International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Visual Storytelling: A Generative Adversarial Networks (GANs) and Graph Embedding Framework

[1]K. Dinesh Kumar, [2]Dr. Sarot Srang, [3]Dr. Dona Valy
[1]Mechatronics and Information Technology, Institute of Technology of Cambodia, Phnom Penh, Cambodia.
vkjdinesh@gmail.com
[2]Mechatronics and Information Technology, Institute of Technology of Cambodia, Phnom Penh, Cambodia.
srangsarot@itc.edu.kh
[3]Department of Information and Communication Engineering, Institute of Technology of Cambodia, Phnom Penh, Cambodia.
dona@itc.edu.kh

Abstract— Visual storytelling is a powerful educational tool, using image sequences to convey complex ideas and establish emotional connections with the audience. A study at the Chinese University of Hong Kong found that 92.7% of students prefer visual storytelling through animation over text alone [21]. Our approach integrates dual coding and propositional theory to generate visual representations of text, such as graphs and images, thereby enhancing students' memory retention and visualization skills. We use Generative Adversarial Networks (GANs) with graph data to generate images while preserving semantic consistency across objects, encompassing their attributes and relationships. By incorporating graph embedding, which includes node and relation embedding, we further enhance the semantic consistency of the generated high-quality images, improving the effectiveness of visual storytelling in education.

Keywords-*:* *GANs, Graph Embedding, Information Extraction, Visual Storytelling*

I. Introduction

According to Allan Paivio's dual coding theory, our brains grasp information more effectively when we blend both verbal and non-verbal elements [1]. Our working memory consists of verbal and visuospatial components, and their integration can significantly improve our understanding. In simpler terms, images can complement text, leading to improved comprehension. Without this cooperation, there is a chance of burdening our working memory and reducing performance. The inclusion of images can substantially boost learning, and mental visualization further facilitates learning and enriches reading comprehension.

The human brain can quickly understand whole images in only 13 milliseconds. This speed helps us look at different things because our eyes shift focus three times each second. Recent research findings suggest that about 65% of people learn better when seeing images instead of just text and speech [2]. Images can quickly explain complex ideas and evoke strong emotions that lengthy explanations cannot. In fact, roughly 75% of the information our brains process is visual [3]. This means that seeing, understanding, and connecting with images helps us understand and remember things better. This insight has significant implications for education, where students can benefit from visual aids in various forms like text-to-images, videos, graphs, cartoons, games, and flashcards. Teachers and students can use these visuals to enhance comprehension and make connections between different elements mentioned in the text.

Not enough research has been done to confirm if words and pictures are the sole way humans remember things.

Most research has focused on words and images and hasn't explored other possibilities. When we don't link a word with an image, it becomes harder to remember that word later on. This is one of the limitations of the dual coding theory. An alternative perspective comes from John Anderson and Gordon Bower, who proposed the propositional theory as a different way to understand how our minds represent knowledge. According to this theory, our minds store knowledge as propositions rather than images. Each proposition represents an object or concept, and the connections between these propositions describe the relationships between these concepts, forming a network of propositions [4].

However, creating a coherent visual narrative remains a challenge in the fields of Natural Language Processing, Computer Vision, and Deep Learning. These areas continuously work to generate high-quality image sequences that maintain semantic consistency while telling engaging stories. Our goal goes beyond image generation; we aim to apply our proposed model in educational settings to boost students' analytical thinking skills through the power of visual storytelling by integrating the text into image generation along with the knowledge graph generation of the text sentence. Our proposed architecture integrates the dual coding theory and the propositional theory to provide an effective platform for students to learn complex concepts by visualizing the text in images and graphs.

This research article covers the following key aspects: (i) The extraction of a list of objects and their attributes, along with the relationships between them, using Natural Language Processing techniques. (ii) The construction of a graph based on the propositional theory. (iii) Graph Embedding techniques to preprocess the data to generate images. (iv) Image Generation using the Generative Adversarial Networks (GANs). (v) The incorporation of graph embedding into image generation while preserving the semantic meaning of the text in the images.

II. VISUAL STORYTELLING

The current landscape of image generation models has seen significant advancements in producing exceptionally high-quality, photorealistic images. Models such as Stable Diffusion, Mid-Journey, DALL E2, and more have demonstrated remarkable capabilities in this regard. Our research, however, is centered on a novel approach that integrates the dual coding theory and the propositional theory with a specific focus on enhancing the educational domain. Our methodology commences with the extraction of information and relationships from natural language text, achieved through the application of Natural Language Processing techniques and Graph Theory principles.

A. Information Extraction

Our research focused on extracting information from English language sentences to create a graph that serves as a graphical representation of natural language. Information extraction is an important component of NLP, responsible for retrieving structured data from unstructured text. Document-level Information Extraction (IE) presents several challenges, including entity coreference resolution, the ability to reason across extensive contexts, and the need for common-sense reasoning, often constrained by specific domains and languages [5].

Each linguistic expression comprises a sequence of words, and the primary goal is to capture specific information elements, including Event Mention (phrases and sentences signifying an event), Event Trigger (the verb denoting the event occurrence), and Event Type (encompassing the array of events referenced within the dataset). Furthermore, it is to identify Augment Mention, which pertains to entities furnishing supplementary details about the event, encompassing the who, what, when, where, and how of the event's occurrence, and Augment Role, denoting the associated argument for the entity. Following the event extraction, the author [6] extracted the relations involved in a sentence by predicting attributes, and relationships between the entities. Intra-sentence Relation is to find the relationship between entities within a sentence and the extracted features are referred to as the local feature. Inter-sentence relation is to find the relationship between the entities across multiple sentences and the features are referred to as the global features.

We possess a variety of datasets for Event Extraction (EE) and Relation Extraction (RE). In the realm of Relation Extraction, there are datasets like Drug-gene-mutation (DGM), Gene-disease Association (GDA), and manually annotated datasets such as CDR, which cover chemicals, diseases, and chemical-induced diseases (CID). Additionally, several RE datasets cater to different domains and languages, including DocRED, RE-DocRED, SciREX, and HacRED [6].

Event Extraction datasets are primarily sourced from the news and financial domains. The news domain provides a wealth of information, covering a wide range of events, including social emergencies and incidents related to human life. In the financial domain, datasets are created to identify financial risks and profitable opportunities. Notable datasets for Relation Extraction include ACE-2005, MUC-4, WikiEvents, Roles Across Multiple Sentences (RAMS), DCFEE, ChFinAnn, and DuEE-FinFin [6].

Furthermore, the author [7] considered metrics such as Precision (P), Recall, and Macro-F1 scores when evaluating Information Extraction. For Relation Extraction, the author [7] employs an additional evaluation metric known as Ign F1, which calculates the F1 score while excluding relational facts shared by the training, development, and test sets.

B. Graph Construction

Our research Graph data is becoming progressively influential in conveying information efficiently to others. For instance, the increasing use of social media networks, recommendation systems, and Geolocation identifications places substantial reliance on graph technologies [8]. Whether consciously or subconsciously, we often elucidate concepts by illustrating them through graphs, identifying elements, and establishing connections between them to elucidate ideas. Our mental processes involve structuring thoughts into visual forms, effectively crafting what can be considered as graphs.

Graphs are powerful tools for understanding and analyzing complex systems in various disciplines. Knowledge graphs have become most popular among researchers as an effective way of conveying information. Within the graph, all things are linked, much like in our daily lives. It serves as the structure of our digital existence, so mastering navigation is vital in this digital age. Graphs illustrate the components of a real-world issue and their connections, showcasing how they interact with each other. Sometimes, representing these relationships in a graph can be simplified to a line with a specific weight, indicating strength or volume.

A graph is composed of nodes (sometimes referred to as vertices) and the links connecting these nodes, which are depicted as edges (or links). Knowledge graphs, often referred to as semantic webs, illustrate connections between entities found in the real world, including objects, events, situations, concepts, and more. This data is stored in a graph database and displayed graphically. Within knowledge graphs, nodes signify interconnected entities, while edges symbolize various relationships between these entities. The primary aim of knowledge graphs is to accumulate and communicate real-world information.

As a result, our approach is highly customized to extract objects, attributes, and their relationships, allowing us to create a graph that represents natural language. This graph can be employed in educational contexts to enhance students' critical and analytical thinking abilities.

A knowledge Graph (KG) mainly contains relationship triples of the form (Head, Relation, Tail) which is represented as (h, r, t). The KG is a graphical representation in which nodes represent the head and tail, and the edges represent the relationship between them. Google and Bing are using the KG to represent the real-world entities in which they compute the semantic similarity with the nodes from the user search queries [11]. By the KG, they enhance the search experience by aligning user search queries with the semantic connection.

Our research focuses on creating images from text with semantic coherence. Both current and forthcoming data will adopt a graph-based format. Hence, our approach leverages graph techniques to transform text into visual graphs. These visual representations are designed to effectively communicate information in educational contexts, enhancing the learning experience for students. Our approach differentiates from the conventional knowledge graph generation pattern, as our goal is to produce photo-realistic images from textual input while maintaining semantic consistency.

C. Embedding Techniques

Our research aims to generate images based on natural descriptions by utilizing Generative Adversarial Networks (GANs). To achieve this, we first convert the input text sequence into text embeddings, which can be done using techniques like GloVe, Skip-gram, and Word2Vec, depending on the application. Since GANs are designed for continuous data and text graphs are discrete in nature, we need to transform the graphical representation of text into an embedding vector. There are various methods available for performing this graph embedding process.

A Graph embedding produces the fixed-length vector representation for each node in the graph. This generated embedding is a lower-dimensional representation of the graph. Graph embedding involves several key components, including the Similarity Function: This function is employed to assess the similarity between nodes within the graph. Encoder Function: It is responsible for producing the node embeddings. Decoder Function: This component plays a role in reconstructing the pairwise similarity. It also serves as a metric to evaluate the quality of the reconstruction [12].

The generated directed graph in which edges have orientation.

$$V : Set\ of\ nodes$$
$$E : Set\ of\ edges$$

$$\phi : E \rightarrow \{(x, y) | (x, y) \in V^2\ and\ x \neq \mathrm{y}\} \quad (1)$$

An incidence function(ϕ) maps each edge to an ordered pair, which is associated with two distinct nodes [22]. In the context of this directed graph, self-loops, such as (x, x), are not permitted. However, the incidence function can be modified to

$$\phi : E \rightarrow \{(x, y) | (x, y) \in V^2\} \quad (2)$$

To avoid ambiguity, self-loops are allowed in this multi-directed graph [22].

Node2Vec is an efficient semi-supervised algorithm designed for learning features in networks. Leveraging techniques pioneered in Natural Language Processing (NLP), it fine-tunes a custom objective function using Stochastic Gradient Descent. The outcome is feature representations that prioritize the preservation of network neighborhoods in a d-dimensional feature space. It does so by employing a 2nd order random walk approach to create network neighborhoods for nodes.

Each node in the graph is like a single word and a random walk like a sentence. They use these "sentences" in a skip-gram model or a continuous bag of words model. This helps us analyze paths from random walks, much like we study sentences in text using regular data-mining techniques.

In Random walk, the source node u, simulates the random walk of fixed length l. ci denotes the ith node in the walk with c0=u. The following distribution is used to generate the nodes ci.

$$P\ (c_i = x\ | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{\mathrm{Z}} & if\ (v, x)\ \in \mathrm{E} \\ 0 & otherwise \end{cases} \quad (3)$$

where π_{vx} is the unnormalized transition probability between nodes v and x, and Z is the normalizing constant [13].

For any graph, G = {V, E} contains the nodes and edges. A is the adjacency matrix of size {N x N}. {X} is the node feature matrix of size {N x M}. A Graph Neural Network (GNN) is a customizable operation that works on all aspects of a graph, including its nodes, edges, and global context. This operation maintains the symmetries of the graph, ensuring that it remains invariant under permutations. Graph Neural Networks follow a sequence of steps, including Locality, Aggregation, and Composition (or Stacking) layers. Locality: This step involves creating node embeddings by considering the local neighborhood of each node in the graph. Aggregation: In the Aggregation step, we use a summation operation. It's important because it provides permutation invariance, meaning that the order of elements doesn't affect the outcome. Every node in the graph contributes to the computation graph. Composition (or Stacking) layers: These layers aim to generate lower-dimensional feature vectors from the higher-dimensional ones. This process is crucial for effectively processing graph data [14].

D. Text to Image Generation

Deep learning models are harnessed to extract advanced features from diverse data distributions, employing multiple layers of artificial neurons. These deep learning models encompass Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), as well as generative models like Variational Auto-encoders (VAE), Generative Adversarial Networks (GAN), and Normalizing Flows (NF).

Generative modeling finds application across various domains, including image denoising, structured prediction, inpainting, and super-resolution. Deep neural networks are instrumental in generative modeling, as they facilitate the modeling of high-dimensional distributions from extensive datasets. However, while this approach yields favorable outcomes in domain-specific tasks, it presents challenges during the training phase. Training accuracy may be adversely affected by fluctuations in data distributions and the complexity of deep learning architectures, making training a non-trivial task.

Generative Adversarial Networks (GAN) to overcome the challenges faced by earlier generative models. In different domains, there has been substantial expansion in the realm of Generative Adversarial Networks (GANs), constituting a vibrant area of research within the field of deep learning. Various GAN variants, including DCGAN, CGAN, Stack GAN, Stack GAN++, WGAN, Big GAN, and others, have demonstrated superior performance. The proliferation of these diverse GAN models has facilitated the creation of numerous domain-specific applications by integrating natural language processing and computer vision. The process of creating images based on textual descriptions is commonly known as Text-to-Image Synthesis, a technique applicable in diverse domains like education and art generation [15]. Utilizing methods like GANs and Autoencoders, it becomes possible to generate images from text descriptions, ensuring a high degree of confidence in the algorithm's comprehension of image features.

However, The GAN can generate images similar to the real data but the limitation is to generate diverse images because they don’t have any specific probability density estimation [15]. To address this, researchers often employ a combination of techniques alongside GANs to enhance their performance. For instance, Variational Autoencoders (VAEs) are effective at generating diverse images, and by integrating them, it's possible to improve both image quality and diversity.

In the world of GANs, we have two key players: the Generator (G) and the Discriminator (D). Think of the Generator as an artist who creates fake images, trying to make them look as real as possible. The Discriminator, on the other hand, acts like an art critic. It's given two pictures: one from the Generator and one real picture. The job of the Discriminator is to tell them apart and become an expert at it. So, we can say the Generator is like a painter of fakes, and the Discriminator is like a detective that can spot the real deal. The basic GAN operation is as below: A generator(G) generates synthetic image G(z) from a random noise vector(z) and the discriminator will differentiate the generated image and the real image. To enhance performance, this basic process can be refined in multiple ways [16]. The objective function of GAN is

$$\min_G \max_D \mathbb{E}_{x\sim P_r}[\log(D(x))] + \mathbb{E}_{z\sim P_z}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (4)$$

Nonetheless, the input x comes from our data distribution (Pr), while z is taken from a Uniform or Gaussian distribution (Pz). To make things more interesting, we can also throw in another input, let's call it c, into the mix with the random noise z. This way, we're generating an image using both G(c, z). Now, this c input can be anything we want, like an object's characteristic or an image class, among other things [16].

The GAN produces a wide range of high-quality images, building upon various modifications to the original GAN structure. However, it's challenging to assess both the generated images and the model's efficiency. This evaluation can be achieved through metrics like the Inception Score (IS) and the Fréchet inception distance (FID). The Inception Score (IS) evaluates the GAN's performance by considering the diversity of generated images and how closely they resemble real scenes. A high IS score is attained when both conditions are met, while a low score is given if either condition is not satisfied. IS scores can vary infinitely, with higher scores indicating better image quality and diversity [17].

The Fréchet inception distance (FID) score assesses the quality of generated images by comparing the statistical features of generated images with real ones from the desired domain. Measures the similarity between feature representations of generated and real images. The FID score of the model is 0 to infinity, lower is better [17].

III. Model Architecture

The goal of this architecture is to produce high-quality images while maintaining semantic consistency with the text description. To achieve the desired outcome, let's break down the process step by step. (i). Node and Relation Extraction, (ii). Graph Generation and Embedding, (iii). Dataset Preparation (iv). Generative Adversarial Networks (GANs).

Our objective is to create an image based on a text description while ensuring that the objects, their attributes, and the relationships between them in the text are semantically consistent in the generated image. To achieve this, we extract information about the objects, attributes, and relationships from the text and construct a graph. This graph serves as an optimized representation of the input, maintaining semantic consistency throughout the process.

A. Node and Relation Extraction

To build the graph, it's essential to understand what a graph is and determine the most appropriate type for our method. Having covered the fundamental graph terminology in Section II, we can delve into the graph construction process. Our approach involves creating a weighted directed graph, where nodes represent objects, and the edges between nodes signify the relationships with corresponding labels.

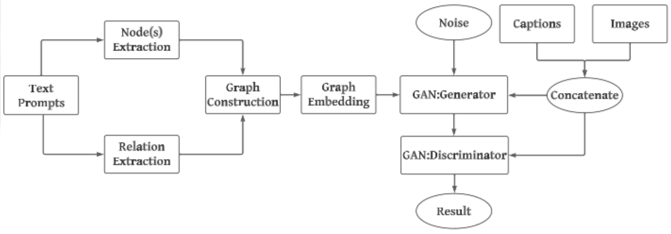


Figure 1:Model Architecture

To gather the information needed for constructing the graph, we utilize a rule-based grammar that extracts relevant details from each sentence.

We employ Natural Language Processing (NLP) techniques, specifically part-of-speech tagging, using the NLTK library to analyze the given text description. As our focus is on the English language and the need for maintaining semantic consistency in generated images, we didn't rely on a specific graph-based dataset or methods like Knowledge Graphs (KG). Instead, our objective is to extract a comprehensive list of objects along with their attributes, encompassing details such as color, shape, and other pertinent characteristics, which we represent as nodes in our framework. Additionally, for the edges in our structure, we extract the actions and spatial relationships between these nodes, guided by the principles of Propositional Theory.

To achieve this, we extract objects and their attributes as nodes using the following grammar:

Nodes: {<DT|JJ|JJR|JJS|RB|RBR|RBS|CD>*<NN. *>+}

In this grammar, nodes encompass determiners, adjectives, adverbs, and nouns. This approach allows us to extract meaningful nodes for maintaining semantic consistency.

Relation: {<IN|VBG|VBN|VBP|VBZ>}

In this process, we specifically capture prepositions and verbs that appear between the nodes in the sentence, which we utilize as edges or relations. By doing so, the extracted nodes and relations work harmoniously to ensure semantic consistency and optimize the text descriptions effectively. This serves as one of the outcomes, which can be employed in the field of education to help students grasp the connections between words in a sentence, sentence structures, and the visual representation of the text.

B. The Graph Generation & Embedding

The nodes and relations are now prepared, and the next step involves constructing the graph. This graph is essential for the random walk process, which aids in identifying the node and relation embeddings, serving as input for the GAN to generate images. During the construction of the text input's graph, we employ sentence segmentation and extract nodes and relations from the sentences to build the graph. The sequence graph is formed for each sentence by connecting the different nodes within the sentence using their relationships. Given that a sentence can have multiple nodes, the graph structure follows a "Source Node: Relation: Destination Node" format, maintaining the sequence order of the sentence. After the sentence's graph has been created, the next task is graph embedding. Among various graph embedding methods, we utilize the Node2Vec technique, which provides graph embedding as node embedding.

In contrast to other data types, graphs lack a natural order or reference point. Sequence data embedding, on the other hand, exhibits a well-defined structure. Even for a given graph, the adjacency matrix may not be unique. The concept of node embedding or node representation involves converting each node into its respective embedding vectors. These embeddings should encompass the entire graph network, node relationships, and other relevant information. The node embedding is decided based on the principles of similarity between the nodes [18].

The procedure is

i. Determine the size of the embedding space. In our case, we are using the 300 as the embedding dimension.

ii. Initialize the embeddings for nodes, edges, or graphs randomly.

iii. Enhance the embeddings iteratively to better represent network similarities.

Graph embedding can be accomplished through three main methods: Factorization-based, Random Walk-based, and Deep Learning-based techniques [19]. In our current approach, we have introduced basic graph embedding into GAN for generating images with semantic consistency. Our future research will explore the integration of various embedding techniques to assess the model's effectiveness.

C. *Dataset Preparation*

In our research, we have made use of two primary datasets: Flicker8k and Oxford-102(Flowers). These datasets include around five captions for each image. Our research approach, founded on theoretical principles, is centered on the creation of caption-image pairs for GAN training. What sets our approach apart is the innovative use of Graph Embedding as input for generating images, and we've tailored our dataset accordingly.

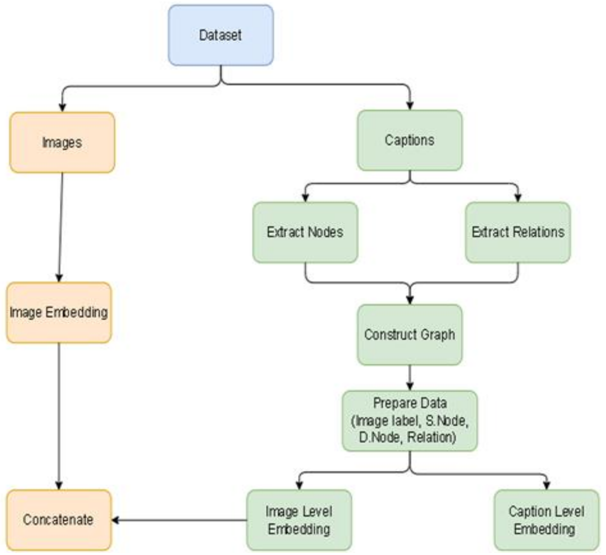


Figure 2: Data preprocessing

To achieve this, we have undertaken the task of constructing graphs for the captions and then combining them with their corresponding images to create pairs for model training. We offer the flexibility of conducting graph embedding at either the caption level or the image level.

When opting for caption-level embedding, we build a distinct graph for each caption, which is later merged with the image embedding. This results in multiple pairs of inputs for each image, based on the number of captions associated with it. This approach provides more precise input for GAN training.

Alternatively, we can group the captions related to each image and create a single graph for the entire group. This process yields a pair that includes the graph embedding and the image.

D. *The Generative Adversarial Networks (GANs)*

The GAN Generator and Discriminator are trained using the dataset's image-level embeddings. In the process, the node embeddings derived from the provided text descriptions, along with random noise, serve as inputs to the GAN generator, resulting in the generation of synthetic images. These synthetic images are then subjected to the discriminator for evaluation and classification.

In our GAN framework, both the Generator and Discriminator are trained using image-level embeddings from our dataset. During this process, the node embeddings extracted from the given text descriptions, in combination with random noise, are employed as inputs to the GAN Generator. This results in the generation of synthetic images, which are subsequently assessed and categorized by the Discriminator.

To advance our GAN model's performance and ensure semantic consistency between text and image, we've updated the loss function. Our novel semantic loss, grounded in the context of our work, hinges on the cosine similarity between the node embedding of the generated image and the node embedding of the associated text description. The semantic loss enhances the GAN training process, fostering the generation of images that exhibit greater semantic coherence.

The semantic loss formula is defined as follows:

Semantic Loss = 1 - μ (Cosine Similarity (Image Node Embedding, Text Node Embeddings))

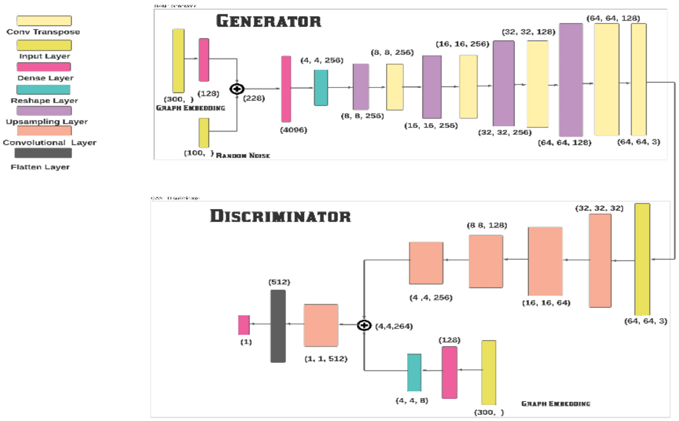


Figure 3: GANs layers architecture

This loss function calculates the semantic loss as 1 minus the mean cosine similarity between the node embeddings of the generated images and the node embeddings of the text descriptions. By incorporating the semantic loss into the

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

standard GAN generator loss, such as Binary Cross-Entropy, we ensure that the images produced by our GAN have semantic consistency with the provided text descriptions. To enhance the GAN's performance, we have been actively exploring the integration of relation embedding, graph embedding, and node embedding.

IV. Result and Discussion

The proposed approach builds upon our base model [20]. The base model has been tested with two main datasets: the Oxford-102 flower dataset and the CUB dataset. Additionally, we assessed the performance of our model using both the Oxford-102 flower dataset and the Flicker8k dataset. Notably, the Flicker8k dataset is known for its inclusion of complex scenes similar to MSCOCO, enabling us to evaluate our model's performance in challenging scenarios. It's worth noting that for both datasets, we had the benefit of five captions available for each image. The Oxford-102 dataset encompasses 8,189 flower images categorized into 102 distinct classes, while the Flicker8k dataset comprises 8,091 images, offering a wide variety of scenes and objects.

We began by grouping the five captions corresponding to each image and subsequently performed part-of-speech tagging on these grouped captions. As previously discussed in Section II, we extracted the nodes and relations from these grouped captions.

In the second stage, we constructed an image-level graph for each image. This decision was driven by the fact that each caption contains multiple nodes and relations. It's important to note that the extracted nodes and relations from the entire captions also include multiple nodes and relations. To build the graph, we used the 'Source Node: Relation: Destination Node' pattern, with a focus on sequence graph construction. We constructed the vocabulary for graph embedding through the described process. Given our intended use in the educational domain, this graph serves as a visual representation of the text. It aids in helping students comprehend sentence structures and encourages deeper thinking about the relationships between nodes within a sentence.

Node embeddings have been generated by applying Node2Vec to the captions, and these embeddings have been combined with the image embeddings using GoogLeNet[20]. The GAN is trained on this paired dataset, incorporating both the node embeddings and image embeddings. The resulting generated images have a fixed size of 64 × 64 × 3.

The given text prompt for image generation follows the previously outlined process to create nodes and their corresponding relationships. Node2Vec is utilized to obtain node embeddings through random walks on the graph. These node embeddings are integrated with randomly generated noise and processed layer by layer to generate the images. The resulting graph embedding is subsequently provided to the discriminator for image differentiation. This marks our initial effort to generate images from the graph, particularly focusing on node embeddings. We are actively extending this approach to enhance image generation with semantic consistency.

The experimental result of our approaches:

Sentence: a group of people riding motor scooters stopped at a stop light.

Nodes: ['a group', 'people', 'motor', 'scooters', 'a stop', 'light']

A flower with a yellow center, surrounded by white petals with a purple shade. (Oxford-102)

Edges: ['of'], ['riding'], ['at']

The graph vocabulary consists of five pairs of nodes, each with its corresponding relation, for a single sentence.

TABLE I. Graph Vocabulary for a sentence

Source Node	Relation	Destination Node
a group	of	people
people	riding	motor
motor	---	scooters
scooters	stopped at	a stop
a stop	---	light

When there is a relationship between nodes, we represent it using red edges accompanied by the respective relation, whereas empty edges are depicted in blue. The color-coding system serves as a visual aid for students to comprehend relationships within the graph. This color coding is applied to each sentence. The node embedding dimension is 300, ensuring that each node possesses a distinct and unique embedding derived from this method. It's a sequence graph, designed for sentence-based graph generation. If we can create a graph for a group of captions, we can enhance its efficiency. Here is the generated image of our model using the node embedding.

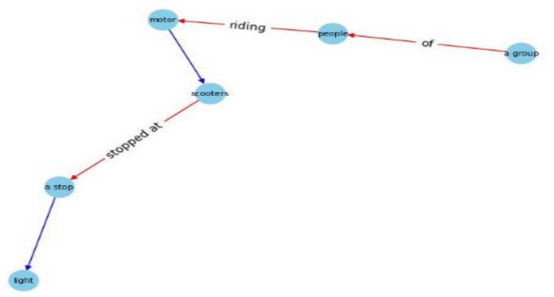
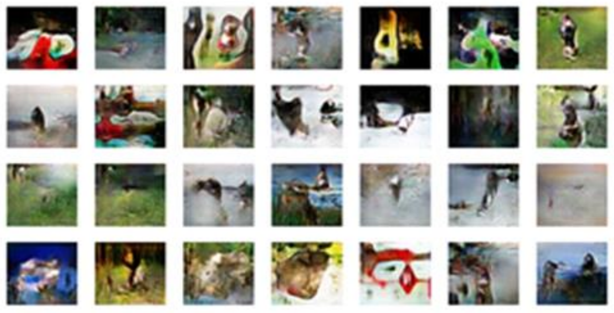


Figure 4: Generated Graph

A Boston Terrier is running on lush green grass in front of a white fence (Flicker8k dataset)



Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

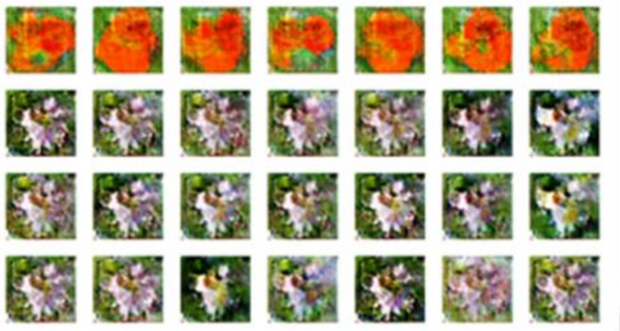


Figure 5: The generated images from the GAN and Graph Embedding architecture

The images shown above were generated during the 400th epoch of training. However, they were not deemed semantically consistent in human evaluations. To enhance the model's performance, the focus now lies on improving the graph embedding, specifically the node embeddings. Strengthening the list of nodes within the generated images is essential. To achieve a more detailed scene from the model, we should explore the inclusion of sentence embeddings and enhance the images by introducing graph embeddings for nodes and relations. This will help enrich the scenes depicted in the generated images. Because of the current constraints in data processing within the GAN, transitioning to graph-based data requires us to improve the graph embedding. This enhancement is important for generating high-quality and semantically consistent images.

V. Conclusion

In conclusion, our approach to visual storytelling focuses on creating an efficient model for generating images and graphs from text descriptions. We construct graphs to represent objects, attributes, and their relationships, guided by information extracted from the text using NLP's part-of-speech tagging. This approach aids students in comprehending sentence structures, keywords, and their interconnections. By training our model with graph data, we aim to enhance the visual storytelling experience by emphasizing objects, relations, and attributes. Leveraging the power of Generative Adversarial Networks (GANs) and graph data, we produce images that maintain semantic consistency. In the future, our research will expand to integrate relation and graph embedding, as well as sentence embedding, with the goal of creating a more effective visual storytelling framework, particularly for educational purposes.

References

[1]. Flávia Schechtman Belham, "How Images and Imagination Can Enhance Learning" GUEST POST: How Images and Imagination Can Enhance Learning — The Learning Scientists

[2[. Vinnie Wong, "What is visual Storytelling? How to engage and Inspire Audience", Aug 31, 2023, What Is Visual Storytelling? How to Engage and Inspire Audiences (piktochart.com).

[3]. Jamal Raiyn, "The Role of Visual Learning in Improving Students' High-Order Thinking Skills", Journal of Education and Practice, www.iiste.org ISSN 2222-1735 (Paper) ISSN 2222-288X , (Online) Vol.7, No.24, 2016.

[4]. Eva Maria Rodríguez "Allan Paivio and His Dual Coding Theory", 21 December 2022, Allan Paivio and His Dual Coding Theory - Exploring your mind.

[5]. Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level Event Extraction via Parallel Prediction Networks. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6298–6308, Online. Association for Computational Linguistics.

[6]. Hanwen Zheng Sijia Wang Lifu Huang, "A Survey of Document-Level Information Extraction", Sep 2023, 2309.13249.pdf (arxiv.org).

[7]. Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. "DocRED: A large-scale document-level relation extraction dataset", In Proceedings of ACL 2019. 2019b.

[8]. Yuanyuan Tian, "The World of Graph Databases from An Industry Perspective", https://arxiv.org/pdf/2211.13170.pdf, Nov 2022.

[9]. Iain Brown, "Decoding the World of Graph Data: Applications, Techniques, and Tools", https://www.linkedin.com/pulse/decoding-world-graph-data-applications-techniques-iain-brown-ph-d-/, September 2023.

[10]. Resul Das, Mucahit Soylu," A key review on graph data science: The power of graphs in scientific studies", Chemometrics and Intelligent Laboratory Systems, ISSN 0169-7439, https://doi.org/10.1016/j.chemolab.2023.104896.

[11]. Adnan Zeb, Summaya Saif, Junde Chen, Anwar Ul Haq, Zhiguo Gong, Defu Zhang," Complex graph convolutional network for link prediction in knowledge graphs", Expert Systems with Applications, Volume 200, 2022, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.116796.

[12]. Graph Embedding, https://neo4j.com/developer/graph-data-science/graph-embeddings/#:~:text=A%20graph%20embedding%20determines%20a,Figure%201.,

[13]. Aditya Grover, Jure Leskovec, "node2vec: Scalable Feature Learning for Networks", August 13 - 17, 2016, DOI: http://dx.doi.org/10.1145/2939672.2939754.

[14]. Sanchez-Lengeling, et al., "A Gentle Introduction to Graph Neural Networks", Distill, 2021

[15]. Haileleol Tibebu, Aadil Malik, Varuna De Silva, "Text to Image Synthesis using Stacked Conditional Variational Autoencoders and Conditional Generative Adversarial Networks", Intelligent Computing: Proceedings of the 2022 Computing Conference, springer, 2022.

[16]. K. Dinesh Kumar, Sarot Srang and Dona Valy, "A Review of Generative Adversarial Networks (GANs) for Technology-Assisted Learning: Solving Teaching and Learning Challenges," 2022 International Conference on

Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2022, pp. 820-826, IEEE Xplore. doi: 10.1109/ICACRS55517.2022.10029021.

[17].K. Dinesh Kumar, Sarot Srang, and Dona Valy, “Evaluating Text-to-Image GANs Performance: A Comparative Analysis of Evaluation Metrics”, International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 8s, pp. 618–627, Aug. 2023. Scopus Index. DOI: https://doi.org/10.17762/ijritcc.v11i8s.7248.

[18]. Yves Boutellier, Node embeddings for Beginners. Node embeddings can be hard in the… | by Yves Boutellier | Towards Data Science, July 2021.

[19]. Introduction to Node Embedding (memgraph.com)

[20].Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswara, “Generative Adversarial Text to Image Synthesis”, Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

[21]. Ming Li, Chi Wai Lai, Wai Man Szeto, “Whiteboard Animations for Flipped Classrooms in a Common Core Science General Education Course”, 5th International Conference on Higher Education Advances (HEAd’19) Universitat Politecnica de Val ` encia, Val ` encia, 2019 ` DOI: http://dx.doi.org/10.4995/HEAd19.2019.9250.

[22]. Wiki2, Graph theory — Wikipedia Republished // WIKI 2