Interactive Sketch Mate

Kavya M Adiga*, Akshaya R*, Shrinidhi S*, Rajasekar V*
*Computer Science and Engineering,
SRM Institute of Science and Technology, Chennai,
India.

***Abstract*—The applications and uses of text-to-image generation are growing every day and a lot of researchers are focusing on building efficient systems to achieve the results. But one main aspect of the system is the text that is given as input. A normal user's text need not be precise and in the proper format as the system can accept. Natural Language Processing is a field that deals with texts and thereby can be used to modify the user's query to satisfy the requirements of the image generating system. The input fed to a simple image generating system, is processed and verified using modules implemented with the help of natural language processing.**

As the name suggests this is an interactive system. Therefore, a user interface is created in which the query is entered by the user. In case of any ambiguities in the query which needs clarification, a question is raised to the user. Based on the response the query is modified to fit the system. The accuracy of the asked query is increased by implementing pronoun replacement. Any obvious conclusions which can be drawn from the previous sentences within the query are resolved on its own. Therefore, this system is an intelligent system that mainly focuses to rectify the text which is given as input in order to increase the efficiency of the model.

Keywords—Pronoun Replacement, Natural Language Processing, Text-to-image generation

I. INTRODUCTION

Visualization is the technique used for representing objects or a situation in the form of an image. It has proven to be an effective mode of communication not only for abstract ideas but also for concrete ones. It helps in getting a thorough understanding of the object or the idea that is being represented. From simple domains such as mathematics and science to complex ones like animation, gaming and computer vision, visualization forms an integral part by reducing the complexity and providing scope for advancements. Early readers can make use of visualization to gain a better understanding of the subject.

A text to image generator based on Geneva GAN was proposed [1] . It used generative adversial network to synthesis images based on the description provided by the user iteratively.

Drawing inspiration from the same, we aim to build a simple text-to-image generator which takes in a description and generates the image based on it. Our task doesn't end here.

We also extend this model by building an interactive system to resolve the ambiguities present in the user queries in order to improve system results. The interactive system is built using an algorithm to perform pronoun replacement, and in addition to that a question-answering module which asks the user to provide clarity for any ambiguity found in the query.

Natural Language Processing is a field of Artificial Intelligence that incorporates the ability to understand, analyze and derive the intended meaning from natural language or human language. NLP is developing rapidly and is being used in various areas. This field has many applications. Some of them are as follows:

- Used in machine translation making the system understand the intended meaning of the sentence and not merely translating word by word.
- Various organizations use NLP in sentiment analysis to analyze customer reviews about their product or service. . It is used in the development of chat bots which is recently being used in various applications like ecommerce, healthrelated query answering, etc.
- NLP used by companies like Yahoo, Google, etc. for spam detection of emails.
- Voice-driven interfaces like Amazon's Alexa and Apple's Siri uses NLP as the main technology to understand and analyze the voice input to perform tasks.
- It helps predict the diseases based on the electronic health record.

There are various libraries in JavaScript which deals with NLP and provides a way to incorporate NLP in projects. One of such libraries is NLP_Compromise. It provides various features to work with words and sentences. It provides ways to conjugate verbs, convert words to its different forms, convert a sentence to a different tense, POS tag each word in a sentence, find a particular part of speech in a sentence using regular expressions, etc.

This is a very helpful tool in our text to image generator as analysing the text plays a major role in achieving the end result.

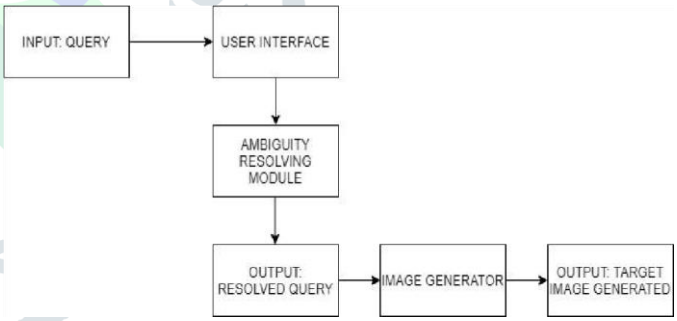

Fig. 1. Architecture diagram for Interactive Sketchmate

There are various approaches to analyzing the text in a text-toimage generator that is being used in previous systems. Most of such systems use machine learning to map the text to images. But in our system, we give importance to the input query as it plays a major role in determining the intended meaning. Hence, we use an approach that can analyze the whole sentence and understand the requirement to place objects in the image.

First, we resolve the possible ambiguities in the sentence and then find the absolute and relative positions of each object to be placed in the image. This helps us find the position of each object. From our NLP module, we have the position with which the coordinates can be determined. With this information, we should be able to generate an image with all the objects mentioned in the query.

We use SVG to draw or place the objects in its proper position. SVG or Scalable Vector graphics are based on

Extensible markup language. This helps describe twodimensional vector graphics. SVG images can be edited or created using a text editor or drawing software. SVG elements can be dynamic with many features like raising an event on mouse-hover or on-click of that element. SVG provides features like drawing basic shapes, placing the given image, changing the width and height of the image, drawing lines and curves, etc.

We have implemented question answering module which helps refine the input query in a way that it resolves ambiguous reference made to one among many objects. The module prompts the user to provide a unique characteristic in order to differentiate the objects.

Stanford coreNLP has coref annotator which is used to implement pronominal and nominal coreference resolution. There are three types of coref annotators provided by Stanford NLP namely Deterministic, Statistical and Neural out of which we tried to apply the dcoref annotator to our text. But this system fails to work for greater number of sentences. Our dataset contains upto 5 sentences, when given to this system worked only upto three sentences and also failed to resolve references made to certain nouns.

Also, a pronoun replacement approach which replaces only the proper noun with the latest occurrence of a pronoun was used in text summarization [2]. We found this approach not suitable for our use case as it fails to work on all nouns, fails to replace pronouns while ensuring that the noun and the pronoun share the same gender and number.

So we have implemented our own rule based algorithm by incorporating the common rules followed in pronoun replacement, keeping in mind the gender and the number of the nouns. We have created a corpus for female and male nouns to be used for the algorithm. We have represented our architecture diagram in Figure 1.

II. LITERATURE SURVEY

A dimension of Natural language processing is text summarization. Replacing proper nouns by pronouns was proposed [2] which helped improve the results of the text summarization process, that earlier considered only word frequency. Text-to-image generation using an iterative approach was proposed [1]. The output or feedback of the previous step was considered for rendering the output of the current iteration. The base system was implemented using a Generative Adversarial Network. A system using Stacked Generative Adversarial Networks was proposed to convert the given text description to image [3] . To reduce the complexity of the process sketch-refinement process was implemented. The image generation process was accomplished in two stages. In Stage I a sketch of basic shapes and colors of the object was obtained resulting in low-resolution images. Stage II generates high-resolution images using the results of Stage I. Cosine similarity is a measure of the similarity between two texts represented in the form of vectors. The cosine value of the two term's vector is considered. To make this metric work well with semantic relations, semantic checking between the vectors is performed as well [4]. This improves the quality of similarity checking. To evaluate the results of the model, Jaccard similarity is used. Jaccard similarity was proposed [5] to evaluate the correct grammar syntax and also to calculate the similarity. Many a time, sentences may not have antecedents. A different perspective to process such sentences without antecedents was proposed [6].

III. OBJECTS USED FOR IMAGE GENERATION

We use two main categories of objects for image generation.

A. Real-Time Objects:

In this category, we represent 6 real-time objects. The objects used are, the image of the sun, girl, boy, apple tree, pine tree, and coconut tree. A brief description of the positions of these objects in the image can be given as input. We treat each object as a unique object hence an object can be placed only once in the image. We can input 1 to 6 sentences for this category of objects. The sun is always placed above all the objects though it is possible to change the orientation and it can be placed relative to any other object. We bring this restriction to make the image look realistic. The results of image generated using these objects are presented in Figures 2, 3 and 4.

Fig. 2. Add a pine tree to the center. Add a boy to the front of it. Add a girl to the left of the pine tree. Add the sun above her.

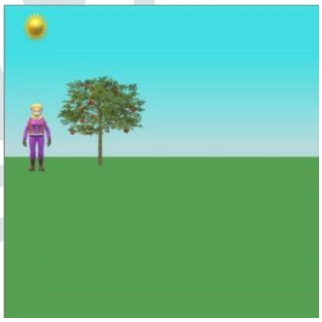

Fig. 3. Add sun on the left. Add a girl on the left below it. Add an apple tree to the right of her.

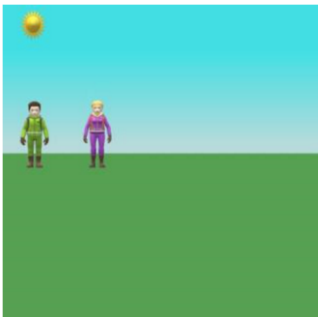

Fig. 4. Add a sun on the left. Add a boy below it. Add a girl to the right of him.

B. Basic 3D shapes:

Another category of objects that finds various applications in the real world is the shape category. So we represent 3 basic 3D shapes: a cube, a cylinder and a sphere, and 7 colors: blue, brown, green, red, yellow, grey and purple. Each shape of a particular color is considered a unique object hence we represent 21 objects in this category. It is possible to provide absolute and positions relative to other objects. The first object is placed at the center to make the relative positioning more clear and provide enough space to accommodate the other objects. The results of image generated using these shapes is presented in Figures 5, 6 and 7.

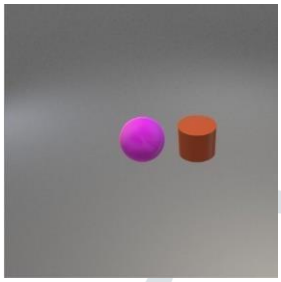

Fig. 5. Add a purple sphere. Add a brown cylinder to the right of it.

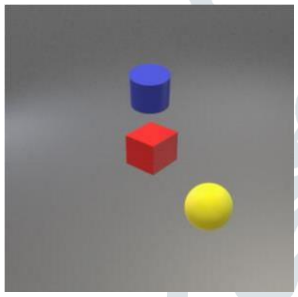

Fig. 6. Add a red cube. Add a blue cylinder behind it. Add a yellow sphere in front of cyan cube to the right of it.

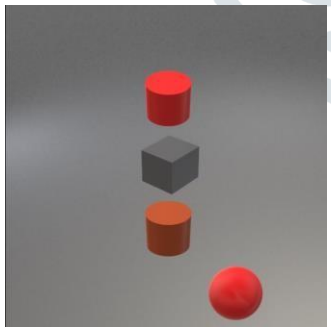

Fig. 7. Add a grey cube. Add a red cylinder behind it. Add a brown cylinder in front of the grey cube. Add a red sphere to the right of grey cube and front of it.

IV. METHODOLOGY

A. QUESTION ANSWERING:

The first stage is the Question Answering module. The user query is given as input. In various scenarios of ambiguity, questions are raised to resolve it. The ambiguities are grouped and resolved in 4 categories.

First case being same objects not allowed to be added in the image to be generated. In such a case, the system gives a warning saying we cannot add 2 similar objects and also prompts a question to enter the changed color or shape. Second case is raising a question when a user wants to enter an object when there already exists more than 2 objects of that kind.

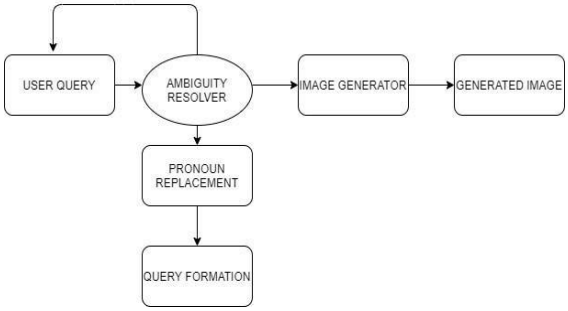

Fig. 8. UML diagram for Interactive Sketchmate

Now the system makes the user exactly specify which object he/she mentioned. Third case is when the user refers to an object which does not exist or was not added previously. The system gives a warning that such an object does not exist and asks for clarity. The final case is when the user refers to an already existing object without specifying its features. In such a case, the system questions the user for clarity about the object specified.

After this phase, the text is free from ambiguity and is given as input to the text-to-image generator. Now with this input the results of the text-to-image generator are enhanced.

B. PRONOUN REPLACEMENT:

We have built an algorithm for the pronoun replacement task, which replaces the pronouns in the user given query and passes the output to the next stage. We have used basic functions in nltk for POS tagging the words and also to find the number of objects. In addition to this, we have built a classifier to detect the gender of a proper noun present in the sentence. There are certain words in the English language which have a neutral gender. These words do have significance and cannot be simply ignored. They have to be assigned a gender according to the pronoun used to refer to it. A separate corpus is maintained for other standard pronouns. The replacement is done abiding a set of rules.

Our algorithm makes use of a set of functions to classify the words better, before replacing the pronoun. We have used pickle, numpy packages and natural language toolkit(NLTK) python library to serve the purpose. We use three dictionaries to classify the nouns into the respective gender namely, for female nouns, male nouns and neutral nouns. The input text is split into sentences and each word of the sentence is POS tagged. The nouns in the sentence are checked for gender and collective noun through a user-defined function and according to this it is added to either one of the dictionaries. Then the pronoun replacement function is invoked. Its role is to find the gender and number of the pronoun found in the sentence by POS tag and then finds the nearest word with the same gender of the pronoun passed to it and returns the word and its position. The pronoun is replaced with the returned word.

C. *DRAWING MODULE:*

The drawing module is created using JavaScript. There are two major components in the drawing module:

1. NLP in JavaScript
2. Drawing using svg **1.**

NLP in JavaScript:

The text from the ambiguity resolver module is the input for this module. The given input is analyzed and the required parameters like the absolute and relative position is obtained. This is done using a JavaScript library called compromise. We use it to identify the objects and the position of the objects. The position can be absolute or relative (to another object on the canvas). The first object is placed at the center. Other objects can be placed to the right, left, behind or in front of other objects. This information obtained from the text is passed used by our sub module to find the coordinates of the object being referenced.

2. Drawing using svg:

The canvas is divided into 5 X 5 matrix. Depending on the position of the object to be placed, the coordinates on the canvas is determined. With the information about the object and the coordinates, it is placed using svg. In this module we import images and place those images on the canvas based on the coordinates obtained. If an object has an absolute direction meaning it has no reference to the coordinates of other objects that was placed, it is assigned a fixed coordinate. Otherwise, the direction is relative therefore the coordinates are calculated with respect to the objects that are already placed. As a result each object is placed at the proper position according to the specification given in the input query. We also intimate the user in case of any object placement failure due to the coordinates being place out of bound because of repeated instruction in one direction.

V. USER INTERFACE

A web application was deployed to show the working of the text-to-image generator. The simple application consists of a home page where the user query is entered. The page shows the results after each stage of processing for better understanding of the process by the user. There are a total of 3 stages, namely question-answering or ambiguity resolver, pronoun replacement and finally the output, the target image.

VI. EXPERIMENTS

Text Similarity generally means to determine how close two documents are to each other. There are two types, lexical and semantic similarity. Lexical similarity means word-level similarity. It can be said that 4 out of 5 unique words are similar in a sentence. Mere word comparison might not be effective. So we have to consider the context also. The semantic similarity, considers meaning of text by breaking the sentences into words.

TABLE I

RESULTS OF METRIC FOR PRONOUN REPLACEMENT USED IN PROPOSED SYSTEM, 1. A GENERAL SENTENCE 2. A SENTENCE FOR THE CASE-REALTIME OBJECTS 3. A SENTENCE FOR THE CASE- BASIC 3D SHAPES

Test sentence	output	Cosine similarity score	Jacard similarity score
Shruti and Ram were at a restaurant. She had a handbag. The waiter gave the bill. He suddenly took the handbag and ran away.	Shruti and Ram were at a restaurant. Shruti had a handbag. The waiter gave the bill. waiter suddenly took the handbag and ran away.	0.9927	1.0
Add sun on the left. Add a boy on the left below it. Add a girl to the right of him.	Add sun on the left. Add a boy on the left below the sun. Add a girl to the right of boy.	0.9826	1.0
Add a grey cube. Add a red cylinder behind it. Add a yellow sphere to right of it.	Add a grey cube. Add a red cylinder behind the grey cube. Add a yellow sphere to right of red cylinder."	0.969 0.866	

TABLE II

RESULTS OF METRIC FOR PRONOUN REPLACEMENT USED IN EXISTING SYSTEM [2], 1. A GENERAL SENTENCE 2. A SENTENCE FOR THE CASE-REALTIME OBJECTS 3 . A SENTENCE FOR THE CASE- BASIC 3D SHAPES

Test sentence	output	Cosine similarity score	Jaccard similarity score
Shruti and Ram were at a restaurant. She had a handbag. The waiter gave the bill. He suddenly took the handbag and ran away.	Shruti and Ram were at a restaurant. Ram had a handbag. The waiter gave the bill. Ram suddenly took the handbag and ran away.	0.9299	1.0
Add sun on the left. Add a boy on the left below it. Add a girl to the right of him.	Add sun on the left. Add a boy on the left below Add. Add a girl to the right of Add.	0.946	0.812
Add a grey cube. Add a red cylinder behind it. Add a yellow sphere to right of it.	Add a grey cube. Add a red cylinder behind. Add a yellow sphere to right of.	0.954	0.842

We have used two significant metrics to evaluate and quantify the results of our work. One being the Cosine similarity [4] and the other, Jaccard similarity [5]. We pass the output from the pronoun replacement module and also the ground truth text to both the algorithm and tabulated the similarity score as mentioned in Table I. We have also passed the same input query to the pronoun replacement approach used in [2] and tabulated the same as mentioned in Table II.

A. EVALUATION METRICS

1) ***Cosine Similarity:*** Our aim is to determine how similar two documents are, given our prominent work is dealing with text. Cosine similarity [4] is one such metric that does the work irrespective of the size of the documents.

Mathematically, it calculates the cosine of the angle measured between two vectors which are projected in a multidimensional space. In our case, the two vectors will be the data structure containing the word counts of the two documents or sentences.

The cosine similarity formula is,

$$\frac{\vec{a}.\ \vec{b}}{\|\vec{a}\|\ \|\vec{b}\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}}$$

$$\text{Cos}\theta = \quad (1)$$

For each sentence in use, Jaccard Similarity takes into account unique words only, unlike cosine similarity which considers the vectors' total length. The Jaccard Similarity does Generally, this type of task considers just the common words not change with the number of occurrences of the same word in between the two documents. In that case, the value keeps a sentence.

increasing with the size of the document. To overcome this shortfall, this metric considers the angle between them. So even if two sentences or documents differ by the euclidean distance because of the large file size, the angle might still be smaller because of their similarity.

The main rule here is that lesser the angle, greater the similarity.

2) ***Jaccard Similarity:*** Jaccard similarity [5] is yet another metric which helps in determining the similarity between two text files. In addition to this, it is useful in cases where duplication does not matter.

The calculation of Jaccard similarity index, helps us find the similarity. If its value is 1, it implies that the two documents are of good similarity whereas 0 implies that they are not at all similar to each other. The metric is defined as the ratio of intersection of two documents to the union of the 2 documents. The formula to calculate Jaccard Similarity index is,

$$J(a,b) = \frac{a \cap b}{a \cup b} \quad (2)$$

where a, b are the two documents to be compared.

VII. CONCLUSION AND FUTURE WORK

The Interactive Sketchmate, as the name suggests is a textto-image generator which interacts with the user to generate the required image. Our work has been done to give justice to this description. This will be helpful to carry out the task of visualization and its related applications.

For future work and research, there are certain aspects of this project, that can be extended further. The drawing module can be built using generative adversial network to support larger variety of objects. Our work can be used as a base for making application for designing architectural plans through commands and also our pronoun replacement algorithm can be incorporated in any NLP task.

[3], [2], [1] [4], [5]

REFERENCES

[1] A. El-Nouby, S. Sharma, H. Schulz, R. D. Hjelm, L. E. Asri, S. E. Kahou, Y. Bengio, and G. Taylor, "Tell, Draw, and Repeat: Generating and Modifying Images Based on Continual Linguistic Instruction," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019, pp. 10303–10311. [Online]. Available: 10.1109/ICCV.2019.01040

[2] S. Urolagin and L. Satish, "Improving the quality of text summarization using pronoun replacement technique," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, May 2017, pp. 1991–1995. [Online]. Available: 10.1109/RTEICT.2017.8256947

[3] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," *International Conference on Computer Vision (ICCV*, 2017.

[4] F. Rahutomo, T. Kitasuka, and M. Aritsugi, "Semantic Cosine Similarity," 10 2012.

[5] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard Coefficient for Keywords Similarity," 03 2013.

[6] R. Filik, A. J. Sanford, and H. Leuthold, "Processing Pronouns without Antecedents: Evidence from Event-related Brain Potentials," *Journal of Cognitive Neuroscience*, vol. 20, no. 7, pp. 1315–1326, 2008.