mathematics

Article

Beyond Event-Centric Narratives: Advancing Arabic Story Generation with Large Language Models and Beam Search

Arwa Alhussain and Aqil M. Azmi *

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; ahussain@ksu.edu.sa
* Correspondence: aqil@ksu.edu.sa

Abstract: In the domain of automated story generation, the intricacies of the Arabic language pose distinct challenges. This study introduces a novel methodology that moves away from conventional event-driven narrative frameworks, emphasizing the restructuring of narrative constructs through sophisticated language models. Utilizing mBERT, our approach begins by extracting key story entities. Subsequently, XLM-RoBERTa and a BERT-based linguistic evaluation model are employed to direct beam search algorithms in the replacement of these entities. Further refinement is achieved through Low-Rank Adaptation (LoRA), which fine-tunes the extensive 3 billion-parameter BLOOMZ model specifically for generating Arabic narratives. Our methodology underwent thorough testing and validation, involving individual assessments of each submodel. The ROCStories dataset provided the training ground for our story entity extractor and new entity generator, and was also used in the fine-tuning of the BLOOMZ model. Additionally, the Arabic ComVE dataset was employed to train our commonsense evaluation model. Our extensive analyses yield crucial insights into the efficacy of our approach. The story entity extractor demonstrated robust performance with an *F*-score of 96.62%. Our commonsense evaluator reported an accuracy of 84.3%, surpassing the previous best by 3.1%. The innovative beam search strategy effectively produced entities that were linguistically and semantically superior to those generated using baseline models. Further subjective evaluations affirm our methodology's capability to generate high-quality Arabic stories characterized by linguistic fluency and logical coherence.

Keywords: arabic natural language generation; generative language models; story generation; story space remodeling

MSC: 68T05; 68T07; 68T50

check for updates

Citation: Alhussain, A.; Azmi, A.M. Beyond Event-Centric Narratives: Advancing Arabic Story Generation with Large Language Models and Beam Search. *Mathematics* **2024**, *12*, 1548. https://doi.org/10.3390/math12101548

Academic Editors: Radu Tudor Ionescu and Faheim Sufi

Received: 27 March 2024
Revised: 29 April 2024
Accepted: 13 May 2024
Published: 15 May 2024

1. Introduction

Computational narratives facilitate the creation of numerous stories with minimal effort, tailoring them to meet users' educational and entertainment needs. However, the methodology behind storytelling is complex, deriving its diversity from choices in plot organization, character development, and thematic framing. Traditional perspectives on story creation typically focus on constructing story plots through the generation of event sequences. This approach has been explored through various methodologies, including early planning models, case-based reasoning frameworks, and machine learning techniques, as discussed by Alhussain and Azmi [1]. Such a focus on event sequences—which include the narrative's inception, conclusion, and the dynamic transitions between states—is practical. It simplifies the computational generation of story plots, making the process more algorithmically accessible. However, while this approach has successfully produced compelling stories, it suffers from a lack of long-range coherence [2] and may diminish suspense due to an underdeveloped dramatic arc [1].

Contrary to traditional approaches, Singh [3] argued that innovative narratives can arise not only from establishing a core story framework but also through the creative adaptation and reimagining of existing narratives. He illustrated how new stories can be crafted using the same narrative structure by infusing fresh elements into the narrative space, thus enhancing the storytelling process. Additionally, Kybartas and Bidarra [4] highlighted the importance of automated space generation in computational narratives and identified a notable gap: the lack of systems dedicated exclusively to generating narrative spaces.

In recent years, large language models (LLMs) have achieved remarkable success across various natural language processing (NLP) tasks. However, prominent LLMs such as GPT-4 [5] and PaLM 2 [6] are not open-source, limiting transparency into their mechanisms and imposing usage restrictions. These models are accessible only through APIs and do not offer the flexibility of fine-tuning. In response, open-source LLMs have been developed to serve the research community. Examples include English LLMs like LLaMA [7] and BLOOM [8], as well as their multilingual counterparts, m-LLaMA [9] and BLOOMZ [10]. Fine-tuning LLMs can be prohibitively expensive, especially when resources are limited. To mitigate this, techniques such as Low-Rank Adapters (LoRAs) [11] have been introduced. LoRAs enable efficient fine-tuning by making only a small portion of the model trainable. This approach significantly reduces the number of parameters that need to be learned and the overall model size, thus facilitating faster fine-tuning and conserving memory resources.

This paper presents an innovative approach to story generation that moves away from the conventional event-centric method. Instead, it emphasizes redefining the narrative space by focusing on the generation of story elements such as places, objects, and characters. By reimagining the story space while maintaining the original sequence of events, this method allows for the creation of new stories without compromising the narrative's dramatic structure. Furthermore, we utilized a LoRA to fine-tune the extensive BLOOMZ model for story generation tasks using an English story corpus. Following this, we used the beginning of the reimagined story as an initial seed for the story space to prompt the fine-tuned BLOOMZ model to generate new narratives.

Although our proposed approach is applicable to any language, we specifically chose to apply it to Arabic due to the language's morphological complexity and the scarcity of available resources. To the best of our knowledge, this work represents the first attempt at generating textual Arabic stories. Generally, Arabic has received limited attention in the field of natural language generation (NLG). This oversight has been observed across various Arabic NLG tasks, including question generation [12,13], conversational systems [14], and image captioning [15].

The following summarizes our key contributions:

- We introduce a new approach to textual story generation for Arabic, helping to bridge a notable gap in Arabic NLG.
- Our methodology departs from traditional event-driven narratives and instead uses advanced models to generate and modify key story elements.
- We employed an efficient fine-tuning approach to fine-tune an extensive 3 billion-parameter BLOOMZ model tailored for generating narratives in Arabic.
- We showcase the adaptability of LLMs in cross-lingual transfer learning, particularly in zero-shot contexts.

The structure of this paper is as follows. Section 2 explores the challenges associated with the Arabic language. Section 3 reviews related works. Section 4 describes the methodologies proposed for Arabic story generation. Section 5 details the experiments conducted and their results. Finally, we conclude the paper in Section 6.

2. Challenges Due to Arabic Language

Arabic, spoken by over 350 million individuals as a native language, holds significant cultural and religious importance, serving as the liturgical language for nearly 1.8 billion Muslims worldwide [16]. It is recognized as the official language in 26 nations and is one of the six official languages of the United Nations. Despite its widespread use and significance, Arabic has received limited attention in the domain of computational language generation, as discussed in the previous chapter. This research represents a pioneering effort to address the challenge of story generation within the Arabic linguistic context. The intricate morphology and syntax of Arabic present substantial obstacles for computational systems in terms of text comprehension and generation. Effective story generation in Arabic necessitates a deep semantic understanding of the narrative and the capability to construct coherent and contextually appropriate stories.

Crafting narrative in Arabic is fraught with challenges, attributed to the language's unique attributes and the complexities entailed in narrative understanding and generation. These multifaceted hurdles include the following:

Morphological Complexity: Arabic's morphological system, characterized by its extensive use of derivation and agglutination, significantly enriches its lexical diversity [17]. This complexity presents formidable challenges for computational models in accurately parsing and understanding words, thereby impacting the coherence of generated story endings.
Semantic Ambiguity: The semantic richness and syntactic flexibility of Arabic often lead to multiple interpretations of sentences, introducing ambiguity in generated story endings [18,19]. This ambiguity, exacerbated by the absence of diacritical marks in Modern Standard Arabic (MSA), necessitates contextual clarity to ensure the intended meaning is conveyed.
Dialectal Variations: The linguistic landscape of Arabic is marked by a plethora of dialects, distinct from MSA, used in formal and written communication [20]. The challenge lies in aligning the generated story endings with the narrative's dialect, a task complicated by the lack of comprehensive resources for these dialects. This study, however, focuses on MSA.
Computational Resource Limitations: The availability of computational resources for Arabic lags behind that for languages like English, with a notable dearth of extensive annotated datasets and sophisticated language models. This shortfall significantly impedes the advancement of effective story generation models for Arabic. The scarcity of Arabic language resources has negatively impacted various research areas, such as text simplification [21], the creation of judicial support systems [22], and the answering of why-questions [23].
Contextual and Cultural Nuances: Arabic storytelling is deeply infused with cultural and contextual subtleties, demanding a comprehensive understanding of its cultural, historical, and social intricacies [17]. Automated story generation systems must thus ensure that stories are not only linguistically accurate but also culturally resonant and contextually relevant, incorporating appropriate cultural references, proverbs, and social norms.
Long-Range Dependencies: In Arabic storytelling, as in other languages, complex plots often hinge on events and characters introduced at the story's outset. This interweaving of elements across the narrative necessitates that automated story generation systems grasp and incorporate these early details to construct a cohesive and fulfilling story arc. For example, a seemingly minor detail, like a magical seed given to a boy, may later prove central to resolving the narrative's climax. Effective story generation requires the system to recognize and utilize such elements from their introduction, ensuring a coherent and complete narrative.

3. Related Work

The field of automatic story generation has a long historical background. Nevertheless, the advent of pretrained language models (PLMs) marked a significant leap forward, rekindling research interest and placing renewed emphasis on the exploration of story generation. Generative PLMs, whether auto-regressive or encoder–decoder, have the ability to generate stories based on a given prompt without additional training. However, the stories they produce often lack higher-level features such as text coherence and logical progression, particularly as the stories become longer [24]. As a result, various research attempts have been undertaken to enhance the quality of stories generated by PLMs.

To facilitate more strategic content planning and support the long-range coherence of stories, the concept of hierarchical story generation has been introduced. This technique involves generating a high-level summary of the story, which then serves as a guiding framework for generating the actual narrative. These high-level summaries can take various forms, including short descriptions [25–27], writing prompts [28], entity mentions [29,30], or plans [31,32].

Numerous studies have endeavored to elevate the storytelling capabilities of LLMs by integrating external commonsense knowledge bases, such as ConceptNet [33] and ATOMIC [34]. These efforts involve either implicitly encoding commonsense knowledge into LLMs through fine-tuning [2,35] or actively querying these knowledge bases and employing the acquired information to explicitly govern the stories generated by LLMs, as showcased in the works of Xu et al. [36], Peng et al. [37], and Ammanabrolu et al. [38].

Vijjini et al. [39] conceptualized interpersonal relationships as latent variables and subsequently employed GPT-2 to generate stories, with the generation process conditioned on both the relationship set and the story context as a prompt. In a different approach, Xie et al. [40] merged two distinct LLMs, BERT [41] and GPT-2 [42], to construct a domain-specific variational autoencoder for story generation. This innovative strategy aimed to strike a balance between providing stories that are not only high quality but also diverse in their narrative content.

The task of story ending generation (SEG) is a specialized domain within the broader field of story generation focusing on the creation of coherent conclusions to narratives by comprehensively understanding the preceding context. While numerous studies have addressed SEG in the English language, exemplified by works such as those by Huang et al. [43], Liu et al. [32], and Wang et al. [44], research in the Arabic language context is scarce, with a notable contribution by Alhussain and Azmi [45]. The scarcity of Arabic language resources, as discussed in Section 2, has necessitated innovative approaches such as the one employed in [45], which leverages cross-lingual transfer learning. This method utilizes multilingual models, including mBART, mT5, and mT0, to facilitate the generation of story endings in Arabic, evaluating their efficacy in both zero-shot and few-shot learning environments.

The introduction of LLMs marked a significant leap forward in various natural language processing (NLP) tasks, including story generation. In a study conducted by Clark et al. [46], the proficiency of non-experts in discerning between stories authored by humans and those generated by LLMs was examined. Their findings revealed that untrained evaluators struggled to differentiate between human-generated and LLM-generated text. Even when evaluators received training specifically aimed at distinguishing between the two, notable improvements in discernment were not observed.

In language models, a prompt is a piece of text given to the model to initiate or guide its generation of subsequent text. It acts as the starting point or context for the model to create coherent and relevant text [47]. Researchers have used techniques called prompt engineering to help LLMs create stories. For example, Xie et al. [48] used a simple method to have GPT-3 write stories. This approach proved that LLMs can write stories that are of notably better quality compared to the best existing story generation models. Moreover, these models can sometimes write as well as human authors, although they often end up copying stories from their training data.

Yao et al. [49] introduced Tree-of-Thoughts (ToT) prompting as an innovative mechanism within prompt engineering to enhance the performance of LLMs. ToT empowers LMs to engage in deliberate decision making by exploring diverse reasoning paths and self-assessing choices to determine the subsequent course of action. Additionally, it facilitates backtracking when necessary to make overarching decisions. Their findings demonstrate that employing ToT to prompt LLMs in creative writing results in more cohesive text, preferred by human evaluators over passages generated by simpler prompting techniques.

Wen et al. [50] constructed a retrieval repository for target conditions to generate few-shot examples for prompting LLMs. They begin by completing the entire task initially, and then iteratively refining and enhancing it over subsequent iterations.

The Re^3, which stands for Recursive Reprompting and Revision [51], tackled the challenges of long-range plot coherence and relevance. This framework improves story generation using iterative revisions, resulting in stories that are more coherent and relevant to the initial premise than those generated directly from the same base model. Expanding on Re^3, DOC, short for Detailed Outline Control [52], further refines this approach by introducing a detailed outliner that generates a hierarchical story outline, along with a detailed controller that adjusts the intensity of event descriptions. DOC has shown improvements over Re^3 in terms of plot coherence, outline relevance, and overall narrative appeal.

In a different approach, LLMs have been used to co-create stories with human writers. This collaboration involves simple turn-taking between the writer and the AI during the story development process. One such tool is TaleBrush [53], a generative story ideation tool that employs line sketching interactions between writers and LLMs. Here, user sketches are translated into prompts using a control module, enabling the LLM to generate stories based on these sketches. This iterative process allows writers to explore and develop stories according to their intentions while drawing inspiration from the stories generated by the LLM. Another example is Wordcraft [54], a text editor that facilitates collaboration between users and LLMs in writing stories, offering novel human-LLM co-writing experiences. For instance, the language model can engage in open-ended conversations about the story, respond to writers' custom requests expressed in natural language, and provide suggestions to assist writers in overcoming creative blocks during the writing process.

Wan et al. [55] employed LLMs in a collaborative human–AI effort during the prewriting stage, which involved the exploration and refinement of ideas prior to drafting. Unlike story generation, prewriting demands a focus on originality and diversity over coherence and structure. Their study revealed that humans primarily lead the process of idea exploration and development. However, it also highlighted a dynamic interplay of initiative between humans and LLMs throughout the collaborative prewriting endeavor. In facilitating decision-making processes, Davis et al. [56] automated the creation of 'what-if' scenarios by directing BERT through simulated conversations to extract a model from a corpus. Their method delivers outcomes akin to those of prior research while drastically reducing the necessity for human intervention in the model-building phase.

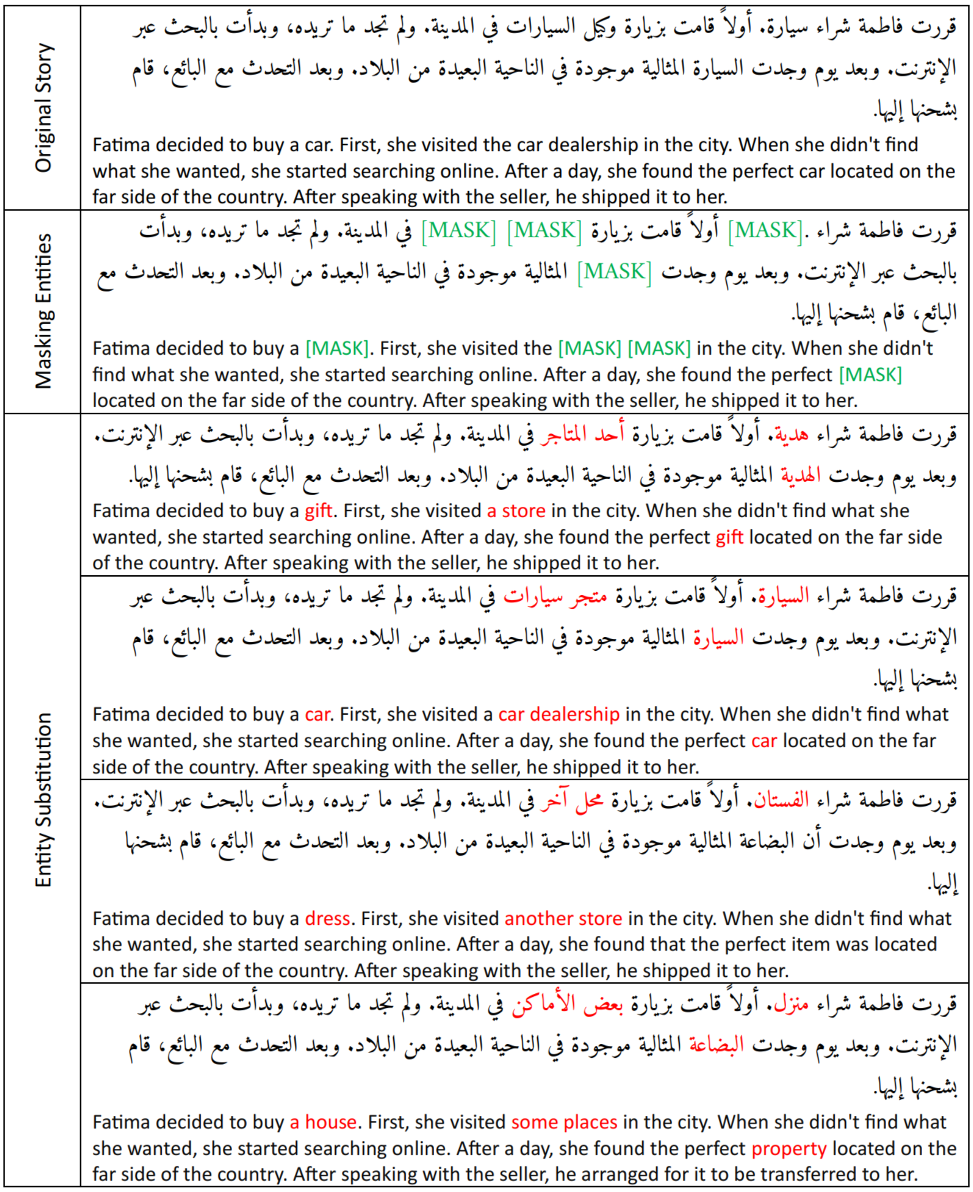
4. Methodology

In this study, our objective was to create Arabic stories by reshaping the space of existing stories. The task can be outlined as follows: starting with a story S, we define the narrative space by identifying its constituent entities $E = \{e_1, e_2, \ldots, e_n\}$. Then, a new story space is created by substituting each e_i, where $i \in [1, n]$. It is imperative that the selection of these new entities is not only linguistically appropriate but also logical in the context of the story. Subsequently, we utilize the newly generated narrative space as a context for generating a new story. This section provides a detailed overview of our methodology.

While Figure 1 shows examples of generating new stories by substituting the entities of an existing story, Figure 2 provides an overview of the entire system. The subsequent subsections offer a thorough exposition of the proposed methodology.

4.1. Story Entity Extraction

Entities in the context of a story can refer to the various elements or objects within the narrative that hold significance or play specific roles. These entities can include characters, geographical locations, physical objects, time expressions, and other elements that contribute to the richness of the story, providing depth, context, and a framework for the narrative. In our research, the emphasis was on extracting characters, physical objects, and geographical locations. However, it is important to note that these entities were treated uniformly without differentiation based on their specific types.



Original Story	قررت فاطمة شراء سيارة. أولاً قامت بزيارة وكيل السيارات في المدينة. ولم تجد ما تريده، وبدأت بالبحث عبر الإنترنت. وبعد يوم وجدت السيارة المثالية موجودة في الناحية البعيدة من البلاد. وبعد التحدث مع البائع، قام بشحنها إليها. Fatima decided to buy a car. First, she visited the car dealership in the city. When she didn't find what she wanted, she started searching online. After a day, she found the perfect car located on the far side of the country. After speaking with the seller, he shipped it to her.
Masking Entities	قررت فاطمة شراء [MASK]. أولاً قامت بزيارة [MASK] [MASK] في المدينة. ولم تجد ما تريده، وبدأت بالبحث عبر الإنترنت. وبعد يوم وجدت [MASK] المثالية موجودة في الناحية البعيدة من البلاد. وبعد التحدث مع البائع، قام بشحنها إليها. Fatima decided to buy a [MASK]. First, she visited the [MASK] [MASK] in the city. When she didn't find what she wanted, she started searching online. After a day, she found the perfect [MASK] located on the far side of the country. After speaking with the seller, he shipped it to her.
Entity Substitution	قررت فاطمة شراء هدية. أولاً قامت بزيارة أحد المتاجر في المدينة. ولم تجد ما تريده، وبدأت بالبحث عبر الإنترنت. وبعد يوم وجدت الهدية المثالية موجودة في الناحية البعيدة من البلاد. وبعد التحدث مع البائع، قام بشحنها إليها. Fatima decided to buy a gift. First, she visited a store in the city. When she didn't find what she wanted, she started searching online. After a day, she found the perfect gift located on the far side of the country. After speaking with the seller, he shipped it to her.
	قررت فاطمة شراء السيارة. أولاً قامت بزيارة متجر سيارات في المدينة. ولم تجد ما تريده، وبدأت بالبحث عبر الإنترنت. وبعد يوم وجدت السيارة المثالية موجودة في الناحية البعيدة من البلاد. وبعد التحدث مع البائع، قام بشحنها إليها. Fatima decided to buy a car. First, she visited a car dealership in the city. When she didn't find what she wanted, she started searching online. After a day, she found the perfect car located on the far side of the country. After speaking with the seller, he shipped it to her.
	قررت فاطمة شراء الفستان. أولاً قامت بزيارة محل آخر في المدينة. ولم تجد ما تريده، وبدأت بالبحث عبر الإنترنت. وبعد يوم وجدت أن البضاعة المثالية موجودة في الناحية البعيدة من البلاد. وبعد التحدث مع البائع، قام بشحنها إليها. Fatima decided to buy a dress. First, she visited another store in the city. When she didn't find what she wanted, she started searching online. After a day, she found that the perfect item was located on the far side of the country. After speaking with the seller, he shipped it to her.
	قررت فاطمة شراء منزل. أولاً قامت بزيارة بعض الأماكن في المدينة. ولم تجد ما تريده، وبدأت بالبحث عبر الإنترنت. وبعد يوم وجدت البضاعة المثالية موجودة في الناحية البعيدة من البلاد. وبعد التحدث مع البائع، قام بشحنها إليها. Fatima decided to buy a house. First, she visited some places in the city. When she didn't find what she wanted, she started searching online. After a day, she found the perfect property located on the far side of the country. After speaking with the seller, he arranged for it to be transferred to her.

Figure 1. Examples of generating new stories by substituting entities in an existing narrative. The '[MASK]' marker, displayed in green, serves to conceal the entities extracted from the original story by the entity extractor. The newly generated entities, displayed in red, are the outputs of the entity substitution model. For ease of understanding, we include an English translation for each example.

We approached the task of extracting entities from the unstructured text of a story as a token classification problem. Inspired by the remarkable performance of BERT [41] in token classification tasks, such as Named Entity Recognition (NER) [41] and Part-of-Speech (POS) tagging [57], we implemented the Story Entity Extraction (SEE) model as a BERT-based token classifier for extracting story entities. A token classification layer was added on top of the pretrained BERT, which consists of a linear layer for token-level classification, and a softmax activation function, to convert the raw scores from the linear layer into a valid probability distribution.

Given that our SEE model does not distinguish between various entity categories, we trained it to classify words as either entities or non-entities. However, understanding that entities could be either individual words or groups of words, we introduced three token classes: "O" for non-entities, "B-Ent" for the first word of an entity, and "I-Ent" for the consecutive words within an entity. Another challenge arose from BERT's use of WordPiece tokenization [58], which splits a single word into multiple tokens. To address this issue, we assigned the same word class to each token of the word.

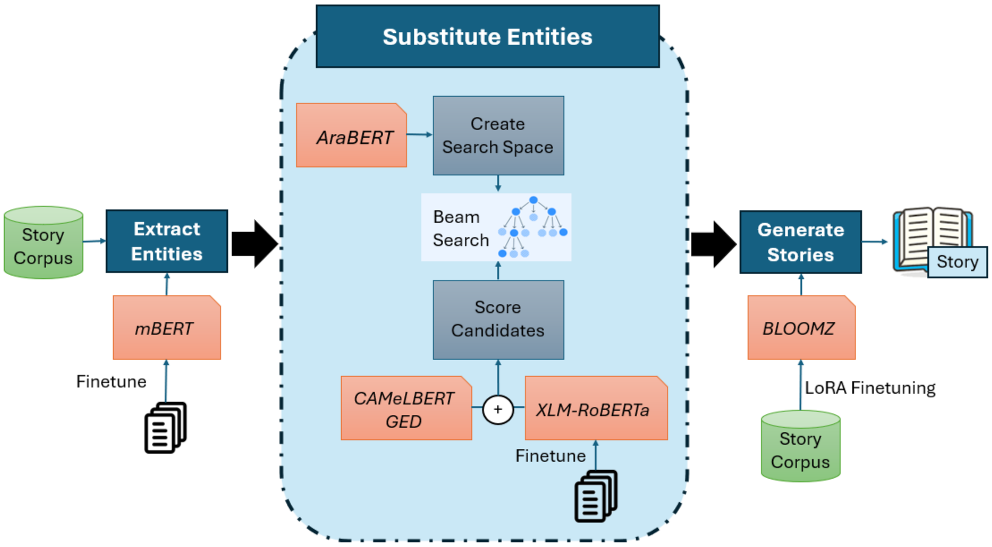


Figure 2. Our proposed story generation system.

Since story entities may occur as named entities (characters and locations), we initially employed two models for tagging entities in the dataset: a NER model [41] and an entity detection model [59]. Nonetheless, preliminary results indicated an improvement in performance when named entities were omitted. Consequently, we opted to exclusively utilize the entity detection model for tagging stories, deferring the treatment of named entities to a subsequent step.

Recognizing that the entity detection model is trained only in English and there is no equivalent model for Arabic, we addressed the lack of Arabic resources by using cross-lingual transfer learning. First, we used the English entity extraction model to identify entities in an English dataset. Then, we initialized our BERT model with pretrained multilingual uncased BERT parameters. After this, we fine-tuned our BERT model using the tagged English dataset. Finally, we applied the fine-tuned model to identify entities in Arabic stories in a zero-shot setting.

4.2. Substituting Story Entities

After extracting story entities, we replace them with new entities to reshape the story space. This section provides a detailed explanation of the entity substitution process.

4.2.1. Generating New Entities

To generate new entities that fit seamlessly into the story context, we implemented a two-step process: (a) First, we masked the entities previously identified in the story by our SEE model, effectively creating blanks within the text; and (b) Next, we utilized the capability of LLMs to fill in these blanks, generating new, contextually appropriate entities in place of the originals. More details below.

(a) Models for blank filling. Large language models, such as BERT [41] and RoBERTa [60], have the ability to fill in the blanks in a given context. This capability results from training the models on large text corpora through the masked language modeling (MLM) task. The goal of the MLM task is to train the model to predict masked (hidden) words within a given text sequence. During training, 15% of tokens in the input text are randomly replaced with a special token called `[MASK]`. The model then learns to predict the original tokens that were replaced with [MASK] based on the surrounding context. By training on the MLM task, the model learns to generate meaningful representations of words and phrases based on their context within a text sequence. However, while the training objective of BERT includes MLM and Next Sentence Prediction (NSP), RoBERTa is pretrained solely on MLM. Additionally, RoBERTa uses dynamic MLM, where the positions of [MASK] tokens are dynamically changed during training. As an English model, RoBERTa outperformed BERT in many downstream tasks. However, this superiority may result from RoBERTa's extensive training, which involves training on longer sequences, increasing the batch size and the length of training, and utilizing 10× training data.

The T5 (Text-To-Text Transfer Transformer) model adopts a text-to-text approach for training, wherein it learns to transform input text sequences into output text sequences. Unlike conventional models like BERT and RoBERTa, T5 is not trained using the typical Masked Language Modeling (MLM) objective. It employs a specialized form of encoder–decoder masked language modeling, enabling it to proficiently fill in blanks within a given context. In T5, rather than using the [MASK] token, the model employs sentinel tokens identified by their prefixes `<extra_id>` as potential masks. Each sentinel token represents a unique mask token for a sentence, starting with `<extra_id_0>`, `<extra_id_1>`, and so forth, up to `<extra_id_99>`. This approach allows T5 to effectively address the masking task while leveraging its text-to-text training paradigm.

To generate new entities, we experimented with available BERT, RoBERTa, and T5 models that support Arabic. Specifically, we used the following models:

1. mBERT [41] is an extension of BERT that has been pretrained on multiple languages. This multilingual approach enables the model to understand and generate text in various languages.
2. AraBERT [61] is a dedicated instantiation of BERT, fine-tuned exclusively for Arabic language processing. Its pretraining process involves a substantial dataset of 70 million Arabic sentences, equivalent to approximately 77 GB of textual data.
3. CAMeLBERT [62] is a collection of BERT models pretrained on Arabic texts with different sizes and variants. We selected CAMeLBERT for MSA to align with the characteristics of our dataset. CAMeLBERT (MSA) was trained on a larger dataset compared to AraBERT. However, it is only available in the base version, comprising 12 layers, whereas AraBERT contains 24 layers.
4. AraT5 [63] is a variant of the text-to-text transformer (T5) model designed specifically for MSA. Our utilization involved the enhanced version, AraT5v2-base-1024.
5. XLM-R, standing for Cross-lingual Language Model–RoBERTa [64], builds upon the RoBERTa framework, having been trained across multiple languages using a wide array of multilingual corpora to achieve language-independent representations. Its efficacy across a spectrum of cross-lingual NLP benchmarks and applications has been notably impressive.

Table 1 presents a comprehensive comparison of the models. Our initial findings indicate that AraBERT surpasses the other models in generating high-quality entities that seamlessly integrate into the story context. Consequently, we employed AraBERT to construct a space of potential entities for the story, as elaborated in the following section.

Table 1. Comparison of the models used for blank filling. Abbreviations used in the table are as follows: Enc-Dec for encoding–decoding; SMLM+NSP for Static Masked Language Modeling + Next Sentence Prediction; DMLM for Dynamic Masked Language Modeling; BPE for Byte Pair Encoding.

						Training		
Model	**Arch.**	**Version**	**Layers**	**Size**	**Encoding**	**Objective**	**Language**	**Data**
AraBERT	Encoder	Large	24	∼369 M	WordPiece	SMLM + NSP	Arabic	77 GB
CAMeLBERT (MSA)	Encoder	Base	12	∼109 M	WordPiece	SMLM + NSP	Arabic	167 GB
mBERT	Encoder	Base	12	∼178 M	WordPiece	SMLM + NSP	Multilingual	—
XLM-R	Encoder	Large	24	∼560 M	BPE	DMLM	Multilingual	2.5 TB
AraT5	Enc-Dec	Base	12	∼368 M	SentencePiece	Text-to-text	Arabic	248 GB

(b) Creating an etity search space. Our entity substitution task shares similarities with MLM. Once we identify story entities, our objective is to replace them with new entities that seamlessly integrate into the story context. To achieve this, we employ a technique akin to MLM. Specifically, we mask these entities using the [MASK] token. Subsequently, we leverage an AraBERT [61] model to predict and fill in the masked tokens. This prediction considers the bidirectional context of the given story, relying on the probability distribution learned during the pretraining phase.
However, it is worth noting a distinction: while the probability of predicting a masked token in BERT-based models is typically independent of other masked tokens, we enhance the entity substitution results by introducing a sequential substitution approach. This involves incorporating previously predicted story entities when predicting subsequent entities, thereby promoting a more coherent and contextually appropriate substitution process. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$ denote the indexes of the masked entities in the story S, where N is the number of masked entities. Let $E = \{e_{\pi_1}, e_{\pi_2}, \ldots, e_{\pi_N}\}$ denote the set of masked entities in S and $X = \{x_1, x_2, \ldots, x_m\}$ denote the set of unmasked tokens representing the non-entities of the story context. Given θ as the model parameters learned during pretraining, the objective of our entity substitution module is to maximize the probability of predicting story entities as follows:

$$\hat{S} = \sum_{n=1}^{N} \log p\left(e_{\pi_n} \mid X, e_{\pi_1}, \ldots, e_{\pi_{n-1}}; \theta\right) \tag{1}$$

Furthermore, we do not rely solely on the single most probable substitution. Instead, our approach involves considering the topmost probable substitutions. The entity generation module is utilized to create an initial search space of possible entities. This approach enables the incorporation of additional evaluation criteria to select the most appropriate entity for the overall story context. Moreover, it aims to foster diverse story generation, allowing a single story to serve as a basis for generating multiple stories.

In our study, we noted instances where the utilization of BERT-based models in substituting story entities resulted in the generation of nonsensical stories. This was a result of the high ambiguity of Arabic language. For example, AraBERT was able to appropriately predict the mask in the sentence “كان محمد يشعر بـ [MASK].”, which translates to “Muhammad was feeling [MASK]”, by returning الصدمة (shocked), التعب (tired), الإحباط (frustration), الارتياح (relief), and الملل (boredom) as the most probable tokens. Nonetheless, when the sentence was rephrased to “شعر محمد بـ[MASK].”, which translates to “Muhammad felt”, the model failed to predict the mask correctly; it returned ‘.’, ‘ع’, ‘(’, ‘)’, and ‘أ’ as the most probable tokens, where more appropriate words such as ‘أمان’ (safety) were predicted as less probable. This may be a result of the ambiguity of the word ‘شعر’, which can be translated to ‘felt’, ‘hair’, or ‘poetry’. In addition, due to the high morphology of Arabic, there were instances where masks were filled with suffixes. For example, to fill the mask in “ذهب محمد إلى بيت[MASK].”, the most probable tokens are ‘+ه’, ‘أبيه’, and ‘+نا’. Although this may be linguistically accepted in an isolated sentence, it does not serve the aim of the entity substitution model in replacing the story entity with a new entity. Therefore, to overcome this limitation, we post-processed the tokens returned by the model to remove stop words, pronouns, and numbers. Furthermore, we evaluated the returned tokens linguistically and logically to select the most appropriate token that serves as a good story entity, as discussed in subsequent sections.

4.2.2. Commonsense Evaluation

While LLMs have shown exceptional performance in various tasks, including story generation, they often struggle with maintaining long-range coherence and commonsense reasoning. This challenge, recognized since the early development of LLMs [24], arises from the limited sequence lengths that transformers can handle due to the complexity of their self-attention operations [65]. Transformers form the core architecture of most LLMs [66]. Despite these limitations, recent studies have shown that LLMs do develop a degree of commonsense reasoning during pretraining [67,68]. Moreover, further enhancements have been achieved by fine-tuning these models on commonsense datasets [5,69,70].

This limitation was clearly observed in our preliminary entity substitution experiments. As the input story context lengthened, numerous logical conflicts emerged. To address this limitation, our approach involved training a dedicated commonsense evaluator to provide guidance during the story generation process.

The XLM-RoBERTa model [64] showed superior performance in both Arabic [71] and multilingual commonsense evaluations [72]. Therefore, we used XLM-RoBERTa to train on an Arabic commonsense dataset, developing our commonsense evaluation module. The commonsense score for each sentence is calculated using a sentence-level vector from the fine-tuned XLM-RoBERTa model. This pooling mechanism gathers information from the entire story context, creating a fixed-size representation that captures contextual information and helps determine the likelihood score of a sentence.

4.2.3. Entity Selection

In the process of selecting the most fitting entity for a specific position in the story, we evaluate the overall story context after substituting each entity. To guide the search toward entities that are more likely to be correct or optimal, we used beam search, where the search space is created by the entity generation module, as discussed in Section 4.2.1. Beam search is advantageous because it explores multiple possibilities simultaneously, preventing the algorithm from being trapped in locally optimal solutions. This is crucial to avoiding selecting entities for the story's beginning that may not harmonize well with subsequent sentences, thus safeguarding the story's overall quality and long-range coherence and preventing system failures in generating subsequent entities.

The scoring function in the context of beam search serves as a pivotal mechanism for evaluating and ranking candidate entities during the story generation process. We propose to assess the quality of a candidate entity by evaluating the story context after substituting the new entity based on two criteria:

1. Commonsense acceptance, which entails the assessment of the generated story context for its plausibility and coherence. The commonsense score is computed through the utilization of the commonsense evaluation module, as elaborated in the preceding section (Section 4.2.2).
2. Linguistic correctness, which assesses how well the generated stories adhere to the rules and structure of the language. It involves evaluating the grammatical accuracy of the candidate stories. For this task, we employed the Arabic grammatical error detection model (CAMelBERT GED) proposed by Alhafni et al. [73]. The linguistic correctness score for a candidate story is calculated as follows:

$$L_{\text{score}} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2}$$

where n represents the number of tokens in the story context as produced by the CAMelBERT GED tokenizer, and X_i denotes the grammatical score of the i-th token, as evaluated by the CAMelBERT GED model within the given context.

The beam search scoring function is expressed as a linear combination of the commonsense acceptance score (CS_{score}) and the linguistic correctness score (L_{score}), given by the formula

$$BeamScore = \alpha CS_{\text{score}} + \beta L_{\text{score}} \tag{3}$$

where α and β are weighting coefficients determining the contribution of each score to the overall beam score.

In addition, we incorporated an element of randomness into the entity selection process to introduce diversity in the generated stories. This is achieved through the application of a temperature (t) adjustment to the entity scores before the final selection of top candidates:

$$ScaledScore_i = \frac{\exp(\log(BeamScore_i)/t)}{\sum_{j=1}^{m} \exp(\log(BeamScore_j)/t)} \tag{4}$$

Here, m represents the beam width, and $BeamScore_i$ is the original beam score of the i-th candidate as expressed in Equation (3).

The temperature adjustment serves as a control mechanism, influencing the degree of randomness introduced into the selection. A higher temperature results in a more uniform distribution of probabilities, making a wider range of entities eligible for selection. On the other hand, a lower temperature emphasizes the differences in scores, leading to a more deterministic selection process.

4.3. Story Completion

After constructing a new story space through the substitution of story entities, this newly created story space serves as a stimulus for generating a new story. To accomplish this task, we utilized BLOOMZ [10], an advanced variant of BLOOM [8] designed for multitask prompted fine-tuning. BLOOMZ, as an autoregressive large language model, was trained to generate text from a given prompt. This training involved vast volumes of textual data, leveraging extensive computational resources on an industrial scale. In addition to BLOOMZ's ability to produce text that closely mimics human-authored content, it exhibits transfer learning capabilities, enabling it to apply acquired knowledge to languages not present in the fine-tuning data, provided that they were part of the pretraining data; its pretraining data encompassed 46 languages, including Arabic, and 13 programming languages.

The remarkable capabilities of BLOOMZ stem from its extensive number of parameters, a characteristic that renders full fine-tuning for downstream tasks resource intensive and time consuming. To address this challenge, various parameter-efficient fine-tuning (PEFT) approaches have been introduced, aiming to enhance the efficiency of fine-tuning large models in terms of parameters and computations. These approaches include adapters [74,75], prefix tuning [76], and Low-Rank Adaptation (LoRA) [11]. These strategies make the process more feasible for the research community and are particularly helpful when working with limited data for a specific task.

In this work, we fine-tuned the largest BLOOMZ model with 3 billion parameters on an English story dataset and used it for Arabic story generation in zero-shot settings. To make fine-tuning efficient and feasible, we utilized Huggingface libraries to employ the following strategies:

1. We quantized the BLOOMZ model to reduce the memory usage.
2. We froze all of the model's parameters while enabling the gradients for the input embeddings. This helped in fine-tuning adapter weights while keeping the model weights fixed.
3. We applied the LoRA approach to BLOOMZ's attention weight modules, as recommended by the original paper [11], with a rank of 4.

Due to BLOOMZ's training methodology involving prompted fine-tuning, we developed a prompt template by incorporating concise and clear natural language instructions into the input story context. Our standardized prompt format is structured as follows:

```
Complete the following Story: ## Context: {story_context} ## Output:
{story_completion},
```

where '{story_context}' represents the input story context provided to the model, and '{story_completion}' represents the generated story output. During the training phase, the first sentence of each story in the training dataset served as the story context, while the remaining sentences were utilized as completions. During inference, the same template is applied, with '{story_context}' filled with the story space created by substituting story entities, as detailed in Section 4.2, while '{story_completion}' is left blank.

5. Experiments and Results

In this section, we detail the datasets used (Section 5.1) and describe the experiments conducted and the results obtained from all sub-models: the story entity extractor (Section 5.2), the commonsense evaluator (Section 5.3), the entity substitution model (Section 5.4), and the story generator (Section 5.5). The entity substitution model employs beam search, while the other sub-models were developed through fine-tuning existing LLMs. We summarized the fine-tuning experimental setup in Table 2 for ease of reference, with detailed explanations provided in each corresponding subsection. For the evaluations of entity extraction, commonsense evaluation, and entity substitution, we utilized automatic metrics. In contrast, the assessment of the generated stories involved both subjective and objective methods.

Table 2. Experimental setup for fine-tuned models.

Model	Base Model	Version	Learning Rate	Batch Size	Epochs	LoRA Rank
Story Entity Extractor	mBERT	Base	5×10^{-5}	16	3	N/A
Commonsense Evaluator	XLM-R	Large	3×10^{-6}	8	4	N/A
Story Generator	BLOOMZ	3B	1×10^{-3}	3	1	4

5.1. Datasets

Due to the absence of Arabic story datasets, we trained and evaluated our models using ROCStories [77], an English dataset commonly utilized in NLP research for tasks related to story generation and comprehension. This corpus comprises five-sentence stories with a title, each logically following everyday topics created by crowd workers. These narratives contain a variety of commonsense causal and temporal relations between everyday events. This dataset is commonly employed in story ending generation (SEG) tasks, as seen in references such as [32,43–45], where a model is presented with four sentences and tasked with generating a fitting fifth sentence. Here is an example story from ROCStories titled "The Test": "Jennifer has a big exam tomorrow. She got so stressed, she pulled an all-nighter. She went into class the next day, weary as can be. Her teacher stated that the test is postponed for next week. Jennifer felt bittersweet about it."

5.1.1. Story Entity Extractor Dataset

For the training of the story entity extractor, we randomly sampled a total of 50,000 stories from the ROCStories dataset, allocating 30,000, 10,000, and 10,000 for the training, validation, and test sets, respectively. However, recognizing that the entity extractor exhibited improved performance with shorter texts, we opted to treat each sentence within a story as an individual input. This restructuring resulted in a dataset with a total size of 243,000. Detailed statistics of the training data are presented in Table 3.

Table 3. Details of the training data for the Story Entity Extraction model.

Data Split	Sentences	Tokens
Training	145,166	1,297,943
Validation	48,706	436,221
Test	49,170	402,624

5.1.2. BLOOMZ Fine-Tuning Dataset

To fine-tune the BLOOMZ model for story generation, we combined the ROCStories 2016 and ROCStories 2017 sets, resulting in a total of 98,161 stories. We excluded the story titles and treated each narrative as consisting of exactly five sentences. On average, each story contained 43.58 words. We used the first sentence of each story as the story context to prompt the model, with the subsequent sentences serving as story completions. This methodology is detailed in Section 4.3, where we describe the process of constructing the BLOOMZ training template.

5.1.3. Entity Substitution Dataset

To evaluate our proposed entity substitution model, we utilized a set of 1000 stories randomly selected from the ROCStories dataset. These stories were translated into Arabic using Google translate (https://translate.google.com/, accessed on 12 September 2023). Subsequently, we masked the identified entities in these stories using our entity extraction model. The Arabic stories with masked entities were then used as a test set to assess the effectiveness of our entity substitution model in filling in these masked entities, compared to baseline models.

5.1.4. Commonsense Evaluation Dataset

To train our commonsense evaluator, we used the Arabic Dataset for Commonsense Validation (Arabic ComVE) [78], which was professionally translated from the original English Commonsense Validation dataset (comVE) [79]. This dataset is the only Arabic benchmark for commonsense validation and was designed to test the ability of models to assess commonsense reasoning. Each entry in the dataset includes two syntactically similar sentences: one that is logically coherent and one that is not. The model was trained to identify the more logical sentence. The dataset consists of 10,000 training examples, 1000 validation examples, and 1000 test examples.

5.2. Entity Extraction Evaluation

To extract story entities, we fine-tuned the pretrained multilingual uncased BERT on our entity extraction dataset, as discussed in previous sections. During training, both the classifier and the BERT model underwent joint training without the constraint of freezing the pretrained parameters. The model was trained for three epochs, with a learning rate of 5×10^{-5}, and a batch size of 16.

The SEE model achieved an overall *F*-score of 96.62%. A detailed breakdown of the evaluation scores for each category is provided in Table 4. Notably, the high scores for the "O" tag highlight the model's robust ability to distinguish between entities and non-entities. Only 5.58% of the "B-Ent" tokens were not detected as entities. The primary limitation lies in the model's accuracy in classifying "I-Ent" tokens. Our analysis revealed that 93.42% of misclassified 'I-Ent' tokens were classified as "B-Ent". However, this discrepancy is inconsequential for our system, as both token classes are treated equivalently.

Table 4. Results of the Story Entity Extraction (SEE) model.

Tag	Precision	Recall	*F*-Score
O	98.09	98.21	98.15
B-Ent	93.32	93.08	93.20
I-Ent	86.95	86.02	86.48

5.3. Commonsense Evaluation

Drawing inspiration from the work of Al-Bashabsheh et al. [71], we fine-tuned the XLM-RoBERTa model [64] on the Arabic ComVE dataset [78] for eight epochs, using a learning rate of 3×10^{-6} and a batch size of 8. Our model achieved an accuracy of 84.3% on the test set, which is 3.1% higher than the result reported in [71]. It is worth noting, however, that the best model performance was achieved after just four epochs of training.

The commonsense score, derived from the sentence-level vector produced by the fine-tuned XLM-RoBERTa model, effectively indicates the logical coherence of input sentences. Table 5 provides an example of these commonsense scores. As the example shows, although the MLM model rated "the kitchen" lower than "a shop" and "school," the commonsense evaluator accurately determined that "the kitchen" is the more logical setting in the given story context.

Table 5. An example of the commonsense scores produced by our evaluation model for various entity substitutions in the story context "كان محمد يشعر بالجوع لذا ذهب إلى[MASK]", which translates to "Muhammad was hungry so he went to [MASK]". Entries are ordered by their probability score.

MLM Top 5 Tokens	MLM Probability Score	Commonsense Score
مطعم (a restaurant)	19.38	4.45
المطعم (the restaurant)	10.41	4.36
محل (a shop)	7.15	0.87
المدرسة (school)	3.83	−3.96
المطبخ (the kitchen)	2.82	3.83

5.4. Entity Substitution Evaluation

For substituting story entities, we implemented our proposed beam search algorithm with a beam width of 5 and temperature t set to 0.2. Additionally, both α and β in the scoring equation were configured to be equal to 1. The evaluation involved comparing the stories generated by our entity substitution model with those produced by the original LLMs listed in Table 1.

To assess the models' proficiency in generating high-quality new entities, we evaluated stories after the substitution of these entities using three criteria:

1. Linguistic analysis: this is one of the earliest automated metrics used to assess the quality of generated stories. In recognizing that high-quality writing inherently minimizes grammatical mistakes, several researchers [80,81] have employed grammar, among other metrics, for evaluating story quality. Following this methodology, we employed linguistic analysis, leveraging the CAMeLBERT Grammatical Error Detection model [73], to assess generated stories.
2. Semantic relatedness: each sentence within a story should maintain coherence with the overall story context [80]. To assess this criterion, we computed the mean cosine similarities between the embeddings of each sentence in the story and the remaining story context. Sentence-BERT [82] was employed to generate the embeddings for the sentences.
3. Originality: ELECTRA models [83] undergo training to differentiate between "real" input tokens and "fake" input tokens generated by another neural network. In our context, we employed the AraElectra [84] discriminator model to identify whether a filled entity is authentic or not. The reported originality score represents the percentage of tokens detected as fake. Consequently, a lower originality score indicates a more favorable result.

Table 6 provides a comprehensive evaluation of our model and baselines in the entity substitution task. Upon comparing the baselines, it becomes evident that AraBERT excelled in semantics and originality, slightly underperforming XLM-R in linguistic metrics. This compelling performance led us to choose AraBERT as the initial model for creating the entities search space, as elaborated in Section 4.2. CAMeLBERT secured the second position with only a marginal difference from AraBERT. On the other hand, our observations revealed that AraT5 tends to generate highly repetitive text, adversely impacting its overall results. While XLM-R exhibited high scores in both linguistic and semantic metrics, its new entities were significant detected by the AraElectra discriminator, potentially influenced by its multilingual nature.

By integrating linguistic correctness and commonsense acceptance into the beam search scoring function, we effectively improved the performance of the entity substitution model. The linguistic score for the stories generated by our model approached near-optimal levels at 99.23%. Furthermore, the semantic relatedness score experienced a significant boost, reaching 76.1%, which is approximately twice the score achieved by our base model, AraBERT. This enhancement in both linguistic and semantic aspects contributed to an overall improvement in the quality of the newly substituted entities, making them more challenging for the AraElectra discriminator to identify as machine-generated, where only 9.88% of the newly generated entities were detected as fake tokens compared to 15.16% of the entities generated by the base model.

Table 6. Evaluation of entity substitution models focused on linguistics and semantics, where higher scores indicate better performance and originality, where lower scores are preferable. The top-performing values in each category are highlighted in bold for clarity.

Model	Linguistics	Semantics	Originality
mBERT	93.70	38.84	35.21
AraBERT	97.83	39.14	15.16
CAMeLBERT	96.90	38.47	17.95
AraT5	95.20	45.36	24.52
XLM-R	98.17	38.31	20.14
Our Model	**99.23**	**76.1**	**9.88**

5.5. Evaluating Generated Stories

Having replaced the story entities, we selected the initial two sentences from each story to encapsulate the newly crafted story space. Subsequently, we used these representations as prompts in the BLOOMZ model [10] to generate complete stories. A total of 250 stories were generated, averaging 6.1 sentences and 41.88 words each. This relatively short story length is attributed to the nature of the stories in ROCStories, the dataset utilized for fine-tuning BLOOMZ, which typically contains an average of 43.58 words per story, as discussed in Section 5.1.2. To evaluate the quality of the generated stories, we employed a combination of subjective and objective metrics, detailed in the following subsections.

5.5.1. Objective Evaluation

By assessing the language quality of the generated stories, we observed a notable absence of mixed-language outputs from BLOOMZ, an occurrence often encountered in zero-shot transfer learning across various generative multilingual models. Additionally, we gauged the linguistic correctness of the generated narratives using the CAMelBERT GED model [73], revealing an impressive average score of 98.78%.

In accordance with the methodology proposed by Yao et al. [31], our analysis included the examination of repetition within the generated stories. We assessed both inter-story and intra-story repetition. For each sentence position i, the inter-story (r_e^i) and intra-story (r_a^i) repetition scores are computed as follows:

$$r_e^i = 1 - \frac{T\left(\sum_{j=1}^{N} s^{ji}\right)}{T_{\text{all}}\left(\sum_{j=1}^{N} s^{ji}\right)} \tag{5}$$

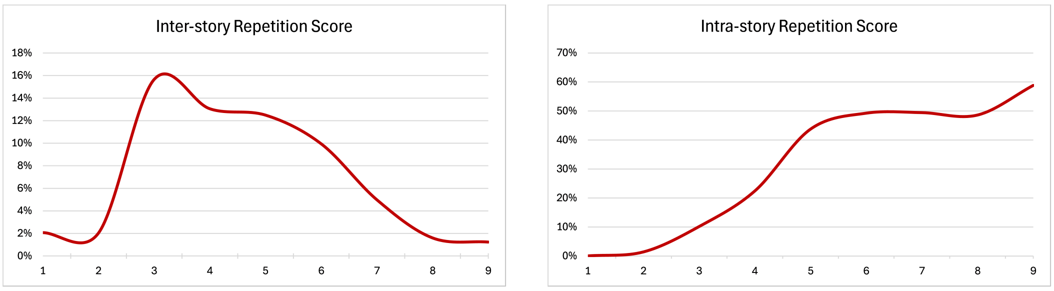
$$r_a^i = \frac{1}{N}\sum_{j=1}^{N}\left[\frac{\sum_{k=1}^{i-1} T\left(s^i \cap s^k\right)}{(i-1)\cdot T\left(s^i\right)}\right]^j \tag{6}$$

where the functions $T(\cdot)$ and $T_{\text{all}}(\cdot)$ denote the number of distinct and total trigrams, respectively. The s^{ji} stands for the i-th sentence in the j-th story; and $s^i \cap s^k$ is the distinct trigram intersection set between sentence s^i and s^k.

Inter-story repetition demonstrates the rate of repetition between stories at a given sentence position, evaluating the model's creativity in avoiding the generation of identical sentences in different stories. Intra-story repetition, on the other hand, represents the average repetition of a sentence compared to previous sentences within the same story.

Figure 3a illustrates the inter-story repetition score. The initial two sentences exhibit very low repetition, as expected, given that these sentences correspond to the input story prompt. In the third sentence, which is the first sentence generated by BLOOMZ, the repetition surges to approximately 16%. Upon scrutinizing repeated triples, we identified the dominant recurring triple as "هناك الكثير من", which translates to "there was a lot of". After the third sentence, the repetition rate consistently diminishes until it reaches negligible repetition scores.

The intra-story repetition score, depicted in Figure 3b, shows that BLOOMZ tends to repeat words within a story as the story lengthens. Furthermore, in numerous instances, BLOOMZ exhibits a tendency to persistently reiterate identical phrases, indicating a more pronounced and problematic form of repetition. Nevertheless, even powerful language models such as GPT-3 [85] can encounter challenges related to repetition. It is commonly noted that these models may unintentionally adhere to patterns, leading to the repetition of specific phrases or structures, particularly when tasked with generating lengthy text. Ongoing efforts are being made to address this issue.



(**a**) (**b**)

Figure 3. The inter- and intra-story repetition scores: (**a**) inter-sentence score; (**b**) intra-sentence score.

5.5.2. Subjective Evaluation

For the subjective evaluation, we engaged the expertise of three evaluators proficient in Arabic language. Their task was to assess the generated stories based on a set of three distinct criteria:

1. Language: covers all language characteristics, including spelling, grammar, and the correct usage of pronouns.
2. Quality: evaluates the story's quality from a human perspective, considering the presence of repetition.
3. Coherence: evaluates the logical and clear connection of ideas within the generated story.

These criteria were assessed using a Likert scale with five levels, and the average of the individual evaluators' results was calculated. Figure 4 displays the results of the subjective evaluation, indicating that the generated stories exhibit high linguistic quality and good coherence. However, their quality as literary pieces was considered to be moderate, with slightly more than half rated as excellent or very good. Only Upon discussing this weakness with the evaluators, it was noted that repetition and lack of engagement were the two factors that most negatively impacted the quality of the stories. Figure 5 illustrates several samples of the generated stories, with the final story serving as an example of a high repetition rate.

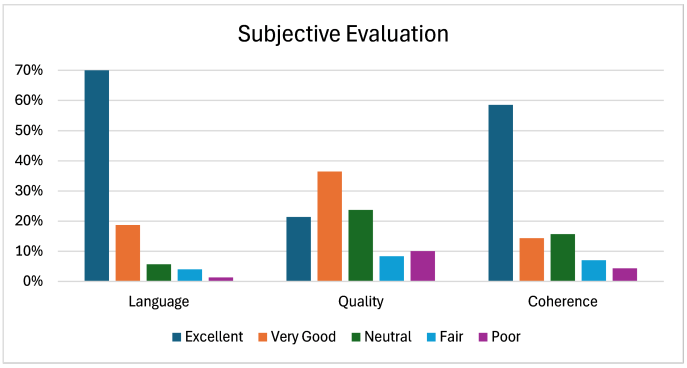
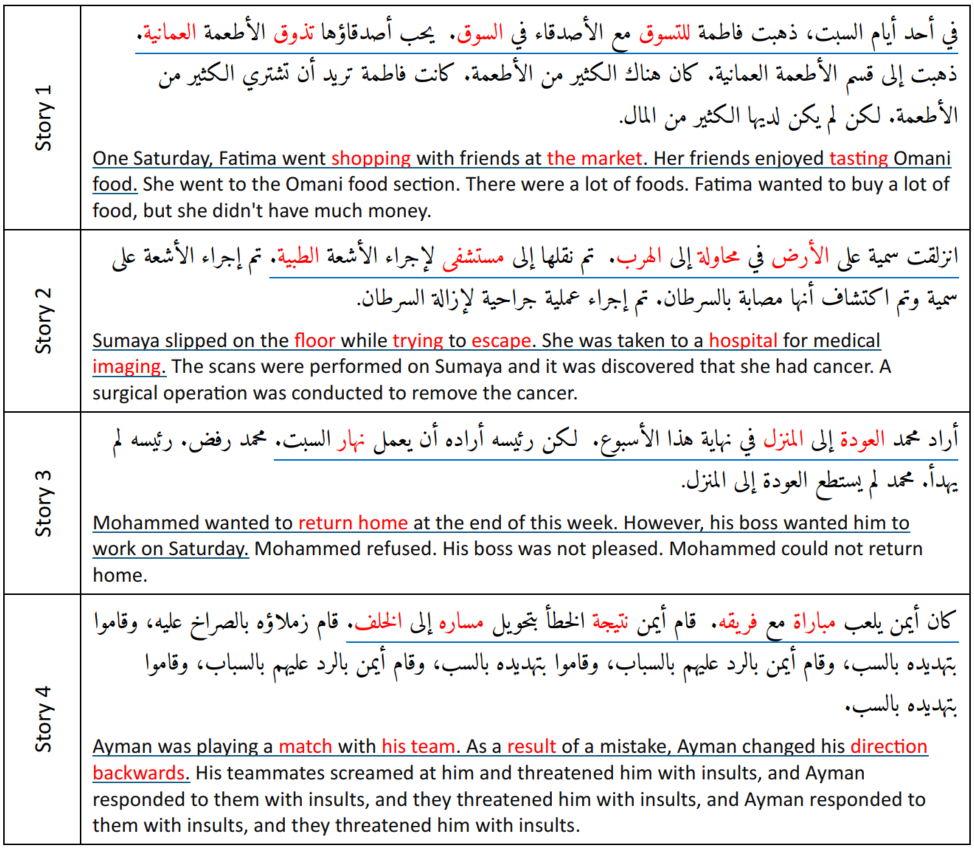


Figure 4. Results of subjective evaluation using a Likert scale with five levels ranging from "Excellent" to "Poor". Each criterion, like language, is normalized to total 100%.



Story 1	في أحد أيام السبت، ذهبت فاطمة للتسوق مع الأصدقاء في السوق. يحب أصدقاؤها تذوق الأطعمة العمانية. ذهبت إلى قسم الأطعمة العمانية. كان هناك الكثير من الأطعمة. كانت فاطمة تريد أن تشتري الكثير من الأطعمة. لكن لم يكن لديها الكثير من المال. One Saturday, Fatima went shopping with friends at the market. Her friends enjoyed tasting Omani food. She went to the Omani food section. There were a lot of foods. Fatima wanted to buy a lot of food, but she didn't have much money.
Story 2	انزلقت سمية على الأرض في محاولة إلى الهرب. تم نقلها إلى مستشفى لإجراء الأشعة الطبية. تم إجراء الأشعة على سمية وتم اكتشاف أنها مصابة بالسرطان. تم إجراء عملية جراحية لإزالة السرطان. Sumaya slipped on the floor while trying to escape. She was taken to a hospital for medical imaging. The scans were performed on Sumaya and it was discovered that she had cancer. A surgical operation was conducted to remove the cancer.
Story 3	أراد محمد العودة إلى المنزل في نهاية هذا الأسبوع. لكن رئيسه أراده أن يعمل نهار السبت. محمد رفض. رئيسه لم يهدأ. محمد لم يستطع العودة إلى المنزل. Mohammed wanted to return home at the end of this week. However, his boss wanted him to work on Saturday. Mohammed refused. His boss was not pleased. Mohammed could not return home.
Story 4	كان أيمن يلعب مباراة مع فريقه. قام أيمن نتيجة الخطأ بتحويل مساره إلى الخلف. قام زملاؤه بالصراخ عليه، وقاموا بتهديده بالسب، وقام أيمن بالرد عليهم بالسباب، وقاموا بتهديده بالسب، وقام أيمن بالرد عليهم بالسباب، وقاموا بتهديده بالسب. Ayman was playing a match with his team. As a result of a mistake, Ayman changed his direction backwards. His teammates screamed at him and threatened him with insults, and Ayman responded to them with insults, and they threatened him with insults, and Ayman responded to them with insults, and they threatened him with insults.

Figure 5. Samples of generated stories, where the underlined text represents the prompt fed into the fine-tuned BLOOMZ model, with substituted entities written in red. English translation is provided for convenience.

6. Conclusions

In conclusion, this work addresses a significant gap in Arabic Natural Language Generation (NLG) by successfully generating stories in Arabic, filling a critical void in the literature and opening new avenues for research and application in this field. This study also highlights the relatively weak predictive capabilities of Arabic large language models (LLMs) due to Arabic's complex morphology and high ambiguity, which we mitigated by post-processing the results from LLMs. Additionally, we demonstrated the capacity of LLMs for cross-lingual transfer learning in zero-shot scenarios, showcasing their versatility and potential for multilingual NLG applications, especially in low-resource settings. We introduced a novel story generation approach that creatively alters the narrative space of existing stories to produce diverse and engaging narratives suited to specific contexts and requirements. This method involves two strategies: generating new stories by infusing new entities without altering the story structure and using remodeled story spaces as prompts for LLMs to generate complete stories. The first strategy preserved the story's suspense and structure, though it restricted creativity in the narratives, while the second, despite generating a larger number of stories and exhibiting more creativity, sometimes compromised the structural integrity of the stories as noted by human evaluators.

Future research directions could delve into developing adaptive algorithms that enhance the integration of cultural contexts and idiomatic expressions specific to various Arabic dialects into the generated stories, thereby improving their authenticity and engagement. Research could also explore the development of more dynamic narrative generation methods that can respond to real-time user feedback, creating interactive and personalized storytelling experiences. Additionally, as AI-generated content continues to grow, investigating methods to mitigate biases and uphold ethical standards in automated story generation becomes paramount. Lastly, enhancing narrative coherence in longer sequences and more complex story arcs would significantly improve the quality of AI-generated texts and their acceptance by broader audiences.

Author Contributions: Conceptualization, A.A. and A.M.A.; methodology, A.A.; formal analysis, A.A.; investigation, A.A.; resources, A.M.A.; writing—original draft, A.A.; writing—review & editing, A.M.A.; supervision, A.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deputyship for Research & Innovation, "Ministry of Education" in Saudi Arabia, for funding this research work through project No. IFKSUOR3-049-4.

Data Availability Statement: The original ROCStories data presented in the study are openly available at https://cs.rochester.edu/nlp/rocstories/, accessed on 21 October 2023.

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

Abbreviations

The following abbreviations are used in this manuscript:

BPE	Byte Pair Encoding;
ComVE	Commonsense Validation Dataset;
GED	Grammatical Error Detection;
LLMs	Large Language Models;
LoRA	Low-Rank Adaptation;
MLM	Masked Language Modeling;
MSA	Modern Standard Arabic;
NER	Named Entity Recognition;
NLG	Natural Language Generation;
NLP	Natural Language Processing;
NSP	Next Sentence Prediction;
PEFT	Parameter-Efficient Fine-Tuning;
POS	Parts-of-Speech;
SEE	Story Entity Extraction;
SEG	Story End Generation;
T5	Text-To-Text Transfer Transformer.

References

1. Alhussain, A.I.; Azmi, A.M. Automatic story generation: A survey of approaches. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 103. [CrossRef]
2. Guan, J.; Huang, F.; Zhao, Z.; Zhu, X.; Huang, M. A knowledge-enhanced pretraining model for commonsense story generation. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 93–108. [CrossRef]
3. Singh, M. The evolutionary and psychological foundations of universal narrative structure. *Open Sci. Framew.* **2019**. [CrossRef]
4. Kybartas, B.; Bidarra, R. A survey on story generation techniques for authoring computational narratives. *IEEE Trans. Comput. Intell. AI Games* **2016**, *9*, 239–253. [CrossRef]

5. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
6. Anil, R.; Dai, A.M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. Palm 2 technical report. *arXiv* **2023**, arXiv:2305.10403.
7. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMa: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
8. Workshop, B.; Scao, T.L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; et al. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv* **2022**, arXiv:2211.05100.
9. Zhu, W.; Lv, Y.; Dong, Q.; Yuan, F.; Xu, J.; Huang, S.; Kong, L.; Chen, J.; Li, L. Extrapolating large language models to non-english by aligning languages. *arXiv* **2023**, arXiv:2308.04948.
10. Muennighoff, N.; Wang, T.; Sutawika, L.; Roberts, A.; Biderman, S.; Scao, T.L.; Bari, M.S.; Shen, S.; Yong, Z.X.; Schoelkopf, H.; et al. Crosslingual Generalization through Multitask Finetuning. *arXiv* **2023**, arXiv:2211.01786.
11. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. *arXiv* **2021**, arXiv:2106.09685.
12. Chergui, N.H.; Mbarek, M.S.A. AQG: Arabic Question Generator. *Rev. D'Intelligence Artif* **2020**, *34*, 721–729.
13. Alhashedi, S.; Suaib, N.M.; Bakri, A. Arabic Automatic Question Generation Using Transformer Model. Technical Report. *EasyChair*. 2022. Available online: https://easychair.org/publications/preprint/tzZ2 (accessed on 26 March 2024).
14. Shamas, M.; El Hajj, W.; Hajj, H.; Shaban, K. Metadial: A Meta-learning Approach for Arabic Dialogue Generation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**, *22*, 172. [CrossRef]
15. Attai, A.; Elnagar, A. A survey on Arabic Image Captioning Systems Using Deep Learning Models. In Proceedings of the 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 17–18 November 2020; pp. 114–119.
16. Azmi, A.M.; Alsaiari, A. A calligraphic based scheme to justify Arabic text improving readability and comprehension. *Comput. Hum. Behav.* **2014**, *39*, 177–186. [CrossRef]
17. Azmi, A.M.; Aljafari, E.A. Universal web accessibility and the challenge to integrate informal Arabic users: A case study. *Univers. Access Inf. Soc.* **2018**, *17*, 131–145. [CrossRef]
18. Mannaa, Z.M.; Azmi, A.M.; Aboalsamh, H.A. Computer-assisted i'raab of Arabic sentences for teaching grammar to students. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *10*, 8909–8926. [CrossRef]
19. Azmi, A.M.; Alnefaie, R.M.; Aboalsamh, H.A. Light diacritic restoration to disambiguate homographs in modern Arabic texts. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *21*, 60. [CrossRef]
20. Azmi, A.M.; Aljafari, E.A. Modern information retrieval in Arabic–catering to standard and colloquial Arabic users. *J. Inf. Sci.* **2015**, *41*, 506–517. [CrossRef]
21. Al-Thanyyan, S.S.; Azmi, A.M. Simplification of Arabic text: A hybrid approach integrating machine translation and transformer-based lexical model. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101662. [CrossRef]
22. Almuzaini, H.A.; Azmi, A.M. TaSbeeb: A judicial decision support system based on deep learning framework. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101695. [CrossRef]
23. Alwaneen, T.H.; Azmi, A.M. Stacked dynamic memory-coattention network for answering why-questions in Arabic. *Neural Comput. Appl.* **2024**, *36*, 8867–8883. [CrossRef]
24. See, A.; Pappu, A.; Saxena, R.; Yerukola, A.; Manning, C.D. Do Massively Pretrained Language Models Make Better Storytellers? In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, 3–4 November 2019; pp. 843–861.
25. Jain, P.; Agrawal, P.; Mishra, A.; Sukhwani, M.; Laha, A.; Sankaranarayanan, K. Story generation from sequence of independent short descriptions. In Proceedings of the SIGKDD Workshop on Machine Learning for Creativity (ML4Creativity), Halifax, NS, Canada, 14 August 2017.
26. Chen, G.; Liu, Y.; Luan, H.; Zhang, M.; Liu, Q.; Sun, M. Learning to generate explainable plots for neural story generation. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **2020**, *29*, 585–593. [CrossRef]
27. Rashkin, H.; Celikyilmaz, A.; Choi, Y.; Gao, J. PlotMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 4274–4295.
28. Fan, A.; Lewis, M.; Dauphin, Y. Hierarchical Neural Story Generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018.
29. Clark, E.; Ji, Y.; Smith, N.A. Neural text generation in stories using entity representations as context. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 2250–2260.
30. Ippolito, D.; Grangier, D.; Callison-Burch, C.; Eck, D. Unsupervised hierarchical story infilling. In Proceedings of the First Workshop on Narrative Understanding, Minneapolis, MI, USA, 7 June 2019; pp. 37–43.
31. Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; Yan, R. Plan-and-write: Towards better automatic storytelling. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7378–7385.

32. Liu, Y.; Huang, Q.; Li, J.; Mo, L.; Cai, Y.; Li, Q. SSAP: Storylines and Sentiment Aware Pre-Trained Model for Story Ending Generation. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **2022**, *30*, 686–694. [CrossRef]
33. Liu, H.; Singh, P. ConceptNet—A practical commonsense reasoning tool-kit. *BT Technol. J.* **2004**, *22*, 211–226. [CrossRef]
34. Sap, M.; Le Bras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N.A.; Choi, Y. Atomic: An atlas of machine commonsense for if-then reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3027–3035.
35. Lin, L.; Cao, Y.; Huang, L.; Li, S.; Hu, X.; Wen, L.; Wang, J. What makes the story forward? inferring commonsense explanations as prompts for future event generation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid Spain, 11–15 July 2022; pp. 1098–1109.
36. Xu, P.; Patwary, M.; Shoeybi, M.; Puri, R.; Fung, P.; Anandkumar, A.; Catanzaro, B. MEGATRON-CNTRL: Controllable Story Generation with External Knowledge Using Large-Scale Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 2831–2845.
37. Peng, X.; Li, S.; Wiegreffe, S.; Riedl, M. Inferring the Reader: Guiding Automated Story Generation with Commonsense Reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 7008–7029.
38. Ammanabrolu, P.; Cheung, W.; Broniec, W.; Riedl, M.O. Automated storytelling via causal, commonsense plot ordering. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 5859–5867.
39. Vijjini, A.R.; Brahman, F.; Chaturvedi, S. Towards Inter-character Relationship-driven Story Generation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 8970–8987.
40. Xie, Z.; Lau, J.H.; Cohn, T. Exploring Story Generation with Multi-task Objectives in Variational Autoencoders. In Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association, Online, 8–10 December 2021; pp. 97–106.
41. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA; pp. 4171–4186. [CrossRef]
42. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
43. Huang, Q.; Mo, L.; Li, P.; Cai, Y.; Liu, Q.; Wei, J.; Li, Q.; Leung, H.f. Story ending generation with multi-level graph convolutional networks over dependency trees. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 13073–13081.
44. Wang, J.; Zou, B.; Li, Z.; Qu, J.; Zhao, P.; Liu, A.; Zhao, L. Incorporating commonsense knowledge into story ending generation via heterogeneous graph networks. In *Proceedings of the International Conference on Database Systems for Advanced Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 85–100.
45. Alhussain, A.; Azmi, A.M. Crosslingual Transfer Learning for Arabic Story Ending Generation. *Indones. J. Comput. Sci.* **2024**, *13*, 1564–1574. [CrossRef]
46. Clark, E.; August, T.; Serrano, S.; Haduong, N.; Gururangan, S.; Smith, N.A. All that's "human" is not gold: Evaluating human evaluation of generated text. *arXiv* **2021**, arXiv:2107.00061.
47. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
48. Xie, Z.; Cohn, T.; Lau, J.H. The Next Chapter: A Study of Large Language Models in Storytelling. In Proceedings of the 16th International Natural Language Generation Conference, Prague, Czechia, 11–15 September 2023; pp. 323–351.
49. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst.* **2024**, *36*. [CrossRef]
50. Wen, Z.; Tian, Z.; Wu, W.; Yang, Y.; Shi, Y.; Huang, Z.; Li, D. GROVE: A Retrieval-augmented Complex Story Generation Framework with A Forest of Evidence. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; pp. 3980–3998.
51. Yang, K.; Tian, Y.; Peng, N.; Klein, D. Re3: Generating Longer Stories with Recursive Reprompting and Revision. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 4393–4479.
52. Yang, K.; Klein, D.; Peng, N.; Tian, Y. DOC: Improving Long Story Coherence with Detailed Outline Control. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 3378–3465.
53. Chung, J.J.Y.; Kim, W.; Yoo, K.M.; Lee, H.; Adar, E.; Chang, M. TaleBrush: Sketching stories with generative pretrained language models. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, Orleans, LA, USA, 29 April–5 May 2022; pp. 1–19.
54. Yuan, A.; Coenen, A.; Reif, E.; Ippolito, D. Wordcraft: Story writing with large language models. In Proceedings of the 27th International Conference on Intelligent User Interfaces, Helsinki, Finland, 22–25 March 2022; pp. 841–852.

55. Wan, Q.; Hu, S.; Zhang, Y.; Wang, P.; Wen, B.; Lu, Z. "It Felt Like Having a Second Mind": Investigating Human-AI Co-creativity in Prewriting with Large Language Models. *arXiv* **2023**, arXiv:2307.10811.
56. Davis, C.W.; Jetter, A.J.; Giabbanelli, P.J. Automatically generating scenarios from a text corpus: A case study on electric vehicles. *Sustainability* **2022**, *14*, 7938. [CrossRef]
57. Sajjad, H.; Durrani, N.; Dalvi, F.; Alam, F.; Khan, A.R.; Xu, J. Analyzing Encoded Concepts in Transformer Language Models. In Proceedings of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies (NAACL), NAACL '22, Online, 10–15 July 2022.
58. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144.
59. Srihari, N.; Mehra, D.; Huang, M.; Varshney, T.; Sawarkar, A.; Onofrio, D. Relation Extraction and Entity Extraction in Text Using NLP. 2021. Available online: https://nikhilsrihari-nik.medium.com/identifying-entities-and-their-relations-in-text-76efa8c18194 (accessed on 21 October 2023).
60. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
61. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 11–16 May 2020; pp. 9–15.
62. Inoue, G.; Alhafni, B.; Baimukan, N.; Bouamor, H.; Habash, N. The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. In Proceedings of the Sixth Arabic Natural Language Processing Workshop; Association for Computational Linguistics: Kyiv, Ukraine (Virtual), 9 April 2021.
63. Nagoudi, E.M.B.; Elmadany, A.; Mageed, M.A.M. AraT5: Text-to-text transformers for Arabic language generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 628–647.
64. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, É.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 8440–8451.
65. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
66. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document transformer. *arXiv* **2020**, arXiv:2004.05150.
67. Zhou, X.; Zhang, Y.; Cui, L.; Huang, D. Evaluating commonsense in pre-trained language models. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 9733–9740.
68. Tamborrino, A.; Pellicanò, N.; Pannier, B.; Voitot, P.; Naudin, L. Pre-training Is (Almost) All You Need: An Application to Commonsense Reasoning. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3878–3887.
69. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
70. Zoph, B.; Bello, I.; Kumar, S.; Du, N.; Huang, Y.; Dean, J.; Shazeer, N.; Fedus, W. St-moe: Designing stable and transferable sparse expert models. *arXiv* **2022**, arXiv:2202.08906.
71. Al-Bashabsheh, E.; Al-Khazaleh, H.; Elayan, O.; Duwairi, R. Commonsense Validation for Arabic Sentences using Deep Learning. In Proceedings of the 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, 21–23 December 2021; pp. 1–7.
72. Lin, B.Y.; Lee, S.; Qiao, X.; Ren, X. Common Sense Beyond English: Evaluating and Improving Multilingual Language Models for Commonsense Reasoning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 1274–1287.
73. Alhafni, B.; Inoue, G.; Khairallah, C.; Habash, N. Advancements in Arabic Grammatical Error Detection and Correction: An Empirical Investigation. *arXiv* **2023**, arXiv:2305.14734.
74. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2790–2799.
75. Lin, Z.; Madotto, A.; Fung, P. Exploring Versatile Generative Language Model Via Parameter-Efficient Transfer Learning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 441–459.
76. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 4582–4597.
77. Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; Allen, J. A corpus and cloze evaluation for deeper understanding of commonsense stories. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 839–849.

78. Tawalbeh, S.; AL-Smadi, M. Is this sentence valid? An Arabic Dataset for Commonsense Validation. *arXiv* **2020**, arXiv:2008.10873.
79. Wang, C.; Liang, S.; Jin, Y.; Wang, Y.; Zhu, X.; Zhang, Y. SemEval-2020 Task 4: Commonsense Validation and Explanation. In Proceedings of the Proceedings of The 14th International Workshop on Semantic Evaluation, Online, 12–13 December 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020.
80. Roemmele, M.; Gordon, A.S.; Swanson, R. Evaluating story generation systems using automated linguistic analyses. In Proceedings of the SIGKDD 2017 Workshop on Machine Learning for Creativity, Halifax, NS, Canada, 13–17 August 2017.
81. Purdy, C.; Wang, X.; He, L.; Riedl, M. Predicting Generated Story Quality with Quantitative Measures. In Proceedings of the AIIDE, Edmonton, AL, Canada, 13–17 November 2018; pp. 95–101.
82. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Hong Kong, China, 3–7 November 2019.
83. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Pre-Training Transformers as Energy-Based Cloze Models. In Proceedings of the EMNLP, Online, 16–20 November 2020.
84. Antoun, W.; Baly, F.; Hajj, H. AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kyiv, Ukraine (Virtual), 9 April 2021; pp. 191–195.
85. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.