AraGPT2 [3], a variation of GPT-2 (Generative Pre-trained Transformer 2), follows a unidirectional model and was developed explicitly for Arabic text generation.

Table 1. Models size

Model	Optimizer	Context size	Embedding Size	Num of heads	Num of layers	Model Size / Num of Params
AraGPT2-base	lamb	1024	768	12	12	527MB / 135M
AraGPT2-medium	lamb	1024	1024	16	24	1.38G/370M
AraGPT2-large	adafactor	1024	1280	20	36	2.98GB/792M
AraGPT2-mega	adafactor	1024	1536	25	48	5.5GB/1.46B

Table 1 presents four different variants of AraGPT2 models, each with two distinct architectures. It provides detailed information about the optimizer used, context size, embedding size, number of heads, number of layers, and the corresponding model size or number of parameters for each model.

2.2. Story Generation

Creating narratives with a coherent plot and theme while infusing creativity and entertainment remains a challenge. A constructive approach to ensuring narrative consistency involves employing a sentence prompt as a guiding framework [4]. This method significantly streamlines the process of generating stories with a uniform plot structure. However, extending these narratives into longer, cohesive texts pose another hurdle. The model must accurately establish a suitable global context and strategically plan content for sustained generation. One effective technique in this regard is the Progressive Generation of Text [5]. This method initiates by generating a sequence of informative words, which subsequently serve as the foundation for stage-by-stage elaboration. Each stage of text generation builds upon the output of the previous stage, leading to a complete and full final story.

3. Dataset and Features

Due to a scarcity of resources for gathering Arabic stories, a collection of approximately 1000 stories were manually collected from online sources. This corpus represents a variety of narratives, each offering a unique perspective and thematic element. To ensure the dataset's integrity, a thorough cleaning process was done, by meticulously reviewing and refining the content. This step was particularly crucial, given our target audience of children. Any stories that deviated from our desired message or contained inappropriate content were removed, ensuring that the corpus maintains its educational, culturally accurate and wholesome essence. After the meticulous cleaning, the corpus was narrowed down to 301 native Arabic stories.

To prepare the text data for training, each individual story was split word by word into a single token, then each word was converted and mapped into its corresponding integer ID. This is done using AutoTokenizer.

4. Methodologies

This proposed system revolves around our fine-tuned story generator, by utilizing the generated story with many other models to create an engaging narrative experience. A title, image description, and background audio description are generated from the story. The image description is then used as input for the image generator, similarly the audio description is used as input for the background audio generator. Finally, the text to speech model generates a voice that reads the story. The proposed system structure is shown in Figure 1.

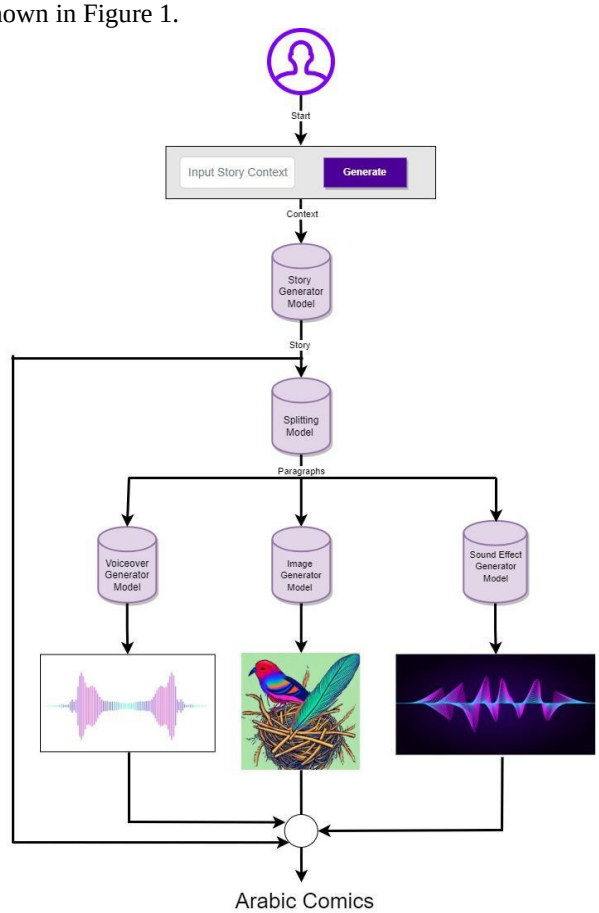

Figure 1. System structure

4.1. Text Generator

With a dataset of only 301 entries, the dataset that will

be used for training is small, thus suggesting the use of a model that does not have a large size to avoid overfitting. Therefore, we will be training and comparing the AraGPT-Base and AraGPT-Medium models, we'll also train on AraGPT-Large cautiously to check for potential overfitting.

Delving into model specifications, AraGPT2-Base stands as the most compact and lightweight among the AraGPT2 models, having only 135 million parameters. It has a 12-layer transformer architecture shown in Figure 2 where each layer incorporates 12 multi-head self-attention mechanisms. This mechanism allows the model to simultaneously focus on various positions within the input text. With an embedding dimension of 768, the model represents each token through a vector of 768 dimensions.

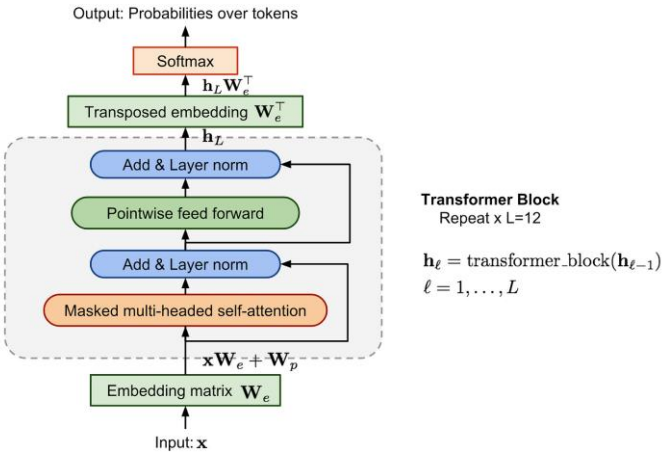

Figure 2. GPT2-Base architecture

Elgeish's GPT2-Medium-Arabic-Poetry was fine-tuned using AraGPT2-Medium and therefore has its same architecture. AraGPT2-Medium possesses increased complexity compared to AraGPT2-Base. It comprises 370 million parameters and features an expanded embedding dimension of 1024. AraGPT2-Medium uses a 24-layer. transformer architecture, with each layer having 16 multi-head self-attention mechanisms. AraGPT2-Large comes with an impressive configuration, featuring 792 million parameters, a 36-layer transformer architecture, and a substantial embedding size of 1280. However, given our limited dataset size of 301, there is a concern that AraGPT2-Large might suffer from overfitting. We will fine-tune, train, and compare these models to see which comes up with the better stories.

4.2. Title Generator

Titles serve a dual purpose beyond just naming the story. They not only fulfill the role of labeling the story, but they could also act as input for other models, such as the image generator model. In the implementation AraT5-base-title-generation, a model made for generating Arabic titles from texts, will be used for title generation. Alternatively, ChatGPT API will be used to create titles and will be compared with the generated titles for AraT5.

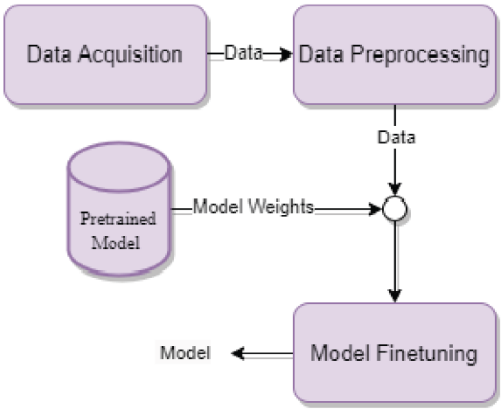

Figure 3. Model fine-tuning

4.3. Image Generator

To aid in crafting a comprehensive and immersive storytelling environment, visual stimuli play an important role in captivating the minds of young readers. As a result, images will be generated to accompany the stories, akin to comics. A fine-tuned Stable-Diffusion v2.0, Anything V4.0, will be used for image generation. The model operates by initially encoding an image into a latent representation. Subsequently, this latent representation is fed into a diffusion model, which incrementally introduces noise to the latent representation until it converges to a random noise vector. Most importantly, the diffusion model considers a text prompt as a conditioning factor, aiding in directing the image generation process. We will compare three different methods to work as inputs for this image generator, we will try the Ara-T5 Title Generator, the mbert2mbert Arabic Text Summarizer model, and finally we'll ask ChatGPT for a description of the image from the story.

4.4. Text to Speech

To enhance the auditory dimension of storytelling, a text to speech component will be integrated into the comics. Different Arabic text to speech (TTS) models will be used to determine which is most fit as a storyteller. Following this, one of the utilized models is the Arabic TTS system (referred to as "الناطق العربي") which is an open-source project that comes from KACST, the project is given in the appendix.

The other model is Tacotrone2 that is a predictive network with recurrent sequence-to-sequence architecture and attention is employed [6]. This network anticipates a series of mel spectrogram frames using an initial character

sequence as input. Additionally, a customized iteration of the WaveNet model is utilized. This modified version generates samples of time-domain waveforms while being conditioned on the mel spectrogram frames that were predicted earlier. An implementation of the approach is available and will be used for the study. Further details and references to this implementation can be found in the appendix section.

Finally, the comparison will also include the GTTS (Google Text-to-Speech) library. GTTS is particularly relevant for this study due to its emphasis on accurate pronunciation, a crucial aspect for young listeners.

4.5. Background Audio Generator

Background audio is another way to further increase the immersion in the stories, it can also assist in setting the tone or mood of the story by the different ambient sounds it can produce. The model that will be used for background audio generation is AudioLDM. AudioLDM takes a text prompt as input and predicts the corresponding audio. It can generate text-conditional sound effects, by utilizing an audio description generated via ChatGPT as input for the model, thus generating the background audio for the comic.

5. Results and Evaluation

5.1 Story generation evaluation

After conducting experiments to fine-tune and evaluate the AraGPT-Base, Elgeish, and AraGPT-Large models for the task of story generation. The evaluation of the models primarily focused on the models' ability to generate coherent and contextually relevant stories. key findings from our experiments are in Figure 4:

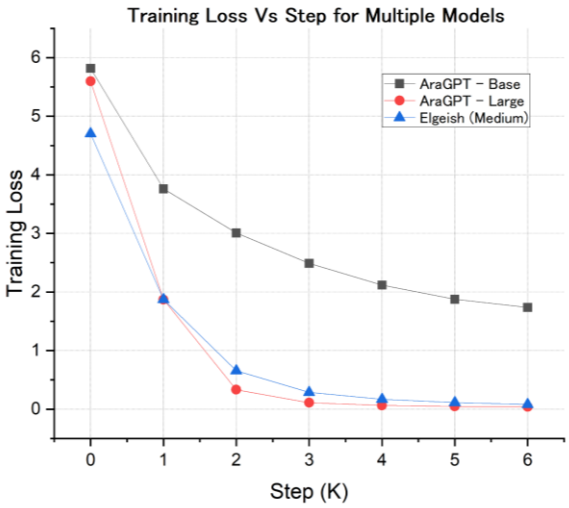

Figure 4. Training loss for multiple models

All three models were able to successfully generate stories that exhibited both coherence and quality, but as is evident in Figure 4, both Elgeish, and AraGPT-Large, for the fine-tuning process. These models were evaluated based on their performance and output. Subsequently, their outputs were extracted as descriptions and further utilized as input for additional models to produce the final product.

Upon careful examination of the provided Figure 4, it becomes evident that AraGPT-Large model and elgeish exhibits signs of overfitting when applied to our dataset, rendering it potentially unsuitable for us, until more data is collected.

Table 2 illustrates the evaluation of the models through an intrinsic evaluation technique, perplexity, that was applied to the test dataset. The perplexity of the models is computed using a stride of 50. AraGPT-Large and Elgeish both achieved a high perplexity score, meaning the model is unpredictable when generating text. Based on their perplexity scores, and figure 4, it is evidently clear that both AraGPT-Large and Elgeish overfit when fine tuning with a small dataset. Therefore AraGPT-Base will be used for story generation.

Table 2. intrinsic models' evaluation

Model	**Perplexity**
AraGPT-Base	25.60
AraGPT-Large	1419.95
Elgeish (Medium)	5958.94

Table 3. Arabic story text generation using different models.

Prompt	GPT - Base	(GPT – Medium Elgeish)	GPT - Large
كان هناك طير جميل	كان هناك طير جميل جدا ذو ريش ملون وجميل، وكان الجميع يحبه ويعشقونه، وفي يوم من الأيام ذهب الطائر الصغير مع أمه إلى حديقة الحيوانات، فوجدوا الطيور بألوانها الزاهية، وأشكالها الجميلة، ولذلك فقد أقبلت جميعها على زيارته، والجلوس معه، والاستمتاع بمشاركته في صنع اليراعة."يحكى أن ملكا كان يحكم مملكة أنام في قارة آسيا في قديم الزمان، وكانت مملكته عبارة عن مملكتين صغيرتين متقاربتين...	كان هناك طير جميل جدا ذو ريش ملون وجميل ، وكان الجميع مفتونين بجماله وجمال ريشه. كان هذا الملك في قديم الزمان ، وكان لهذا الملك أميرة جميلة تدعى باسم (مي - نونج) ، وبالرغم من غناها وجمالها الأخاذ إلا أنها كانت تعاني من حزنها وعدم سعادتها ، وفي صباح أحد الأيام سمعت الأميرة صوتا موسيقيا جميلا ، لقد كان صوت آلة الناي ، حيث امتلأ الجو بنغمات موسيقية حزينة ...	كان هناك طير جميل جدا ذو ريش ملون وجميل ، وكان الجميع مفتونين بجماله وجمال ريشه. كان هذا الملك في قديم الزمان ، وكان لهذا الملك أميرة جميلة تدعى باسم (مي - نونج) ، وبالرغم من غناها وجمالها الأخاذ إلا أنها كانت تعاني من حزنها وعدم سعادتها ، وفي صباح أحد الأيام سمعت الأميرة صوتا موسيقيا جميلا ، لقد كان صوت آلة الناي ، حيث امتلأ الجو بنغمات موسيقية حزينة ...
كان يا مكان في قديم الزمان	كان يا مكان في قديم الزمان كان هناك شيخ اسمه أحمد، وكان هذا الشيخ دائم التباهي أمام الناس، وفي يوم من الأيام ذهب إلى السوق ليشتري بعض الحاجيات، فسأل أحد الرجال عنه، فأخبره بما رأى، فقال: أنا رأيت في السوق خروفا سمينا جدا، فأعجبه حجمه، ولونه أسود، فأمر وزيره...	كان يا مكان في قديم الزمان كان هناك فلاح بسيط لا يمتلك من متاع الدنيا سوى حقله وزوجته وأبنائه الخمسة ، وذات موسم من مواسم الزروع استيقظ الفلاح وبدأ في بذر الحبوب في الحقل وانتظر هطول الأمطار أيام كثيرة ولكن لا أمل. فخرج الفلاح الى الحقل العطشان ونظر الى السماء التي كانت على وشك أن تُلبد بالغيوم وقال لها : تعال يا مطر تعال ...	كان يا مكان في قديم الزمان شاب اسمه علاء الدين، وكان هذا الشاب من عائلة فقيرة، لكنه كان نشيطا وقويا. كان عم علاء الدين شخص أناني ولا يحب إلا نفسه، وفي يوم من الأيام ذهب علاء الدين مع عمه للبحث عن كنز في مغارة، وطلب عمه منه أن ينزل إلى المغارة؛ كي يعتني به، ويحضر له بعض المجوهرات. وبعد أن عاد علاء الدين إلى الكهف، ذهب إلى عمه بحجة أنه يريد أن يطلب مسامحته...

From the table above, various Arabic stories texts were generated providing the same prompts via the fine-tuned models. The generated stories from the large model are overfit as they are almost exactly duplicates from stories in the dataset, and the Elgeish model seems to follow a similar pattern as well, in fact the first 16 words from the first prompt for the Elgeish and the Large model are the same. Therefore, the Base model will be used to

generate stories.

5.2 Prompt Comparison for the Image model

The generation of images depends on a prompt, a method must be chosen to create this prompt. After generating a story, we used three different methods to create prompts for image generation, the results are shown in the table below. (The first two methods generate the text in arabic, they were translated to english using Google Translate)

Table 3. Input methods for image generation

Method	Prompt	Image
Ara-T5 Title Generator	قصة طير : Bird Story	
Mbert2Mbert Arabic Text Summarization	طيران مصنوع من الأرز والخضراوة : Airplane made of rice and vegetables	
ChatGPT-API	A colorful bird in a mountain nest, its magical feathers creating a vibrant haven as people discover nature's secrets.	

The first method, the Ara-T5 Title Generator model, generated a short prompt that lacked enough detail to accurately describe the story, it generated an image that can vaguely fit the story but isn't a precise representation of it. The second method, using the summarization model, generated a completely inaccurate description of the story, and the image had no correlation to the story. The last method, ChatGPT-API, generated a description of the story that visually described the objects in the story. As a result, ChatGPT-API provided the most accurate input for generating an image that closely aligns with the story's content.

5.3 Voice and Sounds

There are two components here, the voice over and background sound effects. For the background sound effects, ChatGPT API is used to describe the sounds that are related to the story, then the prompt goes to the AudioLDM model to generate the background sound. For the text-to-speech model, we compared 3 different models, the open-source project from KACST, Google Text-to-Speech, and Tacotron2, and we found the following. The KACST model generates a robotic voice that we'd prefer not to use, Google TTS also produces a robotic voice but has great pronunciation. The final model, Tacotron2 generated the most realistic voice but had bad pronunciation. Consequently, the Tacotron2 model is chosen for narrating stories. However, to combat the unclear pronunciation we found that we need diacritics to be added to the arabic text. So ChatGPT API will be used to add diacritics to the story before it is used as input to the Tacotron2 model.

After generating these two audio segments, we combined them by allocating 80% of the audio volume to the voice-over component and 20% to the background sounds.

5.4 Final Models Output

Figure 5. Example of generating comics

The successful functioning and generation of satisfactory results can be observed in figure 5 when the outputs of the four models are merged.

6. Conclusion

In conclusion, the study highlights the significance of educating children about Arabic culture through storytelling. The proposed approach explores the utilization of natural language processing (NLP) to create captivating Arabic comics. This involves employing a pre-trained model for the text generation and subsequently fine-tuning it. The text generation model's performance has been assessed through the perplexity which is an intrinsic evaluation and then using the least perplexity to get the model that fits well with our test dataset. Hence, we decided the best model to be used is AraGPT-Base. The developed system comprises voice narration, sound effects generator, image generator, and the generated story. All are combined to create engaging comics for children.

Consequently, this initiative promotes the cultivation of a robust cultural identity and language proficiency among children while fostering a deeper comprehension and appreciation of their traditions. Furthermore, future endeavors may expand upon this work to generate educational animated Arabic videos encompassing disciplines such as science, art, and poetry.

References

[1] Wiseman, S., Shieber, S. M., & Rush, A. M. (2017). Challenges in Data-to-Document Generation. arXiv:1707.08052.

[2] Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based Model for Arabic Language Understanding. arXiv:2003.00104.

[3] Antoun, W., Baly, F., & Hajj, H. (2020). AraGPT2: Pre-Trained Transformer for Arabic Language Generation. arXiv:2012.15520.

[4] Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical Neural Story Generation. arXiv:1805.04833.

[5] Tan, B., Yang, Z., AI-Shedivat, M., Xing, E. P., & Hu, Z. (2020). Progressive Generation of Long Text with Pretrained Language Models. arXiv:2006.15720.

[6] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... Wu, Y. (2017). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. arXiv:1712.05884.

Appendix

KACST implementation:
https://github.com/asrajeh/arabic-tts/blob/master/samples/kacst_ar_asc-festvox.wav
Tacotrone2:
https://github.com/nipponjo/tts-arabic-pytorch