Article

A Pipeline for Story Visualization from Natural Language

Jezia Zakraoui, Moutaz Saleh *, Somaya Al-Maadeed and Jihad Mohamad Alja'am

Department of Computer Science, Qatar University, Doha 2713, Qatar; j.zakraoui@gmail.com (J.Z.); s_alali@qu.edu.qa (S.A.-M.); jaljaam@gmail.com (J.M.A.)
* Correspondence: moutaz.saleh@qu.edu.qa

Abstract: Generating automatic visualization from natural language texts is an important task for promoting language learning and literacy development for young children and language learners. However, translating a text into a coherent visualization matching its relevant keywords is a challenging problem. To tackle this issue, we proposed a robust story visualization pipeline ranging from NLP and relation extraction to image sequence generation and alignment. First, we applied a shallow semantic representation of the text where we extracted concepts including relevant characters, scene objects, and events in an appropriate format. We also distinguished between simple and complex actions. This distinction helped to realize an optimal visualization of the scene objects and their relationships according to the target audience. Second, we utilized an image generation framework along with different versions to support the visualization task efficiently. Third, we used CLIP similarity function as a semantic relevance metric to check local and global coherence to the whole story. Finally, we validated the scene sequence to compose a final visualization using the different versions for various target audiences. Our preliminary results showed considerable effectiveness in adopting such a pipeline for a coarse visualization task that can subsequently be enhanced.

Keywords: scene generation; story visualization; GAN; story understanding; language learning

1. Introduction

During the period of the COVID-19 pandemic, teachers had a full-time schedule to provide regular and online lessons to children, divided into several small groups. Both teachers and students encountered changes in teaching and learning habits, respectively. For instance, preparing a sequence of coherent images to visualize textual stories from an Arabic natural language text is a very challenging problem [1]. On the other hand, using only text-to-image retrieval methods is very inefficient for young children with special educational needs and learning difficulties (Senld). For instance, using retrieved images from diverse search engines to visualize non-common characters and actions from a story often requires enormous manual effort, yet it is more difficult to adapt this to meet each student's effective learning needs. The same applies for aligning images within a story. Sometimes, this task remains completely unresolved. We approached this unresolved issue using a semi-automatic scene sequence task, i.e., a visual story task to facilitate the learning process and inspire teachers, instructors, and students.

However, to create such story visualization efficiently, one needs to convert the story constituents into a sequence of image frames in a proper and coherent way. A sequence of images can illustrate the story events and characters that can contain multiple sentences. The sequence of images is defined as a continuous stream of consistent images that are part of the same story or event, as argued by the authors in [2]. Although visualized stories are difficult to generate in a robust way, they are more comprehensible, memorable, and attractive. Consequently, automatic story understanding and visualization has a broad application prospect in storytelling, while also representing an important step in many computer vision (CV) applications such as children learning natural language vocabularies. Essentially, our goal was to create a sequence of images to visualize an Arabic story where

check for updates

Citation: Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; Alja'am, J.M. A Pipeline for Story Visualization from Natural Language. *Appl. Sci.* **2023**, *13*, 5107. https://doi.org/10.3390/app13085107

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 18 March 2023
Revised: 11 April 2023
Accepted: 17 April 2023
Published: 19 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

the text was extended from sentence level to paragraph level for continuous visualization. In prior studies on text-to-image generation [3–5], the same sentence may have a significantly different generated image while depending largely on the contextual information; therefore, it is also necessary to pass the essential contextual information from the story text to the image generation framework. For instance, considering the sentences given in Figure 1, Figure 1b will vary widely without the context of the story, i.e., without the Figure 1a.

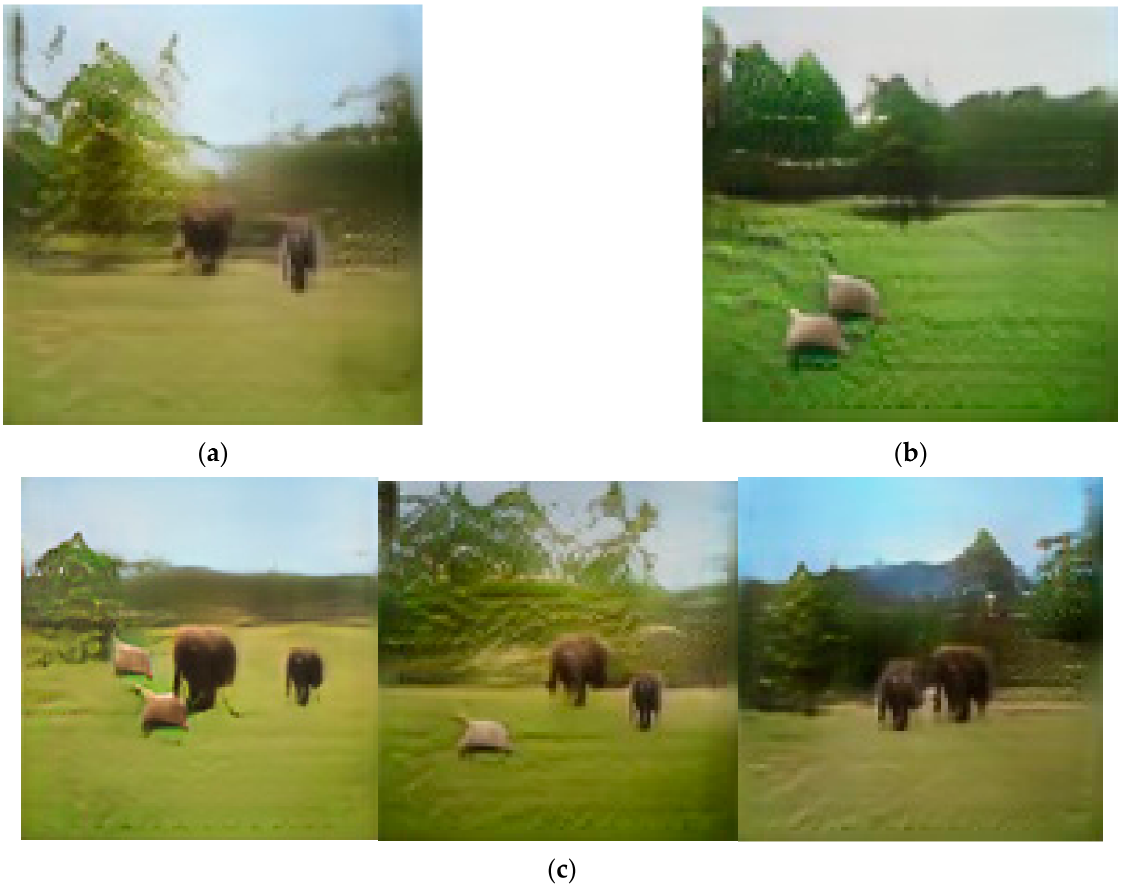


Figure 1. Input sentences (translated from Arabic to English) and correspondingly generated single images and an image sequence: (**a**) The elephants are standing in a grassy field; (**b**) There are sheep around them; (**c**) The sheep approached the elephants.

To tackle the problem of objects and event extraction, we applied scene graphs [6] to represent the detailed semantics of each sentence from the story text. A scene graph (SG) is defined as a graph-based semantic representation having nodes and edges. The nodes represent objects, and the edges represent relationships between them. For simplicity, we rewrote each complex sentence in a simple form such as (object, attribute) or (object, relationship, object) tuples. This step abstracted away most of the lexical and syntactic challenges of natural language in the process.

To tackle the second challenge with image synthesis, we used an automatic image generation framework to allow different versions, namely single images and a sequence of images for common and non-common actions, respectively. However, the challenge here was how to display the logic narrative flow of the sequence of images to visualize the story characters and events in a coherent way. Specifically, the appearance of objects and the layout in the background must evolve as per the story narrative flow. Our method can be considered as a fast solution to visualize non-common characters and actions with multiple images whenever needed.

An emerging trend in CV combining deep learning models is regarded as a possible solution for approaching our task. Among these models, generative models construct scenes from sentences, either from short textual descriptions [5] or short dialog [7]. Additionally, previous studies have assumed that a space that synthesizes both vision and language modalities are indispensable to the performance of any text-to-image synthesis [5]. Notably, recent studies using generative adversarial networks (GANs) have presented good results. However, GAN cannot achieve expected results when the image to be generated contains multiple objects. Indeed, such a requirement is more challenging when multiple objects with complicated relationships and different locations are to be presented in the image [8]. Consequently, complex scene generation is still in the development stage and has not been elaborated upon.

We extended our previous study [9], which attempted to generate sequences of images. Despite producing visual sequences that capture the relevant content of the input text, i.e., characters and events, our previous method was limited to the extracted entities and relationships that exactly matched the model vocabulary, thereby ignoring other content from the input text. To tackle this issue, in this extended study, we added a vocabulary mapping module. Another limitation of our previous study was that the text–image alignment method was made by consecutively aligning the images, which showed sharp changes in visual content between the frames. In this study, we employed a multimodal similarity function to dynamically align the images in a sequence based on their similarity scores to the input text. Moreover, uncommon actions in the input text are hard to visualize and are left behind due to many reasons such as their rarity in the dataset; thus, we believed that a decomposition of such actions in a detailed image sequence could facilitate the visualization of such actions. Furthermore, we compared our method with two state-of-the-art models for generating images.

In this context, we proposed a framework based on a text-to-image approach and CLIP to generate and return the best image in the sequence corresponding to the input text. More specifically, the framework took the text as an input, generated a sequence of images, and highlighted the images whose CLIP embedding was most similar to the input text. Notably, we started with an NLP task, i.e., a story parsing task using handcrafted syntactic rules, followed by entities and relation extraction, and vocabulary mapping. Then, a semantic representation using SG was built upon all of the resulting triples denoted as (object, relation, object). Afterward, an image sequence generator for generating images from SG was applied. Subsequently, CLIP was used for image production and input text embedding, followed by computing similarity scores. Further, we evaluated the produced text–image alignments using different metrics.

In contrast to previous studies that have focused on single image generation, we applied detailed image sequence generation for non-common actions using a pre-trained model on a visual genome dataset [10] under the PyTorch framework [11]. Finally, we applied the CLIP [12] similarity function as a metric to check the sentence-level coherence to generated an image sequence, and it computed the cosine similarity between the feature vectors of the story sentences and each of the images. A higher similarity meant a closer match between the story sentence and the corresponding images. Based on these scores, the images were reranked to form an image sequence.

The rest of the paper is organized as follows: Section 2 describes the main approaches to scene generation, Section 3 specifically presents our method, Section 4 discusses the experimental setup, Section 5 shows our evaluation and obtained results, while Section 6 concludes the paper.

2. Related Studies

Early studies on text visualization and illustration [13–17] traditionally relied upon manually annotated image repositories collected from search engines using image retrieval techniques [18,19], and by using images produced by users [20]. Retrieval-based approaches compare texts and images across modalities [21] using different techniques such a canonical correlation analysis [22]. Specifically, text-to-image systems use retrieval methods that focus on the matching of text and images. In addition, these studies have relied on massive amounts of labeled data, as stated by the authors in [23]. One of the early story visualization attempts was the story-picturing system [13]. The system retrieved landscape and art images from online repositories to illustrate ten short stories. It used keywords from the stories and image descriptions to match the linking between the images using the similarity function. A comparative study of early story illustrations, visualization systems, and tools can be found in [24].

A method worth mentioning was proposed by Huang et al. [14], using VizStory, as a visualization system of fairy tales, to transform the input texts to representative pictures. The system selected keywords from segments in the stories, while relevant pictures were searched for using online resources based on their tags. Finally, to represent the main ideas of the original segments, the final pictures were composed. Afterward, the authors built in a visual storytelling dataset (VIST) that was useful for image-in-sequence to story-in-sequence generation [25], thereby initiating the visual storytelling task.

Alternatively, the studies of the authors in [2,26] attempted to visualize a story with image sequences. The former proposed to enhance the single sentence representation with a global coherence vector and apply global and region matching to retrieve an image for each sentence. The latter proposed a framework with a story-to-image retriever. It selected relevant and inspirative cinematic images and used a storyboard creator that further refined and rendered the images to improve the relevancy and visual consistency. Both authors worked on VIST datasets to evaluate their work. Despite the method given by the authors in [26] scene images with a high resolution and multiple foreground objects were generated; however, it only used cartoon characters where the structures and shapes were poor, resulting in poor image quality.

Recently, Fang et al. [27] used the shooting time order and the storyline behind the images to construct a narrative collage image. First, they considered a set of semantic salient objects from each representative image for object extraction. Then, they used an image canvas according to layer graphs and scene graphs to visualize the extracted objects. Finally, they synthesized a new narrative collage image. More recently, Fang et al. [28] proposed a comprehensive text-to-image synthesis pipeline. They used segmented background scene image and foreground objects from the COCO dataset to generate complex and high-resolution scene images. Finally, they applied the constrained Markov chain Monte Carlo method to generate the optimal positions and scales for all foreground objects to look more realistic. However, these methods rely heavily on image retrieval and fail to generate images with a realistic look, since they just focus on text understanding, object selection, and text–object matching.

With the advances in CV using GANs [29], which are a more powerful class of implicit generative models, they have been successfully applied to various image synthesis methods such as text-to-image synthesis from short textual descriptions [3–5,30–32]. A key task in text-to-image generation is understanding longer and more complex input text, as in our case. Story visualization, however, is different from short textual descriptions, which places more emphasis on semantic coherency rather than simple descriptive text. A story text can contain different scene changes, many objects, different backgrounds, etc. An interesting study [33] has demonstrated dialogue-to-image generation, where the input was a complete dialogue session rather than a single sentence. However, this method was simply a text–image concatenation task and used a coarse sentence condition that, as a consequence, limited its overall performance.

Lee et al. [23] proposed the StoryGAN model to tackle the above the story visualization challenge. Their model employed a context encoder to track the story narrative flow. It used two discriminators; one at the story level and the other at image level to enhance image quality and the consistency of the generated images. However, well-known difficulties in training generative models such as instabilities in the training procedure [34] has limited these studies of specific domains, such as cartoon characters [23]. The study of Zeng et al. [35] enhanced the latter study in several and significant ways, particularly in relation to image quality and consistency. First, they integrated a universal sentence encoder to incentive compliance of the generated images with textual descriptions. Second, they incorporated an attention-driven word feature into their model, making it more realistic in terms of image details. Finally, they introduced an image patches discriminator to determine whether parts of the image were real. However, this work was limited in its scope since only cartoons could be considered and the quality of the generated images needed further improvement.

More recently, Song et al. [36], Li et al. [37], and the authors in [38] improved upon StoryGAN [23] to emphasize the continuity between consecutive frames in generated video as well as to enhance the quality and relevance of the generated images. More recently, the authors proposed an approach [39] that decomposed the task of story visualization into three phases, namely semantic text understanding, object layout prediction, and image generation and refinement. In contrast to our study, only captions were considered, and only a single image was generated at each step. In addition, their model used two-stage image generation, aka StackGAN. A different model called *Text2Scene* has been proposed by Tan et al. [40]. It is a sequential framework [41] where, at every time point, it learns to generate objects and their associated attributes by attending to the words in the input text and the status of the current generated scene. This approach, however, is restricted to the composition of tasks of abstract scenes and object layouts. On the other hand, the quality of the generated image is usually not stable in most cases. Subsequently, it is difficult to directly apply generative models in complex and real-life scenarios such as scene generation for stories in the wild [42].

3. The Proposed Pipeline Framework

Image generation for the task of story visualization aims to generate representative and coherent images to convey the semantic in a given story text. This is a challenging task since it requires a deep understanding of the objects involved in the story as well as their mutual interactions, and semantical connections and co-relations. In this context, we proposed a framework consisting of (i) an NLP task followed by (ii) a semantic representation using SG, (iii) an image sequence generator for generating images from SG, and (iv) CLIP for producing images and input text embeddings followed by computing similarity scores. Further, we evaluated the produced text–image alignments using different metrics. Moreover, we compared our approach based on scoring images according to their semantic relevance to the input text. In the following section, we have presented the main components of the proposed story visualization pipeline, as depicted in Figure 2. The architecture consists of four consecutive parts as shown below:

1. **Natural language processing**: The first step was the language model where we applied a preprocessing pipeline, machine translation, tokenization, stop-word removal, co-reference resolution, and semantic parsing, i.e., the task of mapping natural language text into its semantic representation using a scene graph parser.
2. **Relation extraction and vocabulary mapping**: The second step involved constructing scene graphs of extracted triples so that the text was transformed into a directed graph G = (O; R) of objects O (nodes) and their relations R (edges).
3. **Image sequence generation**: The third task was image sequence generation where we generated images from scene graphs for all mapped triplets using two different modes.
4. **Text–image alignment**: Finally, we applied CLIP similarity function to produce previsualizations with different sequences. The instructor could examine each image sequence and choose whether to use the single image version or the detailed image sequence.

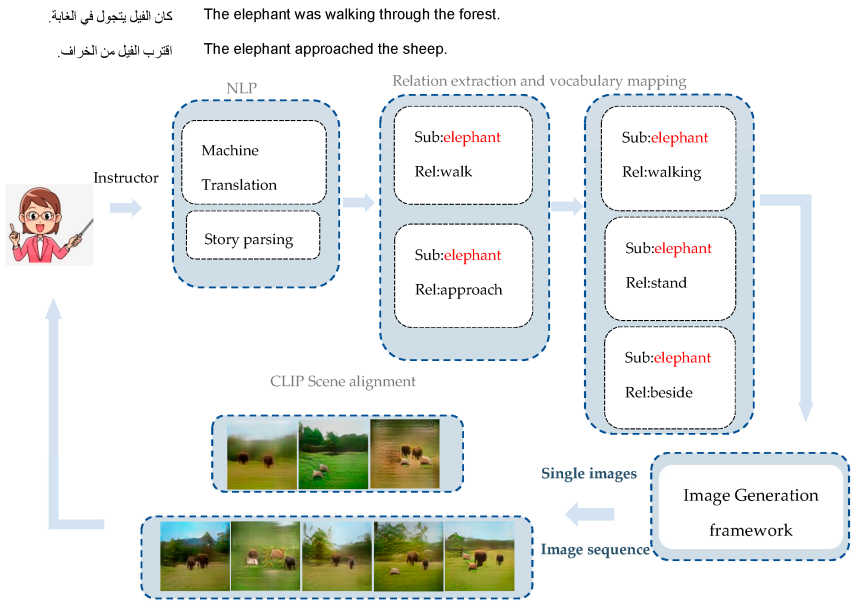


Figure 2. The overall pipeline of the proposed approach: a story text is piped into an NLP module to the first MT, which preprocesses and parses the sentences into scene graph triples. Then, the triples are mapped to model vocabulary to generate single images and image sequences. After applying CLIP model, the instructor is able to adjust the synthesized image sequences by choosing whether to use the single image version or the detailed image sequence.

3.1. Natural Language Processing (NLP)

We considered mainly children's stories featuring animals. After translating the stories from Arabic to English, we extracted the characters and scene objects that were necessary for visualization, including the relationships between them. Then, we proceeded with a neural coreference resolution of the pronouns to prepare the text as simply as possible for the next step. A pre-trained neural model NeuralCoref [43] was used to replace the ambiguous mention of pronouns with its corresponding nominal pronoun.

We obtained a set of relationships based on form (subject; relation; object) by using a scene graph parser. In many cases, the obtained list of relationships was noisy, for instance, objects may have multiple relationships, a passive form, a plural form, etc. To prepare the list for further processing, we pre-processed the list using different rules.

We defined a list of entities to record characters and scene objects that appeared in the text. We traversed every relationship and confirmed whether the involved relation and the entities existed in the vocabulary list. If they existed, they were appended in the *relationships* and *entities* list, respectively. Otherwise, we use word2vec-based (https://code.google.com/archive/p/word2vec/ (accessed on 10 February 2023)) similarity function to find the nearest token in the model vocabulary list. Finally, a dictionary output was created that included two lists, *entities* = $[o_i, o_j, \ldots]$ and *relationships* = $[[x_i, r, x_j], \ldots]$, where x_i is the index of o_i, x_j is the index of o_j in from the entities' list, and $r \in R$ is the set of model relationship categories. The described process is shown in Algorithm 1.

Algorithm 1: ParseStory parses input text and extracts triples as characters, entities, and relations

```
Input = AS: Arabic Story, args []: list of access parameters, vocabulary []
Output = DrawTriples {}, SceneGraphTriples {}, entities [], relationships []
Begin
    rawtext = translateQCRI (AS, args)
    rawtext = coreference_resolution (rawtext)
    docx = nlp (rawtext)
    SceneGraphTriples = sng_parser (docx)
    for relation in SceneGraphTriples ['relations']
        if (relation in vocabulary) then
            $x_i$, $x_j$ = relation ['subject'], relation ['object']//indices for both involved entities
            $o_i$, $o_j$ = SceneGraphTriples ['entities'].value ($x_i$, $x_j$)//get both involved entities
            if ($o_i$, $o_j$ in vocabulary)
                entities.append ($o_i$, $o_j$)
                relationships.append ([$x_{i,}$ relation, $x_j$ ])
            end
        else
            $o_i$, relation, $o_j$ = get_mapping ($o_i$, relation, $o_j$)//vocabulary mapping
            entities.append ($o_i$, $o_j$)
            relationships.append ([$x_{i,}$ relation, $x_j$ ])
        end
    end
    DrawTriples ['entities] = entities
    DrawTriples ['relationships'] = relationships
    return DrawTriples
end
```

3.2. Relation Extraction and Vocabulary Mapping

We considered phrases that described the main animal characters' behavior. We also focused on some of their common and uncommon basic behaviors. Table 1 shows some common sample phrases used in this work as well as their related actions. It is worth noting that animal behavior that is not listed is considered to be non-common animal behavior. From the resultant phrases of the previous step, we obtained all of the triples in the form <*object, relationship, object*> using a scene graph parser. Due to practical reasons, it was not possible to create images for all of the extracted triples from the story text. Due to this restriction, as in the case of the visual genome dataset [10], the vocabulary mapping used a semantic similarity based on word2vec to find the nearest tokens from the model vocabulary, as shown in Algorithm 2.

Algorithm 2: Get_mapping extended extracted entities and relations with model vocabulary

```
Input = vocabulary [], triples []
Output = similar_triples []
Begin
    model = gensim.models.Word2Vec (vocabulary, size = 100, min_count = 1, sg = 1)//initialize
        from Gensim library (https://radimrehurek.com/gensim/models/word2vec.html
        (accessed on 12 February 2023))
    for entity in triples:
        top_similar = model.wv.most_similar (positive = entity, topn = 1)//get most similar token to
            entity
        ... similar_triples.append (top_similar)
    end
    return similar_triples
end
```

Table 1. An excerpt from stories' details related to the above sentences.

Sentences	Noun-Phrases	Dependency Parsing	Scene Graph Triples
The elephants are standing in a grassy field	The elephants	nsubj	<elephants, in, field>
	A grassy field	pobj	
The sheep are running behind the elephants	The sheep	nsubj	<sheep, behind, elephants>
	The elephants	pobj	
The sheep approached the elephants	The sheep	nsubj	<sheep, approached, elephants>
	The elephants	dobj	

Thus, the mapping also helped us to map non-common actions such as "*approach*" to the similar common action in the list such as "*stand*" and "*walk*". If we failed to find a match, we checked for a mapping while including the verb's preposition such as "*close to*", "*next to*", etc. For instance, for the tokens of the sentences mentioned earlier in Figure 1, we computed their similarities with the terms in the vocabularies and took the maximum value among them all. As an example, the token "*elephants"* was mapped to the term *"elephant"* with a similarity value of 1.0; however, the token *approached* was mapped to the term *"stand"* with a similarity value of 0.1, using the word2vec similarity function.

3.3. Image Generation

We split the image generation step into two main tasks. One task tackled the generation of a single image to visualize sentences in isolation. The second task was directed towards the detailed generation of image sequences, i.e., multiple images that were highly coherent with the whole story. After, obtaining the objects and relationships that composed the scene graph, we used a graph convolution network [11] composed of several graph convolution layers to process the scene graph.

Single image generation. We generated images from scene graph triples of actions and characters using a pre-trained model for PyTorch [11]. Basically, the architecture consisted of three main modules: a graph convolution network (GCN), a layout prediction network (LN) and a cascade refinement network (CRN). First, the GCN took a scene graph as an input and produced an embedding label vector output for each object. Then, these object embedding vectors were used by LN to compute a scene layout by predicting a segmentation mask and bounding box for each object. Given a scene layout, the CRN was then responsible for generating an image that respected the object relations in the scene layout. Finally, discriminators were used to generate realistic output images by adversarially training the image generation network against a pair of image discriminator networks and an object discriminator network. The generated realistic output images were adversarially trained by the image generation network against a pair of discriminator networks $\mathrm{D}_{\mathrm{image}}$ and $\mathrm{D}_{\mathrm{object}}$ to minimize the weighted sum of six losses [11]. The discriminator $\mathrm{D}_{\mathrm{image}}$ attempted to classify its input x as real or fake by maximizing the following objective:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x\sim p_{real}} \log D(x) + \mathbb{E}_{x\sim p_{fake}} \log(1 - D(x)) \quad (1)$$

where $x \sim p_{real}$ is the ground truth image and $x \sim p_{fake}$ is the fake image that is generated using the generator network. The discriminator $\mathrm{D}_{\mathrm{object}}$ guarantees that the generated objects are identifiable by predicting the object's category. Both $\mathrm{D}_{\mathrm{object}}$ and the generator network attempt to maximize the probability that $\mathrm{D}_{\mathrm{object}}$ correctly classifies objects [11].

Image sequence generation. Non-common actions are typically hard to illustrate. To enable a fair visual understanding of such actions, it was necessary to decompose these actions into simple ones. This process enabled us to employ more detailed images rather than only one image. However, the decomposition for actions has only been explored for humans [44,45], even though representative actions with structured representations could lead to improved action recognition in general. Therefore, we applied an image generation mode [11] to generate sequences of images rather than single isolated images. This is because the image sequence can give more details to support the visual understanding of complex actions. For instance, in Figure 1, it is hard to visualize the action "*approaching*" using a single image only; therefore, it is necessary to generate a sequence of images that decompose the flow of this action into several frames, similar to the way that humans actively perceive ongoing actions, i.e., a phenomenon referred to as event segmentation theory [46].

Specifically, for this category of actions, we generated sequences of images shot by shot using progressive additions of objects and relations. Where the input text described only one object, it was rendered in almost the middle of the scene. On the other hand, complex images were rendered by starting with simple characters and progressively adding others to build up to more complex images.

3.4. Text–Image Alignment

Once we generated all of the images, we subsequently computed the cosine similarity using CLIP feature vectors between the story text and each of the generated images. In CLIP, a visual encoder and a text encoder encode an input image and text independently, and the dot-product between the two encoder's output was used as the "alignment score" between the input image and text based on following Formula (2):

$$logits = X_{image}\ X_{text}^{T}\ \times e^{\tau} \tag{2}$$

where X_{image} image and X_{text}^{T} are normalized encoders outputs for the image and the text, respectively, and τ is a learned temperature parameter [12]. The CLIP model, which was already trained over an extremely large number of images, was capable of generating semantic encodings for arbitrary images without additional supervision.

Finally, an automatic alignment image sequence was suggested based on the CLIP scores. The instructor could choose whether to use the single image version or the detailed image sequence. He/she could then refine the image sequence by reordering and skipping frames, etc.

4. Experimental Setup

At this stage, we began by preprocessing the input stories as the input data set. We considered 80 short and simple phrases from Arabic stories in the animal domain [9]. We translated them from Arabic to English, and the selected 20 stories had 80 key phrases. The distribution of objects was consistent in number, where each object possessed five different actions. We selected phrases with a simple narrative structure to introduce concepts using animal characters and their common behaviors such as *running, eating, jumping*, etc., as well as non-common behavior *such as approaching, covering, looking at*, etc. The characters, objects, location, and background were explicitly mentioned in the text and were realistic. In the experimental set up, we further applied the following steps:

Story parsing was applied; it included coreference resolution, part-of-speech tagging, dependency parsing, relation extraction using linguistic patterns, and scene graph parser (https://github.com/vacancy/SceneGraphParser (accessed on 17 March 2023)). For example, we considered the sentences "*The elephants are standing in a grassy field. The sheep are running behind them. The sheep approached the elephants*". After applying the coreference resolution using NeuralCoref (https://github.com/huggingface/neuralcoref (accessed on 25 February 2023)) and manual adjustment, we obtained the following final representation for the sentences "*The elephants are standing in a grassy field. The sheep are running behind the elephants. The sheep approached the elephants*". Table 1 shows the story parsing outputs. Of note, *nsubj*, *pobj*, *dobj*, and *iobj* denoted the nominal subject, object of a preposition, direct object, and indirect object, respectively.

1. To handle out-of-vocabulary words besides those in the training data set, we applied simple vocabulary mapping using word2vec, a pre-trained word embedding model,

to find the nearest vocabulary of the extracted triples in the trained model vocabulary. For example, the token *approached* was mapped to the terms *stand* and *beside*, with similarity values of 0.1 and 0.2, respectively.

2. We used all extracted relation triples and their mapped tokens to generate 128 × 128 images using the same configuration as in the *sg2im* model [11]. The sg2im model is pretrained on the visual genome dataset [10], a dataset with 108,077 128 × 128 images annotated with scene graphs. Each image has an average of 21 objects, 18 attributes, and 18 pairwise relationships between objects where animal categories are included, in addition to their visual relationships. All experiments were executed with Pytorch 0.4, CUDA v10, Cudnn v7, and Python 3. We generated single images and sequences of images for common and non-common actions, respectively depending on the type of actions. Table 2 provides excerpts of such phrases and the correspondingly generated images.
3. We applied object detection using the *PixelLib* (https://pixellib.readthedocs.io/en/latest/ (accessed on 2 March 2023)) model for all generated images. The detected object indicated whether the character mentioned in the story line also appeared in the image frame. We exploited the output of the model to estimate, to some extent, the coherence of the image sequence with the whole story text. We scored each correct image frame and summed the final score for each sequence.
4. Finally, we arrange the generated images for each story sequentially, as produced by the CLIP score, using two different versions: the single image version and image sequences, as can be seen in Table 3 below.

Table 2. An excerpt of phrase (containing non-common actions) and corresponding generated single images versus image sequences.

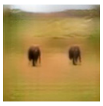
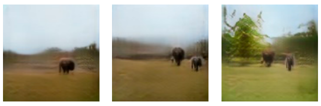
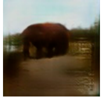
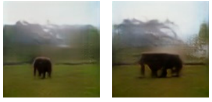
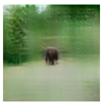
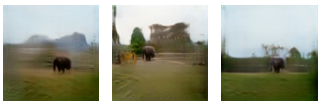
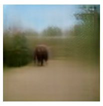
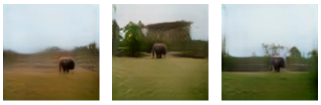
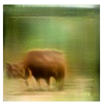
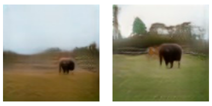
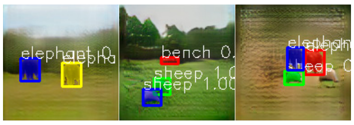
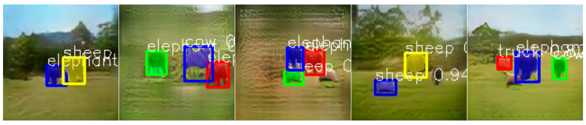
Id	Non-Common Phrases	Generated Single Image	Generated Image Sequence
1	Elephants approaching		
2	Elephant looking at sky		
3	Elephant attaching to tree		
4	Elephant covering in tree		
5	Elephant carrying wood		

Table 3. Single images, image sequences, and corresponding object detection details.

Generation Mode	Object Detection Mask	Character Relevance
Single image		1.2 correct images (counting only correct images, 1 + 0 + 1 = 2)
Image sequence		2.3 correct images (counting only correct images, 1 + 1 + 1 + 0 + 0 = 3)

5. Evaluation and Results

The evaluation of story visualization is complex due to the generative nature of the method. We conducted both quantitative and qualitative assessments as follows. First, we compared our method with BigGAN + CLIP [12] and Dall-E [47], two state-of-the-art models for generating images from user prompts. In these models, each image prediction was actually the result of an optimization process where the latent space of the generator directly maximized the CLIP score between the generated image and the description.

5.1. Quantitative Results

We demonstrated the image sequence quality of our method in a score-based manner with regard to two aspects: character relevance and the semantic relevance of the generated images. For our approach, we adopted a character relevance score and CLIP similarity score between the story text and each of the images.

Character relevance: Inspired by the studies of [23,35], we selected five of the most common characters and actions. Specifically, we selected the following animal characters, *elephant*, *sheep*, *cow*, *zebra*, *and giraffe*, each with five actions representing their behavior and relations. The results obtained from the experiment with each story character, e.g., *elephants*, and *sheep*, is summarized in Table 3. The five continuous images form a visualization version corresponding to a single story. For each image sequence version, we counted each image frame as correct if the characters mentioned in the input sentence appeared in the corresponding image frame, according to the object detection model. For instance, in Figure 1, in the second image (from left to right) the elephant character was not present in the image, so this image was counted as being incorrect. Since the ground truth of the object segmentation was unavailable in the visual genome dataset, we exploited a pre-trained salient object detection model to detect objects from all generated images. The object detection task gave an indication if the character mentioned in the story line also appeared in the image frame.

Semantic relevance: We measured the semantic relevance between the generated image and the story text features for each generated image sequence using CLIP. The images with the highest scores were marked with red borders and selected for final visualization, see also Table 4. We computed the sentence similarity score as local consistency (Table 4) and the story similarity score as global consistency (Table 5).

Table 4. Comparison with state-of-the-art model for real-world image synthesis for a one-sentence story sample: the images with the highest scores were marked with red borders and selected for final visualization.

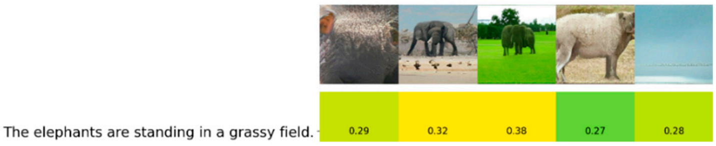
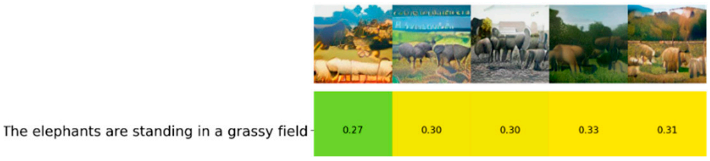
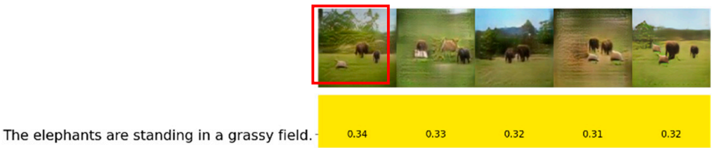
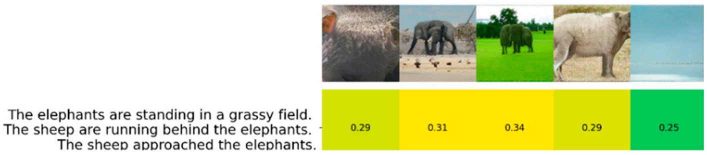
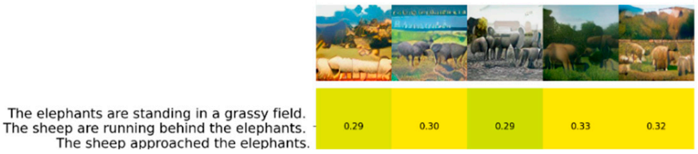
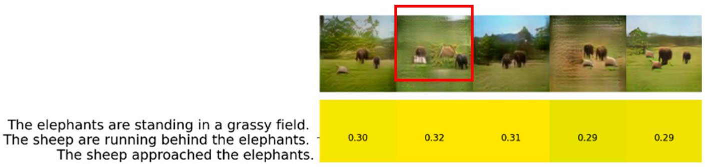
Method	Generated Images	CLIP Score
BigGAN + CLIP Radford, et al. (2021) [12]	 The elephants are standing in a grassy field. 0.29 0.32 0.38 0.27 0.28	0.30
Dall-E Ramesh, et al. (2021) [47]	 The elephants are standing in a grassy field 0.27 0.30 0.30 0.33 0.31	0.30
Our	 The elephants are standing in a grassy field. 0.34 0.33 0.32 0.31 0.32	0.32

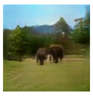
Table 5. Comparison with state-of-the-art model for real-world image synthesis for a story sample: the images with the highest scores were marked with red borders and selected for final visualization.

Method	Generated Images	CLIP Score
BigGAN + CLIP Radford, et al. (2021) [12]	 The elephants are standing in a grassy field. The sheep are running behind the elephants. The sheep approached the elephants. 0.29 0.31 0.34 0.29 0.25	0.29
Dall-E Ramesh, et al. (2021) [47]	 The elephants are standing in a grassy field. The sheep are running behind the elephants. The sheep approached the elephants. 0.29 0.30 0.29 0.33 0.32	0.30
Our	 The elephants are standing in a grassy field. The sheep are running behind the elephants. The sheep approached the elephants. 0.30 0.32 0.31 0.29 0.29	0.30

5.2. Qualitative Results

We evaluated the visual quality of image sequences, generated image sequences that contained multiple scene objects, and visually inspected them. Table 6 shows a scenario applied on the sentences from Figure 1. The results showed that image sequences that were coherent and consistent were preferred over any image sequence, according to our early evaluation. Consistent image sequence indicates visual similarity between images, while coherent image sequences show common characters in the story in terms of overall appearance.

Table 6. A scenario showing the results after vocabulary mapping step.

Characters	Actions and Relation	Resulted Triples	Generation Mode	Generated Images
Two Elephants	Standing in	<elephant, standing in, field>	Single	
Two Sheep, two elephants	Walking on	<sheep, walking on, field>	Multiple	
	Behind	<sheep, behind, elephant>		
Two Sheep, two elephants.	Stand	<sheep, stand, field>	Multiple	
	Beside	<sheep, beside, elephant>		

Our proposed story visualization pipeline saved us time and manual effort in delivering a robust visualization that is ready to use in schools under certain pandemic conditions. We further demonstrated the effectiveness of the proposed pipeline in more complex scenarios such as inter-related sentences and non-common actions. On one hand, using coreference resolution simplified such sentences so that relation extraction reflected the whole sentence meaning, including the story context embedded in previous neighbor sentences. In addition, identifying non-common actions supported the provision of detailed images, while the decomposition of such actions into simple spatio-temporal actions was helpful in explaining how objects and their relationships change as such action occurs.

Robustness: Since our testing set contained sentences of different types, it could exaggerate the contributions of the relation extraction task. Therefore, we resolved this issue by splitting them into two groups. One group included the stories as they are, while the other group included only sentences with co-referenced pronoun resolution. We report that relation extraction performance significantly improved in the second group. This is due to the simplification of inter-related and complicated sentence into multiple simpler sentences, each having a single action along with its participant characters, making it straightforward to extract necessary relations and actions.

Quality of different versions: Concerning the obtained results, we selected some examples from our test set which are shown in Tables 2 and 6, together with the generated images. In the single generated images, for actions such as *standing on*, *attaching*, *etc.*, we can clearly see that the single images visualized the characters and the actions in some cases. However, they were misleading for other non-common actions such as *covering*, *looking at*, etc., as they required more supporting detailed images. In contrast, for the sequence of images, it was observed that non-common actions such as *approaching* were decomposed by starting with simple graphs and progressively building up to more additional details. The addition of objects caused the shift of related objects so that the relationships were respected. However, many images capturing the same type of events can be vastly different in their visual structures, such as those seen in row#2 and row#3 in Table 6. Adding more images promoted the understanding of the input sentence; in contrast, using a single image with cluttered objects resulted in a crowded image plane. Moreover, we observed that using one version instead of the other version was correct based on challenge of using characters only.

Semantic consistency: CLIP guaranteed that each selected image was locally consistent since each selected image matched its corresponding sentence semantically by choosing a higher CLIP score. Our method is of global and local relevance, achieving the highest average rank in comprehensive relevance compared to two state-of-the-art image synthesis methods used for real-world scenarios. Visual examples are shown in Tables 4 and 5, where our method outperformed these two models in terms of global semantic consistency.

Character relevance: The evaluation of the story character classification results indicated that all characters mentioned in the story also appeared in any frame of the image, as was observed from the calculated character relevance score. Thus, this result also proves the effectiveness of our method in maintaining story character accuracy. Essentially, generating image sequences with a higher story coherence score can better comply with the story text, in addition to supporting visual learning

Image quality: Regarding the quality of the generated images, however, the promising results were still limited to generating a few categories of objects. For general stories where multiple objects co-exist with complex relationships, the realism and diversity of the generated images are not satisfactory and remain to be improved in relation to many aspects. Though experimentation with CLIP, the semantic relevance between the two modalities was enhanced.

To reduce the difficulty of synthesizing complex scenes in any real-world setting, we aimed to enrich our pipeline to cover a wide range of characters, objects, and diverse actions. However, we still faced some limitations and technical problems with the image generation task such as the low quality of the synthetized images. Likewise, the generated images still contained many obvious visual artifacts; therefore, models trained for this task are still far from being deployed in any real-world setting. Nevertheless, our work strongly argues that text visualization through a single image only will not produce a meaningful visualization to help with understanding stories. However, proposing a better solution that combines automatic image sequence generation and semi-manual adjustment can ensure flexibility and safety in the learning process.

6. Conclusions

We presented a pipeline overview to illustrate the use of Arabic story text with a sequence of generated images as a fast solution to support distance learning in schools. In summary, we applied an NLP module to process the story text and to obtain an appropriate semantic representation of the main characters, common events, and actions in each sentence. Extensive experimental results on an in-domain visual story test set demonstrated the effectiveness of the proposed pipeline, while the image generation framework was applied to complete the final visualization. Despite the challenges associated with evaluating such systems, our preliminary results showed considerable effectiveness in the adoption of such a pipeline for a coarse visualization task that can be subsequently enhanced. In addition, we expect our contributions to assist with the visualization of stories with a higher image quality when considering more detailed information regarding characters, objects, and relationships.

We are now positioned to conduct an Arabic story annotation effort, followed by implementation of the story visualization, following the outlined task modules detailed previously. Our pipeline and implementation details are algorithmically comprehensible. We anticipate state-of-the-art computer vision and language generation methodologies will provide a number of baselines for Arabic story visualization. For instance, to compare a computer vision algorithm that may over-identify objects against one focused on a specific story domain. Our pipeline allows us to easily prompt for different narrative versions and audiences. In the future, it will be necessary to compare different narrative sequences of images in terms of the cognitive and perception degree of students. Evaluation and release of the final image sequence must take into consideration the narrative goal and audience to ensure a flexible and safe learning environment. In addition, the evaluation must balance the correctness of the action flow, as well as the coherency of the generated story visualization. In particular, new quantitative and qualitative metrics for such tasks must be developed.

In the future, we would like to process more complex and meaningful text with multiple paragraphs. We would also extend the work to produce more professional and intelligent components to support the whole proposed pipeline. Indeed, such as pipeline for a story visualization task can be extended to a video generation task, which is more challenging in terms of the temporal spatial consistency of the video content.

Author Contributions: Conceptualization, J.Z., M.S. and J.M.A.; Methodology, J.Z. and M.S.; Software, J.Z.; Validation, J.Z.; Writing—original draft, J.Z.; Writing—review & editing, M.S., S.A.-M. and J.M.A.; Supervision, M.S., S.A.-M. and J.M.A.; Project administration, S.A.-M.; Funding acquisition, S.A.-M. and J.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was made possible by an NPRP grant #10-0205-170346 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used during this study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Zakraoui, J.; Elloumi, S.; Alja'Am, J.M.; Ben Yahia, S. Improving Arabic Text to Image Mapping Using a Robust Machine Learning Technique. *IEEE Access* **2019**, *7*, 18772–18782. [CrossRef]
2. Ravi, H.; Wang, L.; Muniz, C.; Sigal, L.; Metaxas, D.; Kapadia, M. Show Me a Story: Towards Coherent Neural Story Illustration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7613–7621. [CrossRef]
3. Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; He, X. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1316–1324.
4. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1947–1962. [CrossRef] [PubMed]
5. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative Adversarial Text to Image Synthesis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016.
6. Johnson, J.; Krishna, R.; Stark, M.; Li, L.J.; Shamma, D.; Bernstein, M.; Fei-Fei, L. Image retrieval using scene graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3668–3678.
7. El-Nouby, A.; Sharma, S.; Schulz, H.; Hjelm, D.; El Asri, L.; Ebrahimi Kahou, S.; Bengio, Y.; Taylor, G.W. Keep Drawing It: Iterative language-based image. In Proceedings of the Neural Information Processing Systems (NeurIPS) Visually-Grounded Interaction and Language (ViGIL) Workshop, Montreal, QC, Canada, 7 December 2018.
8. Tobias, H.; Stefan, H.; Stefan, W. Generating Multiple Objects At Spatially Distinct Locations. In Proceedings of the International Conference on Learning Representations (ICLR), Washington, DC, USA, 30 May–2 June 2019.
9. Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; Alja'am, J.; Abou El-Seoud, M. Visualizing Children Stories with Generated Image Sequences. In *Visions and Concepts for Education 4.0. ICBL 2020. Advances in Intelligent Systems and Computing*; Auer, M.E., Centea, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; p. 1314. [CrossRef]
10. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D.A.; et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73. [CrossRef]
11. Johnson, J.; Gupta, A.; Fei-Fei, L. Image Generation from Scene Graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1219–1228.
12. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sutskever, I. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July2021.
13. Joshi, D.; Wang, J.Z.; Li, J. The Story Picturing Engine—A system for automatic text illustration. *ACM Trans. Multimed. Comput. Commun. Appl.* **2006**, *2*, 68–89. [CrossRef]
14. Huang, C.-J.; Li, C.-T.; Shan, M.-K. VizStory: Visualization of Digital Narrative for Fairy Tales. In Proceedings of the 2013 Conference on Technologies and Applications of Artificial Intelligence, Taipei, Taiwan, 6–8 December 2013.
15. Zakraoui, J.; Al Jaam, J.M. A Dynamic Illustration Approach For Arabic Text. In Proceedings of the IEEE 10th GCC Conference & Exhibition (GCC), Salmiya, Kuwait, 19–23 April 2019.
16. Andrej, K.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
17. Krstajić, M.; Najm-Araghi, M.; Mansmann, F.; Keim, D.A. Story Tracker: Incremental visual text analytics of news story development. *Inf. Vis.* **2013**, *12*, 308–323. [CrossRef]

18. Dureja, A.; Pahwa, P. Image retrieval techniques: A survey. *Int. J. Eng. Technol.* **2018**, *7*, 215–219. [CrossRef]
19. Banharnsakun, A. Artificial bee colony algorithm for content-based image retrieval. *Comput. Intell.* **2020**, *36*, 351–367. [CrossRef]
20. Radiano, O.; Graber, Y.; Mahler, M.; Sigal, L.; Shamir, A. Story Albums: Creating Fictional Stories From Personal Photograph Sets. *Comput. Graph. Forum* **2017**, *37*, 19–31. [CrossRef]
21. Gu, Y.; Wang, C.; Ma, J.; Nemiroff, R.; Kao, D.L.; Parra, D. Visualization and recommendation of large image collections toward effective sensemaking. *Inf. Vis.* **2016**, *16*, 21–47. [CrossRef]
22. Hardoon, D.R.; Szedmak, S.; Shawe-Taylor, J. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Comput.* **2004**, *16*, 2639–2664. [CrossRef] [PubMed]
23. Yitong, L.; Zhe, G.; Yelong, S.; Jingjing, L.; Yu, C.; Yuexin, W.; Lawrence, C.; David, C.; Jianfeng, G. StoryGAN: A Sequential Conditional GAN for Story Visualization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
24. Zakraoui, J.; Saleh, M.; Aljaam; Jihad, M. *Text-to-Picture Tools, Systems and Approaches: A Survey. Journal of Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 78, pp. 22833–22859.
25. Huang, T.-H.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R.; He, P.K.X.; Ba-tra, D.; Zitnick, L.; et al. Visual Storytelling. 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. 233–1239.
26. Chen, S.; Liu, B.; Fu, J.; Song, R.; Jin, Q.; Lin, P.; Qi, X.; Wang, C.; Zhou, J. Neural Storyboard Artist: Visualizing Stories with Coherent Image Sequences. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), 21–25 October 2019*; Association for Computing Machinery: New York, NY, USA, 2019.
27. Fang, F.; Yi, M.; Feng, H.; Hu, S.; Xiao, C. Narrative Collage of Image Collections by Scene Graph Recombination. *IEEE Trans. Vis. Comput. Graph.* **2017**, *24*, 2559–2572. [CrossRef] [PubMed]
28. Fang, F.; Luo, F.; Zhang, H.-P.; Zhou, H.-J.; Chow, A.L.H.; Xiao, C.-X. A Comprehensive Pipeline for Complex Text-to-Image Synthesis. *J. Comput. Sci. Technol.* **2020**, *35*, 522–537. [CrossRef]
29. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
30. Wenbo, L.; Pengchuan, Z.; Lei, Z.; Qiuyuan, H.; Xiaodong, H.; Siwei, L.; Jianfeng, G. Object-driven Text-to-Image Synthesis via Ad-versarial Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12166–12174.
31. Tingting, Q.; Jing, Z.; Duanqing, X.; Dacheng, T. MirrorGAN: Learning Text-to-image Generation by Redescription. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1505–1514.
32. Han, Z.; Tao, X.; Hongsheng, L.; Shaoting, Z.; Xiaogang, W.; Xiaolei, H.; Dimitris, M. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5908–5916.
33. Shikhar, S.; Dendi, S.; Vincent, M.; Samira, E.K.; Yoshua, B. ChatPainter: Improving Text to Image Generation using Dialogue. *arXiv* **2018**, arXiv:1802.08216.
34. Tim, S.; Ian, G.; Wojciech, Z.; Vicki, C.; Alec, R.; Xi, C. Improved techniques for training gans. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), New York, NY, USA, 5–10 December 2016; pp. 2234–2242.
35. Gangyan, Z.; Zhaohui, L.; Yuan, Z. PororoGAN: An Improved Story Visualization Model on Pororo-SV Dataset. In Proceedings of the 3rd International Conference on Computer Science and Artificial Intelligence, Beijing, China, 6–8 December 2019.
36. Song, Y.-Z.; Tam, Z.-R.; Chen, H.-J.; Lu, H.-H.; Shuai, H.-H. Character-Preserving Coherent Story Visualization. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; Volume 12362.
37. Li, C.; Kong, L.; Zhou, Z. Improved-StoryGAN for sequential images visualization. *J. Vis. Commun. Image Repre-Sentation* **2020**, *73*, 102956. [CrossRef]
38. Maharana, A.; Hannan, D.; Bansal, M. Improving Generation and Evaluation of Visual Stories via Semantic Consistency. *arXiv* **2021**, arXiv:2105.10026.
39. Zakraoui, J.; Saleh, M.; Al-Maadeed, S.; Jaam, J.M. Improving text-to-image generation with object layout guidance. *Multimed. Tools Appl.* **2021**, *80*, 27423–27443. [CrossRef]
40. Tan, F.; Feng, S.; Ordonez, V. Text2Scene: Generating Compositional Scenes From Textual Descriptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6703–6712.
41. Ilya, S.; Oriol, V.; Quoc, V.L. Sequence to sequence learningwith neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2 (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
42. Zakraoui, J.; Saleh, M.; Asghar, U.; AlJa'Am, J.M.; Al-Maadeed, S. Generating Images from Arabic Story-Text using Scene Graph. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 469–475.

43. Wolf, T.; Ravenscroft, J.; Chaumond, J.; Rebo, M. Coreference Resolution in Spacy with Neural Networks. HuggingFace. 2018. Available online: https://github.com/huggingface/neuralcoref (accessed on 15 January 2021).
44. Jingwei, J.; Ranjay, K.; Li, F.-F.; Juan Carlos, N. Action Genome: Actions As Compositions of Spatio-Temporal Scene Graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10236–10247.
45. Lillo, I.; Soto, A.; Niebles, J.C. Discriminative Hierarchical Modeling of Spatio-temporally Composable Human Activities. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 812–819. [CrossRef]
46. Kurby, C.A.; Zacks, J.M. Segmentation in the perception and memory of events. *Trends Cogn. Sci.* **2008**, *12*, 72–79. [CrossRef] [PubMed]
47. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 18–24 July 2021.

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.