6
mBART-Large	12	680 M	250 K	3	6
AraBART	6	139 M	50 K	3	6

4.4. Evaluation

Because there is more than one perfect summary for a single document or collection of documents, evaluating a summary could be challenging. In fact, there is a great deal of debate about what constitutes a good summary [1]. There are two methods for assessing the generated summary. The initial one is human-based; in this way, the human concentrates on the main sentences from the message and afterward contrasts them and the produced synopsis. However, it is an impractical way since it is emotional and requires a great deal of time and exertion. Then again, the program-based assessment is quicker and relies upon clear assessment estimates such as review, accuracy, and F-score. ROUGE is the most well-known robotized measure utilized in text summarization, which represents Recall-Oriented Understudy for Gisting Evaluation [52]. Assessing the nature of the produced summaries by contrasting them with their references. ROUGE-1 and ROUGE-2 measure the overlap between unigrams and bigrams, respectively, whereas ROUGE-L and ROUGE-LSUM work similarly to determine the lengthiest common subsequence between two pair of texts, respectively, with and without splitting sentences into new lines [53].

For the purpose of evaluating the models' accuracy performance, the F-Score (Equation (1)) is calculated as the harmonic mean of precision and recall. By dividing the total number of true positive outcomes (number of words shared by or overlapped between both summaries) by the total number of true positive (all words in the reference summary) results, precision (P) is determined. The recall (R) is calculated by dividing the total number of relevant results (all words in the outcome summary) by the number of true positives (number of words shared by/overlapped between both summaries).

$$F_{score} = 2 \times \frac{P \times R}{P + R} \tag{1}$$

where

$$P = \frac{True_{Positive}}{True_{Positive} + False_{Positive}} \tag{2}$$

and

$$R = \frac{True_{Positive}}{True_{Positive} + False_{Negative}} \tag{3}$$

5. Results

Tables 2–4 summarize the results of the compared TLMs-based Arabic ATS on the ANA, AHS, and WikiHow datasets, respectively. We evaluate and compare each TLM with the various ROUGE metrics on each utilized dataset.

The second form of comparison is between a baseline model that is not utilizing TLMs for ATS to shine a light on the superiority of TLMs.

We opted to compare one of the PEGASUS models, which reported the best results in this comparison study against a Bidirectional LSTM (BiLSTM) [13] model. The BiLSTM has reported the most promising results in Arabic ATS as compared to previously proposed models in their extensive study, hence its selection here. It has been trained with 256 hidden states, a word embedding with 128 dimensions, a decoder of 512 states, learning and assembly rates of 0.15 and 0.1, respectively, 300 epochs and AdaGrad [54] as an optimization approach. Tables 5 and 6 summarize this comparison for AHS and ANA datasets, respectively.

Table 2. Comparison among models with the ANA dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSUM
	F-Score %			
mT5	83.34	58.58	76.85	77.78
PEGASUS-XSum	**88.89**	**75.75**	84.57	84.88
PEGASUS-Large	88.27	73.74	83.95	84.25
mBART-Large	26.23	8.82	25.92	25.92
AraBART	85.83	70.90	**85.01**	**85.01**

Table 3. Comparison among models with the AHS dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSUM
	F-Score %			
mT5	48.74	31.22	46.45	46.58
PEGASUS-XSum	**66.70**	**58.41**	**66.50**	**66.50**
PEGASUS-Large	58.37	48.63	58.16	58.16
mBART-Large	27.70	9.81	27.70	27.70
AraBART	34.74	17.50	34.08	34.08

Table 4. Comparison among models with the WikiHow dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSUM
	F-Score %			
mT5	53.16	25.54	50.66	51.00
PEGASUS-XSum	94.49	88.63	94.45	94.41
PEGASUS-Large	**94.62**	**88.72**	**94.58**	**94.58**
mBART-Large	31.91	5.90	30.45	30.58
AraBART	44.12	12.18	42.16	42.12

Table 5. Comparison among baseline and selected TLM model on the AHS dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-L
	F-Score %		
BiLSTM	51.49	12.27	34.37
PEGASUS-XSum	**66.70**	**58.41**	**66.50**

Table 6. Comparison among baseline and selected TLM model on the ANA dataset.

Models	ROUGE-1	ROUGE-2	ROUGE-L
	F-Score %		
BiLSTM	44.28	18.35	32.46
PEGASUS-XSum	**88.89**	**75.75**	**84.57**

6. Discussion

For the ANA dataset, shown in Table 2, PEGASUS-XSum and PEGASUS-Large, which are the two used versions of PEGASUS, report the best results on ROUGE-1 and ROUGE-2 with a good margin but were slightly beaten by AraBART on ROUGE-L and ROUGE-LSUM. Even though AraBART has a quarter size of parameters compared to the other models, it is still reporting the best or comparable results on all metrics on the ANA dataset because it is solely pre-trained and fine-tuned for Arabic ATS. mBART seems to be struggling irrespective of the used metric, which is also the case with the other two datasets, as we see later.

Table 3 presents the obtained results with AHS dataset. It shows that for this comparison, PEGASUS models report the top two results, but PEGASUS-XSum demonstrates superior performance. In contrast to the ANA results, AraBART appears to be struggling with the AHS dataset managing only to score half of what was achieved by PEGASUS-XSum. Results of a similar nature were also obtained in Table 4 with the WikiHow dataset. In particular, the PEGASUS family tends to outperform other models. PEGASUS-LARGE reports the best performance scoring 95% in most metrics. Both mT5 and AraBART perform relatively well on some metrics but are not being able to achieve good results on ROUGE-2. It is also worth noting that the struggle is continuing with mBART.

The TLMs-based ATS, PEGASUS surpasses the baseline model with a big margin regardless of the used datasets or the evaluation metric. These particular results justify the rapidly growing use of TLMs for ATS systems.

Overall, according to the results detailed above, we notice that because of its nature and its dedication to the same type of tasks put in question, for abstractive text summarization, the PEGASUS models with the two used versions (PEGASUS-Large and PEGASUS-XSum) manage to obtain the best results. In the case of the BART multilingual version, mBART, the results are yet to be compared with superior models. However, the Arabic version, AraBART, shows many improvements on all datasets, especially with ANA. The highest reported results of the compared models were obtained with WikiHow datasets with the PEGASUS family. Furthermore, that might be explained by the nature of the models, as well as the length of the summary as an input at the time of training and its nature (e.g., title, highlight).

7. Conclusions

This paper offers a thorough comparative analysis between state-of-the-art TLM-based Arabic ATS models (e.g., mBART, mT5, PEGASUS, and AraBART) on various text summarization datasets, including Arabic News Articles (ANA), WikiHow, and Arabic Headline Summary (AHS). Precisely, the work presented in this paper makes three main contributions in total. A complete comparison analysis of all Arabic and Arabic-supported multilingual ATS systems that are based on abstractive TLMs was provided with multiple assessment metrics.

It also utilized various Arabic datasets currently available for abstractive ATS, including Arabic Headline Summary (AHS) and Arabic News Articles (ANA), to carry out a full comparison. Moreover, we conducted an empirical analysis of the effect of adjusting the TLMs for Arabic ATS on the output summary along with a comparison against deep-learning-based baseline approaches. The experimental results revealed that PEGASUS family models outperform the other TMLs compared and studied and showed superiority against the baseline deep-learning approach. The PEGASUS models with the two employed versions (PEGASUS-Large and PEGASUS-XSum) managed to obtain the best results because of their nature and the fact that they are dedicated to the same kind of tasks as those in question—abstractive text summarization. As part of our future work, we plan to focus our efforts on multimodal ATS as it is proven that using information from the visual modality, multimodal summarizing can raise the quality of the resulting summary.

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

Funding: The researchers would like to thank the Deanship of Scientific Research, Qassim University, for funding the publication of this project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Saggion, H.; Poibeau, T. Automatic text summarization: Past, present and future. In *Multi-Source, Multilingual Information Extraction and Summarization*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 3–21.
2. Rahul; Rauniyar, S.; Monika. A survey on deep learning based various methods analysis of text summarization. In Proceedings of the 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–28 February 2020; pp. 113–116.
3. Fejer, H.N.; Omar, N. Automatic Arabic text summarization using clustering and keyphrase extraction. In Proceedings of the 6th International Conference on Information Technology and Multimedia, Putrajaya, Malaysia, 18–20 November 2014; pp. 293–298.
4. Syed, A.A.; Gaol, F.L.; Matsuo, T. A survey of the state-of-the-art models in neural abstractive text summarization. *IEEE Access* **2021**, *9*, 13248–13265. [CrossRef]
5. Siragusa, G.; Robaldo, L. Sentence Graph Attention For Content-Aware Summarization. *Appl. Sci.* **2022**, *12*, 10382. [CrossRef]
6. Allahyari, M.; Pouriyeh, S.; Assefi, M.; Safaei, S.; Trippe, E.D.; Gutierrez, J.B.; Kochut, K. Text summarization techniques: A brief survey. *arXiv* **2017**, arXiv:1707.02268.
7. Witte, R.; Krestel, R.; Bergler, S. Generating update summaries for DUC 2007. In Proceedings of the Document Understanding Conference, Rochester, NY, USA, 26–27April 2007; pp. 1–5.
8. Fatima, Z.; Zardari, S.; Fahim, M.; Andleeb Siddiqui, M.; Ibrahim, A.; Ag, A.; Nisar, K.; Naz, L.F. A Novel Approach for Semantic Extractive Text Summarization. *Appl. Sci.* **2022**, *12*, 4479.
9. Elsaid, A.; Mohammed, A.; Fattouh, L.; Sakre, M. A Comprehensive Review of Arabic Text summarization. *IEEE Access* **2022**, *10*, 38012–38030. [CrossRef]
10. Boudad, N.; Faizi, R.; Thami, R.O.H.; Chiheb, R. Sentiment analysis in Arabic: A review of the literature. *Ain Shams Eng. J.* **2018**, *9*, 2479–2490. [CrossRef]
11. Kamal Eddine, M.; Tomeh, N.; Habash, N.; Le Roux, J.; Vazirgiannis, M. AraBART: A Pretrained Arabic Sequence-to-Sequence Model for Abstractive Summarization. *arXiv* **2022**, arXiv:2203.10945.
12. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
13. Al-Maleh, M.; Desouki, S. Arabic text summarization using deep learning approach. *J. Big Data* **2020**, *7*, 1–17. [CrossRef]
14. See, A.; Liu, P.J.; Manning, C.D. Get to the point: Summarization with pointer-generator networks. *arXiv* **2017**, arXiv:1704.04368.
15. Wazery, Y.M.; Saleh, M.E.; Alharbi, A.; Ali, A.A. Abstractive Arabic Text Summarization Based on Deep Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 1566890. [CrossRef]
16. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
17. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. 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, 5–10 July 2020; pp. 7871–7880.
18. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
19. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
20. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
21. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
22. Xu, S.; Zhang, X.; Wu, Y.; Wei, F. Sequence level contrastive learning for text summarization. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 11556–11565. [CrossRef]

23. González, J.Á.; Hurtado, L.F.; Pla, F. Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. *Inf. Process. Manag.* **2020**, *57*, 102262. [CrossRef]
24. Meškelė, D.; Frasincar, F. ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. *Inf. Process. Manag.* **2020**, *57*, 102211. [CrossRef]
25. Kahla, M.; Yang, Z.G.; Novák, A. Cross-lingual fine-tuning for abstractive Arabic text summarization. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Online, 1–3 September 2021; pp. 655–663.
26. Zaki, A.M.; Khalil, M.I.; Abbas, H.M. Deep architectures for abstractive text summarization in multiple languages. In Proceedings of the 2019 14th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 17–18 December 2019; pp. 22–27.
27. Edmundson, H.P. New methods in automatic extracting. *J. ACM* **1969**, *16*, 264–285. [CrossRef]
28. Mohan, M.J.; Sunitha, C.; Ganesh, A.; Jaya, A. A study on ontology based abstractive summarization. *Procedia Comput. Sci.* **2016**, *87*, 32–37. [CrossRef]
29. El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **2021**, *165*, 113679. [CrossRef]
30. Hou, L.; Hu, P.; Bei, C. Abstractive document summarization via neural model with joint attention. In Proceedings of the National CCF Conference on Natural Language Processing and Chinese Computing, Dalian, China, 8–12 November 2017; pp. 329–338.
31. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
32. Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H. Distraction-based neural networks for document summarization. *arXiv* **2016**, arXiv:1610.08462.
33. Gu, J.; Lu, Z.; Li, H.; Li, V.O. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv* **2016**, arXiv:1603.06393.
34. HUB, C.; LCSTS, Z. A Large Scale Chinese Short T e xt Summarization Dataset. In Proceedings of the Procee-dings of the 2015 Conference on Em pirical Methods in Natural Language Proces sing. Lisbo n: ACL, Lisbon, Ptugal, 17–21 September 2015; Volume 2, pp. 1967–1972.
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
36. Elmadani, K.N.; Elgezouli, M.; Showk, A. BERT Fine-tuning For Arabic Text Summarization. *arXiv* **2020**, arXiv:2004.14135.
37. Abu Nada, A.M.; Alajrami, E.; Al-Saqqa, A.A.; Abu-Naser, S.S. Arabic text summarization using arabert model using extractive text summarization approach. *Int. J. Acad. Inf. Syst. Res.* **2020**, *4*, 6–9.
38. El-Haj, M.; Koulali, R. KALIMAT a multipurpose Arabic Corpus. In Proceedings of the Second Workshop on Arabic Corpus Linguistics (WACL-2), Lancaster, UK, 22 January 2013; pp. 22–25.
39. Al-Abdallah, R.Z.; Al-Taani, A.T. Arabic single-document text summarization using particle swarm optimization algorithm. *Procedia Comput. Sci.* **2017**, *117*, 30–37. [CrossRef]
40. Bhat, I.K.; Mohd, M.; Hashmy, R. Sumitup: A hybrid single-document text summarizer. In *Soft Computing: Theories and Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 619–634.
41. Martin, L.; Muller, B.; Suárez, P.J.O.; Dupont, Y.; Romary, L.; de La Clergerie, É.V.; Seddah, D.; Sagot, B. CamemBERT: A tasty French language model. *arXiv* **2019**, arXiv:1911.03894.
42. Safaya, A.; Abdullatif, M.; Yuret, D. KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, Barcelona, Spain, 12–13 December 2020.
43. Antoun, W.; Baly, F.; Hajj, H. Arabert: Transformer-based model for arabic language understanding. *arXiv* **2020**, arXiv:2003.00104.
44. Inoue, G.; Alhafni, B.; Baimukan, N.; Bouamor, H.; Habash, N. The interplay of variant, size, and task type in Arabic pre-trained language models. *arXiv* **2021**, arXiv:2103.06678.
45. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. *Techincal Rep. OpenAI* **2018**, 1–12. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 14 November 2022).
46. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
47. Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 726–742. [CrossRef]
48. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 11328–11339.
49. Hasan, T.; Bhattacharjee, A.; Islam, M.S.; Samin, K.; Li, Y.F.; Kang, Y.B.; Rahman, M.S.; Shahriyar, R. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. *arXiv* **2021**, arXiv:2106.13822.
50. Ladhak, F.; Durmus, E.; Cardie, C.; McKeown, K. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. *arXiv* **2020**, arXiv:2010.03093.
51. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv* **2020**, arXiv:2010.11934.

52. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*; ACL Anthology: Barcelona, Spain, 2004; pp. 74–81.
53. Rouge, L.C. A package for automatic evaluation of summaries. In Proceedings of the Proceedings of Workshop on Text Summarization of ACL, Barcelona, Spain, 25–26 July 2004.
54. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.