Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 09, No. 01, March 2023.

EFFECTIVENESS OF ZERO-SHOT MODELS IN AUTOMATIC ARABIC POEM GENERATION

Mohamed El Ghaly Beheitt and Moez Ben HajHmida*

(Received: 25-Oct.-2022, Revised: 9-Jan.-2023, Accepted: 13-Jan.-2023)

ABSTRACT

Text generation is one of the most challenging applications in artificial intelligence and natural-language processing. In recent years, text generation has gained much attention thanks to the advances in deep-learning and language-modeling approaches. However, writing poetry is a challenging activity for humans that necessitates creativity and a high level of linguistic ability. Therefore, automatic poem generation is an important research issue that has attracted the interest of the Natural Language Processing (NLP) community. Several researchers have examined automatic poem generation using deep-learning approaches, but little has focused on Arabic poetry. In this work, we exhibit how we utilize various GPT-2 and GPT-3 models to automatically generate Arabic poems. BLEU scores and human evaluation are used to evaluate the results of four GPT-based models. Both BLEU scores and human evaluations indicate that fine-tuned GPT-2 outperforms GPT-3 and fine-tuned GPT-3 models, with GPT-3 model having the lowest value in terms of poeticness. To the best of the authors' knowledge, this work is the first in literature that employs and fine-tunes GPT-3 to generate Arabic poems.

KEYWORDS

Natural-language processing (NLP), Natural-language generation (NLG), Deep learning, Transformer, GPT-2, GPT-3, Arabic poems.

1. INTRODUCTION

Natural-language Generation (NLG) is a challenging topic that has piqued the interest of the Natural Language Processing (NLP) community [1]. Poem generation is an example of NLG that is particularly interesting due to its unique characteristics. One of the most challenging tasks in NLG is automatic poem generation, since poetry is an art form. Nowadays, many researchers are motivated to contribute to automatic poem generation [2][3][4][5][6][7][8][9][10]. Stochastic search [11] and statistical machine-translation models [12] are traditionally recommended for this task. Lately, deep neural networks have been utilized to generate natural-sounding poetry [13]-[14]. Although these models appear to be promising, they are severely limited in various ways. Previous research, for example, frequently fails to retain topic coherence [15]-[16] and increase word variety [14], both of which are essential qualities of poems. Compared to the efforts made in English and Chinese for NLG, Arabic deep-learning applications for NLG, especially Arabic-poetry generation, remain limited.

Arabic poetry is the oldest type of Arabic literature. Poetry has represented the most profound sense of Arab self-identity and collective past and future ambitions in the Arabic literary tradition. Arabic poetry is frequently classified into classical and modern poetry (also named free poetry) [17]. Appropriately, any poetry written in the classical form is referred to as "traditional poetry "or" vertical poetry" as long as it follows the traditional form and structure and the vertical parallel composition of its two components known as hemistichs. A hemistich is a half-line of verse, followed by a second hemistich, making together a verse unit. On the other hand, modern poetry differs from traditional poetry in terms of form, structure, rhyme and subject matter. Arabic poetry is defined as rhymed, metered speech. Table 1 exposes examples of verses from Arabic poetry.

Large language models, like GPT-3 [18], are tens of gigabytes in size, trained on terabytes of text data and have mostly been designed for text generation. GPT-3 was introduced by OpenAI as the largest language model when compared to state-of-the-art language models, with 175 billion trainable parameters [18]. Apart from raw-text generation, GPT-3 can also generate poems, codes, stories, …etc. Brown et al. [18] demonstrated that GPT-3 excelled in various NLP tasks with few-shot learning and even without any fine-tuning.

M. El G. Beheit and *M. Ben HajHmida (Corresponding Author) are with University of Tunis El Manar, Tunis, Tunisia. Emails: medghaly.beheitt@fst.utm.tn and moez.benhajhmida@enit.utm.tn

GPT-2 was succeeded by GPT-3, which utilizes the same transformer architecture. A major distinction between the two models is their size; GPT-3 has 175 billion parameters, significantly more than GPT-2 with 1.5 billion parameters. We are challenged to measure the effectiveness of fine-tuning GPT-2 compared to the use of a zero-shot or few- shot GPT-3 model in the case of Arabic-poem generation. In this work, we compare our previously proposed fine-tuned GPT-2 model [19] to state-of-the-art models as well as some variants of GPT-3 model on Arabic-poem generation. In this comparison, we use automatic evaluation; namely, BLEU scores and human evaluation. To the best of our knowledge, this is the first work in literature that employs GPT-3 to generate Arabic poems.

Table 1. Example of verses from Arabic poems from Arab poet Abu al-Tayyib Ahmad ibn Al-Husayn Al-Mutanabbi [17].

Arabic verses	**English translation**
أنا الذي نظر الأعمى إلى أدبي وأسمعت كلماتي من به صمم	I am who made the blind see my art and made the deaf hear my words
وإذا كانت النفوسُ كباراً تعبت في مرادِها الأجسامُ	If the souls were great The bodies would be tired in achieving their will

We structure the paper as follows. Section 2 describes related state-of-the-art approaches. Section 3 introduces the architectures of models used in this study and Section 4 presents the proposed models. Section 5 illustrates the evaluation methods and we detail experimental results and provide a useful discussion on poem-generation capabilities in Section 6. Finally, Section 7 shows conclusions and future work.

2. RELATED WORK

Poetry composition is undoubtedly the most difficult of the text-generation sub-tasks, because poetry must be written elegantly and ideally according to a particular context. Over the last few decades, automated poem generation has been a popular research area. However, most of the work on poem generation is accomplished in Chinese and English. This section covers recent approaches for generating poems in Chinese and English, as well as a few works in languages similar to Arabic, such as Persian, and finally cites a few existing works in the Arabic language.

Yi et al. developed a model to generate coherent Chinese poetry with a flexible clear description of the poem's topic [20]. They built an encoder-decoder framework based on a bi-directional recurrent neural network (Bi-RNN) with an attention mechanism. They also tested the model on three styles of poetry. The quatrain style is known to be the most difficult form, which is a pair of matching couplets, each line consisting of five or seven syllables. In this work, authors used a corpus of 71,000 quatrains to train a model, while 1,000 quatrains were used for the test. The Working Memory Model was utilized to write a poetry line while keeping the previous line in mind. The previous line is saved in local memory and will be concatenated with the following line. The results of this model were compared to those of state-of-the-art models by poetry experts. The model obtained higher scores on Coherence (3.57) and Relevance (3.77), indicating that it produced poems of higher quality and cohesiveness.

Liu et al. [21] presented a rhetorically controlled encoder-decoder to develop modern Chinese poetry. This model employs a continuous latent variable as a rhetoric controller in an encoder to record distinct rhetorical patterns. Then, it integrates rhetoric-based mixtures while generating modern Chinese poetry. In this model, word embedding, rhetoric label embedding and hidden state sizes are set at 128, 128 and 128, respectively. The latent variable has 256 dimensions and a single-layer decoder is employed. The human evaluation results show that this method achieves the best results in terms of the Meaningfulness (3.2) and Rhetorical Aesthetics (3.5) metrics. Also, experiments reveal that this model can generate Chinese poetry with convincing metaphors and personification.

Deng et al. [22] presented a novel iterative polishing framework for highly competent Chinese poem generation in this research. An encoder-decoder structure is used to generate a poem draft in the first stage. Following that, the authors suggested Quality-Aware Masked Language Model (QA-MLM) to be used to polish the document in terms of linguistics and literalness. QA-MLM can use a multi-task learning system to identify whether polishing is required based on the poetry draft. In this approach, they used corpus for training and testing the models consisting of approximately 130,525 poems with a total of 905,790 lines. BERT was selected as the encoder with 12 layers and initialized with the parameters pre-trained by [23] and the 2-layer transformer decoder was selected for poem generation.

Human and automatic evaluations were performed and the findings show that this approach effectively improves the performance of encoder-decoder structures.

In [24], the authors proposed a generate-retrieve-then-refine paradigm for poetry generation based on the creative pro- cess of humans. It allows a generative model to benefit from generated draft and retrieval outcomes. To increase coherence, the authors employ bidirectional sentence-level context from previously generated lines and draft lines. In addition, they present the "refining vector," which is distilled by the fantastic word recognition process to develop newer and more unique expressions. The authors collected a 263,669 modern Chinese poetry dataset with 9,209,186 sentences. In this approach, they employ an encoder-decoder model with word2vec embeddings. The word embedding size is 128 and the recurrent hidden layers of the encoder and decoder have 128 hidden units and 4 layers. They used the Adam algorithm [25] to train the model, with a batch size of 512 and a learning rate of 3e-4. The results of experiments on Coherence (3.98), Impressiveness (3.86) and Poeticness (3.40) reveal that this model surpasses baselines in terms of consistency and novelty.

Lau et al. [26] built a model based on a variant of an LSTM encoder-decoder with attention [27] for composing English quatrains (Shakespeare-like sonnets). The authors developed a joint design of three neural networks that capture language, rhyme and meter to construct quatrains. They used 3,355 sonnets to train and test these models. They also assessed the quality of generating quatrains using crowdsourcing and expert assessment. According to crowdsourcing and expert evaluations, the poems generated matched the sonnet structure, but lacked readability and coherence.

Santillan and Azcarraga [28] described a method for generating English poems from a given input poem seed utilizing transformers [29] and doc2vec embeddings [30]. This technique uses a pre-trained model, which is then fine-tuned using a poem dataset and assesses generated poem output using a cosine similarity score from a doc2vec model. The corpora used in this approach are divided into two datasets, which are 50-200 characters long (short poem set) and those longer than 200 characters (long poem set), each comprising 58,955 poems, respectively. Moreover, all transformer training employed the same hyper-parameters, such as a learning rate of $2e^{-5}$ and a batch size of 2. This approach ensures strong cohesiveness between the output and the given input text according to the results.

Van de Cruys introduced an automatic poetry-generation system trained just on standard, non-poetic text [10]. The system employs a recurrent neural encoder-decoder architecture that incorporates poetic and topical constraints by changing the neural network's output probability distribution to create potential verses. Then, the best verse is chosen for inclusion in the poem using a global optimization framework. The authors used an encoder and decoder model consisting of two GRU layers with a hidden state of size 2048. The model parameters are optimized using stochastic gradient descent with an initial learning rate of 0.2 and a batch size of 64. They also trained the system in English and French with a training corpus of 500 million words for each language, then performed human evaluations in both languages. The results show that the system can generate plausible poetry with high Fluency (3.64) and Coherence (3.41) scores as well as with Meaningfulness (3.27) and Poeticness (3.86).

Bena and Kalita [31] proposed a novel method for generating English poems. They fine-tuned a pre-trained language model GPT-2 [32] to generate poems that express emotion in readers and dream poetry. They classified emotional poems and dreamed text to impact automatic natural-language production in creating poetry. To accomplish this job, they created a meaning for emotion-eliciting material using a word-level emotion lexicon, which was subsequently utilized for training different GPT-2 models. The authors pre-trained the OpenAI-released GPT-2 model on a dataset of first-person dream narratives to teach the network the language of poems. The model was then fine-tuned using a dataset of 20,000 dreams. They rate the proposed model on three qualities: Quality1 (The poem is generally a first-person expression), Quality 2 (The primary substance of the poetry is a dream or vision) and Quality 3 (The poem tells or predicts an experience or event). With scores no lower than 3.2 on the Likert scale for all three qualities, the human evaluation demonstrates that the fine-tuned GPT-2 performed well in generating dream poems [31].

In [33], the authors present an LSTM-based model for generating Persian poems. They trained the model on a dataset of Ghazaliat-e-Hafez and Ghazaliat-e-Saadi[1]. One challenge in using machine learning to generate Persian poetry is the complex grammar of the language. Persian language grammar includes

[1] The Mohammad Qazvini/Gha¯sem Ghani 1941 edition.

various inflections and declensions that can alter word form and meaning. To tackle this challenge, the authors pre-processed and cleaned the dataset before feeding it into the model. Then, they trained the model to predict the following word in a sequence based on the context of the preceding words. The authors conclude that training the model for a larger number of epochs leads to improved poem quality. However, no significant evaluation was presented.

Talafha and Rekabdar [34]-[35] were among the first to apply deep learning to generate Arabic poems. In [34], they proposed a two-phased approach to generate Arabic poems. In the first phase, they generate the first line of the poem by utilizing Bi-GRU (Bi-directional Gated Recurrent Unit) model. In the second phase, they use a modified Bi-GRU encoder-decoder model with hierarchical neural attention to produce the following lines of the poem. In a more recent work [35], Talafha and Rekabdar propose a poetry-generation model with enhanced phonetic and semantic embeddings. In this work, they propose a three-phased approach. First, they use a word-embedding model that represents verses' rhyme and rhythm besides the context. These embeddings offer information on each word's phonetics and its vectorized word representation. Second, they extract N keywords representing the sub-themes of the N verses to generate, where each keyword is to generate one new verse. In the third phase, they use a Bi-GRU encoder-decoder model with word-level and verse-level attention to generate the first verse. Then, they use a hierarchical sequence-to-sequence model to produce the next verses. This last work approach assumes that the number of the extracted keywords is equal to the number of verses to generate. If the input is too short, it will be impossible to generate enough verses. The authors in both works evaluated the generated poetry using BLEU scores and human review. According to human evaluation, their models produced fair-quality poetry.

Recently, Hakami et al. [36] studied using the GPT-2 model to generate Arabic poems. The authors fine-tuned the GPT-2 model, which had already been pre-trained on English corpora. The fine-tuning dataset included 34,466 Arabic verses acquired manually from aldiwan website[2]. In terms of the BLEU-1 score (0.56) and human evaluation (0.5 in Meaning and Coherence), the generated GPT-2 model performed poorly.

Despite the poor results obtained in [36], the effectiveness of the GPT-2-based model in the case of English language poem generation motivated us in [19] to build a GPT-2 model for Arabic poem generation. In the following, we compare the fine-tuned GPT-2-based model to the GPT-3 model and fine-tuned GPT-3 model in Arabic poem generation.

3. GENERATIVE PRE-TRAINED TRANSFORMER

Word-embedding techniques, such as word2vec [37], are able to capture the semantic meaning of words ignoring how the semantics varies across linguistic contexts. However, language models provide a contextualized embedding that improves the representational power of word embeddings. In language modeling, as a first step, a neural network is trained on a large non-annotated corpus of text. During the training, the neural network learns how to recover masked (missing) words or produce the next words. In the second step, we keep the first layers of the neural network and train again the weights of the lasting layers on a smaller specific dataset. These two phases are called the pre-training and fine-tuning phases.

In masked language modeling, like BERT [23], the neural network fills a sub-set of masked words based on all others. This technique is not suitable for text-generation tasks where the network produces the next words based on the previous sequence of words. OpenAI's Generative Pre-trained Transformer (GPT) proposes to pre-train on the standard task of language modeling: predicting the next word in the sequence. GPT models are efficient in text-generation tasks, such as summarization or producing text based on a prompt. In this section, we introduce the three variants of GPT models.

3.1 GPT-1

Radford et al. proposed in [38] the first variant of Generative Pre-trained Transformer (GPT-1) that achieved state- of-the-art results in 9 out of 12 NLP tasks. They employed a semi-supervised learning approach using unsupervised pre-training and supervised fine-tuning.

[2] https://www.aldiwan.net

First, a language-modeling objective is used to maximize the following likelihood given an unsupervised corpus of tokens U = {u_1,..., u_n} to learn the parameters of a neural network:

$$L_1(U) = \sum_i logP(u_i \mid u_{i-k}, \ldots, u_{i-1}; \Theta) \quad (1)$$

where k is the context window size and the conditional probability P is represented by a neural network with parameters Θ. Stochastic gradient descent [39] is used to train these parameters.

Second, a supervised objective adapts the learned parameters to a particular target task. This objective aims to maximize the likelihood of a given labeled dataset D, where each instance comprises a sequence of input tokens, $x^1, \ldots, x^m$, along with a label y:

$$L_2(D) = \sum_{(x,y)} logP(y \mid x^1, \ldots, x^m) \quad (2)$$

Radford et al. also discovered that adding language modeling as an auxiliary objective to fine-tuning improves learning by enhancing the generalization of the supervised model and accelerates convergence. Therefore, they specifically optimize the following objective with a weight λ (set to 0.5):

$$L_3(D) = L_2(D) + \lambda * L_1(D) \quad (3)$$

The GPT-1 model is recognized as "task agnostic", since it is not limited to a single NLP task, but provides a generalizable architecture that can be adapted to multiple NLP tasks with few adjustments. In terms of architecture, GPT-1 employs a transformer architecture based on a 12-layer decoder (without a decoder) and a self-attention mechanism (12 attention heads). GPT-1 is trained on the BooksCorpus dataset [40] that holds over 7000 unpublished books.

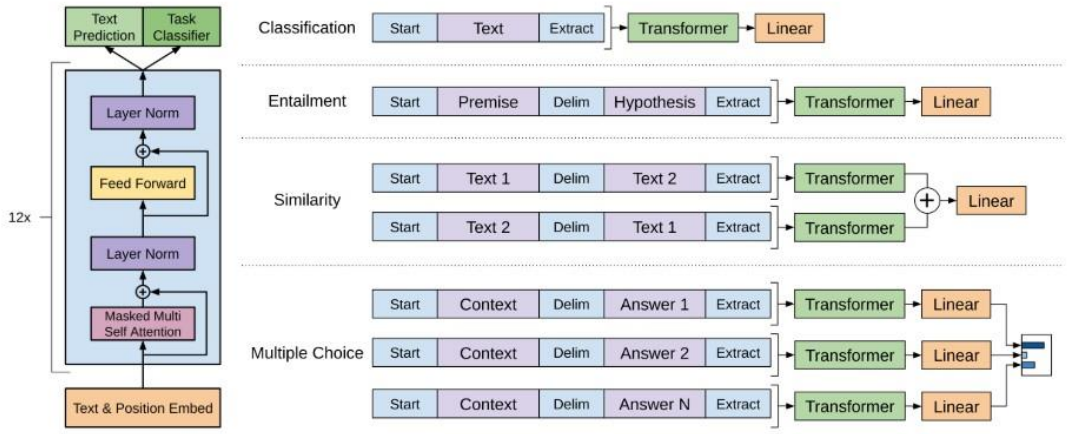

Figure 1. (left) GPT-1 architecture and (right) input transformations for supervised fine-tuning on various tasks [38].

As illustrated in Figure 1, GPT-1 architecture is similar to the decoder-only transformer in [29]. In the network, input tokens $U = (u_{-k}, \ldots, u_{-1})$ are processed through W_e, a token embedding matrix. The activities are then routed through a stack of decoder blocks composed of a multi-headed self-attention layer, a position-wise feedforward layer and a normalization layer:

$$h_0 = UW_e + W_p$$
$$h_l = decoderblock(h_{l-1}) \; \forall i \in [1, n]$$
$$P(u) = softmax(h_n W_e^T)$$

where n is the number of layers and W_p is the position embedding matrix.

Table 2 shows experimental details of the unsupervised pre-training and supervised fine-tuning phases.

Table 2. Hyper-parameters used in GPT-1.

Hyper-parameter	Unsupervised pre-training	Supervised fine-tuning
Max. sequence length	512	512
Batch size	64	32
Learning rate	$2e^{-4}$	$6.25e^{-5}$
# Epochs	100	3

3.2 GPT-2

GPT-2 (Generative Pre-trained Transformer 2) is an unsupervised transformer-based generative language model created by OpenAI [32]. A language model is a machine-learning model that predicts the next word in a given sentence using probability distributions. Using unsupervised methods, language models build many characteristics representing spelling and grammar norms. Unsupervised learning approaches look for patterns in a dataset rather than attempting to find a relationship between the data. GPT-2 was trained using a large corpus (WebText) containing 40 GB of text [32]. This model uses BPE (Byte-Pair Encoding) for encoding text as a sequence of tokens [41]. BPE encoding is a type of sub-word encoding that exists between the character and word levels. Typically, the most common pairs of consecutive bytes of data are encoded as single tokens. However, rare pairs will be encoded as sequences of tokens. This way of encoding catches the recurrent sub-words with specific meanings, like the superlative suffix 'est' in biggest, oldest, …etc. In the Arabic language, suffixes are more common. Arabic suffixes are used as attachable pronouns like تا, which means dual female (they).

With a few structural changes, the GPT-2 architecture closely resembles the GPT-1 model. GPT-2 adds additional layer normalization after the final self-attention block, moves layer normalization to the input of each sub-block and raises context size from 512 to 1024 tokens. Table 3 summarizes the hyper-parameters used to build the various GPT-2 models.

The GPT-2 architecture has demonstrated its ability to represent the English language and has achieved state-of-the- art tasks, such as machine translation, summarization and question-answering. This model is available in four different sizes: small (117 million parameters), medium (345 million parameters), large (762 million parameters) and extra-large (1.5 billion parameters).

Achievements of the GPT-2 model illustrate that training on larger datasets with a larger number of parameters increased the language model's capacity to understand tasks and outperform the state-of-the-art on many tasks in zero-shot. Furthermore, according to the research, as the model's capacity expanded, so did its performance in a log-linear pattern [32].

Table 3. Hyper-parameters used for the four GPT-2 models [32].

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

3.3 GPT-3

In 2020, OpenAI released Generative Pre-trained Transformer 3 (GPT-3) [18]. GPT3 is transformer-based and has the same architecture as GPT-2. More specifically, GPT-3 uses the same updated initialization and pre-normalization used by GPT-2. In addition, GPT-3 architecture includes 96 layers with 96 attention heads in each layer. The context window size was increased from 1024 tokens in GPT-2 to 2048 tokens in GPT-3. GPT-3 was trained on a large and diverse dataset of 499 billion tokens, including Common Crawl, web texts, books and Wikipedia (45TB of text data). With 175 billion parameters, GPT-3 was 10 times larger than any previous language model and had 100 times more parameters than GPT-2. GPT-3 is proposed to be used for all NLP tasks without the need for gradient updates or fine-tuning. GPT-3 performs well on various NLP tasks, including translation, question-answering and cloze tasks [18]. GPT-3 is available in eight versions with a different number of trainable parameters, as listed in Table 4.

GPT-3 shows strong performance on many NLP tasks and benchmarks in the zero-shot, one-shot and few-shot settings, in some cases nearly matching the performance of state-of-the-art fine-tuned systems.

GPT-3 was not made available; instead, access was to be allowed *via* an API, giving the model's developers more control over its use. At the time of writing, the API was under beta testing. However, the API is typically used to start the model by providing a prompt and some introductory text.

Table 4. Sizes, architectures and learning hyper-parameters of the GPT-3 models [42].

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3 B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7 B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7 B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0 B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0 B	96	12288	96	128	3.2M	0.6×10^{-4}

4. PROPOSED MODELS FOR POEM GENERATION

In this section, we describe the pre-training data, the training setup and the fine-tuning setup that we used to build a GPT-2 model for poem generation. We also report on the setup that we used to fine-tune two variants of the GPT-3 model.

4.1 GPT-2 Based Model

In our previous work [19], we proposed a fine-tuned GPT-2 model capable of generating Arabic poetry. However, GPT-2 is only trained in English, while GTP-3 is trained in several languages, including Arabic. To build a GPT-2 model, we went through the two phases of the training process: pre-training and fine-tuning. In the pre-training phase, we used two publicly available corpora Khaleej-2004 [43] and Watan-2004 [44]. Khaleej-2004 is an MSA (Modern Standard Arabic) corpus collected from thousands of articles downloaded from Akhbar Al Khaleej, an online newspaper. The corpus contains 5,690 documents, totaling more than 2 million words. Watan-2004 is another MSA corpus composed of nearly 20,000 documents containing more than 9 million words. To fine-tune our model, we used the Arabic poetry dataset[3] that was scrapped completely from aldiwan website[4]. The Arabic poetry dataset has 55K poems for over 540 poets from 9 different eras. Table 5 summarizes the number of words and unique words for each dataset used in building our fine-tuned GPT-2 model.

After collecting the data, we had to pre-train our model on Arabic corpora. We used the small versions of the pre- trained GPT-2 Tokenizer and Model from the Transformers Library (Hugging Face[5]). This Library provided us with the tokenizer structure needed as well as with pre-trained model weights. Rather than starting with a random network, we trained our GPT-2 model in Arabic with weights already trained in English. Next, we trained a BPE tokenizer on the Arabic corpus using the Tokenizers' Library (Hugging Face), where we obtained a vocabulary of 50K tokens. Then, we used Google Colab to pre-train our GPT-2 model on the Khaleej-2004 and Watan-2004 corpora. The pre-training was executed on an NVIDIA Tesla T4 (16 GB) GPU for 25 epochs. The pre-training lasted about 32 hours.

For the Arabic poem-generation task, we fine-trained our pre-trained model on poems from the Arabic poetry dataset. We used the same GPU (used in the pre-training stage) for 6 epochs to fine-tune our model. The fine- tuning lasted 12 hours. Table 6 shows the hyper-parameter values used in the pre-training and the fine-tuning steps.

Table 5. Summary of the used datasets [19].

Dataset	#Words	#Unique Words
Khaleej-2004	2.482K	122K
Watan-2004	9.813K	291K
Total pre-training	12.229K	413K
Arabic poetry	6.933K	2.060K

Table 6. Hyper-parameters used to build a GPT-2 Arabic poem generator [19].

Hyper-parameter	Pre-training	Fine-tuning
Max. sequence length	1024	1024
Batch size	24	24
Learning rate	$3e^{-5}$	$3e^{-5}$
# Epochs	25	6

[3] https://www.kaggle.com/ahmedabelal/arabic-poetry
[4] https://www.aldiwan.net
[5] https://huggingface.co/

4.2 GPT-3 Based Models

Since GPT-3 is pre-trained on a corpus containing Arabic language and we are not able to pre-train such a big model, we decided to fine-tune the GPT-3 models without the pre-training phase. Through GPT-3 API, OpenAI offers the possibility to build new models by fine-tuning GPT-3 models. We used the provided API to fine-tune GPT-3 on Arabic poems. OpenAI has publicly released four versions of GPT-3: Ada, Babbage, Curie and Davinci. The fastest model is Ada, while the most capable model is Davinci. For reasons of computation cost, we fine-tuned only two versions of GPT-3: Ada and Davinci models. We fine-tuned the Ada and Davinci models during 4 epochs on 5,223 poems and 114 poems from an Arabic poetry dataset, respectively. More precisely, we fine-tuned the Ada model on 10% of the data used to fine-tune GPT-2. For the Davinci model, we used 0.2% of the Arabic poetry dataset. This fine-tuning is considered as a few-shot fine-tuning, as claimed by GPT-3 authors [18]. Few-shot fine-tuning is a method of adapting a pre-trained model to a new task using a small amount of labeled data [18]. It involves updating the weights of the pre-trained model on the new task, poem-generation task in our case. This technique is often used when it is difficult or expensive to fine-tune the model on a big amount of data. In Table 7, we detail the hyper-parameter values used in this fine-tuning.

Table 7. Proposed models' fine-tuning configurations.

Models	Max. seq. len.	Batch size	Learning rate	# Epochs	# Poems	Model size
GPT-2	1024	24	$3e^{-5}$	6	55,000	117M
GPT-3 Ada	2048	64	0.1	4	5,223	2.7B
GPT-3 Davinci	4000	64	0.1	4	114	175B

5. EVALUATION

We evaluate fine-tuned GPT-2, GPT-3 Davinci, fine-tuned GPT-3 Ada and fine-tuned GPT-3 Davinci on the task of poem generation based on a set of two input verses. Each model is asked to generate the N following verses of the two verses given as input. To build the evaluation inputs, we randomly picked five Arabic poems not included in the fine-tuning dataset. From each poem, we extract two verses to be used as input. Table 8 displays the five Arabic verses used as evaluation inputs.

Table 8. Arabic verses used as evaluation inputs.

Input verses' samples	**Input verses samples in English**
أنا الذي نظر الأعمى إلى أدبي وأسمعت كلماتي من به صمم أنام ملء جفوني عن شواردها ويسهرُ الخلقُ جراها ويختصم	I am the one whose verse is seen (even) by the blind and whose words are heard (even) by the deaf I enjoy my sweet repose, not concerning myself with poetry, whereas others burn the midnight oil, in endless literary disputes
أراك عصي الدمع شيمتك الصبر أما للهوى نهي عليك ولا أمر؟ بلى أنا مشتاق وعندي لوعة ولكن مثلي لا يذاع له سر	I see you holding back the tears, your habit is patience Does not love has on you a prohibition or an order? Yes, I miss with a burning desire, but someone like me doesn't spread secrets
لا يكتم السر إلا من له شرف والسر عند كرام الناس مكتوم السر عندي في بيت له غلق ضلت مفاتيحه والباب مردوم	No one keeps a secret except those with honor and the secret by generous people is kept The secret to me is in a house that has a lock its keys are lost and the door is buried
قالت غلبتك يا هذا فقلت لها لم تغلبيني ولكن زدتني كرماً بعض المعارك في خسرانها شرف من عاد منتصراً منها أو انهزما	She said I have beaten you, so I told her you didn't beat me, but you gave me more generousity To loose some battles is an honor for both who returned as winners and those who were losers
في مدخل الحمراء كان لقاؤنا ما أطيب اللقيا بلا ميعاد عينان سوداوان في حجريها تتوالد الأبعاد من أبعاد	At the entrance of Alhambra we have met how delightful it is to meet without a rendezvous Two dark eyes: in their depths distances give birth to distances

For each input, each model is asked to generate the N following verses, with N taking the values of 2, 4 and 6. The quality of the generated verses will be evaluated regarding the meaning and the rhyme of the verses fed as inputs.

To generate Arabic poems using GPT-3 Davinci, fine-tuned GPT-3 Ada and fine-tuned GPT-3 Davinci, we use the OpenAI Beta Playground[6]. The Playground is a web-based application that allows users to quickly test the prompts and get familiar with how the API works. We use BLEU scores and human evaluation to evaluate the verses generated by each model. In the following, we introduce both evaluation methods.

5.1 BLEU Scores

BLEU (Bilingual Evaluation Understudy) scores [45] are commonly used in machine translation (MT) to compare reference and candidate texts. The BLEU-1, BLEU-2, BLEU-3 and BLEU-4 represent the number of unigrams, bigrams, trigrams and 4-grams, respectively. Each gram reflects the number of words selected from both texts and compared to one another. A BLEU score has a value ranging from 0 to 1. In earlier studies, including [46][47][48], BLEU scores were also used to evaluate poem generation. A better generated poem usually achieves a higher BLEU score, as it shares more n-grams with the referenced poem. Therefore, we use BLEU scores to evaluate the verses generated by the fine-tuned GPT-2, GPT-3 Davinci, fine-tuned GPT-3 Ada and fine-tuned GPT-3 Davinci models. For the poems generated by the four models in two, four and six verses, we calculated the BLEU-1, BLEU-2, BLEU-3 and BLEU-4 scores. We used test data from the Arabic poetry dataset containing 7,136 verses as a reference to compute these scores.

5.2 Human Evaluation

To evaluate the quality of generated poems, we performed a human evaluation following previous works [19], [46], [34]. We asked three human experts to judge 240 verses generated by fine-tuned GPT-2, GPT-3 Davinci, fine-tuned GPT-3 Ada and fine-tuned GPT-3 Davinci models. This evaluation is based on four criteria: Fluency, Coherence, Meaning and Poeticness. Fluency checks whether the generated poem is grammatically satisfactory. Coherence inspects whether the generated poem is thematically coherent. Meaning measures how meaningful the content of a generated poem is. Poeticness estimates the features of poetry in the generated poem. These four criteria are rated by the human evaluators on a scale ranging from 1 (worst) to 5 (best).

Table 9. Inter-annotator agreement.

Variable	Krippendorff's α
Fluency	0.90
Coherence	0.81
Meaning	0.79
Poeticness	0.85

We employ Krippendorff's α [49] Inter-Annotator Agreement (IAA) to estimate annotation reliability. Krippendorff's α is predicated on the notion that predicted agreement is computed by looking at the total distribution of ratings, regardless of who issued them. Table 9 shows Krippendorff's α calculated for each dimension. Table 9 indicates that the reliabilities ranged between 0.90 and 0.79, showing that the annotators' judgments were consistent.

6. EXPERIMENTS

We first evaluated our fine-tuned GPT-2 model against the state-of-the-art models for Arabic poem generation: Vanilla RNN, LSTM, GRU, RNN EncoderDecoder (with and without attention) and Bi-GRU with hierarchical neural attention. Then, we evaluated our model against GPT-3 models: GPT-3 Davinci, fine-tuned GPT-3 Davinci and fine-tuned GPT-3 Ada. The results are detailed and discussed in the following sub-sections.

6.1 GPT-2 against State-of-the-art Models

6.1.1 BLEU Scores

We evaluate our fine-tuned GPT-2 results of BLEU scores in generating two verses against the work

[6] https://beta.openai.com/playground

of Talafha and Rekabdar in [34]-[35]. The fine-tuned GPT-2 model is better than other state-of-the-art models for BLEU-1, BLEU-2, BLEU-3 and BLEU-4, as shown in Table 10.

We observe that all BLEU scores are low. Since there are multiple ways to compose a poem given two verses (see Appendix A), we believe it is rational to obtain low BLEU scores. This observation can be confirmed by the scores decreasing from BLEU-1 to BLEU-4. The probability of obtaining shared n-grams between the generated poem and the reference poem (BLEU-n) decreases when the number of words (n) composing the n-grams increases.

Table 10. BLEU comparison with Talafha and Rekabdar's [34]-[35] work.

Models	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**
Vanilla RNN	0.0211	0.0199	0	0
LSTM	0.1522	0.1124	0.0081	0.0013
GRU	0.1512	0.1139	0.0084	0.0021
RNN EncoderDecoder (without attention)	0.2513	0.1539	0.0740	0.0510
RNN EncoderDecoder (with attention)	0.3010	0.2110	0.0911	0.0801
Bi-GRU with hierarchical neural attention [34]	0.4122	0.3144	0.204	0.1092
Phonetic CNN_sub-word embedding model [35]	0.5301	0.4010	0.3001	0.1500
Fine-tuned GPT-2	**0.8726**	**0.5572**	**0.3418**	**0.2001**

6.1.2 Human Evaluation

We also compared the fine-tuned GPT-2 results of the human evaluation in generating two verses with the work of Talafha and Rekabdar [34]-[35]. Table 11 reports the results of this comparison. Results in Table 11 show that in terms of Poeticness and Fluency, the fine-tuned GPT-2 model outperforms the other models. In terms of Coherence and Meaning, the fine-tuned GPT-2 model obtains acceptable results compared to the other models. This can be explained by the fact that Talafha and Rekabdar's work is focused on specific topics: love and religion. Contrasting our work, the covered topics are multiple and include most of the Arabic poetry topics. We also recall that in Talafha and Rekabdar's work, each verse is generated from a keyword. In comparison, our model is only constrained by the verses in inputs.

Table 11. Fine-tuned GPT-2 compared to the work of Talafha and Rekabdar [34]-[35].

Models	**Fluency**	**Coherence**	**Meaning**	**Poeticness**
Vanilla RNN	0.1	0.8	0.7	0
LSTM	0.3	0.9	0.8	0.1
GRU	0.3	1.0	1.0	0.2
RNN Encoder- Decoder (without attention)	2.0	1.5	2.4	0.3
RNN Encoder- Decoder (with attention)	2.3	2.5	2.7	0.4
Bi-GRU with hierarchical neural attention [34]	2.1	3.2	3.5	0.9
Phonetic CNN_sub-word embedding model [35]	2.7	**3.3**	**3.6**	2.5
Fine-tuned GPT-2	**3.2**	2.8	2.4	**4.0**

6.2 GPT-2 against GPT-3 Models

6.2.1 BLEU Scores

Table 12 shows that the fine-tuned GPT-2 model outperformed the GPT-3 Davinci, the fine-tuned GPT-3 Ada and the fine-tuned GPT-3 Davinci models for BLEU scores in generating two, four and six Arabic poem verses. We think that fine-tuning on poem dataset forces the model to generate text closer to the poetic context. The fine-tuned GPT-3 Davinci model achieved better BLEU scores than the raw GPT-3 Davinci model. We also observe that GPT-3 Davinci obtains better BLEU scores than fine-tuned GPT-3 Ada when generating 2 verses, while fine-tuned GPT-3 Ada performs better in 4- and 6-verse generation. These observations show that a fine-tuned smaller model has a better generation performance than a non-fine-tuned larger model.

Table 12. The BLEU scores of different GPT models.

Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4
2-verse generation				
GPT-3 Davinci	0.6846	0.4066	0.2404	0.1375
Fine-tuned GPT-2	**0.8726**	**0.5572**	**0.3418**	**0.2001**
Fine-tuned GPT-3 Ada	0.6668	0.3903	0.2298	0.1310
Fine-tuned GPT-3 Davinci	0.7075	0.4200	0.2515	0.1449
4-verse generation				
GPT-3 Davinci	0.6548	0.3853	0.2272	0.1297
Fine-tuned GPT-2	**0.8456**	**0.5151**	**0.3064**	**0.1764**
Fine-tuned GPT-3 Ada	0.7103	0.4248	0.2532	0.1455
Fine-tuned GPT-3 Davinci	0.7288	0.4374	0.2611	0.1502
6-verse generation				
GPT-3 Davinci	0.5648	0.3278	0.1927	0.1095
Fine-tuned GPT-2	**0.8211**	**0.5123**	**0.3095**	**0.1796**
Fine-tuned GPT-3 Ada	0.7060	0.4188	0.2485	0.1424
Fine-tuned GPT-3 Davinvi	0.7200	0.4336	0.2586	0.1486

6.2.2 Human Evaluation

Automatic evaluation metrics like BLEU scores are fast and cost-effective measurements of the quality of poem-generation models. However, as poetry is an art form subject of human appreciation, human judgment is the benchmark to assess the quality of the generated poems. The results of the human evaluation of GPT-2 and other GPT-3 models are listed in Table 13.

Table 13. The results of human evaluation of different GPT models.

Models	Fluency	Coherence	Meaning	Poeticness
2-verse generation				
GPT-3 Davinci	2.7	2.1	2.3	1.4
Fine-tuned GPT-2	3.2	**2.8**	2.4	**4.0**
Fine-tuned GPT-3 Ada	3.1	2.4	2.5	2.3
Fine-tuned GPT-3 Davinci	**4.0**	2.5	**3.0**	2.5
4-verse generation				
GPT-3 Davinci	2.9	**2.4**	2.4	1.7
Fine-tuned GPT-2	3.0	2.1	2.3	**3.5**
Fine-tuned GPT-3 Ada	3.0	2.2	2.2	2.4
Fine-tuned GPT-3 Davinci	**3.5**	2.2	**2.5**	2.3
6-verse generation				
GPT-3 Davinci	3.5	2.1	1.9	1.9
Fine-tuned GPT-2	3.4	2.0	1.9	**3.7**
Fine-tuned GPT-3 Ada	3.4	2.3	2.5	2.7
Fine-tuned GPT-3 Davinci	**3.6**	**2.4**	**2.6**	2.1

Results in Table 13 show that the fine-tuned GPT-2 scored higher on Poeticness than GPT-3 and fine-tuned GPT-3 in generating two, four and six Arabic poem verses. Regarding Fluency, Coherence and Meaning, the four models performed similarly. We also notice from Table 13 that the two fine-tuned GPT-3 models get a better result in Poeticness than GPT-3 without fine-tuning when generating four and six verses. The GPT-3 Davinci (without fine-tuning) has the lowest score in Poeticness compared to the other models. On the other hand, the fine- tuned GPT-2 model performance degrades in Coherence and Meaning, but maintains good performance in terms of Fluency and Poeticness when generating more verses. Albeit, in the fine-tuned GPT-3 Davinci, we find that Poeticness decreases when generating more verses.

Human-evaluation results show that fine-tuning GPT-3 Davinci model improved its Poeticness

capabilities, since it achieved better Poeticness scores than the raw GPT-3 Davinci model. We also observe that fine-tuning improved GPT-3 Davinci Meaning capabilities, maintaining similar Fluency and Meaning scores. In addition, the smallest model, fine-tuned GPT-2, obtained the best Poeticness scores in all runs. Human-evaluation observations agree with the findings of the automatic evaluation (BLEU scores). These observations show that the size of the model is less significant than fine-tuning in the case of automatic poem generation.

Table 14 presents a sample of outputs generated by the four models given the same input. The two verses used as input belong to the Kamil meter (بحر الكامل). The usual form of this meter is as follows:
| uu – u – | uu – u – | uu – u – |
where "–" represents a long syllable, "u" a short syllable and "uu" one long or two short syllables.

Table 14. Examples of verses generated by the different GPT models.

Input		
	Arabic verses	English translation
	في مدخل الحمراء كان لقاؤنا ما أطيب اللقيا بلا ميعاد عينان سوداوان في حجريهما تتوالد الأبعاد من أبعاد	At the entrance of Alhambra we have met how delightful it is to meet without a rendezvous Two dark eyes: in their depths distances give birth to distances
Outputs		
Model	Arabic verses	English translation
Fine-tuned GPT-2	قد كان في البحر البهيم مسلها بالركب حين تكون للأوراد يا ليل يا ليل انتبهوا على ما كان في بغداد من أعراس	It was in the rough sea, a place on the way, when you are for the source of water Oh night, oh night, pay attention to what was in Baghdad weddings
GPT-3 Davinci	ليس مع فكرة الصداقة شيء غريب ما اطيب اللقيا بلا ميعاد دايخة الحرب بين الأحزان والأغراب الحزن في الحرب إذا ما فتح الصدر	There is nothing strange about the idea of friendship how delightful it is to meet without a rendezvous The bitterness of the war between the sorrows and the strangers sadness in war is when the chest is opened
Fine-tuned GPT-3 Ada	في حمراء حفاظن كما أطيبهم وفي شرب بدران أكثار حراد في حمراء ريحانات تتوالد في حسنات حميانات تتوالد	In Hamra, we have preserves because they are the best and in Badran drinking more Harad In Hamra, Rihanat are born In good deeds, fevers are born
Fine-tuned GPT-3 Davinci	أحبابي وأصدقائي وإخواني أشوفهم وأنا أسمع منهم أصوات أحبابي المنادية وأنا أعيش في دار الحمراء	My lovers, my friends and my brothers I see them and I hear them The calling voices of my beloved and I live in the Hamra house

The example generated by the fine-tuned GPT-2 model (see Table 14) perfectly maintains the same meter of the input, while the other models do not maintain this meter. We notice that the fine-tuned GPT-3 Davinci model generated better rhyme than GPT-3 Davinci model. We also observe that except GPT-3 Davinci model, all the models generated love verses. This capability to generate the same thematic verses is enabled during the fine-tuning phase. Another remarkable fact is the reuse of the word "Hamra" ("الحمراء") by the two fine-tuned GPT-3 models. In free text generation, the model generates the next word based on the previous context (input text). In such a case, the model effort focuses on optimizing the meaning of the generated text. However, in the context of poem-generation, the model effort focuses on the rhyme, the thematic and the meaning of the previous context (input verses, in poetry). The fine-tuning phase is responsible for adapting the model to the underlying task. This is confirmed by the observations on the generated verses illustrated in Table 14. We observed that with fewer parameters and well defined fine-tuning, our fine-tuned GPT-2 model outperformed larger models with less fine-tuning in the task of Arabic poem generation.

7. CONCLUSION

In this paper, we tackled an automatic Arabic poem-generation task. We competed for different deep neural network architectures to emphasize the significance of the model size and the fine-tuning phase in the case of Arabic poem generation. We showed in the state-of-the-art review that the Generative

Pre-trained Transformer (GPT) architecture fits best for automatic poem generation. We presented and compared four models for automatic Arabic poem generation: fine-tuned GPT-2, raw GPT-3 Davinci, fine-tuned GPT-3 Ada and fine-tuned GPT-3 Davinci. We proposed to evaluate these four models on two, four and six verses of Arabic poem generation. We used BLEU scores and human evaluation to assess the quality of the poems generated by the four models.

Experiments revealed that the poeticness capability strongly depends on the quality of fine-tuning phase, even for very large models. The fine-tuned GPT-2 model measured higher BLEU scores than all the GPT-3 models. The human evaluation shows that the smaller version of GPT-2 fine-tuned on Arabic poems outperforms the most capable version of GPT-3 Davinci in the quality of poem generation. Human evaluation also shows that Ada, the minor publicly available version of GPT-3 fine-tuned on a small dataset, performs better than GPT-3 Davinci without fine-tuning in terms of Poeticness, which is the most critical criterion in poem generation.

We conclude that the size of the model is less significant than fine-tuning in the case of automatic poem generation. Thereafter, researchers with limited computation resources should not be discouraged by the size of the latest models. A trade-off between the model size and its text generation performance is still possible through fine- tuning, which is less resource-intensive. Nevertheless, it must not be forgotten that other architectures exist. In future work, we plan to explore other deep neural network architectures to improve the quality of the generated Arabic poems. However, many challenges need to be addressed to achieve effectiveness. We also plan to focus on particular topics of Arabic poems to enhance the Coherence and the Meaning of the generated poems.

REFERENCES

[1] S. Subramanian, S. Rajeswar, F. Dutil, C. Pal and A. Courville, "Adversarial Generation of Natural Language," Proc. of the 2nd Workshop on Representation Learning for NLP, pp. 241–251, 2017.

[2] R. Yan, H. Jiang, M. Lapata, S.-D. Lin, X. Lv and X. Li, "i, Poet: Automatic Chinese Poetry Composition through a Generative Summarization Framework under Constrained Optimization," Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence, pp. 2197-2203, 2013.

[3] A. Das and B. Gambäck, "Poetic Machine: Computational Creativity for Automatic Poetry Generation in Bengali," Proc. of the Int. Conf. on Innovative Comp. and Cloud Comp. (ICCC), pp. 230–238, 2014.

[4] H. G. Oliveira and A. Cardoso, "Poetry Generation with PoeTryMe," Proc. of Computational Creativity Research: Towards Creative Machines, Part of the Atlantis Thinking Machines Book Series (ATLANTISTM), vol. 7, pp. 243–266, Springer, 2015.

[5] M. Ghazvininejad, X. Shi, Y. Choi and K. Knight, "Generating Topical Poetry," Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing, pp. 1183–1191, Austin, Texas, 2016.

[6] M. Ghazvininejad, X. Shi, J. Priyadarshi and K. Knight, "Hafez: An Interactive Poetry Generation System," Proc. of ACL 2017, System Demonstrations, pp. 43–48, Vancouver, Canada, 2017.

[7] D. Singh, M. Ackerman and R. Y. Pérez, "A Ballad of the Mexicas: Automated Lyrical Narrative Writing," Proc. of the 8ht Int. Conf. on Computational Creativity (ICCC), [Online], Available: http://ilitia.cua.uam.mx:8080/jspui/handle/123456789/442, 2017.

[8] L. Xu, L. Jiang, C. Qin, Z. Wang and D. Du, "How Images Inspire Poems: Generating Classical Chinese Poetry from Images with Memory Networks," Proc. of the 32nd Conf. on Artificial Intelligence, vol. 32, Article No. 689, pp. 5618–5625, 2018.

[9] A. Zugarini, S. Melacci and M. Maggini, "Neural Poetry: Learning to Generate Poems Using Syllables," Proc. of the Int. Conf. on Artificial Neural Networks, pp. 313–325, DOI: 10.1007/978-3-030-30490-4_26, Springer, 2019.

[10] T. Van de Cruys, "Automatic Poetry Generation from Prosaic Text," Proc. of the 58th Annual Meeting of the Associ. for Computational Linguistics, pp. 2471–2480, DOI: 10.18653/v1/2020.acl-main.223, 2020.

[11] C.-L. Zhou, W. You and X. Ding, "Genetic Algorithm and Its Implementation of Automatic Generation of Chinese Songci," Journal of Software, vol. 21, no. 3, pp. 427–437, 2010.

[12] J. He, M. Zhou and L. Jiang, "Generating Chinese Classical Poems with Statistical Machine Translation Models," Proc. of the 26th AAAI Conference on Artificial Intelligence, vol. 26, no. 1, pp. 1650–1656, DOI: 10.1609/aaai.v26i1.8344, 2012.

[13] Q. Wang, T. Luo and D. Wang, "Can Machine Generate Traditional Chinese poetry? A Feigenbaum Test," Proc. of the Int. Conf. on Brain Inspired Cognitive Systems, Part of the Lecture Notes in Computer Science Book Series (LNAI), vol. 10023, pp. 34–46, Springer, 2016.

[14] J. Zhang, Y. Feng, D. Wang, Y. Wang, A. Abel, S. Zhang and A. Zhang, "Flexible and Creative Chinese Poetry Generation Using Neural Memory," arXiv: 1705.03773, DOI: 10.48550/arXiv.1705.03773, 2017.

[15] Z. Wang, W. He, H. Wu, H. Wu, W. Li, H. Wang and E. Chen, "Chinese Poetry Generation with Planning

Based Neural Network," arXiv:1610.09889, DOI: 10.48550/arXiv.1610.09889, 2016.

[16] X. Yang, X. Lin, S. Suo and M. Li, "Generating Thematic Chinese Poetry Using Conditional Variational Auto-encoders with hybrid decoders," arXiv: 1711.07632, DOI: 10.48550/arXiv.1711.07632, 2017.

[17] Z. N. Abdel-Malek, Towards a New Theory of Arabic Prosody, 5th Edn., ISSN: 0258-3976, Tajdid Online Forum for Facilitating Arabic Studies, 2019.

[18] T. Brown, B. Mann, N. Ryder et al., "Language Models Are Few-shot Learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[19] M. E. G. Beheitt and M. B. H. Hmida, "Automatic Arabic Poem Generation with GPT-2," Proc. of the 14th Int. Conf. on Agents and Artificial Intelligence (ICAART 2022), vol. 2, pp. 366-374, 2022.

[20] X. Yi, M. Sun, R. Li and Z. Yang, "Chinese Poetry Generation with a Working Memory Model," Proc. of the 27th Int. Joint Conference on Artificial Intelligence (IJCAI'18), pp. 4553–4559, 2018.

[21] Z. Liu, Z. Fu, J. Cao, G. de Melo, Y.-C. Tam, C. Niu and J. Zhou, "Rhetorically Controlled Encoder-Decoder for Modern Chinese Poetry Generation," Proc. of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1992–2001, Florence, Italy, 2019.

[22] L. Deng, J. Wang, H. Liang et al., "An Iterative Polishing Framework Based on Quality Aware Masked Language Model for Chinese Poetry Generation," Proc. of the AAAI Conference on Artificial Intelligence, pp. 7643–7650, 2020.

[23] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 4171–4186, 2019.

[24] L. Shen, X. Guo and M. Chen, "Compose Like Humans: Jointly Improving the Coherence and Novelty for Modern Chinese Poetry Generation," Proc. of the 2020 IEEE Int. Joint Conf. on Neural Networks (IJCNN), pp. 1–8, Glasgow, UK, 2020.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv: 1412.6980, 2014.

[26] J. H. Lau, T. Cohn, T. Baldwin, J. Brooke and A. Hammond, "Deep-speare: A Joint Neural Model of Poetic Language, Meter and Rhyme," Proc. of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 1948–1958, 2018.

[27] D. Bahdanau, K. H. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," Proc. of the 3rd Int. Conf. on Learning Representations (ICLR 2015), arXiv: 1409.0473, 2015.

[28] M. C. Santillan and A. P. Azcarraga, "Poem Generation Using Transformers and doc2vec Embeddings," Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN), pp. 1–7, Glasgow, UK, 2022.

[29] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention Is All You Need," arXiv: 1706.03762, DOI: 10.48550/arXiv.1706.03762, 2017.

[30] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," Proc. of the Int. Conf. on Machine Learning (PMLR), pp. 1188–1196, arXiv: 1405.4053, 2014.

[31] B. Bena and J. Kalita, "Introducing Aspects of Creativity in Automatic Poetry Generation," arXiv: 2002.02511, DOI: 10.48550/arXiv.2002.02511, 2020.

[32] 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.

[33] M. H. Moghadam and B. Panahbehagh, "Creating a New Persian Poet Based on Machine Learning," arXiv e-prints, pp. arXiv–1810, DOI: 10.48550/arXiv.1810.06898, 2018.

[34] S. Talafha and B. Rekabdar, "Arabic Poem Generation with Hierarchical Recurrent Attentional Network," Proc. of the IEEE 13th Int. Conf. on Semantic Comp. (ICSC), pp. 316–323, Newport Beach, USA, 2019.

[35] S. Talafha et al., "Poetry Generation Model *via* Deep Learning Incorporating Extended Phonetic and Semantic Embeddings," Proc. of the IEEE 15th Int. Conf. on Semantic Comp., pp. 48–55, USA, 2021.

[36] A. Hakami, R. Alqarni, M. Almutairi and A. Alhothali, "Arabic Poems Generation Using LSTM, Markov-LSTM and Pre-trained GPT-2 Models," Computer Science & Information Technology (CS&IT), vol. 11, no. 15, pp. 139–147, 2021.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," arXiv:1310.4546, DOI: 10.48550/arXiv.1310.4546 2013.

[38] A. Radford, K. Narasimhan, T. Salimans et al., "Improving Language Understanding by Generative Pre-training," [Online], Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018.

[39] H. Robbins and S. Monro, "A Stochastic Approximation Method," The Annals of Mathematical Statistics, vol. 22, no. 3, pp. 400–407, 1951.

[40] Y. Zhu, R. Kiros, R. Zemel et al., "Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books," Proc. of the IEEE Int. Conf. on Computer Vision (ICCV), pp. 19–27, DOI: 10.1109/ICCV.2015.11, 2015.

[41] R. Sennrich, B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," arXiv: 1508.07909, DOI: 10.48550/arXiv.1508.07909, 2015.

[42] T. B. Brown, B. Mann, N. Ryder et al., "Language Models Are Few-shot Learners," arXiv: 2005.14165,

DOI: 10.48550/arXiv.2005.14165, 2020.
[43] M. Abbas and K. Smaili, "Comparison of Topic Identification Methods for Arabic Language," Proc. of Int. Conf. on Recent Advances in Natural Lang. Process. (RANLP), pp. 14–17, Borovets, Bulgaria, 2005.
[44] M. Abbas, K. Smaïli and D. Berkani, "Evaluation of Topic Identification Methods on Arabic Corpora," Journal of Digital Information Management, vol. 9, no. 5, pp. 185–192, 2011.
[45] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311–318, Philadelphia, USA, 2002.
[46] J. Li, Y. Song, H. Zhang, D. Chen, S. Shi, D. Zhao and R. Yan, "Generating Classical Chinese Poems *via* Conditional Variational Autoencoder and Adversarial Training," Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing, pp. 3890–3900, Brussels, Belgium, 2018.
[47] R. Yan, "i, Poet: Automatic Poetry Composition through Recurrent Neural Networks with Iterative Polishing Schema," Proc. of the 25th Int. Joint Conf. on Artificial Intell. (IJCAI-16), pp. 2238–2244, 2016.
[48] X. Zhang and M. Lapata, "Chinese Poetry Generation with Recurrent Neural Networks," Proc. of the 2014 Conf. on Empirical Methods in Natural Lang. Process. (EMNLP), pp. 670–680, Doha, Qatar, 2014.
[49] K. Krippendorff, Content Analysis: An Introduction to Its Methodology, 3rd Edition, Sage Publications, 2013.

APPENDEX. MULTIPLE WAYS TO COMPOSE A POEM

This Appendix accompanies our fine-tuned GPT-2 experiments in Section 6. It illustrates the various ways to compose a poem from a given input. More explicitly, as illustrated in Table 15, we feed the model two verses as input and ask the model to generate the two following verses. The generated verses should keep the rhyme scheme of the verses fed as input. Each output listed in Table 15 represents a different model run to generate the two following verses of the input listed in Table 15.

Table 15. Examples of multiple Arabic verses generated given two verses.

	Arabic verses	**English translation**
Input		
	يا منزل الآيات و الفرقان بيني وبينك حُرمة القرآن اشرح به صدري لمعرفة الهدى واعصم به قلبي من الشيطان	O revealer of verses and furqan between me and you is the sanctity of Quran Open my chest with it to know the guidance and protect my heart from the devil
Outputs		
Output 1	يا ليت شعري قد بكى زمن الأسى والقلب في قلبي هوى الإنسان أيام كانت أمّةٌ لكعبة شعرهم فأراها وهو الرّدى في آن	I wish my poetry had cried on the time of sorrow and the heart in my heart is the love of the human The days when it was a nation for the Kaaba of they poetry and he saw them and he was the one who was at the same time
Output 2	وأملأ جمال القلب في بغداد ما ينجو من الأغصان والرضوان واسلم هديت الشعر من خمر الهوى يا من له ما أنت في الإنسان	And fill the beauty of heart in Baghdad what survives of the branches and the Radwan And thanks I gave the poetry from the wine of love o you who have what you are in human
Output 3	يا سيدي هذا الذي أحيا به قلبي الذي أهواه من إيماني إن كان ظني من سواك فإنني من بعد فقدك في ذرى الأزمان	Sir, this is what I live by my heart that I love from my faith If I think of anyone other than you, then I'm after losing you in the pinnacles of time

ملخص البحث:

تُعدّ كتابة الشِّعر تحدّياً لبني البَشر من حيث حاجتها الى الإبداع والى مستوى عالٍ من القدرة اللغوية، لذا فإنّ التّوليد الأوتوماتيكي للشِّعر مجالاً مهماً للبحث جَذَب اهتمام مجتمع معالجة اللغات الطبيعية. فقد فحص العديد من الباحثين التّوليد الأوتوماتيكي للشِّعر باستخدام طُرُق التّعلُّم العميق، إلا أنّ القليل من تلك البحوث ركّز على الشِّعر العربي.

في هذا البحث، نَعْرض كيفية الاستفادة من نماذج GPT-2 و GPT-3 لتوليد الشِّعر العربي أوتوماتيكياً. وقد تمّ استخدام أربعةٍ من هذه النّماذج ومن تمّ تقييمها آليّاً وبشرياً. وقد أثبتت نتائج التّقييم بِنَوْعَيْهِ فاعلية نموذج GPT-2 ذي الضّبط الدّقيق على غيره من النّماذج المستخدمة، وكانت نتائج نماذج GPT-3 الأقلّ من حيث الشّاعِرية. وفي حُدود عِلْم الباحثين، فإنّ هذه الدّراسة هي الأولى من بين أدبيات البحث التي وظّفت نماذج GPT-3 ذات الضّبط الدّقيق لتوليد الشِّعر العربي.