IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 November 2023, accepted 25 December 2023, date of publication 29 December 2023,
date of current version 8 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3348410

RESEARCH ARTICLE

Arabic Narrative Question Answering (QA) Using Transformer Models

MOHAMMAD A. ATEEQ[1,2], SABRINA TIUN[1], HAMED ABDELHAQ[2], AND NAWRAS RAHHAL[2]

[1]Faculty of Information Science and Technology, Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[2]Department of Computer Science, An-Najah National University, Nablus 00970, Palestine

Corresponding authors: Mohammad A. Ateeq (mhmd.abed.ateeq@gmail.com) and Sabrina Tiun (sabrinatiun@ukm.edu.my)

This work was supported by Universiti Kebangsaan Malaysia under Grant FRGS/1/2020/ICT02/UKM/02/1.

ABSTRACT The Narrative question answering (QA) problem involves generating accurate, relevant, and human-like answers to questions based on the comprehension of a story consisting of logically connected paragraphs. Developing Narrative QA models allows students to ask about inconspicuous narrative elements while reading the story. However, this problem remains unexplored for the Arabic language because of the lack of Arabic narrative datasets. To address this gap, we present the Arabic-NarrativeQA dataset, which is the first dataset specifically designed for machine-reading comprehension of Arabic stories. This dataset consists of two parts: translation of an English NarrativeQA dataset and a collection of new question-answer pairs based on Arabic stories. Furthermore, we implement the Arabic-NarrativeQA system using the Ranker-Reader pipeline, exploring and evaluating various approaches at each stage to identify the most effective ones. To avoid the need for an extensive data collection process, we utilize cross-lingual transfer learning techniques to leverage knowledge transfer from the English Narrative QA dataset to the Arabic-NarrativeQA system. Experiments show that incorporating cross-lingual transfer learning significantly improved the performance of the reader models. Furthermore, the question's evidence information provided in the Arabic-NarrativeQA dataset enables the learnable rankers to effectively identify and select the pertinent paragraphs. Finally, we examine and categorize challenging questions that require a deep understanding of the stories. By incorporating these question types into the introduced dataset, we show that existing reading comprehension models struggle to answer them, and further model development should be conducted. To promote further research on this task, we make both the Arabic-NarrativeQA dataset and the pre-trained models publicly available.

INDEX TERMS Arabic question answering, answer generation, cross-lingual transfer learning, reading comprehension, narrative QA.

I. INTRODUCTION

Question answering(QA) is one of the most challenging problems in Natural Language Processing (NLP). The main task of any QA system is to understand a given passage of text and then answer a set of questions posed in natural language. QA systems follow one of two paradigms based on the way the answer is created:

- Extractive QA: The system extracts the answer from the provided documents (Knowledge Base). More specifically, the output is span (i, j), where i and j are the start and end positions of the extracted answer within a paragraph, respectively.
- Generative QA: The model produces expressive and free text based on the context. In this case, the generated answer is not a snippet in the document; hence, it is closer to human-generated answers and easier to understand.

The task of Generative QA is more challenging than that of extraction QA. Extractive QA requires a model that can predict the exact start and end positions of a given answer within a paragraph. On the other hand, the Generative QA model faces the significant challenge of producing a cohesive sequence of words from a dictionary containing a potentially

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

vast vocabulary of tens of thousands of words. Therefore, it generates text more freely which leads to a common issue where they produce text that's not accurate or doesn't match the input. This problem is addressed in the open-domain question answering [1], and text summarization [2].

In this study, we aimed to help young children to develop their narrative comprehension abilities. Deep learning techniques are utilized to build a QA system that can answer questions on Arabic stories after understanding and analyzing the narrative elements: places, events, entities, and the relations between them. As a result, children can use this system to ask about any inconspicuous narrative elements while reading their stories.

In the past few years, Arabic Question-Answering (QA) models have achieved remarkable results on the Holy Qur'an [3], open domain question answering (ODQA) [4], [5] and Arabic conversational systems [6]. However, the ability to comprehend complex contextual information remains an unresolved problem, particularly when it comes to answering questions about narrative elements in stories [7]. Although there has been a lack of significant advancements in this challenging task [7], it is still not addressed in Arabic.

The Narrative QA problem involves generating a correct, relevant, and human-like answer to a given question after comprehension of a story that includes a series of consecutive and logically related narrative paragraphs. One possible approach to solve this problem is to train a generative model such as Bart [8] or T5 [9] on the entire story, along with the question to generate the answer [10], [11]. However, processing lengthy stories is considered a significant challenge because of the demanding GPU memory requirements of the model. One potential solution followed by [7], [12], and [13] is the Ranker-Reader pipeline. It consists of two main phases: (1) retrieving the most relevant paragraphs to the input question and (2) passing the concatenation of retrieved paragraphs along with the question to a reader model such as Bart [8] and T5 [9] to generate the answer. Training the ranker model requires having pairs of questions and associated paragraphs containing the relevant information for those questions. However, the manual labeling of such data is time-consuming. Therefore, the labels can be generated using heuristics approaches such as pseudo distance supervision signals [7] or by weakly supervised learning such as utilizing the attention activations from the reader model [13]

In this study, we recorded relevant paragraphs for each question in the constructed Arabic-NarrativeQA dataset. Therefore, we can utilize these labels to train a Bert-based classifier efficiently, as illustrated in the ranker block of Fig. 1.

1) Ranking Block: This block is responsible for assessing the relevance of each paragraph within the story in relation to a specific question. Formally, given a question q and a set of paragraphs $P = P_r \cup P_i$, where each paragraph $p \in P_r$ is relevant to q and each $p \in P_i$ is irrelevant. The goal of this component is to select a subset of paragraphs $P_{\text{sel}} \subset P_r$ such that the paragraphs in P_{sel} attain the highest relevance scores with respect to q. Each paragraph $p_j \in P$ is concatenated with question q to generate paragraph-question pairs (p_j, q) for $1 \leq j \leq n$, where n represents the total number of paragraphs. Subsequently, the encoder block computes context vector CV_j for each (p_j, q) pair. The final layer (linear + sigmoid) calculates $P(CV_j)$, which can be considered as the relevance score between the question q and paragraph p_j. Finally, the ranker selects $p_j \in P$ that maximizes $P(CV_j)$.
2) Reader Block: This is responsible for comprehending and understanding the content of the selected paragraphs to generate a precise, free-form, and correct answer to a given question. The Reader block should be able to comprehend the underlying meaning of the textual information in the selected paragraphs and identify the relevant context to the given question. Let q represents a given question and $P_{\text{sel}} = \{ps_1, ps_2, \ldots, ps_k\}$ be the set of k selected paragraphs obtained from the ranking block. The two main parts are:
   - Encoder: It takes the concatenation of k selected paragraphs P_{sel} and the question q as input and computes a context vector CV. This vector captures the relationships, semantics, and important details within the narrative text of selected paragraphs.
   - Decoder: The decoder processes the contextual information obtained from the encoder, employs the attention mechanism to focus on relevant parts of the narrative, and generates a token of the answer at each decoding step. Formally, it uses the context vector CV and generates a hidden state $H(s_l)$ for $1 \leq l \leq L$, where L is the length of the generated answer. Each $H(s_l)$ captures the current context and the information required for token prediction. Subsequently, in each decoding step, the linear and softmax layers utilize $H(s_l)$ to calculate the token probabilities $P(t_v)$ for $1 \leq v \leq m$, where m denotes the vocabulary length. More details about the attention mechanism and the interconnection between the encoder and decoder are provided in Section II-A

In modern natural language processing (NLP) applications, it is common to utilize transfer learning, which initially involves training a model on a data-rich task and then fine-tuning it for a downstream task of interest [14]. This technique avoids the need for extensive data collection and enhances the generalization capabilities of the model [14]. Subsequently, we fine-tuned the pretrained multilingual models [15], [16] on the narrative QA dataset.

Cross-lingual transfer learning is a technique for the adaptation of a model, initially trained for a specific task in a monolingual context, to enhance its capability for generalization across diverse languages [17], [18]. The principal

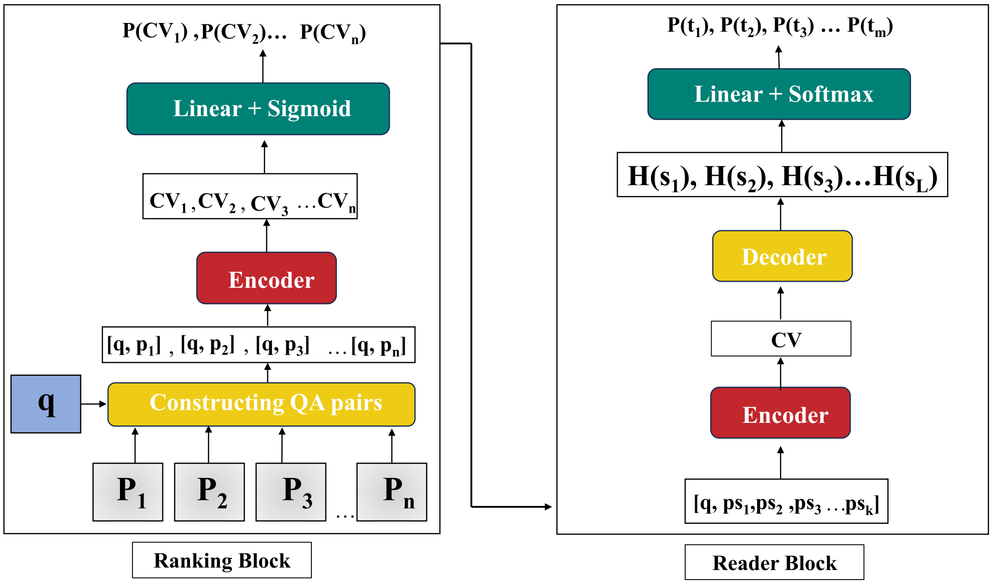

Figure 1. The network architecture of the narrative QA system.

objective is to build shared multilingual representations of text by aligning the representations across languages [17], [19]. This process consists of three main phases [19]: (1) developing multilingual models, such as mt5 [15] and mbart [16] (2) refining the multilingual model by fine-tuning it for a specific task in the source language, and (3) applying the fine-tuned model to the target language in zero-shot settings or applying another fine-tuning phase on a small-scale dataset in the target language. In [20], it is demonstrated that the inclusion of a pre-trained multilingual language model significantly reduces the size of the dataset required to achieve good performance by 80%. In the context of this study, we demonstrate that a two-stage fine-tuning process of multilingual models, such as mt5 [15] and mbart [16], when initially applied to the English narrative dataset and subsequently fine-tuned on a limited subset consisting of Arabic translations for 13% of the original dataset, yields results comparable to those obtained by translating the entire dataset and fine-tuning the model on it without cross-lingual settings. This method mitigates the requirement for translation costs or the costly acquisition of language-specific annotations for the target language.

Despite the similarity of Narrative QA to open-domain question answering (ODQA), progress in Narrative QA lags behind [7], [21], [22]. The challenges associated with Narrative QA can be summarized as follows [7], [21], [22]:

- Diverse writing styles in narrative stories require a deeper level of comprehension compared to the formal texts in Wikipedia and news articles.
- Finding evidence from the story is not a trivial task as the paragraphs of a story exhibit greater semantic similarities compared to Wikipedia articles.
- A reliable narrative QA system should address two types of questions: Explicit and Implicit. In the explicit type, the questions can be answered directly from the text as a span of words. However, implicit questions are more difficult to answer because the answer cannot be found directly from the text. Instead, the model must generate free-form answers [10]. The presence of free-form answers requires the employment of Generative QA, which is considered more challenging than Extractive QA. According to [23], most existing QA models extract the answer as a span of words from a document. However, the generation of fluent and complete-sentence answers is still in its infancy, especially in non-English languages.
- There are typically logical connections between different paragraphs. This requires comprehending the paragraphs and the interactions between them to answer challenging questions.

The narrative QA problem is not addressed for the Arabic language as there are no Arabic narrative datasets available that can be utilized for fine-tuning a transformer model. There is a need for an Arabic narrative dataset that includes well-designed, reliable, and valid questions. Moreover, questions should assess narrative comprehension rather than simple text matching or memorization.

We selected Arabic as our context because of its extensive range of distinct variations and widespread usage. However, in future works, we will investigate the feasibility of extending our model to other languages.

Our contribution in this research is three-fold:

1) Introducing Arabic-NarrativeQA as the first machine reading comprehension dataset on Arabic stories. We make this dataset publicly available[1] to encourage

[1]https://www.kaggle.com/datasets/mohammadateeq/arrabic-narrative-qa

state-of-the-art research on Arabic Narrative Comprehension tasks.

2) Establishing a solid state-of-the-art performance by fine-tuning published Pre-trained Language Models (Arabert [24], mT5 [15] and mBART [16]) on Arabic-NarrativeQA dataset. Arabert [24] was fine-tuned to rank paragraphs based on their relevance to the given question. Generative models mT5 [15] and mBART [16] were fine-tuned to generate a correct, relevant, and human-like answer from the top-ranked paragraphs.
3) Employing cross-lingual transfer learning techniques to facilitate the transfer of knowledge from the English Narrative QA dataset, which represents a high-resource language, to Arabic, a low-resource language.

The rest of the paper is organized as follows. A literature review in the area of Narrative QA is explored in Section II. Section III describes the methodology employed in this study. In Section IV, we present our results and findings. Finally, Section V provides concluding statements and future directions.

II. LITERATURE REVIEW

Due to the importance of QA applications, the QA field has received significant attention in recent years. In the context of narrative QA, the answers to some questions can not be explicitly found in the story. Therefore, generative models should be employed to generate free-form, coherent, and contextually appropriate answers.

A. GENERATIVE MODELS FOR QUESTION ANSWERING (QA)

The task of Generative QA can be described as generating an accurate, coherent, and natural response in natural language when provided with the context and the question [25].

In the past few years, some researchers have focused on developing generative QA systems using Recurrent Neural Networks (RNNs), in particular, encoder-decoder generative models. Fig. 2 provides an overview of the generative QA system based on an encoder-decoder RNN model [25]. Initially, the questions were divided into a sequence of tokens. Consequently, each token was transformed into a vector representation using an embedding block. The encoder block transforms the generated embedding into fixed-length vectors by applying a series of computations, resulting in the generation of an intermediate state for each token, where each state is updated by the output of previous hidden states. The final hidden state of the encoder network outputs the context vector that captures the meaning of the input sentence. The main function of the decoder is to generate a response by utilizing the context vectors produced by the encoder. The decoder employs a mechanism called attention to calculate the alignment scores between each decoder's hidden state and that of the encoder [26]. These alignment scores can be utilized to capture the importance of each

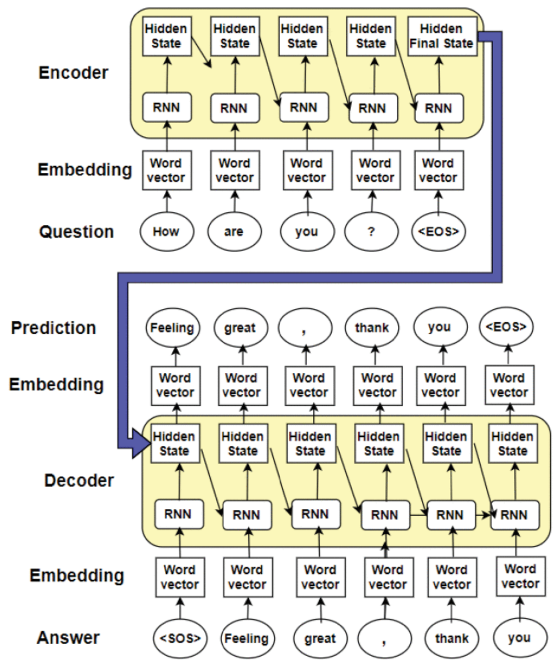

Figure 2. RNN-based encoder-decoder model for generative QA systems [25].

encoder's hidden state with respect to the token that is currently being processed in the decoding stage. Attention weights (alignment scores) are applied to the encoder's hidden states, producing a context vector that is combined with the decoder's current hidden state to predict the next output token.

One of the main problems in the RNN-based architecture in Fig. 2 is the generation of generic or inconsistent answers. This issue usually occurs when using the cross-entropy loss function to train a Seq2Seq model. Using that loss function leads to adjusting the weights of the model targeting to minimize the error and find the most likely answer which may sometimes be generic or inconsistent. Authors in [27] addressed this issue by reformulating the generative QA problem as multi-task learning (MTL) problem: The main task is building a generative QA using the Seq2Seq model. The second task is a binary QA classification to determine whether the provided QA pair is actually matched or not. Both tasks are simultaneously learned using shared parameters. The classification labels from the second task are utilized as the main guide for the generation of word sequences for the answers. In terms of the technique employed, the cross-entropy loss function is replaced with an MTL loss function which is basically the summation of the loss of the answer generation task and the loss of the classification task.

Another main problem of RNN-based encoder-decoder models is Sequential Processing. RNN-based sequence model produces the output of each hidden state after processing all previous states, making the model impractical when processing sequences with long inputs. This problem is resolved by transformer architecture which is originally proposed in [26].

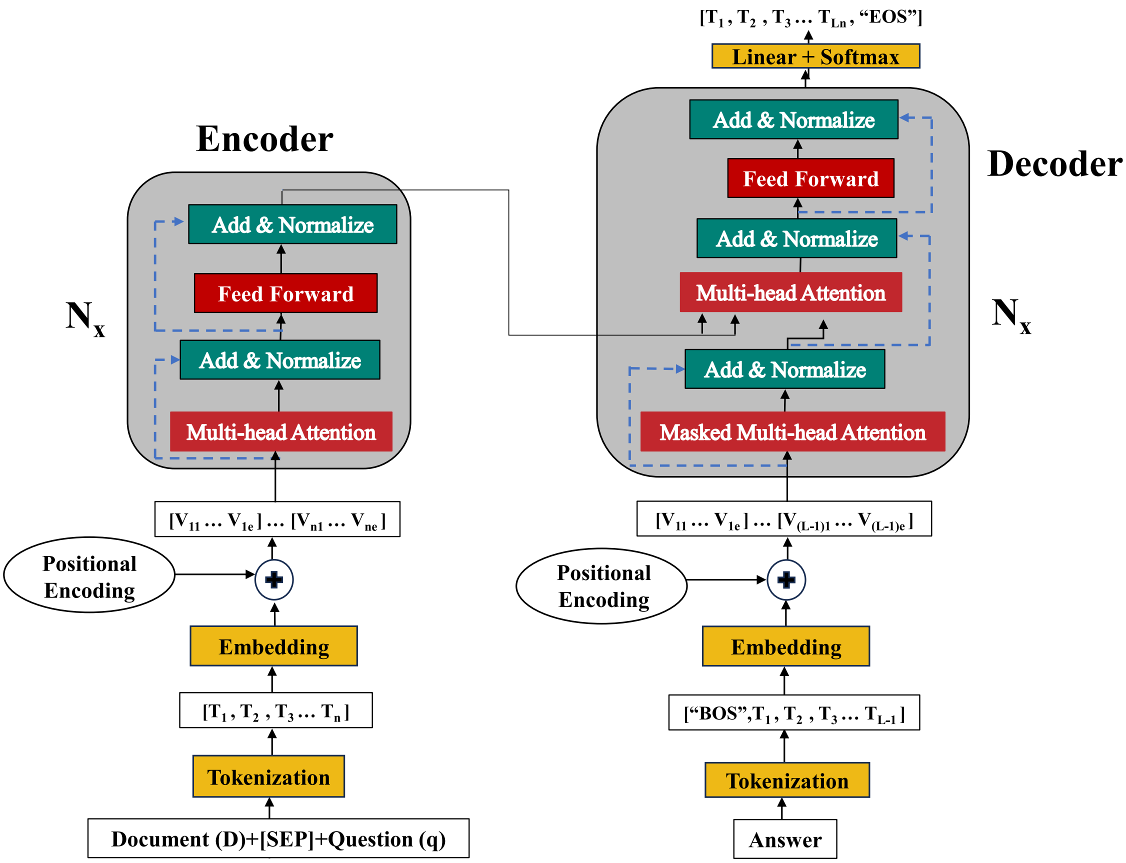

Figure 3. Transformer-based encoder-decoder model for generative QA system.

Fig. 3 provides an overview of the Transformer-based encoder-decoder model for the generative QA system.

The tokenization block splits the concatenated text of the question and document into smaller units called tokens. The list of tokens is mapped into a sequence of numerical tokens (T_1-T_n) that can be processed by the model.

The embedding component maps each token (T_1-T_n) to its corresponding vector representation. The embedding component consists of the embedding matrix which is a two-dimensional matrix (N*D), where N is the vocabulary size and D is the embedding size which determines the dimensionality of token embedding. Each row from the matrix corresponds to the token's dense vector which captures the semantic and syntactic information of the token. These embedding vectors are learned during the training process.

The positional encoding component adds information about the relative positions of the tokens into the input sequence. The positional encoding vectors are computed based on the position of each token in the input sequence, and they have the same dimensionality as the token-embedding vectors. Element-wise addition is applied to add the positional encoding vector to the token embedding.

The primary purpose of the encoder stack is to provide contextual representations that capture the relationships between tokens for each token. Each encoder in the stack consists of two main components: a multihead self-attention and feedforward network (FFN). Multi-head attention computes the attention weights that capture the importance of each token to the others. The attention weights are multiplied by each token embedding to produce a contextualized representation for each token. The main purpose of FFN is to introduce nonlinearity into the model, which allows the capture of more complex patterns. Adding more encoders to the stack allows the model to capture more complex dependencies between tokens and to understand the long-range relation between them. The normalization layer helps the model to mitigate overfitting and vanishing gradient problems by ensuring a consistent distribution of inputs to each layer [26].

Similar to the encoder, the decoder uses multihead self-attention mechanisms. However, it utilizes another attention layer to attend to all positions in the output of the encoder. This helps the decoder to focus on different tokens of the concatenation of the document and question while generating the answer.

The output of the decoders is fed into a linear layer with softmax activation, which plays an important role in generating the output tokens by transforming the high-dimensional vectors that are generated by the decoders into a lower-dimensional space (the number of output neurons from this layer is equal to the vocabulary size). The Softmax layer transforms the output of the linear layer into a probability distribution over the vocabulary.

In the inference phase, the decoder generates answers token-by-token. In each iteration, it predicts the subsequent token based on the context derived from the previous tokens along with the embeddings of the input token. Subsequently, the predicted token is added to the output list. This process begins with the "BOS" token (which represents the start of the sequence) to generate the first token. Consequently, it will be repeated until the "EOS" token (which represents the end of the generated sequence). In the decoding phase, a mask is applied to the multihead self-attention layer to ensure that each position can only attend to previous positions.

For more detailed information about each block, readers can refer to the original paper [26] or follow-up tutorials in [2] or .[3]

T5 [9] is developed based on the Transformer architecture. It is pre-trained using a "masked language modeling" objective, where specific tokens in the input sequence are masked, and the model is trained to predict the masked tokens. By leveraging the concept of transfer learning, we can fine-tune the T5 model to achieve good results when constructing a generative question-answering (QA) system.

B. ENGLISH NARRATIVE QA

Within the domain of narrative QA, questions require a deep comprehension of the story and analyzing the narrative elements of the story: places, events, entities, and the relations between them. This requires producing human-like written answers (answers can not be extracted as a span from the story) by synthesizing information from multiple sections of the story's content. Therefore, it is crucial to employ generative QA models to generate free-text, coherent, and contextually suitable answers.

It is worth noting that only a small proportion of reading comprehension datasets specifically address the understanding of the narrative text. NarrativeQA [21], TellMeWhy [28], and FairytaleQA [10] are the only available narrative datasets that include free-form answers.

- NarrativeQA [21]: It is a large-scale reading comprehension dataset, where the answers are in a free-form format. It contains approximately 47,000 question-answer pairs and around 1,600 stories.
- TellMeWhy [28]: It comprises 30,519 why-questions, each accompanied by three "gold standard" free-form answers. Each entry in the dataset includes a short story, a corresponding question, and three possible answers.
- FairytaleQA [10]: It is specifically designed to evaluate the narrative comprehension skills of students from kindergarten to eighth grade. One advantage of this dataset, in comparison to others, is that it is created by experts rather than being generated through crowd-sourcing. It consists of 10,580 questions that are derived from 278 children-friendly stories and cover 7 narrative elements.

[2]http://jalammar.github.io/illustrated-transformer

[3]http://nlp.seas.harvard.edu/2018/04/03/attention.html

However, all of the aforementioned datasets are limited to the English language. Hence, a significant research gap exists in the development of QA models capable of understanding the narrative test for languages other than English.

Different approaches [7], [12], [21], [29], [30] were conducted to build the English Narrative-QA systems. Whereas the models in [7], [12], and [21] have been developed to address diverse question types, other studies [29], [30] target understanding event reasoning in the narrative text. Questions that address event reasoning are more challenging [29], [30]. Furthermore, they often require fine-tuning large language models, such as GPT-3, or injecting external knowledge into the model to provide some hints in generating the answer.

The study conducted in [21] aimed to enhance the reader's comprehension ability through the implementation of NarrativeQA [21]. This approach involves training a model to effectively answer more complex questions, which necessitated a deep understanding of the narrative, while simultaneously excluding questions that could be answered solely based on local context similarity or global term frequency. The findings of their study reveal that, despite the humans can answer the questions easily, reading comprehension (RC) models encounter obstacles in achieving satisfactory performance.

In [7] and [21], researchers have conducted an experimental assessment comparing extractive and generative reading comprehension (RC) models. The generative models outperformed the extractive counterparts. This can be attributed to the generative models' ability to answer both explicit and implicit questions effectively. On the other hand, the extractive models are only capable to answer the explicit questions.

The narrative QA system was established through the implementation of a Ranker-Reader pipeline, as outlined in the works of [7] and [12]. The ranker component can be implemented using non-learnable methods, such as computing the cosine similarity between the question vector and the paragraph vector, followed by the selection of the top-k paragraphs based on the highest scores. Different Neural-Network-Based embedding techniques can be employed for generating vector representations of both the questions and paragraphs. Some of these techniques compute embedding vectors based on individual words such as Word2Vec [31] and GloVe [32]. On the other hand, more sophisticated models like BERT-based architectures [33] capture context at the sentence or even document level. The strengths and limitations of each technique were discussed in [34], [35], and [36]. Nevertheless, learnable approaches such as fine-tuning the BERT [33] model significantly improved the performance of the ranker block [12].

Many generative [8], [37] and extractive [33] pre-trained models were studied in [7] and [12] to build the Reader component. A significant insight highlighted in [12] emphasizes the utilization of pre-trained language models, specifically GPT2 [37] and BERT [33], to improve the performance of the reader block. This involves employing transfer learning techniques and fine-tuning these pre-trained models on the NarrativeQA dataset [21]. Notably, this approach yielded superior results compared to the results obtained from [21], where different deep network models were trained exclusively on the Narrative QA dataset.

Processing lengthy paragraphs within a story is considered a significant challenge in building narrative QA systems. When the answer is scattered in n paragraphs, the concatenation result of the n paragraphs is then provided as input to the reader model. Training reader models like BART [8] and T5 [9]) on such lengthy paragraphs requires a substantial amount of GPU memory. One potential approach called Fusion-in-Decoder (FiD) [38]. It tackles this problem by initially concatenating each paragraph with the given question and inputting them into the encoder part of the generative model to produce a question-aware vector for each question-paragraph pair. These vectors from all paragraphs are then merged together and provided as input for the decoder part to predict the answer. Many alternative approaches were also discussed in [7].

C. ARABIC NARRATIVE QA

In general, Arabic generative QA models are still in their early stages. Among 18 studies that addressed developing Arabic Chatbots, only one of them used natural language generation (NLG) techniques to generate human-like responses [39]. However, the dataset utilized in [39] was constructed from automatically translated conversations from other English datasets, Furthermore, it includes only question-answer pairs (without context), which can not be sufficient for building a generative QA model that aims to achieve a deeper level of text comprehension. GEN-TYDIQA [23] dataset is the first work(and the only one to the best of our knowledge) proposed to address the building of generative QA models for non-English languages. Whereas this dataset includes human-generated answers, it has only 859 question-answer pairs for the Arabic language and it is not available to the public to date.

As mentioned in section II-B, the datasets for narrative comprehension are limited to the English language. Hence, a significant research gap exists in the development of QA models capable of comprehending Arabic stories. The new Arabic Narrative-QA dataset should include well-designed, reliable, and valid questions. Furthermore, the questions should asses narrative comprehension rather than simple text matching or memorization.

III. METHODOLOGY

In the present research, the implementation of the NarrativeQA system is attained through the employment of the Ranker-Reader pipeline(as illustrated in Fig. 1). During the progression of this research, we explored and evaluated various approaches at each stage of the Ranker-Reader pipeline, aiming to determine the most efficient ones.

To address the scarcity of Arabic-Narrative data, we introduced the Arabic-NarrativeQA dataset. Additionally, to avoid the need for extensive data collection and annotation, two transfer learning methods were employed:

1) Utilizing the pre-trained multilingual models such as mT5 [15] and mBART [16] (multilingual versions of T5 [9] and BART [8] respectively). These pre-trained models are capable of transferring the knowledge acquired during their initial training on extensive datasets [15], [16], enabling them to generalize effectively. This characteristic is valuable when fine-tuning them on the Arabic-NarrativeQA dataset.
2) Cross-lingual transfer learning: By fine-tuning multilingual models on English Narrative-QA datasets, the model gains the ability to comprehend English stories and acquire the knowledge necessary to find answers to input questions. This knowledge can be efficiently transferred by fine-tuning the same model on a small-scale Arabic NarrativeQA dataset.

A. ARABIC-NARRATIVE QA DATASET

To build the Arabic-NarrativeQA dataset, the collection process involves two main phases:

- Translating the stories and question-answer pairs from the FairytaleQA dataset [10] into Arabic. This involved processing 275 stories and 9605 question-answer pairs. A native Arabic speaker translated 70 stories and their corresponding 3,000 question-answer pairs. Moreover, we conducted a validation phase to confirm the grammatical and semantic accuracy of each question and its corresponding answer. Furthermore, the Google Cloud Translate API[4] was employed to translate 205 stories and 7,489 question-answer pairs. In the subsequent sections, we will use the notations "Arabic-NarrativeQA-T", "Arabic-NarrativeQA-T-N", and "Arabic-NarrativeQA-T-G" to denote the complete dataset resulting from translation, the portion translated by the native speaker, and the portion translated using Google API, respectively.
- Gathering new question-answer pairs based on Arabic stories: The 34 Arabic stories used in this collection were sourced from various online websites. Three native Arabic speakers participated in collecting a total of 1000 question-answer pairs. Each individual contributed by writing challenging questions and their corresponding answers after reading the story thoroughly. Each story is divided into multiple paragraphs. The paragraphs that have the answer to each question were also recorded. Detailed information regarding the number of tokens per story, token count per paragraph, token count per question, token count per answer, number of paragraphs per story, question count per story, and question count per paragraph are provided in Table 3. In the subsequent sections, we will use

[4]https://cloud.google.com/translate

"Arabic-NarrativeQA-C" notation to refer to this part of the Arabic-NarrativeQA dataset.

FairytaleQA [10] is selected for two main reasons: Firstly, it encompasses details regarding which paragraphs provide answers to the given questions. This valuable information facilitates supervised fine-tuning of BERT-based models to determine the relevance score of each paragraph to its respective question. Secondly, the FairytaleQA dataset [10] was built by a group of educational experts who specifically addressed seven narrative elements while writing the questions.

Each question in the Arabic-NarrativeQA dataset is also labeled as a local or summary. Local questions can be answered by reading a single paragraph. In contrast, the questions that are labeled with "summary" requires synthesizing information from multiple paragraphs. The questions are further categorized based on whether the answers could be explicitly or implicitly determined from the corresponding paragraph. Explicit questions concern obvious story facts and can be found directly in the text. On the other hand, implicit questions require summarizing and making inferences based on information that is not explicitly stated in the text. Answering implicit questions may involve paraphrasing the text or drawing conclusions based on the story context. Therefore, implicit questions require advanced comprehension capabilities. The weights of implicit to explicit questions are nearly 1:2 and 1:3 in the Arabic-NarrativeQA-C, and Arabic-NarrativeQA-T datasets, respectively. Some examples of implicit and explicit questions, their answers, and the context are illustrated in Table 1.

Moreover, the questions are classified according to their intent, falling into one of the following categories:

1) **Setting**: Questions concerning the place, time, or overall context in which the story events occur.
2) **Character**: Questions that describe the characters, or identify characteristics of the story's characters.
3) **Prediction**: Questions that use evidence from a sequence of events or actions to unfold how a character may behave within the story.
4) **Causal relationship**: Questions pertaining to cause-and-effect connections between different story events or actions.
5) **Outcome resolution**: Questions that ask about the resolution or result of conflicts, troubles, or challenges faced by characters of the story.
6) **Feeling**: Questions that try to uncover the emotional conditions, reactions, states, or experiences of the characters within the story events.
7) **Action**: Questions pertaining to events, behaviors, decisions, and actions that characters perform in the story.

To ensure accuracy in the collection of the Arabic-NarrativeQA-C dataset, the records associated with each story and its corresponding question-answer pairs were initially created by one native speaker and subsequently reviewed by at least one of the other two speakers. An overview of the statistics pertaining to the Arabic-NarrativeQA-C dataset, as well as the distribution of question attributes, is presented in Table 2.

B. CROSS-LINGUAL TRANSFER LEARNING

The lack of high-quality Arabic narrative data presents a significant challenge. To address this problem without the need for extensive data collection and annotation, cross-lingual transfer learning can be employed when there is limited training data available in the target language (Arabic in our case). This study investigates the effectiveness of cross-lingual transfer learning in constructing an Arabic narrative QA system using different multilingual models. This approach has yielded promising results when employed to build an Arabic task-oriented dialogue system [18]. Consequently, we fine-tuned mT5 [15] and mBART [16] models on English stories and question-answer pairs (FairytaleQA [10]). Subsequently, we proceeded with an additional fine-tuning phase on the Arabic-NarrativeQA dataset.

C. ARABIC-NARRATIVE QA SYSTEM

Referring to Section I and Fig. 1, the Ranker-reader pipeline was used to implement the NarrativeQA System. Regarding the Ranker block, we experimented with three approaches to determine P_{sel}:

1) Approach#1: We employed a TF-IDF-based similarity approach for paragraph selection. TF-IDF is a text representation technique commonly used in NLP tasks [4], [40]. Initially, stop words were removed from the text. Following the method described in [4], a TF-IDF vector Vp_j was computed for each $1 \leq j \leq n$, where n represents the total number of paragraphs. Consequently, each paragraph p_j is represented by a TF-IDF vector Vp_j. Furthermore, question q is represented by a TF-IDF vector Vq. To determine the most relevant paragraphs, we computed the similarity score for each paragraph, p_j with respect to question q. Finally, we determine P_{sel} using (1).
$$\underset{j}{\operatorname{argmax}} \left(\frac{Vp_j \cdot Vq}{\|Vp_j\| \cdot \|Vq\|} \right) \tag{1}$$
2) Approach#2: We adopted a methodology similar to Approach#1, with a modification involving the utilization of Arabic pre-trained encoders [24], [41] and multilingual pre-trained language models [42] to extract the vector representations for each paragraph p_j and question q. Notably, this approach involves obtaining these vectors without fine-tuning the model on the Arabic-NarrativeQA dataset. To achieve this, we employed mean pooling on the output of the model for each token to derive the desired representations.
3) Approach#3: In this approach, we concatenate each paragraph p_j with question q to generate paragraph-question pairs (p_j, q) for each $1 \leq j \leq n$, where

Table 1. Examples of implicit and explicit questions, their answers, and the context.

Context	Question	Answer	Type
ما كاد عمر يخرج من غرفته حتى سمع صوتا عاليا يقول له : كل عام وانت بخير يا عمر فقد راى اباه يقف امامه ، ومعه الدراجة الهوائية ، وراى امه تقف امامه ، ومعها السيارة ذات العجلات الاربع . سلم عمر على والديه وهو مسرور ، ثم راى اخاه احمد يقف امامه، ومعه القطار الطويل . اما اخته سعاد فقد احضرت له الطائرة .	من قالْ كل عام وأنت بخير يا عمر لعمر؟ Who said "Happy Birthday to you" to Omar?"	والده ووالدته واخوته His parents, his sister and his brother	Implicit
Omar left his room then he heard a loud voice saying to him: Happy New Year, Omar. He saw his father standing in front of him, with the bike and he saw his mother standing in front of him, with her four-wheeled car. Omar greeted his parents and he was pleased, then he saw his brother Ahmed standing in front of him, with him the long train. As for his sister Souad, she brought him the plane	ماذا أحضر أحمد لعمر؟ What did Ahmed bring to Omar?"	القطار الطويل. The long train"	Explicit

Table 2. Distribution of questions per category.

Category	Count	Percentage (%)
Attributes		
character	108	10.8
causal relationship	151	15.1
action	222	22.2
setting	94	9.4
feeling	157	15.7
prediction	58	5.8
outcome resolution	210	21
Explicit vs Implicit		
explicit	668	66.8
implicit	332	33.2
Local vs Summary		
Local	883	88.3
Summary	117	11.7

Table 3. Statistics of the Arabic-NarrativeQA-C dataset. The mean, standard deviation, minimum, and maximum are reported for the following variables: tokens count per story, tokens count per section, tokens count per question, tokens count per answer, sections count per story, questions count per story, and questions count per section.

Variable	Mean	S.D.	Min	Max
# sections per story	5.7058	2.6293	2	14
# tokens per story	643.1212	356.1700	185	1585
# tokens per section	112.4278	50.0208	37	338
# questions per story	29.4117	15.6883	9	81
# questions per section	5.1989	3.1346	1	14
# tokens per question	9.4934	2.5889	3	21
# tokens per answer	8.3715	5.3367	1	31

n represents the total number of paragraphs. Subsequently, we followed the binary classification approach and fine-tuned the Bert-based models to determine the relevance of each (p_j, q) pair. Finally, the paragraphs are ranked based on the output probability generated by the classification head of the fine-tuned BERT-based model. The number of irrelevant paragraphs is greater than that of relevant paragraphs. To address the issue of class-imbalanced data, an undersampling approach [43] was employed. In addition, various other methods for handling this problem were explored in [43].

With respect to the reader component (Fig. 1), we fine-tuned two multilingual models: mT5 [15] and mBART [16]. These models have been published in various versions, each characterized by differences in model size. Considering the GPU capacity available to us (16 GB), we opted for the "mT5-Base" variant, which consists of 580 million parameters. In addition, we selected the "mBart-Large" variant, which comprises 665 million parameters. Both mT5 [15] and mBART [16] models consist of both encoder and decoder layers, which grant them the capability to generate text. This characteristic is in contrast to Bert-based models, which only consist of encoder layers. Involving the decoder layers in mT5 [15] and mBART [16] enables them to perform tasks, such as text summarization, language translation, and text generation. In both models, the same set of parameters was employed during the training process, including a learning rate of 0.00004, weight decay of 0.01, maximum input length of 800 tokens, batch size of 2, and a total of six epochs. Additionally, the Ranker component was configured to select the top-3 paragraphs. As mentioned in III-B, in cross-lingual settings, we fine-tuned the mT5 [15] and mBART [16] models on English stories and question-answer pairs (FairytaleQA [10]). In this phase, a learning rate of 0.00004 was used for fine-tuning both models. Subsequently, we proceeded with an additional fine-tuning phase on the Arabic-NarrativeQA dataset. In this step, learning rates of 0.00001 and 0.000002 were used to fine-tune the mT5 [15] and mBART [16] models, respectively. Table 4 summarizes the experimental settings- used for fine-tuning the reader model.

IV. RESULTS AND DISCUSSION

As mentioned in Section III-A, the Arabic-NarrativeQA dataset consists of two main parts: translation of the English FairytaleQA dataset and a novel collection of question-answer pairs specifically focused on Arabic narratives. In the subsequent sections, we use the annotations "Arabic-NarrativeQA-T" and "Arabic-NarrativeQA-C" to refer to the first and second parts, respectively. All the experiments in Sections IV-B and IV-C utilize only

Table 4. Summary of the experimental settings for fine-tuning the reader model. The notation "–>" means that the model is fine-tuned in two stages. The parameter values on the left and right sides represent the first and second stages, respectively.

Settings	DataSets	Models & Parameters
Without Cross-lingual	Arabic-NarrativeQA	mT5(lr = 0.00004) mBART (lr = 0.00004)
Cross-lingual	FairytaleQA -> Arabic-NarrativeQA	mT5(lr = 0.00004 -> 0.00001) mBART(lr = 0.00004 -> 0.000002)

the Arabic-NarrativeQA-T part. Nevertheless, Arabic-NarrativeQA-C was used to conduct the experiments described in Section IV-D.

A. EVALUATION METRICS

ROUGE metrics are widely employed to evaluate the quality of text generated automatically by comparing it against ground-truth text produced by humans. After reviewing a number of previously-published research studies [7], [10], [21], [22], we selected the ROUGE-L [44] as the primary evaluation measure for evaluating the performance of the reader component. It measures the ratio between the length of the longest common subsequence shared by the reference text and the generated text, relative to the maximum length of either the reference text or the generated text. In addition to ROUGE-L, we also consider Rouge-1 and Rouge-2 for evaluating the reader model. The primary difference between these Rouge variants is the specific n-gram order considered during the evaluation process. The equations for all Rouge variants are provided in [44].

The evaluation of the ranker in previous studies [7], [21], [22] cannot be conducted separately. This is because NarrativeQA dataset [21], used in these studies lacks the mapping between a question and its respective paragraph containing the answer. In contrast, the Arabic-NarrativeQA dataset includes this mapping information. To evaluate the performance of the ranker independently, we used the Recall@K evaluation metric, that is, the ratio between relevant paragraphs retrieved by the ranker and the total number of relevant paragraphs. K is set to three, as the total number of retrieved paragraphs by the employed ranker equals three. The precision metric was not utilized in this study, as the employed ranker uniformly propagates the top-3 paragraphs to the reader, even if the paragraphs at positions 2 and 3 are not relevant.

B. EFFECTIVENESS OF CROSS-LINGUAL TRANSFER LEARNING

This section studies the effectiveness of cross-lingual transfer learning to improve the performance of multi-lingual pre-trained models [15], [16] in Arabic narrative comprehension. Basically, this section addresses the following questions:

- How effective is cross-lingual transfer learning when applied to multi-lingual models in developing the narrative comprehension models?
- To what degree does the size of the Arabic-NarrativeQA-T impact the performance of these models?

The results of fine-tuning two multilingual models, namely mT5 [15] and mBART [16], using various dataset sizes, are presented in Table 5. These results demonstrate the effect of increasing the size of the fine-tuning data on model performance, both with and without the utilization of cross-lingual learning. The first objective of this section is to investigate the efficacy of cross-lingual transfer learning. This is achieved by comparing the RougeL scores obtained from fine-tuning the multilingual models (mT5 [15] and mBART [16]) on Arabic-NarrativeQA-T, with and without the application of cross-lingual transfer learning. Furthermore, the second objective is to assess the influence of increasing the fine-tuning dataset size on model performance, both in the presence and absence of cross-lingual learning.

The Arabic-NarrativeQA-T utilized in this section consists of two parts: Arabic-NarrativeQA-T-N and Arabic-NarrativeQA-T-G (more details are provided in Section III-A). To ensure linguistic precision, both the validation and test sets are derived from NarrativeQA-T-N. The validation set comprises 15 stories with 544 question-answer pairs, whereas the test set encompasses 10 stories with 340 question-answer pairs. The evaluation metric employed in this study is RougeL. The notations "V" and "T" indicate that the evaluation is performed on the validation and test sets, respectively. It is important to highlight that all stories and their corresponding question-answer pairs in the validation and test datasets are excluded from the English-FairytaleQA dataset [10] during the cross-lingual transfer learning phase.

To address the first objective, we analyze the impact of the cross-lingual transfer learning phase on the performance of the mT5 and mBART models. Table 5 provides an overview of the results. We observe that incorporating the cross-lingual transfer learning phase into the models (mT5 and mBART) significantly improved the performance of both models on both the validation and test datasets. This performance boost is attributed to the model's ability to leverage the knowledge acquired from the English-FairytaleQA dataset [10] and effectively transfer it during fine-tuning on the Arabic-NarrativeQA-T dataset. Remarkably, even with a relatively small training dataset consisting of only 9 Arabic stories and 161 question-answer pairs, the mT5 model achieved promising results. Specifically, when cross-lingual transfer learning was employed, the mT5 model achieved RougeL scores of 0.295 and 0.323 on the validation and test datasets, respectively. In contrast, without the pre-step of cross-lingual transfer learning, the mT5 model only achieved RougeL scores of 0.01 and 0.049 on the validation and test sets, respectively.

To address the second objective, we analyze the results presented in Table 5, which offer valuable insights into the relationship between data size and model performance. The results reveal that increasing the size of the fine-tuning dataset significantly enhances the models' performance. Specifically, when the cross-lingual transfer learning phase is not utilized, the mT5 model exhibits a noticeable performance improvement. The RougeL scores for the validation and test sets

Table 5. Fine-tuning multilingual models using varying dataset sizes. "G" notation means that translations are performed using Google Translate API. The Arabic-NarrativeQA-T dataset is used for fine-tuning and evaluation. The validation set consists of 15 stories and 544 question-answer pairs, whereas the test set comprises 10 stories and 340 question-answer pairs. The results of increasing the dataset size on the multilingual models before and after applying the cross-lingual transfer learning step are reported. All recorded evaluation measures are RougeL scores. The notation (V) denotes the RougeL score on the validation data, whereas (T) represents the RougeL score on the test data.

#Stories	#QA Pairs	Without Cross-Lingual				With Cross-Lingual			
		mt5 (V)	mt5 (T)	mbart (V)	mbart (T)	mt5 (V)	mt5 (T)	mbart (V)	mbart (T)
9	161	0.01	0.049	0.186	0.201	0.295	0.323	0.275	0.278
14	292	0.065	0.068	0.21	0.21	0.298	0.321	0.296	0.3
19	577	0.073	0.104	0.238	0.262	0.296	0.320	0.295	0.311
24	755	0.077	0.079	0.243	0.268	0.3	0.335	0.304	0.317
34	1384	0.147	0.156	0.256	0.283	0.317	0.344	0.309	0.329
45	2116	0.224	0.199	0.277	0.295	0.322	0.345	0.311	0.337
205(G)	7489(G)	0.327	0.372	0.319	0.328	0.343	0.379	0.34	0.33

increase substantially from 0.01 and 0.049 to 0.327 and 0.372, respectively. Similarly, mBART also demonstrates a similar performance enhancement, with RougeL scores increasing from 0.186 and 0.201 to 0.319 and 0.328 for the validation and test sets, respectively. In contrast, after the application of the cross-lingual fine-tuning phase, the magnitude of this performance boost diminishes notably. More specifically, for the mT5 model, the RougeL scores for the validation and test sets increase from 0.295 and 0.323 to 0.343 and 0.379, respectively. Likewise, the mBART model demonstrates an increase in RougeL scores from 0.275 and 0.278 to 0.34 and 0.33 for the validation and test sets, respectively. This reduction in the magnitude of the performance enhancement can be attributed to the fact that the models already possess the necessary knowledge to find the answers from the English stories. Consequently, the models only require a small subset of the Arabic-NarrativeQA-T dataset to transfer this knowledge to build an Arabic Narrative-QA model.

Fig. 4 demonstrates the correlation between the training dataset size for the mBART model and its performance, as measured by the RougeL metric, on the test set of NarrativeQA-T. The results indicate that by using a subset of 34 stories from NarrativeQA-T-N in combination with cross-lingual transfer learning, we achieve a RougeL score equivalent to that obtained when training the model on the entire NarrativeQA-T dataset without cross-lingual settings. This implies that translating only 13% of the English-FairytaleQA dataset [10] to Arabic is sufficient to achieve strong performance in a cross-lingual setting. This finding supports the conclusion that NarrativeQA-T-N, comprising 70 stories created by native Arabic speakers, is a suitable foundation for building a Narrative QA system. Furthermore, the aforementioned findings strongly suggest that cross-lingual transfer learning can be effectively employed to achieve comparable performance with a limited size of the fine-tuning set. Therefore, this method mitigates the requirement for translation costs or the costly acquisition of language-specific annotations for the target language.

In summary, if we have a dataset in a high-resource language like English and we want to build a model for a similar task but in a different language, cross-lingual transfer learning is an effective technique to make it work. Instead of resource-intensive methods such as collecting a large-scale dataset in the target language or translating the entire dataset to train a monolingual model, cross-lingual transfer learning can be utilized to fine-tune a multilingual model on a small-scale dataset.

C. COMPARATIVE ANALYSIS OF VARIOUS APPROACHES FOR RANKER AND READER

As mentioned in Section III, the implementation of the NarrativeQA system is realized by employing the Ranker-Reader pipeline. The evaluation of different approaches at each stage of the Ranker-Reader pipeline is presented in Table 6. The first column of the table holds the employed ranker method, categorized into three distinct approaches mentioned in Section III (separated by a horizontal line in Table 6). Specifically, Approach #1 utilizes TF-IDF, Approach #2 comprises AraBERT [24], AraELECTRA [41], multilingual-BERT [33], and multilingual-distiluse [45], whereas Approach #3 involves AraBERT-Fine-Tuned. In the last group, denoted as True-Ranker, only the relevant paragraphs are passed to the reader component. The second column holds two generative multilingual models utilized to implement the Reader. The cross-lingual transfer learning phase is applied, and only a portion of Arabic-NarrativeQA-T that is translated by a native speaker is employed for fine-tuning and validation. The fine-tuning set consists of 45 stories and 2116 question-answer pairs, the validation set comprises 15 stories and 544 question-answer pairs, and the test set includes 10 stories and 340 question-answer pairs. Evaluation measures for the ranker on the validation and test sets are denoted as Recall@K(V) and Recall@k(T), respectively. As the employed ranker consistently passes the top-3 paragraphs to the reader, K is set to 3. The columns Rouge1(V), RougeL(T), Rouge2(V), Rouge2(T), RougeL(V), and RougeL(T) represent the evaluation measures for the reader on the validation (indicated as "V") and test (indicated as "T") sets.

Due to the high degree of semantic similarity among paragraphs within a story, the task of retrieving evidence from narrative paragraphs becomes more difficult [7], [21],

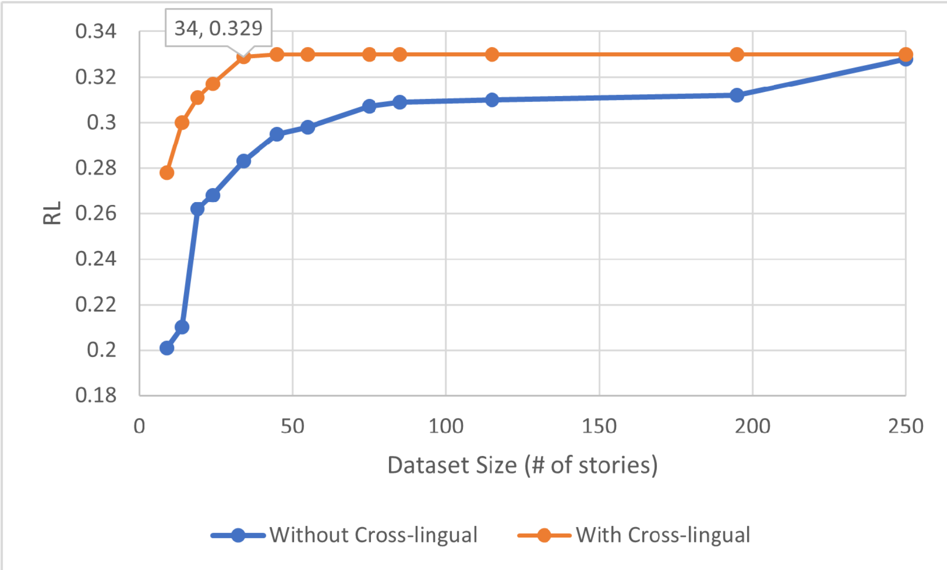

Figure 4. The correlation between the training dataset size and the performance with and without cross-lingual settings.

Table 6. The experimental results achieved by utilization of diverse approaches for the ranker and reader components. The first and second columns represent the employed ranker and reader respectively. The Arabic-NarrativeQA-T-N is utilized. The training set consists of 45 stories and 2116 question-answer pairs, the validation set comprises 15 stories and 544 question-answer pairs, and the test set includes 10 stories and 340 question-answer pairs. Recall@K(V) and Recall@k(T) columns hold the evaluation measures for the ranker on the validation and test sets, respectively. The columns Rouge1(V), RougeL(T), Rouge2(V), Rouge2(T), RougeL(V), and RougeL(T) represent the evaluation measures for the reader on the validation (indicated as "V") and test (indicated as "T") sets.

Ranker	Reader	Recall@K(V)	Recall@K(T)	Rouge1(V)	Rouge1(T)	Rouge2(V)	Rouge2(T)	RougeL(V)	RougeL(T)
TF-IDF	mT5	0.73	0.8	0.246	0.288	0.137	0.144	0.243	0.281
TF-IDF	mBART	0.73	0.8	0.234	0.268	0.125	0.134	0.232	0.266
AraBERT	mT5	0.515	0.674	0.187	0.234	0.098	0.113	0.186	0.231
AraBERT	mBART	0.515	0.674	0.181	0.24	0.087	0.108	0.178	0.237
AraELECTRA	mT5	0.255	0.417	0.127	0.153	0.047	0.059	0.125	0.151
AraELECTRA	mBART	0.255	0.417	0.121	0.164	0.039	0.051	0.119	0.163
multilingual-BERT	mT5	0.475	0.637	0.177	0.216	0.086	0.109	0.175	0.211
multilingual-BERT	mBART	0.475	0.637	0.162	0.212	0.075	0.091	0.16	0.209
multilingual-distiluse	mT5	0.549	0.692	0.18	0.226	0.086	0.114	0.178	0.224
multilingual-distiluse	mBART	0.549	0.692	0.184	0.233	0.087	0.116	0.181	0.233
AraBERT-Fine-Tuned	mT5	0.891	0.917	0.286	0.32	0.164	0.174	0.284	0.316
AraBERT-Fine-Tuned	mBART	0.891	0.917	0.266	0.294	0.149	0.142	0.264	0.289
True-Ranker	mT5	1.0	1.0	0.323	0.351	0.18	0.19	0.322	0.345
True-Ranker	mBART	1.0	1.0	0.314	0.341	0.179	0.185	0.311	0.337

[22]. In our experimentation, Approach#1 and Approach#2 fail to yield satisfactory results as shown in the first two groups of Table 6. These similarity-based approaches determine the most relevant paragraphs by calculating the cosine similarity score between the vector representation of each paragraph and the vector representation of the given question. However, Approach#1 and Approach#2 differ in their corresponding methods of computing the vector representation. To effectively retrieve paragraphs that are most relevant to the given question, we follow a learnable approach by fine-tuning the AraBERT model [24] using the question's evidence information provided in the Arabic-NarrativeQA dataset. This approach yields promising results, with Recall scores of 0.891 and 0.917 on the validation and test sets respectively.

The findings derived from the data presented in Table 6 indicate that errors emanating from the ranker have a cascading effect on the reader. Fine-tuning the mT5 Reader and the utilization of the True-Ranker results in RougeL values of 0.322 and 0.345 for the validation and test sets, respectively. However, when the Ranker fails to retrieve the question's evidence from the story, the RougeL scores decrease significantly. Specifically, using AraBERT-Fine-Tuned Ranker results in reducing the RougeL scores to 0.284 and 0.316 for the validation and test sets, respectively. Similarly, employing the TF-IDF Ranker results in a further decline in the RougeL scores to 0.243 and 0.281 for the validation and test sets, respectively.

Finally, results reveal that Arabic pre-trained encoders fail to effectively generate the vector representations of paragraphs and questions in a zero-shot setting. Nevertheless, the TF-IDF vector representation produced superior results. Thus, other works should be conducted to enhance the capability of Arabic pre-trained encoders in generating embedding vectors for Arabic text. Furthermore, this highlights the necessity of incorporating question evidence information in Arabic-NarrativeQA for fine-tuning the encodes.

D. EVALUATION OF COLLECTED DATASET

This section presents a performance evaluation of the mT5 model [15] on the Arabic-NarrativeQA-C dataset. The results are reported in Table 7. The Arabic-NarrativeQA-C dataset consists of three sets: set#1 (17 stories with corresponding 518 question-answer pairs), set#2 (nine stories with corresponding 283 question-answer pairs), and set#3 (eight stories with corresponding 199 question-answer pairs). The sets #1, #2, and #3 were used for fine-tuning, validation, and testing, respectively. A binary representation was utilized to indicate whether to include a fine-tuning step on the FairytaleQA dataset [10] (cross-lingual transfer learning), Arabic-NarrativeQA-T, or Arabic-NarrativeQA-C. Specifically, the first column indicates whether the mT5 model was fine-tuned on FairytaleQA [10] (1 for fine-tuned, 0 for not fine-tuned), whereas the second column represents the fine-tuning of the Arabic-NarrativeQA-T. Additionally, the third column indicates whether the model is fine-tuned on set#1 of Arabic-NarrativeQA-C. ROUGE-L [44] metric was used to evaluate the model on the validation and test sets of the Arabic-NarrativeQA-C dataset.

The first row of Table 7 shows that fine-tuning the base mT5 model solely on set#1 of Arabic-NarrativeQA-C produces Rouge-L scores of 0.154 and 0.18 on the validation (set#2) and test (set#3) sets, respectively. However, because of the limited size of set#1, this approach is not optimal and requires the inclusion of a cross-lingual transfer learning step. This is demonstrated in the second row, where adding the cross-lingual transfer learning phase results in better performance, with Rouge-L scores of 0.337 and 0.366 on the validation and test sets, respectively. Row#3 shows that replacing the cross-lingual transfer-learning step with the fine-tuning step on the Arabic-NarrativeQA-T does not lead to any performance boost. Furthermore, combining both steps (fine-tuning the model as indicated in row#4) does not further improve the results because the model has already acquired knowledge on how to answer the questions on the English stories during fine-tuning on the FairytaleQA dataset. Therefore, the additional step of fine-tuning the model on the Arabic-NarrativeQA-T did not provide any additional knowledge to the model.

The results in the last two rows of Table 5 indicate that the model was not overfitted with the Arabic-NarrativeQA-T dataset. Nevertheless, it can handle unseen data effectively and generalize well with the NarrativeQA-C dataset. More specifically, row#5 shows the evaluation results of fine-tuning the base mT5 model on the Arabic-NarrativeQA-T dataset, and testing it with the Arabic-NarrativeQA-C dataset. In this

Table 7. Evaluation of the mT5 model on Arabic-NarrativeQA-C. The first column denotes whether the mT5 model is fine-tuned on English FairytaleQA (1 for fine-tuned, 0 for not fine-tuned), and the second column represents the fine-tuning on the Arabic-NarrativeQA-T. Additionally, the third column indicates whether the model is fine-tuned on set#1 of Arabic-NarrativeQA-C.

cross-lingual	translated	collected	validation	test
0	0	1	0.154	0.181
1	0	1	0.337	0.366
0	1	1	0.295	0.328
1	1	1	0.318	0.334
0	1	0	-	0.32
1	1	0	-	0.329

setting, the mT5 model achieves a 0.32 RougeL score on the test set (because the model is not fine-tuned on the Arabic-NarrativeQA-C, the result for the validation set is not reported). Furthermore, adding another phase of fine-tuning on Arabic-NarrativeQA-C adds a small margin of improvement (from 0.32 to 0.328) as illustrated in row#3.

The combination of cross-lingual = 1, translated = 0, and collected = 0 is excluded because it leads to the ''accidental translation'' [15] problem. When the mT5 [15] model is exclusively fine-tuned on the English-FairytaleQA dataset, it tends to mistakenly translate a portion of its predictions into English. Therefore, it is crucial to fine-tune the model on a set of Arabic stories to mitigate this problem. The following is an example of this problem:

- Question:
  كيف تطورت علاقة الصداقة بين التمساح والقرد بمرور الوقت؟
- Translation of the question: How did the friendship between the crocodile and the monkey develop over time?
- Model Output: It becomes
  صديقين حميمين يتحدثان ويحكيان القصص لبعض
- Translation of model output: It becomes two best friends talking and telling stories to each other.
- Correct Answer:
  أصبح التمساح بعدها يزور القرد كل يوم إلى أن أصبح الحيوانان صديقين حميمين
- Translation of the correct answer: The crocodile then visited the monkey every day until the two became good friends.

The above example demonstrates the problem of ''accidental translation'', where the model's output consists of Arabic and English tokens. Nevertheless, the semantic representation of the English tokens is inherently correct. In other words, substituting English tokens with their corresponding Arabic translations significantly improved the accuracy of the model's output.

An additional phase of the assessment involves a comparison between the mT5 model and ChatGPT.[5] For this evaluation, the identical test set (set#3 of Arabic-NarrativeQA-C) is used to assess ChatGPT's performance in comprehending

[5]https://chat.openai.com/

Table 8. Evaluating the generated answers based on the Rouge-L measure. The question, mT5's output, ChatGPT's output, and the correct answer were reported in the first, second, third, and fourth columns respectively. Subsequently, the Rouge-L scores for mT5 and ChatGPT were computed and recorded in the fifth and sixth columns, respectively.

Question	mT5Output	ChatGPTOutput	Correct Answer	mT5R	GPTR
ما هو الدرس الذي يمكن تعلمه من قصة الفيل أبو الفداء؟ What is the lesson that can be learned from the story of the elephant Abu al-Fida?	كيف تستغل وقته وقوتله العضلية والعقلية في مساعدة الناس How to use his time and muscle and mental strength to help people	الدرس الذي يمكن تعلمه من قصة الفيل أبو الفداء هو أهمية العطاء والمساعدة في جعل العالم مكاناً أفضل، وأن قوتنا وقدراتنا يمكن أن تكون سبباًا في تغيير حياة الناس للأفضل. The lesson that can be learned from the story of Elephant Abu al-Fida is the importance of giving and helping to make the world a better place, and that our strength and abilities can change people's lives for the better.	كيف نستغل وقتنا وقوتنا العضلية والعقلية في مساعدة الناس How to use our time, muscle and mental strength to help people	0.88	0.10
بماذا شعر الفيل بعد طرد اللصوص؟ What did the elephant feel after expelling the thieves?	شعر بالسعادة لأنه استغل قوته في شيء مفيد He felt happy that he had used his strength for something useful.	بعد طرد اللصوص، شعر الفيل بالسعادة والرضا لأنه استغل قوته ووقته في شيء يكون مفيداً ومنفعاًا للآخرين. After driving away the thieves, the elephant felt happy and satisfied because he used his strength and time for something that would be useful and beneficial to others.	بالسعادة Happy.	0.222	0.105
ما هو سبب قحل الوادي؟ What is the reason for the aridity of the valley?	جشعها Their greed	سبب قحل الوادي هو جشع الجداء وتناولهم كل العشب الأخضر دون ترك شيء للآخرين The reason for the barrenness of the valley is the greed of the goats and their eating all the green grass without leaving anything for others	بسبب جشعها فلقد تناولت كل العشب ولم يتبق منه شيء يذكر Because of their greed, they ate all the grass and there was nothing left of it	0.16	0.48

Arabic stories. ChatGPT is evaluated in a zero-shot setting, where the model is prompted with question-story pairs and generates the answer without fine-tuning it on set#1 of Arabic-NarrativeQA-C. The ROUGE-L score obtained for the ChatGPT is 0.232. In comparison, mT5 achieved a higher ROUGE-L score of 0.366 on the same test set (as indicated in Table 7), indicating a comparatively enhanced comprehension ability and generating more accurate answers.

E. PERFORMANCE ANALYSIS

Whereas the ROUGE-L metric is widely utilized for evaluating machine-generated text, it does not consider synonymous expressions that carry equivalent semantic meanings. As a result, when a model produces an answer that is semantically correct but employs certain synonyms, the ROUGE-L score may be lower. Table 8 presents some acceptable answers despite their relatively low Rouge-L scores. In the first example, the mT5 model demonstrates a notable overlap with the tokens present in the reference answer. On the other hand, ChatGPT generates a semantically correct answer by including synonymous terms, such as the word "abilities" mentioned in Table 8. As a result, the mT5 achieves a notably improved Rouge-L score. In the second example, both the mT5 and ChatGPT models provide correct answers when prompting feelings of the elephant. Nevertheless, both models continue to generation answers to clarify the factors affecting this feeling, leading to relatively low Rouge-L scores for both models. Finally, in the third example, the mT5 model generates a succinct answer when queried regarding the reason for the aridity of the valley. Consequently, mT5 obtains a relatively low Rouge-L score.

An additional phase of the investigation is conducted to identify the characteristics of the questions that the models struggle to answer. These challenging questions are classified into three primary categories:

- Questions that ask about **implicit feeling and emotion**
- The correct answer for the question is **scattered in many paragraphs**
- Questions that require **multi-hop reasoning**: Answer this type of question, requires a deeper level of reasoning, collecting, and combining disparate fragments of information from the story to generate the answer.

Table 9. **Identifying some challenges in finding the answers. The question, mT5's output, ChatGPT's output, and the correct answer were reported in the first, second, third, and fourth columns respectively. Subsequently, the specific challenge associated with each question is recorded in the fifth column.**

Question	mT5Output	ChatGptOutput	Correct Answer	Problem
كيف يتأثر التمساح عاطفياً بقراره بقتل القرد رغم صداقتهما؟ How is the crocodile emotionally affected by his decision to kill the monkey despite their friendship?	يشعر بأنه أحمق لأنه خسر صديقاً ومخزوناً من الفاكهة اللذيذة الحلوة. He feels like a fool for losing a friend and a supply of sweet, delicious fruit.	التمساح يتأثر عاطفياً بقراره بقتل القرد رغم صداقتهما، فهو يعلم أن هذا القرار سيؤثر بشكل كبير على علاقتهما ويفقده صديقاً عزيزاً عليه. The crocodile is emotionally affected by his decision to kill the monkey despite their friendship, as he knows that this decision will greatly affect their relationship and cause him to lose a dear friend.	يشعر بالأسف والحزن He feels sorry and sad	Implicit Feeling
من ماذا يتكون الحساء الذي أعده الجندي؟ What does the soup prepared by the soldier consist of?	القليل من الجزر few carrots	الحساء الذي أعده الجندي كان يحتوي على مجموعة من المكونات مثل الملفوف والبطاطس والبصل وشجيرات الجزر ولحم البقر والفطر والشعير. The soup the soldier prepared contained an array of ingredients such as cabbage, potatoes, onions, carrot bushes, beef, mushrooms, and barley.	الماء والملح والفلفل والملفوف والجزر والبصل والبطاطا ولحم البقر والفطر والشعير Water, salt, pepper, cabbage, carrots, onions, potatoes, beef, mushrooms, and barley	Answer scattered in many paragraphs
ماذا حدث بين القرد والتمساح عندما عاد القرد لشجرة الجامون؟ What happened between the monkey and the crocodile when the monkey returned to the jamon tree?	قفز القرد بعيداً عن ظهر التمساح وتسلق حتى وصل إلى مكانه الآمن في الشجرة The monkey jumped off the crocodile's back and climbed to his safe place in the tree	عندما عاد القرد لشجرة الجامون، حاول التمساح إغرائه بالعودة معه بذريعة تقديم شكر زوجته له على الفاكهة، لكن القرد رفض وأخبره أنه لن يعود مرة أخرى. When the monkey returned to the jamon tree, the crocodile tried to lure him back with him under the pretext of thanking his wife for the fruit, but the monkey refused and told him that he would not return again.	القرد أخبر التمساح أنه لن يثق به أبداً مرة أخرى أو يعطيه ثمراً من شجرته. وطلب منه أن يذهب بعيدا ولا يعد The monkey told the crocodile he would never trust him again or give him fruit from his tree. He asked him to go away and not come back	Multi-Hop Reasoning

Table 9 shows some of these questions along with the corresponding answers generated by mT5 and ChatGPT, the ground-truth answer, and the specific challenge associated with each question.

The first row demonstrates one of the questions that requires the model to conclude the emotional state associated with a particular situation. When emotional feelings are not explicitly mentioned in the text, the task of answering such questions becomes more difficult. The question asks "how the crocodile is emotionally affected by the loss of his friendship with the monkey?". The mT5 model generated a response that closely answers the question, capturing what the crocodile was thinking about himself after the incident but without explicitly mentioning the emotional feeling itself. On the other hand, ChatGPT focused on paraphrasing the question without directly answering the emotional state.

The second row of Table 9 shows a question regarding the ingredients of the soup that are scattered in multiple paragraphs. Answering this question entails comprehending two paragraphs to accurately enumerate all the ingredients. Water, salt, pepper, cabbage, and carrots are scattered in the first paragraph, whereas other ingredients are scattered in the second paragraph. The mT5 extracted only the final ingredient that was explicitly mentioned in the first paragraph. Although ChatGPT, was unable to enumerate all the ingredients, it yielded an answer that was closer to the ground truth. After a thorough debugging process, we identified a potential cause of this issue. The ranker erroneously retrieved an unrelated paragraph and ranked it in the first position. The two relevant paragraphs containing the necessary ingredients were ranked in the second and third positions, respectively. As illustrated in Section III, only the top-3 paragraphs are passed to the reader due to GPU limitations. Consequently, the paragraph with crucial ingredients at the third position was trimmed, as the concatenation of all paragraphs (in the ranking order) exceeded the maximum input length of 800 for the reader. Owing to the trimming of important information from the input of the reader, the generated answer does not contain all ingredients. To address this issue, a future study will be conducted to increase the maximum input length of the reader model (utilizing Fusion-in-Decoder (FiD) [38]). Consequently, the critical content was prevented from being trimmed.

The last example in Table 9 demonstrates the need for multihop reasoning to answer challenging questions. In particular, the question "What happened between the monkey and the crocodile when the monkey returned to the jamon tree?" relies on the model's ability to understand the sequence of events and the interactions between the monkey and the crocodile before and after the incident.

وافق القرد وقفز على ظهر التمساح...

وبهذا تحرك الصديقان نحو النهر الواسع العميق.

وعندما أصبح الصديقان بعيدين عن ضفة النهر وشجرة الجامون، قال التمساح للقرد أنا آسف جداً ... يؤسفني أن أقول لك إنني مضطر إلى قتلك، رغم أنني سأفتقد أحاديثنا. فكر الفرد بسرعة وقال ...ولكنني تركت قلبي خلفنا في شجرة الجامون. هل تظن أن بإمكاننا العودة حتى آخذه؟ صدّق التمساح القرد وعاد سريعاً لشجرة الجامون. قفز القرد بعيداً عن ظهر التمساح وتسلق حتى وصل إلى مكانه الآمن في الشجرة.

وقال له ظننت أنك صديقي. ألا تعرف أن قلوبنا موجودة داخلنا؟ لن أثق بك أبداً مرة أخرى أو أعطيك ثمراً من شجرتي. اذهب بعيداً ولا تأتِ هنا مرة أخرى.

شعر التمساح بأنه أحمق لأنه خسر صديقاً ومخزوناً من الفاكهة اللذيذة الحلوة. وأنقذ القرد نفسه لأنه فكر بسرعة. ومنذ ذلك اليوم، لم يثق في التمساح أبداً مرة أخرى.

*"**The monkey agreed and jumped on the crocodile's back**... With this, the two friends **moved towards the wide, deep river**. And when the **two friends were far from the bank of the river and the jamon tree**, the crocodile said to the monkey, "I am very sorry... I am sorry to tell you that **I must kill you**, though I shall miss our conversations." The monkey thought quickly and said, "... but I left my heart behind in the jamon tree. Do you think we can come back to get it?" The crocodile believed the monkey and quickly **ran back to the jamon tree**. **The monkey jumped off the crocodile's back and climbed to his safe place in the tree**. He told him, "I thought you were my friend. Don't you know that our hearts are within us? **I will never trust you again or give you fruit from my tree. Go away and don't come here again.**" The crocodile felt foolish for losing a friend and a supply of sweet, delicious fruit. And the monkey saved himself because he thought quickly. From that day on, he never trusted the crocodile again."*

The sequence of events is bolded in the paragraph. The event of climbing the tree occurred immediately after returning to it. However, this event is incorrect because the question inquired about the interaction between the monkey and the crocodile at that particular moment. The correct subsequent event is when the monkey said, "I will never trust you again or give you fruit from my tree." However, to decide whether this event is the correct one and to generate a human-readable answer, a crucial step involves understanding the pronouns, and all preceding events must be considered.

It is observed that the mT5 model ignores a specific aspect of the query, namely the relationship "between the monkey and the crocodile." Instead, the model generated an answer related to the event of the monkey returning to the jamon tree. By contrast, ChatGPT generates novel events that do not entirely exist in the entire story.

V. CONCLUSION AND FUTURE WORK

This study addressed the challenging task of Narrative Comprehension in the Arabic language. The primary gap is the scarcity of available Arabic narrative datasets. This gap was filled by introducing the Arabic-NarrativeQA dataset. Two paths were followed to construct this dataset: Translating the stories and question-answer pairs from FairytaleQA dataset into Arabic and collecting new question-answer pairs on Arabic stories. To the best of our knowledge, this is the first machine-reading comprehension dataset tailored specifically for Arabic stories.

The Arabic-NarrativeQA system was implemented using the Ranker-Reader pipeline, with the exploration and evaluation of various approaches and models at each stage. Selecting the paragraphs that are most relevant to a question is a more challenging task because of the high degree of semantic similarity among the paragraphs within a story. To address this issue, a learnable ranker was implemented. Consequently, the AraBERT model was fine-tuned using evidence information provided in the Arabic-NarrativeQA dataset. Regarding the reader component, two multilingual generative models were employed to generate the answer after comprehending the selected paragraphs passed by the ranker.

Moreover, the effectiveness of cross-lingual transfer learning was investigated to improve the model performance. Experiments showed that incorporating the cross-lingual transfer learning step into the models (mT5 and mBART) significantly improved the performance of both models. This refers to the model's ability to leverage the knowledge acquired from the English stories and effectively transfer it during fine-tuning on the Arabic-NarrativeQA dataset.

By introducing the Arabic-NarrativeQA dataset and making it publicly accessible, the aim is to encourage advanced research in Arabic Narrative Comprehension tasks. The Arabic-NarrativeQA dataset and techniques employed in the Ranker-Reader pipeline are expected to serve as a foundation for future research and to facilitate the development of more sophisticated models that can accurately and contextually answer questions based on Arabic stories.

The current study addressed the limitations of the implemented readers, which process only three paragraphs from the ranker owing to the available GPU memory. Consequently, the reader models may fail to generate correct answers if the required information is implicitly spread across more than three paragraphs. To overcome this problem, future work will explore fusion-based techniques for processing several paragraphs. Specifically, each paragraph is concatenated with a question and input into the encoder part of the generative model. This process produces question-aware vectors for each question-paragraph pair, which are then fused together and utilized as input for the decoder to generate more accurate answers.

VI. ETHICAL CONSIDERATIONS

In the data collection phase, each contributor read the entire story to ensure that the stories were appropriate within Arab culture. Phrases that could be considered inappropriate, potentially harmful, or culturally sensitive were manually replaced or removed. The stories were exported mainly from two websites.

1) hindawi.org A non-profit organization that aims to spread knowledge and culture among Arabic speakers. By referring to the privacy policy, the content is free for learning purposes.
2) mawdoo3.com By referring to point#4 of the privacy policy, the content can be exported for non-profit educational purposes.

Regarding the FairytaleQA dataset [10], the authors explicitly stated that it's for public research use. In the context of this study, both the dataset and fine-tuned models are accessible to other researchers for further model development and non-profit educational purposes.

References

[1] S. Liu, D. Wang, X. Li, M. Huang, and M. Ding, "A copy-augmented generative model for open-domain question answering," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 435–441.

[2] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 1906–1919.

[3] R. Malhas and T. Elsayed, "Arabic machine reading comprehension on the holy Qur'an using CL-AraBERT," *Inf. Process. Manage.*, vol. 59, no. 6, Nov. 2022, Art. no. 103068.

[4] K. Alsubhi, A. Jamal, and A. Alhothali, "Deep learning-based approach for Arabic open domain question answering," *PeerJ Comput. Sci.*, vol. 8, p. e952, May 2022.

[5] H. Mozannar, E. Maamary, K. El Hajal, and H. Hajj, "Neural Arabic question answering," in *Proc. 4th Arabic Natural Lang. Process. Workshop*, Florence, Italy. Stroudsburg, PA, USA: Association for Computational Linguistics, Aug. 2019, pp. 108–118.

[6] A. Fuad and M. Al-Yahya, "Recent developments in Arabic conversational AI: A literature review," *IEEE Access*, vol. 10, pp. 23842–23859, 2022.

[7] X. Mou, C. Yang, M. Yu, B. Yao, X. Guo, S. Potdar, and H. Su, "Narrative question answering with cutting-edge open-domain QA techniques: A comprehensive study," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 1032–1046, Sep. 2021.

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880.

[9] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[10] Y. Xu, D. Wang, M. Yu, D. Ritchie, B. Yao, T. Wu, Z. Zhang, T. Li, N. Bradford, B. Sun, T. Hoang, Y. Sang, Y. Hou, X. Ma, D. Yang, N. Peng, Z. Yu, and M. Warschauer, "Fantastic questions and where to find them: FairytaleQA—An authentic dataset for narrative comprehension," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, Dublin, Ireland. Stroudsburg, PA, USA: Association for Computational Linguistics, May 2022, pp. 447–460.

[11] W. Peng, W. Li, and Y. Hu, "Leader-generator net: Dividing skill and implicitness for conquering FairytaleQA," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2023, pp. 791–801.

[12] X. Mou, M. Yu, B. Yao, C. Yang, X. Guo, S. Potdar, and H. Su, "Frustratingly hard evidence retrieval for QA over books," in *Proc. 1st Joint Workshop Narrative Understand., Storylines, Events*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 108–113.

[13] G. Izacard and E. Grave, "Distilling knowledge from reader to retriever for question answering," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–13.

[14] S. Ruder, M. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Tuts.*, A. Sarkar and M. Strube, Eds., Minneapolis, MN, USA. Stroudsburg, PA, USA: Association for Computational Linguistics, Jun. 2019, pp. 15–18.

[15] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "MT5: A massively multilingual pre-trained text-to-text transformer," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.* Stroudsburg, PA, USA: Association for Computational Linguistics, Jun. 2021, pp. 483–498.

[16] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 726–742, Dec. 2020.

[17] M. M'hamdi, D. S. Kim, F. Dernoncourt, T. Bui, X. Ren, and J. May, "X-METRA-ADA: Cross-lingual meta-transfer learning adaptation to natural language understanding and question answering," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.* Stroudsburg, PA, USA: Association for Computational Linguistics, Jun. 2021, pp. 3617–3632.

[18] A. Fuad and M. Al-Yahya, "Cross-lingual transfer learning for Arabic task-oriented dialogue systems using multilingual transformer model mT5," *Mathematics*, vol. 10, no. 5, p. 746, Feb. 2022.

[19] B. M´uller, Y. Elazar, B. Sagot, and D. Seddah, "First align, then predict: Understanding the cross-lingual ability of multilingual BERT," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 2214–2231.

[20] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 878–891.

[21] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette, "The NarrativeQA reading comprehension challenge," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 317–328, Dec. 2018.

[22] E. Kalbaliyev and K. Sirts, "Narrative why-question answering: A review of challenges and datasets," in *Proc. 2nd Workshop Natural Lang. Gener., Eval., Metrics (GEM)*, 2022, pp. 520–530.

[23] B. Müller, L. Soldaini, R. Koncel-Kedziorski, E. Lind, and A. Moschitti, "Cross-lingual open-domain question answering with answer sentence generation," in *Proc. 2nd Conf. Asia–Pacific Chapter Assoc. Comput. Linguistics 12th Int. Joint Conf. Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2022, pp. 337–353.

[24] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, With Shared Task Offensive Lang. Detection*, H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, and H. Mubarak, Eds. Marseille, France: European Language Resource Association, May 2020, pp. 9–15. [Online]. Available: https://aclanthology.org/2020.osact-1.2

[25] K. Palasundram, N. M. Sharef, K. A. Kasmiran, and A. Azman, "Enhancements to the sequence-to-sequence-based natural answer generation models," *IEEE Access*, vol. 8, pp. 45738–45752, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[27] Y. Huang and T. Zhong, "Multitask learning for neural generative question answering," *Mach. Vis. Appl.*, vol. 29, no. 6, pp. 1009–1017, Aug. 2018.
[28] Y. K. Lal, N. Chambers, R. Mooney, and N. Balasubramanian, "TellMe-Why: A dataset for answering why-questions in narratives," in *Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Aug. 2021, pp. 596–610.
[29] Y. K. Lal, N. Tandon, T. Aggarwal, H. Liu, N. Chambers, R. Mooney, and N. Balasubramanian, "Using commonsense knowledge to answer why-questions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Abu Dhabi, United Arab Emirates. Stroudsburg, PA, USA: Association for Computational Linguistics, Dec. 2022, pp. 1204–1219.
[30] U. Katz, M. Geva, and J. Berant, "Inferring implicit relations in complex questions with language models," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2022, pp. 2548–2566.
[31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.
[32] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, vol. 1, 2019, pp. 4171–4186.
[34] R. H. Chassab, L. Q. Zakaria, and S. Tiun, "Automatic essay scoring: A review on the feature analysis techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 10, pp. 252–264, 2021.
[35] M. H. Ahmed, S. Tiun, N. Omar, and N. S. Sani, "Short text clustering algorithms, application and challenges: A survey," *Appl. Sci.*, vol. 13, no. 1, p. 342, Dec. 2022.
[36] M. Suhaidi, R. A. Kadir, and S. Tiun, "A review of feature extraction methods on machine learning," *J. Inf. Syst. Technol. Manage.*, vol. 6, no. 22, pp. 51–59, Sep. 2021.
[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.
[38] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Apr. 2021, pp. 874–880.
[39] T. Naous, C. Hokayem, and H. Hajj, "Empathy-driven Arabic conversational chatbot," in *Proc. 5th Arabic Natural Lang. Process. Workshop*, 2020, pp. 58–68.
[40] S. M. Al-Ghuribi, S. A. M. Noah, and S. Tiun, "Unsupervised semantic approach of aspect-based sentiment analysis for large-scale user reviews," *IEEE Access*, vol. 8, pp. 218592–218613, 2020.
[41] W. Antoun, F. Baly, and H. Hajj, "AraELECTRA: Pre-training text discriminators for Arabic language understanding," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, Kyiv, Ukraine. Stroudsburg, PA, USA: Association for Computational Linguistics, Apr. 2021, pp. 191–195.
[42] W. Wang, G. Chen, H. Wang, Y. Han, and Y. Chen, "Multilingual sentence transformer as a multilingual word aligner," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, Abu Dhabi, United Arab Emirates. Stroudsburg, PA, USA: Association for Computational Linguistics, Dec. 2022, pp. 2952–2963.
[43] A. Y. Taha, S. Tiun, A. H. A. Rahman, and A. Sabah, "Multilabel over-sampling and under-sampling with class alignment for imbalanced multilabel text classification," *J. Inf. Commun. Technol.*, vol. 20, no. 3, pp. 423–456, 2021.
[44] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004, pp. 74–81.
[45] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*. Stroudsburg, PA, USA: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

MOHAMMAD A. ATEEQ received the B.Sc. and M.Sc. degrees in computer engineering from Birzeit University, Palestine, in 2015 and 2019, respectively. He is currently pursuing the Ph.D. degree in computer science with Universiti Kebangsaan Malaysia (UKM). His research interests include natural language processing and computer vision.

SABRINA TIUN is currently an Associate Professor with the Faculty of Information Science and Technology (FTSM), Universiti Kebangsaan Malaysia (UKM). She is also a member of the Asian Language Processing (ASLAN) Research Group, Centre of Artificial Intelligence (CAIT). She is the Head of the Program of CAIT Postgraduate, UKM. Her research interests include natural language processing and computational linguistic to speech processing.

HAMED ABDELHAQ received the Ph.D. degree in computer science from Heidelberg University. He finds his passion in data mining, NLP, and deep learning. He is currently an Assistant Professor with An-Najah National University, Palestine, where he teaches in the computer science apprenticeship CAP Program-AI track. Driven by a curiosity to explore the applications of AI, his work aims to explore the opportunities where AI techniques can support and improve other fields.

NAWRAS RAHHAL received the B.S. degree in computer systems engineering from Palestine Technical University, Palestine, in 2018, and the M.S. degree in computer science from Arab American University, Palestine, in 2022.