Arabic Automatic Story Generation with Large Language Models

Ahmed Oumar El-Shangiti[ξ] **Fakhraddin Alwajih**[λ]
Muhammad Abdul-Mageed[λ,ξ,ϕ]
[λ] The University of British Columbia
[ξ] Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
[ϕ] Invertible AI
ahmed.oumar@mbzuai.ac.ae, muhammad.mageed@ubc.ca

Abstract

Large language models (LLMs) have recently emerged as a powerful tool for a wide range of language generation tasks. Nevertheless, this progress has been slower in Arabic. In this work, we focus on the task of generating stories from LLMs. For our training, we use stories acquired through machine translation (MT) as well as GPT-4. For the MT data, we develop a careful pipeline that ensures we acquire high-quality stories. For our GPT-4[1] data, we introduce crafted prompts that allow us to generate data well-suited to the Arabic context in both Modern Standard Arabic (MSA) and two Arabic dialects (Egyptian and Moroccan). For example, we generate stories tailored to various Arab countries on a wide host of topics. Our manual evaluation shows that our model fine-tuned on these training datasets can generate coherent stories that adhere to our instructions. We also conduct an extensive automatic and human evaluation comparing our models against state-of-the-art proprietary and open-source models. Our datasets and models will be made publicly available at https://github.com/UBC-NLP/arastories.

1 Introduction

Storytelling is an essential human skill that serves to transmit knowledge, impart values, and connect individuals through tales of daily experiences. It is utilized in education, where teachers harness children's natural affinity for stories to foster cognitive and literacy development. Additionally, stories and legends, viewed as cultural heritage, are passed down through generations by parents, enriching the culture and preserving traditions.

The role of storytelling also extends beyond its traditional roots; it acts as a vital connection between the primitive oral language skills in early childhood and the advanced language abilities associated with literacy. As such, the task of automatic story generation presents numerous benefits across different fields. In entertainment, it allows for the efficient creation of diverse narratives (Xie and Riedl, 2024). In education, tailored stories can be crafted to address the unique needs of learners. In gaming, interactive storytelling significantly enhances user engagement and enjoyment (Patel et al., 2024). And these are only a few application domains.

Progress in natural language processing (NLP) technologies, particularly with large language models (LLMs) such as GPT-4 and Gemini, has made automatic story generation both viable and effective, producing stories with notable fluency and coherence. While substantial efforts have been made to advance automatic story generation in English using generative models, the development of such technologies for Arabic has been limited due to a scarcity of Arabic short story data and minimal focus from the research community.

In this study, we present a novel approach to automatic story generation utilizing the powerful Arabic LLM, AraLLaMA (Alwajih et al., 2024). We enhance AraLLaMA through fine-tuning with both translated and synthetic datasets to optimize its story-generating capabilities. We explore two fine-tuning strategies: one involving direct application of a synthetic dataset produced by GPT-4, and another beginning with an analogous synthetic dataset translated from English. Additionally, we extended the model's utility by fine-tuning it with data from two Arabic dialects, enabling the generation of stories in both Modern Standard Arabic (MSA) and these two dialects. The efficacy of our model is assessed through human evaluation, which confirmed its ability to produce coherent and fluent narratives as per specified instructions.

Our contributions are manifold, summarized as follows:

[1] GPT-4 refers to GPT-4-0125-preview

Proceedings of The Second Arabic Natural Language Processing Conference, pages 140–152

1. We introduce powerful models capable of generating coherent and fluent stories in MSA and two Arabic dialects.

2. We offer a newly created framework for Arabic automatic story evaluation based on LLMs.

3. We develop two novel datasets for automatic story generation: one consisting of translated narratives from the TinyStories (Eldan and Li, 2023) dataset, which was meticulously curated, and another comprising a synthetic dataset created using GPT-4, featuring narratives in MSA and two dialects.

4. We compare two distinct fine-tuning methods on AraLLaMA against AceGPT-7B (Huang et al., 2024), GPT-3.5, and Command-R[2], powerful open source and proprietary models using extensive automatic and human evaluations.

The remainder of this paper is organized as follows: Section 2 provides a review of prior studies focusing on the task of automatic story generation. Section 3 details the creation of our datasets. In Section 4, we outline our prompt design. In section 5 we detail our different experiments. Results and key insights from our comparative analysis of our fine-tuned models against various commercial and open-source models are discussed in Section 6. The paper concludes with Section 7.

2 Related Work

2.1 Early Work on Story Generation.

Jain et al. (2017) is an early work on generating coherent stories, experimenting with two paradigms: Statistical Machine Translation (SMT) and Deep Learning. SMT treats story generation as a translation task, while Deep Learning uses Recurrent Neural Networks (RNNs) to encode sequences of input descriptions into hidden representations, which are then transformed into detailed summaries. They evaluate their models using BLEU, ROUGE-L, and human evaluation. Fan et al. (2018) propose a hierarchical model that first generates a story premise using a convolutional language model (Dauphin et al., 2017) and then a seq2seq model to create a story that follows the premise. They incorporate gated multi-scale attention and model fusion to improve prompt adherence.

Akoury et al. (2020) introduce the STORIUM dataset[3] and fine-tune GPT-2-medium (Radford et al., 2019) for generating short story scene entries, motivated by GPT-2's 1024-token context window. Plug-and-Blend (Lin and Riedl, 2021) consists of a Blending Generative Model (BGM) and Planner for controllable story generation. BGM facilitates controlled continuations, while Planner specifies control parameters based on topic descriptions and story sections. The authors fine-tune GPT-2-large (Radford et al., 2019) on ROCStories (Mostafazadeh et al., 2016) and use pre-trained GeDi (Krause et al., 2020) as the guiding model, evaluating fluency and fidelity through human evaluation.

2.2 LLM Story Generation.

Mirowski et al. (2022) propose using a 70B Chinchilla LLM called Dramatron for generating long narratives, such as full scripts and screenplays, through prompting, prompt chaining, and hierarchical generation. Dramatron supports collaborative writing and was qualitatively assessed via co-writing sessions and interviews with 15 industry professionals.

Yang et al. (2022) propose the Recursive Reprompting and Revision (Re3) framework automatically generates longer stories without human intervention, distinguishing it from previous approaches. Re3 comprises four modules: Plan, Draft, Rewrite, and Edit. The Plan module creates a story plan using GPT-3 (Brown et al., 2020) to add details to a given premise. The Draft module generates story continuations by recursively prompting GPT-3, dynamically updating the prompt with information from the plan and story. The Rewrite module reranks alternate continuations to select the best ones, and the Edit module ensures factual consistency with earlier parts of the story. Re3 operates in a zero-shot manner, allowing it to generate longer stories without domain constraints.

Patel et al. (2024) propose a creative storytelling framework with two components: the story generation model and the Action Discriminator model (AD LLM). They train these models in a feedback loop called SWAG. Initially, a prompt is used to generate the first paragraph, which is then fed into the AD LLM with actions (e.g., "add suspense") to produce the best continuation. This process is repeated until the story reaches the desired length.

[2] https://dashboard.cohere.com/playground/chat?

[3] https://storium.com/

The model is trained using Direct Preference Optimization (DPO) (Rafailov et al., 2023), with preference data generated by GPT-4 (OpenAI et al., 2024) and Mixtral-8×7B (Jiang et al., 2024). GPT-4 samples are chosen, while Mixtral-8×7B samples are rejected. Evaluation is conducted using both human and GPT-4 assessments.

Xie and Riedl (2024) introduces a method for generating suspenseful stories with LLMs using iterative prompting based on psychological and narratological theories of suspense. This zero-shot approach does not require pre-existing story corpora. Human evaluations demonstrate the effectiveness of this technique in crafting engaging suspenseful stories, and controlled studies explore factors influencing readers' perception of suspense.

Radwan et al. (2024) introduces SARD, a tool with a visual drag-and-drop interface for creating multi-chapter stories using advanced large language models. Wordcraft (Yuan et al., 2022) is a web application for story writing that combines a text editor with controls for prompting an LLM to perform various story-generation tasks.

2.3 Evaluation of Story Generation in Literature

There are basically two types of evaluations for story generation in the literature: human evaluation and automatic evaluation. We explore these evaluation methods in the following subsections:

2.3.1 Human Evaluation

Akoury et al. (2020) integrate their fine-tuned model into the STORIUM collaborative storytelling platform, where real authors can query the model to generate suggested story continuations. The authors could edit the generated text by adding or deleting content. The edited stories were collected along with ratings from the authors' on properties such as relevance, fluency, coherence, and likability. They also propose a new automatic metric called User Story Edit Ratings (USER), inspired by the longest common subsequence (LCS) of the ROUGE metric (Lin, 2004), which measures how much of the generated text is preserved in the edited version.

The authors of Re3 (Yang et al., 2022) ask workers from Amazon Mechanical Turk to rate Re3-generated stories against GPT-3 and GPT-3 fine-tune on stories from the WritingPrompts dataset. The evaluation criteria include interestingness, coherence, fluency, human-likeness, and relevance. Workers also identify shortcomings in the generated stories, such as disfluency, repetitiveness, confusing inconsistencies, and narration problems. Re3 outperforms all baselines on almost all criteria.

Xie and Riedl (2024) rely on a pool of three human studies to evaluate their framework for suspenseful story generation. In the first study, human judges compare stories generated by their approach against those generated by a strong baseline (ChatGPT) based on suspense, novelty, enjoyment, logical sense, and naturalness. The second study involved ablations on their system compared against the full system. In the third phase, participants reviewed the story's structure to verify internal processes. Ninety participants assessed 30 story pairs, with each pair reviewed by 30 participants. Their approach outperforms all baselines on all criteria except for a 56% tie with ChatGPT on naturalness.

The authors of (Patel et al., 2024) evaluate their story generation method using human judges and GPT-4. Surge AI employees assessed 50 stories generated by the Llama-2-7B and Mistral-7B models, enhanced by the SWAG technique, against four baselines: the end-to-end approach, a random selection method, GPT-3.5-turbo, and GPT-4-turbo. Evaluations focused on interestingness, surprise, and coherence. The findings show a preference for SWAG-enhanced stories over conventional methods by both human judges and GPT-4, with SWAG models winning 61.5

Wang et al. (2024) compare Weaver's variations against other open-source and proprietary LMs, including GPT-4, GLM-4, ERNIE-Bot-4.0, and Gemini-pro. Evaluations by human professionals and GPT-4 were based on creativity, style, relevance, and fluency. Weaver-Ultra was preferred 1576 and 1657 times out of 3540 samples by humans and GPT-4, respectively.

2.3.2 Automatic Evaluation

Evaluating creative writing such as story generation is a challenging task. Jain et al. (2017), one of the earlier works on neural-based story generation, uses machine translation metrics (BLEU-4, METEOR, TER, and ROUGE-L) to evaluate story generation. The overall results were low, with SMT-based methods scoring better on BLEU-4 than seq2seq models, despite being less coherent. The scores were 3.5 and 1.98 for SMT and seq2seq models, respectively, indicating that n-gram-based metrics are not suitable for creative writing judgment. GPT-Eval, an evaluation framework based

on GPT-4 (Eldan and Li, 2023), takes in a story and provides a general assessment and a score out of 10 in four criteria: grammar, creativity, consistency, and age group.

2.4 Common Datasets for Story Generation

To provide an overview of the resources used in story generation research, we summarize the most common datasets used to build automatic systems for generating stories in Table 1. These datasets vary in size, nature, availability, and average length of the stories they contain. The datasets include human-generated stories as well as those created by advanced language models like GPT-3.5 and GPT-4. They are valuable resources for training and evaluating story-generation models.

2.5 Arabic Story Generation

The task of automatic story generation is uninvestigated in the Arabic NLP community. However, (Alhussain and Azmi, 2024) utilize cross-lingual transfer learning to address the scarcity of Arabic data in Story Ending Generation (SEG) task by leveraging English story corpora.

3 Data Collection

We compile data from different resources. We first translate 1.13M English stories generated by GPT-4 alongside their prompts from the *TinyStories* dataset (Eldan and Li, 2023) using Google translate API.[4] To ensure that we have only high-quality translation, we apply a filtering strategy based on multilingual sentence embeddings (Feng et al., 2022) and remove the story pairs whose embedding similarity is less than 92%.

3.1 Filtering Strategy

With the aim to train only on high-quality data, we apply Algorithm 1 to our dataset. The final threshold was 92%. And we were able to maintain 545K samples which represents 48.3% of the translated data.

3.2 Generated Data

We also generate our own stories from GPT-4-Turbo API using a carefully designed set of prompts and features (see Section 4). The dataset generated with our prompt template is in three Arabic varieties, namely MSA, Moroccan, and Egyptian. We tested the ability of GPT-4-Turbo to generate other Arabic dialects, but the generated content was mostly MSA. For this reason, we decided to limit our work to the above-mentioned varieties. We generate $1,000$ stories for each variety, making a total of $3,000$ stories. We also create 20 additional prompts for evaluation. We provide an example of stories generated by GPT-4-Turbo using our custom prompt template in Figure A.1.

[4] translate.googleapis.com

Algorithm 1 Filtering Stories Based on Similarity Score

Require: Stories dataset D, Similarity threshold t, Minimum word count $m = 50$

1: Remove stories shorter than m words
2: Sort stories based on similarity score
3: Filter out stories whose similarity with the original story is less than the threshold t
4: Get a human in the loop to manually check some random samples
5: Set a new threshold t'
6: Repeat steps 2-5 until satisfactory translation quality is achieved

4 Prompt Design

Prompting is an approach employed by users to interface with LLMs (White et al., 2023). It functions as the primary mode of communication with these models, effectively serving as the input language that LLMs are designed to interpret and respond to. The efficacy of the generated output is significantly correlated with the quality and structure of the input prompt. This relationship underscores the critical role that prompt engineering plays in optimizing LLM performance and output relevance. In our context, the prompt can be conceptualized as a set of instructions or parameters that guide the LLM's for the Arabic story generation process. The prompt's composition, including its specificity, clarity, features, and relevance to the desired output, directly influences the model's ability to generate appropriate and accurate stories that adhere to our instructions. This causal relationship between prompt quality and output quality highlights the importance of developing sophisticated prompting strategies to fully leverage the capabilities of LLMs for Arabic story generation.

4.1 Initial Investigation

When we started this study, we had three prompting choices. We either prompt in English, Arabic, or

Dataset name	Size	Nature	Available	Avg Length
Huang et al. (2016)	$41K$	Human	No	21.2 Tokens
ROCStories	$50K$	Human	Yes	5 sentences
WritingPrompts	$300K$	Human	Yes	734.5 Words
STORIUM	$6K$	Human	Yes	$19K$ Tokens
TinyStories	$5M$	GPT-4/GPT-3.5	Yes	
Weaver (Wang et al., 2024)	$500K$	GTP-4/GPT-3.5	No	—
SWAG (Patel et al., 2024)	$20K$	GPT-4/Llama-2-7B/Mistral-2-7B	No	$5K$ Words

Table 1: Description of the stories datasets

dialect for dialectal stories. Recent studies have shown that LLMs perform better when prompted in English compared to other languages (Etxaniz et al., 2023; Kadaoui et al., 2023). However, we still wanted to test these claims and verify if they hold in our case. For this reason, we query GPT-4 with prompts in English and Arabic respectively and compared the corresponding generated stories. In this exploratory stage, we manually looked into each story and checked the creativity, fluency, and instruction following of the model. This pilot study did not reveal any critical differences between both languages in terms of fluency and creativity of the Arabic-generated stories.

For further investigation, we carry out the same experiment comparing MSA prompts versus dialectal prompts for dialectal story generation. This time, we ran 10 samples to AlDi (Keleg et al., 2023) (5 generated with an MSA prompts while the other 5 are generated with dialectal prompts) to quantify the level of dialectness of each generated story. The scores were 81.84% and 81.6% for the percentage of dialectness of the content generated with MSA and dialectal prompts respectively. Given that the difference are not significant, we decided to create prompts in each Arabic variety. This choice is mainly motivated by our intent to make the story and the prompt uniform as well as making it easier for the user to write their prompt directly in the chosen variety without a need to be fluent in MSA. Next, we describe the details of our prompt template.

4.2 Prompt Template

We design our prompt template with two goals in mind. These are (i) to ensure high quality of the generated output and (ii) make the generated output as diverse as possible. To ensure the variety of the generated stories, we carefully design a set of 12 features: *{age, place, end of story, dialogue, number of characters, moral of the story, topic, country, season, activity, emotion, plot twist}*. Our template is designed in such a way that each feature has a probability p of appearance in a particular prompt. Meaning some features might be present in a prompt while others are not. Except for the following features where they appear in each prompt: *age, number of characters*, and *country*. Based on our preliminary observations, GPT-4 is able to generate coherent stories from dialectal prompts (i.e., Egyptian and Moroccan). Hence, we ask two native speakers to translate our prompt template, originally written in MSA to Egyptian and Moroccan dialects. This prompt template is used to generate MSA and dialectal stories from GPT-4 and later to fine-tune our models.

5 Experiments

We conduct two sets of experiments:

1. Directly fine-tuning on the generated data from GPT-4 Turbo-preview.

2. Fine-tuning on translated data, followed by further fine-tuning on data generated with the GPT-4-Turbo-preview model.

The details of each experiment are described next.

5.1 Supervised Fine-Tuning (SFT)

We instruct fine-tune AraLLaMa-2-base (Alwajih et al., 2024) using a diverse Arabic instruction tuning dataset generated with our custom prompt template. AraLLaMa-2-base is a 7B parameter model based on Llama-2 (Touvron et al., 2023), continually pre-trained on Arabic data. AraLLaMa-2 has shown superior performance compared to other Arabic LLMs such as AceGPT-7B (Huang et al., 2024) and Jais-7B (Sengupta et al., 2023), hence we adopt it for our experiments. For computational

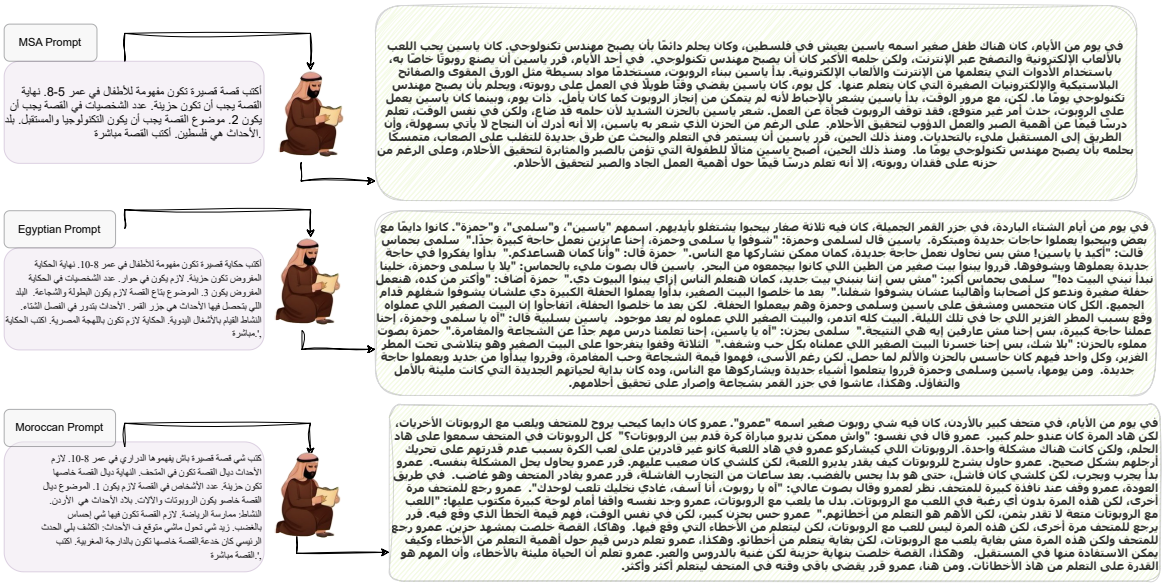

Figure 1: Samples of different stories generated with our models for the three Arabic Varieties.

efficiency, all our models are trained with Huggingface PEFT library (Mangrulkar et al., 2022). In each experiment, our base model, AraLLaMa-2 is quantized in 4-bit precision and then a new QLoRA layer (Dettmers et al., 2023) is added. During our experiments, we keep the base model frozen and update the QLoRA layer only. The instruction fine-tuning is performed by updating a newly added QLoRA layer, with α set to 16, r set to 64, QLoRA layer dimension set to 64, gradient accumulation at 10, batch size equal 1. We use as optimizer AdamW (Loshchilov and Hutter, 2019), a dropout is equal to 10%, a learning rate set to $4 * 10e - 5$. We train for 20 epochs. This training took approximately 5.5 hours on a single Nvidia A100 GPU. We call the model artifact resulting from this training **Model A**.

5.2 Two-Step Fine-Tuning

This experiment is divided into two steps. First, we instruct fine-tune AraLLaMa-2-base on large-scale translated data from the *TinyStories* dataset (Eldan and Li, 2023) for $15,000$ steps. The second step is taking the trained model from the previous step and further instruct fine-tuning it on a smaller dataset from experiment 1 (Section 5.1). The hyperparameters are the same as in the previous experiment. The overall training took about 17.5 hours on the Nvidia A100 GPU. We provide examples of stories generated with our models across all three Arabic varieties in Figure 1. We call the model artifact resulting from this training **Model B**.

6 Evaluation

You are an expert in Arabic language, its dialects, and storytelling. I would like your help in evaluating a story written by a student based on a set of instructions. You are expected to give a score out of five based on the following features:
Fluency: How smooth and natural the text is, including appropriate grammar, vocabulary, and sentence structure.
Coherence: The logical connection and flow of sentences and ideas, making the text easy to understand.
Following Instructions: How well the text adheres to the provided instructions or task requirements.
Consistency: How consistently accurate and uniform the information and style are throughout the text.
Variety: How well does the model generate story in the required Arabic variety. Give the scores directly without explanations or additions. I will first give you the instructions on which the story was based, followed by the story written by the student. Remember, I want the evaluation directly without explanation.

Figure 2: The prompt we pass to GPT-4-Turbo to evaluate stories generated with different models.

Evaluating generative tasks remains an open problem in the AI community. However, in our study, we follow previous works such as (Eldan and Li, 2023) in adapting GPT-4 as an evaluator for model performance. We also conduct an extensive human evaluation trying to understand different model capabilities and how the evaluation of GPT-4 compares to that of human judges. We next describe our two evaluation strategies.

Dialect	Model	Model Size	Fluency	Coherence	Ins Following	Consistency	Variety
	Model A	7B	4.0	3.94	4.21	4.0	3.18
	Model B	7B	3.94	4.0	4.0	4.12	3.15
	GPT-3.5	Unk	3.95	3.9	4.16	4.05	3.66
	Command-R	35B	4.05	4.16	4.22	3.88	3.46
	AceGPT-Chat	7B	3.94	4.00	3.89	3.95	3.33
	Model A	7B	3.55	3.65	3.45	3.40	2.30
	Model B	7B	3.75	3.75	3.60	3.65	2.30
	GPT-3.5	Unk	3.74	3.95	3.48	3.63	2.52
	Command-R	35B	3.68	3.78	4.31	3.73	2.84
	AceGPT-Chat	7B	3.72	3.77	3.61	3.72	1.55
	Model A	7B	3.65	3.70	4.15	3.55	2.60
	Model B	7B	3.79	3.84	4.10	3.73	2.63
	GPT-3.5	Unk	3.83	4.0	4.0	3.72	2.66
	Command-R	35B	3.94	3.94	4.4	3.72	3.27
	AceGPT-Chat	7B	3.8	3.9	3.85	3.9	2.45

Table 2: Results of our Two Models Across three Arabic varieties scored by GPT-4. **Model A** is AraLLaMa-base instruction fine-tuned on data generated from GPT-4-Turbo. **Model B** is AraLLaMa-base fine-tuned on the translated data then on the data generated from GPT-4-Turbo. The scores are on a scale of five points. The best scores are highlighted in green. The size denotes the model size in billions, GPT-3.5 size is unknown.

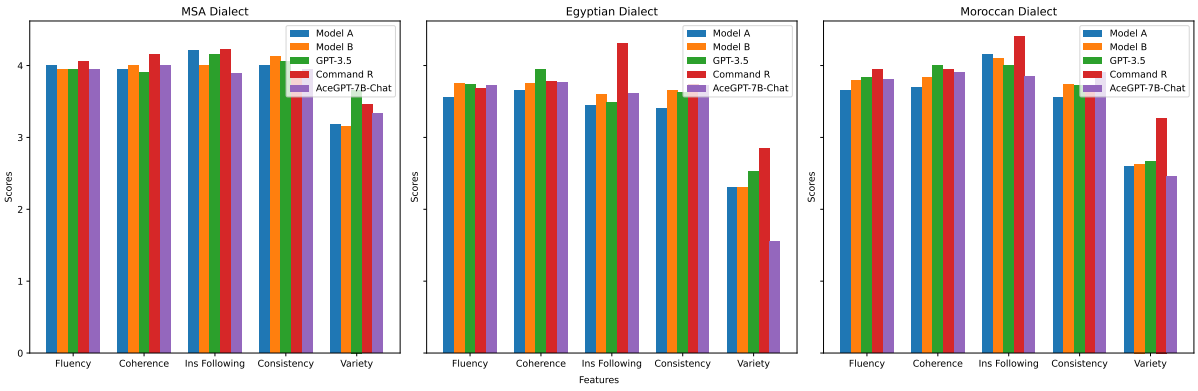

Figure 3: Models performance across the MSA, Egyptian, and Moroccan varieties.

6.1 GPT-4 As a Judge

We evaluate our models on five criteria scored by GPT-4. We design a comprehensive prompt that works as follows: given the original story prompt plus the corresponding generated story, we ask GPT-4 to act as an Arabic language expert and assign a score out of five on the following criteria:

- **Fluency:** The degree to which the text reads smoothly and naturally, with appropriate grammar, vocabulary, and sentence structure.
- **Coherence:** The logical connection and flow between sentences and ideas, making the text easy to understand.
- **Instruction Following:** The extent to which the text meets the given instructions or task requirements.
- **Consistency:** The degree to which the information and style within the text remain uniform and accurate throughout.
- **Variety:** How well the model generates a story in the required Arabic variety.

We find that GPT-4 tends to score the MSA content higher than dialectal ones, even if the task is to generate a dialectal story. To mitigate this issue, we added this last feature where we explicitly ask GPT-4 if the generated content follows the required

Arabic variety specified in the prompt or not, and to which degree. Figure 2 demonstrates the prompt we pass to GPT-4 for evaluation.
We evaluate 20 new prompts by asking the models to generate 20 corresponding stories and then pass the prompt plus story to GPT-4 for evaluation. We compare our two models against three other open and proprietary models. Namely, we compare against GPT-3.5, Command-R[5], and AceGPT-7B-Chat (Huang et al., 2024).

It is pertinent to note that we experimented with other strong open-source models, such as LLaMA-3-70B-Chat (Touvron et al., 2023) and Mixtral-8x7B (Jiang et al., 2024) (accessed through an API), but these models failed to adhere to our instructions. This failure highlights the superiority of our models over these strong baselines. Furthermore, we could not compare our models against larger Arabic LLMs, such as Jais-30B (Jain et al., 2017) and AceGPT-13B (Huang et al., 2024), due to computational constraints. In other words, we limited our comparisons to 7B models unless an API was available.

Table 2 depicts the results of each tested model according to GPT-4. As we clearly see from Table 2, the overall results gap between the models is very narrow. In addition, both our model A and model B are very competitive with larger models even though they are an order of magnitude smaller. Model A performs well in *Instruction Following* across all three varieties and shows strong *Consistency* and *Fluency* in MSA. Model B exhibits better Consistency in MSA and Moroccan, shows strong Fluency and *Coherence* in Egyptian, and relatively lower *Variety* scores.

Model B which was exposed to additional training steps on translated data, performs better than Model A on *almost all metrics across dialects*, which proves indeed the intuition behind training on more data does help. Comparative results suggest that there might be opportunities for further fine-tuning or learning from stronger models such as Command-R since this model shows strong performance across multiple metrics. Our results demonstrate that Command-R is the strongest baseline, and the most consistent model across metrics and varieties. Even in the *Variety* feature where the performance of other models falls, Command-R achieves a score as high as 3.27. Command-R outperforms even GPT-3.5 while being only a fraction of its size, suggesting that the model size is not everything and the quality of data does help. We can see from Table 2 that almost all metrics drop for dialectal varieties compared to MSA. This can be directly linked to the lack of Arabic dialectal data that LLMs have been exposed to during the pre-training stage. We included bar charts in Figure 3 for more details.

[5] https://dashboard.cohere.com/playground/chat?

6.2 Human Evaluation

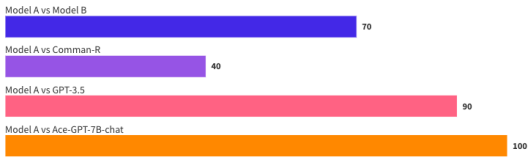

Figure 4: **Model A** vs. other models in MSA human evaluation. Numbers reflect the number of times Model A is preferred over other models.

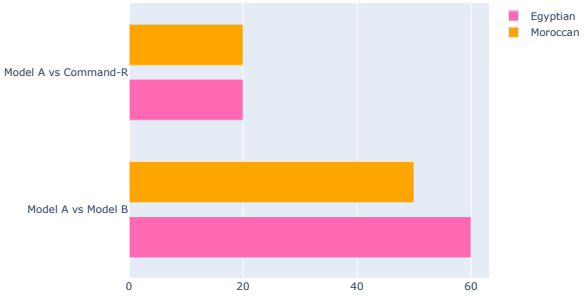

Figure 5: Human comparison of Model A vs Model B and Command-R on Moroccan and Egyptian dialectal story generation. the numbers reflect the number of times Model A was preferred over other models in percentage.

Pilot evaluation. We carry out a pilot investigation where a native speaker of Arabic with knowledge of different dialects inspects stories generated by different models and comes up with observations. The expert finds that Model A tends to generate longer stories compared to Model B. Both models A and B are less perfomant in the Moroccan dialect compared to the Egyptian dialect. AceGPT-7B-Chat (Huang et al., 2024) and GPT-3.5 fail to generate dialectal stories, even when explicitly prompted to do so.
Extensive human comparison of different models. We ask four Arabic native speakers to rank ten stories generated by different models based on the following criteria: *Instruction Following*, *Fluency*, and *Variety Adherence*. However, since

AceGPT-7B-Chat and GPT-3.5 failed to generate dialectal content, we include these models only in the MSA part of the human evaluation task (i.e., we exclude them from the dialectal evaluation). We thus compare our two models, model A and model B, to Command-R. We ask an Arabic native speaker to compare different models to each other, finding our Model A to surprisingly be almost always better than GPT-3.5(90%). Our model A also outperforms Command-R(35B) 40% of times, always outperforms AceGPT-7B-Chat, and Model B 70%. More details are in Figure 4.

For the human evaluation of dialects, we compare our models against Command-R, which was the only model other than ours able to generate dialectal content. Our experts find that Model A to be able to outperform Command-R 20% of the time for both dialects and outperforming Model B 50% and 60% on Moroccan and Egyptian dialectal stories, respectively (Figure 5 for visualization).

7 Conclusion

In this paper, we present the first LLM-based study on automatic Arabic story generation. Our study takes as its target MSA and two Arabic dialects (Egyptian and Moroccan). For our purpose, we translate and generate datasets based on a custom prompt template. We fine-tune our models on these datasets, comparing against both equally- and bigger-sized models. Through extensive automatic and human evaluation, we empirically show our models' superiority to strong baselines. In the future, we plan to fine-tune bigger models on larger datasets. We also plan to include more dialects in our training, for wider coverage.

Limitations

This study has the following limitations:

- **Compute constraints.** Due to computational limitations, we restricted ourselves to models with a maximum size of seven billion parameters or those with an available API.
- **Limited data.** Our training dataset consisted of only 3,000 samples of high-quality data generated by GPT-4. In the future, we are planning to generate more data with the newly released GPT-4o.
- **Lack of error analysis:** We believe carrying out an error analysis would benefit our work. In particular, we observe that GPT-4 does not fully adhere to our instructions 100% of the time during the generation of training data. This could lead to issues in the data generated using this model and an error analysis could uncover any such limitations. Future work should take this into acount.

Ethical Considerations

Similar to other generative models, our model can reflect the bias in its data. Any use of the model should take this into account.

Acknowledgments

We acknowledge support from Canada Research Chairs (CRC), the Natural Sciences and Engineering Research Council of Canada (NSERC; RGPIN-2018-04267), the Social Sciences and Humanities Research Council of Canada (SSHRC; 435-2018-0576; 895-2020-1004; 895-2021-1008), Canadian Foundation for Innovation (CFI; 37771), Digital Research Alliance of Canada,[6] and UBC ARC-Sockeye.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. Storium: A dataset and evaluation platform for machine-in-the-loop story generation. *Preprint*, arXiv:2010.01717.

Arwa Alhussain and Aqil Azmi. 2024. Crosslingual transfer learning for arabic story ending generation. *Indonesian Journal of Computer Science*, 13(2).

Fakhraddin Alwajih, El Moatez Billah Nagoudi, Gagan Bhatia, Abdelrahman Mohamed, and Muhammad Abdul-Mageed. 2024. Peacock: A family of arabic multimodal large language models and benchmarks. *Preprint*, arXiv:2403.01031.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

[6] https://alliancecan.ca

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. *Preprint*, arXiv:1612.08083.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english? *Preprint*, arXiv:2308.01223.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Juncai He, Ziche Liu, Zhiyi Zhang, Junying Chen, Jianquan Li, Benyou Wang, Lian Zhang, Ruoyu Sun, Xiang Wan, Haizhou Li, and Jinchao Xu. 2024. Acegpt, localizing large language models in arabic. *Preprint*, arXiv:2309.12053.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.

Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. 2017. Story generation from sequence of independent short descriptions. *Preprint*, arXiv:1707.05501.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Karima Kadaoui, Samar M. Magdy, Abdul Waheed, Md Tawkat Islam Khondaker, Ahmed Oumar El-Shangiti, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Tarjamat: Evaluation of bard and chatgpt on machine translation of ten arabic varieties. *Preprint*, arXiv:2308.03051.

Amr Keleg, Sharon Goldwater, and Walid Magdy. 2023. ALDi: Quantifying the Arabic level of dialectness of text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10597–10611, Singapore. Association for Computational Linguistics.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation. *Preprint*, arXiv:2009.06367.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zhiyu Lin and Mark Riedl. 2021. Plug-and-blend: A framework for controllable story generation with blended control codes. *Preprint*, arXiv:2104.04039.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. 2022. Co-writing screenplays and theatre scripts with language models: An evaluation by industry professionals. *Preprint*, arXiv:2209.14958.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. 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*, pages 839–849, San Diego, California. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin,

Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Zeeshan Patel, Karim El-Refai, Jonathan Pei, and Tianle Li. 2024. Swag: Storytelling with action guidance. *Preprint*, arXiv:2402.03483.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ahmed Y. Radwan, Khaled M. Alasmari, Omar A. Abdulbagi, and Emad A. Alghamdi. 2024. Sard: A human-ai collaborative story generation. *Preprint*, arXiv:2403.01575.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura,

Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Tiannan Wang, Jiamin Chen, Qingrui Jia, Shuai Wang, Ruoyu Fang, Huilin Wang, Zhaowei Gao, Chunzhao Xie, Chuou Xu, Jihong Dai, Yibin Liu, Jialong Wu, Shengwei Ding, Long Li, Zhiwei Huang, Xinle Deng, Teng Yu, Gangan Ma, Han Xiao, Zixin Chen, Danjun Xiang, Yunxia Wang, Yuanyuan Zhu, Yi Xiao, Jing Wang, Yiru Wang, Siran Ding, Jiayang Huang, Jiayi Xu, Yilihamu Tayier, Zhenyu Hu, Yuan Gao, Chengfeng Zheng, Yueshu Ye, Yihang Li, Lei Wan, Xinyue Jiang, Yujie Wang, Siyu Cheng, Zhule Song, Xiangru Tang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024. Weaver: Foundation models for creative writing. *Preprint*, arXiv:2401.17268.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *Preprint*, arXiv:2302.11382.

Kaige Xie and Mark Riedl. 2024. Creating suspenseful stories: Iterative planning with large language models. *Preprint*, arXiv:2402.17119.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: Story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 841–852, New York, NY, USA. Association for Computing Machinery.

A Example Appendix

This is an appendix. We provide an addition organized as follows:

- Training Examples A.1

A.1 Training Examples

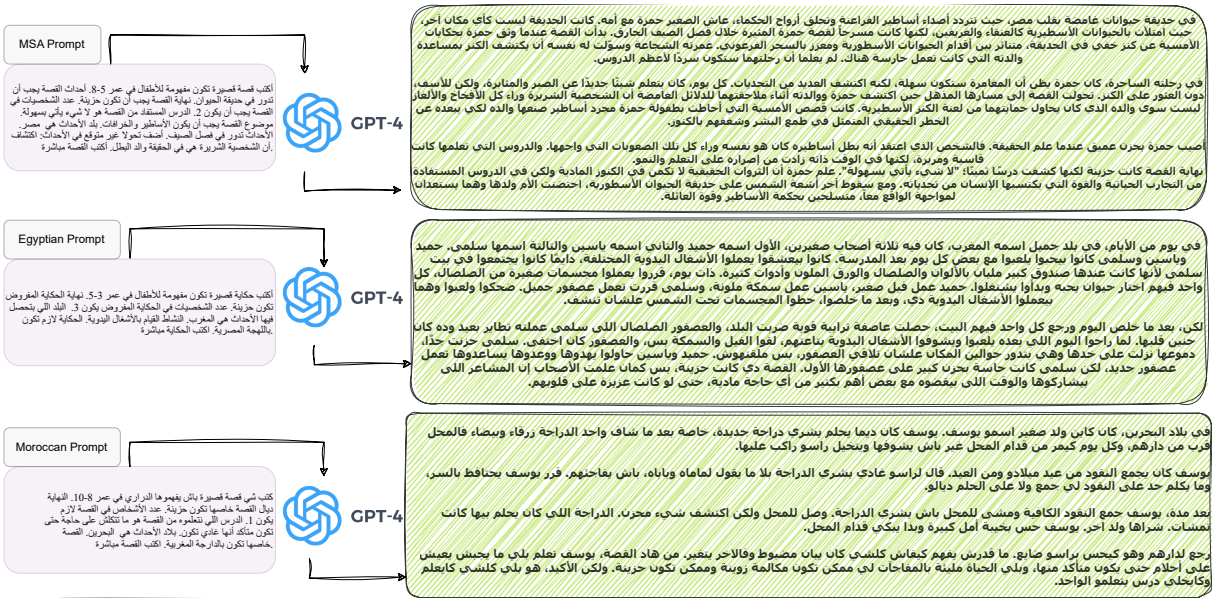

Figure 6: Example of our training samples generated with GPT-4-Turbo. The figure depicts prompts and their corresponding stories in three Arabic varieties: MSA, Egyptian, and Moroccan dialects, correspondingly.