Framework For Proactive Visualization Of Text Based Narrative Using NLP

Arnold Gabriel Benedict, Samiksha Shukla

Abstract: Language is an essential mode, not only for human communication—but also for thinking. A story is conveyed or a report of an incident is being told, humans perceive the conveyed information in the form of visual insights. The increasing advancements in the field of artificial intelligence can help with the same in machines. This paper reflects on the internalization of stories from a cognitive perspective and outlines a scalable framework for supporting the visualization of narrative text data. This paper leverages natural language processing (NLP), probabilistic modelling of discourse knowledge, information extraction of narrative components (who, where, when, what) and the narrative visualization. The graphics knowledge base storage structure has been redesigned to obviate the necessity of having a larger database for all graphics entity. With the developed framework, any user can input unrestricted natural language for the dynamic generation of animated scenes. This provides users with direct visual output in response to their natural language input. This tool can potentially impact the way humans interact with computers and expand a completely new way of understanding conversations.

Index Terms: Conditional Batch Normalization (CBN), Natural Language Processing (NLP), Probabilistic Modelling, Part-Of-Speech (POS), Tokenization.

1. INTRODUCTION

SCIENTISTS are fully devoted to understanding the gap between human cognition and artificial intelligence. The notion about how unintelligent components like transistors etc can help the machine mimic a human brain was always questioned. This led to the emergence of a new domain called cognitive science. Most scientists struggled to make machines intelligent. Enhancing the quality of human life was always the main motive behind every invention. Making machines to enhance human capabilities is much relevant than making machines to think independently. Communication is both cognitive and social because it is a way to exchange information via a common medium [1]. Moreover, the psychology of language was given more attention because it dealt with the cognitive perspective than later. Understanding and viewing a particular message from the same perspective as the source is the most difficult linguistic problem since the dawn of humans. The visual approach in human communication gave its best to make humans view the scenario from a single viewpoint. During the dawn of this idea, extensive research about this was conducted. During which it was recognized as a novel work. Not much work was done in this domain. But, this extensive research on this domain yielded much understanding on NLP and human cognition. In 1998, Microsoft has shown some results related to this work [2]. But that project was shut down due to a few shortcomings, mainly computational resources and other linguistic limitations. This work has overcome those limitations and has redesigned entire architecture which far robust than the one which was developed 20 years from now.Today, the world has a vast number of written materials in the form of journals, novels, books and nonetheless articles and reports which are generated daily in huge numbers. There are tools available to analyse, categorize and summarize the given materials within seconds. The technology has grown to the level that it could summarize an event using just a single image input. All forms of data can be identified by a value-based homogeneous data structure [3]. There is no existing tool when it comes to representing these written contents in the form of real-world entities. That is, visually representing a written narrative by procuring its graphical information. There are just different studies available in this domain with merely non-usable applications. The idea is to link Graphics and NLP. Under the approach, natural language input is analysed by the NLP engine which passes on to the graphics component all the information necessary to render appropriate graphics those that match the story that is being entered previously. An important feature of the tool is that it can be extended by users both in terms of the number of graphics available for illustrating stories and in terms of the link between words and graphics. For instance, users can drag and drop two-dimensional images onto the tool to create two-dimensional graphics and have automatically displayed in three-dimensional space. In this way, developers can personalize the graphics environment according to their use-case. This relieves the burden of having to build up a large repository of graphics before using it. At the same time, it addresses the problem of the unlimited nature of natural language. The integration of NLP and Graphics makes it possible to have a series of animated graphics generated dynamically based on a user's storyline.

This idea was originally put across to motivate children to write stories using the computer. It focuses on the creation of an environment in which children would enjoy writing stories and thereby enhance their reading and writing abilities. The tool was intended to make unnecessary the distracting chore of searching for just the right picture to illustrate a story. Story Maker is fun to use. It provides children with instant gratification while encouraging them to read and writeConverting a narrative to a visual representation using real-world entities was always a challenge and merely any usable application was developed. This study has taken care of most of these challenges and gives a usable application. The following are the objectives that have been identified like Probabilistic Modelling of discourse knowledge in sentence processing [4], Parameter based Object-Action interaction, Using of first-order logic to compress sentence, Framework for interfacing NLP with primitive Graphic elements, Scene generation based on the given NLP input. The main objective is to develop a framework which acts as an interface between unrestricted natural language and the graphics system. Therefore, making it flexible from both sides. The natural language input can be changed to any language

- *Arnold Gabriel Benedict, CHRIST (Deemed to be University), Faculty of Engineering Bangalore, India e-mail: arnold.g.benedict@gmail.com*
- *Samiksha Shukla, CHRIST (Deemed to be University), Lavasa Campus, Pune, India e-mail: samiksha.shukla@christuniversity.in*
- *samiksha.shukla@christuniversity.in*

desired or the graphics system can be changed from 2D graphics to 3D graphics depending on the use case by a developer. This framework will act as the base layer for any given application developed over this framework.

2 RELATED WORK

Many works have been in progress in the field of realistic image synthesize or image manipulation. Recent progress in Generative adverbial networks has shown in improvement in image manipulation using semantic information around layout selected [5]. Taesung Park et al [6] had worked on generating photorealistic images from a semantic input layout alone. The layout is a raw input over a canvas where its generative models draw over. Generating high-quality photorealistic images or videos from static single text input was always challenging. Yet, many extensive research has been done in this field [7, 8]. Moreover, capabilities of Generative Adversial networks (GANs) have been applied to video modelling giving promising results. Walking and transforming in unexplored scene and generating elements in real time is also being worked on. In contrast to it, more study has to incorporated so that not only generating basic properties of the real world elements but also being creative within the reality boundary [9]. Recently, scientists are using GANs for single sentence text interpretations [10] and Conditional Batch Normalization (CBN)[11, ?] for the same. One of the work includes creating boxes as a layout and refining the boundary based the feedback given by the generator module 13]. Use of CNN-based discriminator shows an improved layout generation [14]. Even though, the layout generation is created for still images, it has constraint for up scaling to video generation, as the continuity of frames is difficult to achieve. And moreover giving general understanding of the language of a text before training a neural network can enhance the training quality. One method for mapping this sematic layout from a natural language in the form of a story to generate the scene is shown in this paper.

3 METHODOLOGY

This work has been modelled into two basic unit. First unit deals with natural language and its processing. Hence, the robust programming language to implement is python because of its huge number of in-built packages which can perform all the basic NLP operations along with package named Natural Language Tool Kit (NLTK). Second unit deals with the graphics processing and graphics generation. Since the data structure and its communication is in the form of objects, Java is used because of its simplicity in object oriented-ness. Moreover, for better handling of graphics, a package called processing was used. It helped in easier prototyping of the work. The communication between Java and python was done using jython interface.

4 DESIGN

This project consists of two main components: (a) The NLP engine, and (b) The Graphics engine. The input is given in the form text, this text can be either raw text from a story or text which is extracted from a speech. Sentences are extracted from the text, which is used to create an XML formatted output which is fed onto the graphics engine. The output is generated in such a way that the actor, actions and their interactions are defined properly. The graphics engine receives this information and updates the world map by adding the features. This engine gets the pre-stored information about the world map from a knowledge base. Then, a scene is a generator on the screen based on the graphics element it picks. There is an additional unit which validates the generated scene, to check whether the subject is focused or not. For instance, the narrative says "The boy is crying". A boy ([actor:"BOY_01"]) with a weeping face ([action:"CRYING"]) is displayed as part of the current scene by default graphics of a boy. Later, the narrative mentions the colour of the boy's shirt is 'blue'. The graphics engine will understand the boy from the discourse knowledge, i.e, ["BOY_01"] is wearing a 'shirt' and the colour of the shirt is 'BLUE' ([ac- tor:"BOY_01"sub:"BODY"="SHIRT"; adjective="BLUE"]). NLP generated output can be written as:

<Actor>BOY_01
<sub>
<object>BODY</object>
<attr>SHIRT</attr>
<adjective>BLUE</adjective>
</sub>
</Actor>

For the demonstration of the designed architecture, a User Interface (UI) is created to insert 2D graphics elements in the knowledge base. During the research, the analogy between parts of speech and graphics operations were identified. The data structure of words and their graphics properties is designed to achieve version scalability. The UI design, development and coding is user-friendly based on concept of design pattern to achieve maximal usability. The Level 2 or the in-depth architecture is given in figure 1. All the design components are discussed in this section.

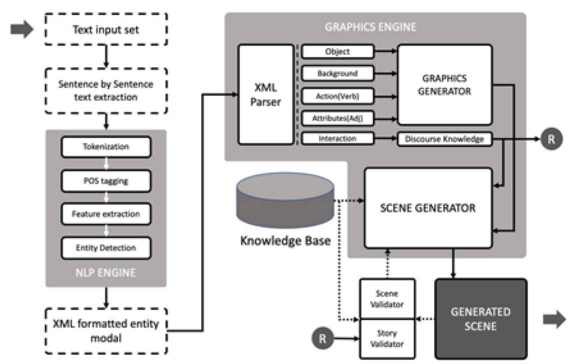


Fig. 1: *Architecture Diagram of the proposed system*

4.1 Knowledge Base

For every word which is encountered by the NLP engine, like actors, actions, inter- actions, background etc. are classified on the basis of basic parts of speech. Namely, Noun, Verb, Adjective/Adverb, Preposition with the background setting. The structure of storage of these graphical elements has been designed in a robust fashion so that each and every word could be mapped with some graphical properties.

4.2 Object-Noun

Any noun such as man, cat, tree etc. can be formed by an independent graphical element or by a collection of other object-noun words. For instance, an object called 'King' is to be added, which is a collection of two other objects mainly 'man' and 'crown'. So there is no need to make completely new element, it is just required to use pre-existing elements. Therefore, eliminating redundant copies of similar elements. To implement such feature of storing non-redundant elements, it was required to design a data structure which satisfies all graphical properties to display a noun. Following shows the designed structure for storing object-nouns

```
<id>0020</id>

<default>
<word>ABC</word>
<word>Synonym1</word>
<word>Synonym2</word>
<scale>0.73333335</scale>
<textture>017</texture>
</default>
<obj>
<load>
<img>00020</img>
<xywh>10,10,120,120,</xywh>
<angle>0.0</angle>
<scale>1.0</scale>
</load>
<load>
<set>0018</set>
<xywh>10,10,120,120,</xywh>
<angle>0.0</angle>
<scale>1.0</scale>
</load>
<link>00029:1::00020:7</link>
</obj>
<custom>...</custom>
```

• id tag: It is used to store the filename for that word set. It is generated serially as new words are created under noun section.
• default tag: It is used to store all the default variables like default texture, universal scale with respect to other variables in the world, additional synonym words which uses the same.
• obj tag: It is used when an element has to be added to the object creator using img tag or to use previously created words using set tag.
• img tag: stores the id of the new graphical element which is stored in the directory “DATA/obj/objectPNG”. set tag stores the id of previously added word which is stored in the directory “DATA/obj/objectMETA”.
Here, for each img and set tags are dedicated tags to carry information related to the placement of each components. xywh tag helps to map each element in 2D plane(x, y are the coordinates and w, h are the dimensions). angle and scale tag helps to add angles and scales to each object.
• custom tag: It helps us to add custom animation during each transformation of graphics element to make the scene more realistic.
Moreover, the system should understand what is plural or singular.

4.3 Background-Noun

Background-Noun usually represents the setting of the scene. For instance, if the scene is happening in the woods/forest, the scene should generate the trees, bushes, clouds and the color of the land should be muddy. In this study, the background is a collection of different object-noun placed for particular area, under well-defined depth and with appropriate texture. Following shows how background-noun metadata is designed and stored for figure 2.

```
<id>0030</id>
<default>
<word>ABC</word>
<word>Synonym1</word>
<word>Synonym2</word>
</default>
<land>
<load>
<set>0010</set>        #Trees
<z-begin>10</z-begin>
<z-end>40</z-end>
<noise>0.03</noise>
</load>


<load>
<set>0014</set>#Bushes
<z-begin>10</z-begin>
<z-end>50</z-end>
<noise>0.03</noise>
</load>
<load>
<texture>0014</texture> #Mud
<z-begin>10</z-begin>
<z-end>50</z-end>


</load>
</land>
<sky>
<load>
<set>0012</set>#Clouds
<z-begin>10</z-begin>
<z-end>40</z-end>
<noise>0.03</noise>
</load>
<load>
< texture >0011</ texture >      #BlueSky
<z-begin>10</z-begin>
<z-end>40</z-end>
</load>
</sky>
```

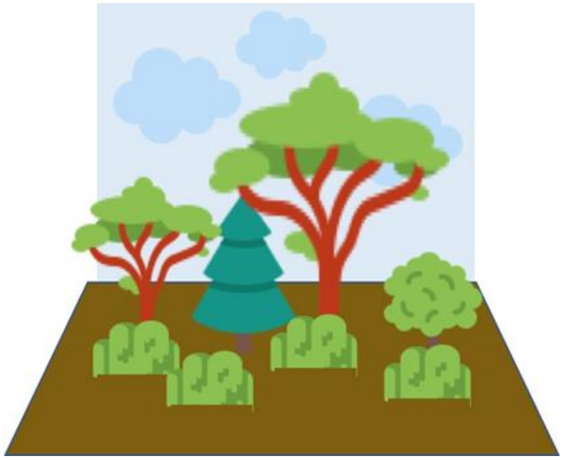


***Fig. 2**: Generated Graphics for Background-Noun sample structure*

- id tag: default, set and word tags are same as object-noun.
- land tag: sky tags: Every background element should in the sky or on the land. So the loading of each object should happen in the respective tag.
- noise tag: It is used inside all set tags. This value is responsible in setting the distribution of the elements appearing in a particular frame change.
- texture tag: Here, this tag means you are setting a land or sky textures.
- z-begin, z-end tags: These tags helps the elements to set its boundary on land or water along its depth.

Whenever, the object interacts with the background, the object should move towards the subject and blend in the background. But the object and subject becomes the part of the foreground. This is called relative transformation as shown in Figure 3.

4.4 Adjectives

Most adjectives doesn't require any graphical representation of itself. Hence the knowledge base for adjectives is very less. Descriptive and quantitative adjectives are the types which needs graphical properties to visualize it. For example, behavioural adjective like 'annoying' has no particular visual required, but on the other hand, a descriptive adjective such as "tall" can be represented using graphics properties. Based on survey, the attributes required for an adjective can be limited to overlay color or texture, transparency of the overlay, scale of the element and frame rate of animation.

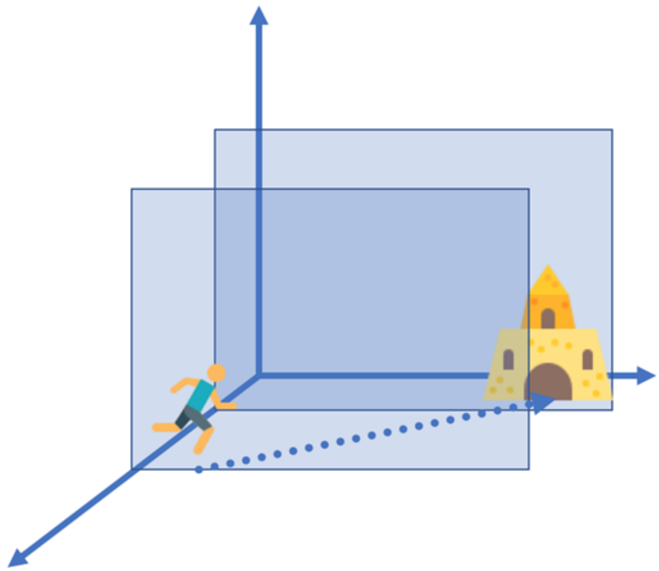


***Fig. 3**: Relative Transformation*

For instance, consider an egg. If the text contain "golden egg", there is a colour overlay over the egg of colour of gold. Moreover, if the egg is "broken" or "rotten", it could be visually represented as overlay of texture on already existing element called "egg" as shown in figure 4. Adjectives like 'big', 'small', 'tall' etc. requires attributes such as scaling or stretching. Adjectives like 'clear', 'transparent', 'translucent' etc. require transparency attribute. Adjective which represents speed of an action such as 'slow', 'fast' etc. can be represented as frame refresh rate of the element. The structure to store adjective data can be in the form as given below:
Word # t=xxx(or # c = xxxxxx) #1.0#0.72#1.5

- Word: This contains the adjective
- Texture: If the adjective requires a texture, then the texture id is mapped to the field
- Color: if the adjective requires a color, then the color hexacode is mapped to the field
- Transparency: The value can range from 0.0 to 1.0 (opaque to transparent)
- Scale: Includes the scaling factor other than the default.

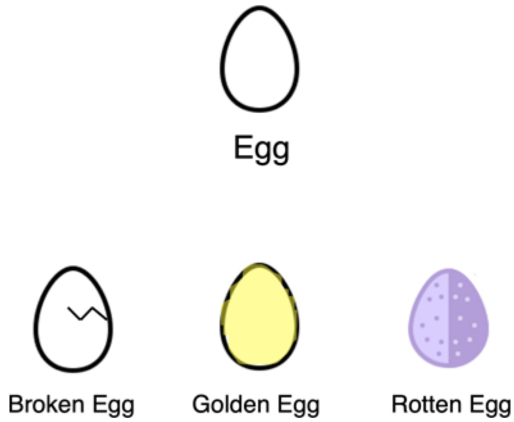


Fig. 4: *Adjective can be represented in the form of color or texture fill.*

- Frame refresh rate: The value represents the speed with respect to the original frame rate.

Transformation of objects (Morphing) is required when there is a state change in a particular objects. For instance, in the case presented here, the entire story was happening around a "healthy egg", and suddenly the narrative says "the egg broke". In this case, there should be a transformation of objects, i.e., the overlay which is drawn should have a pre-defined transition. Fading transition is given for this in this work.

4.5 Verbs
Verbs are also called doing words. Therefore, any form of transformation is occurred when a verb is involved. Any verb can the deal with a combination of translation, rotation, and/or scaling. Translation in these verbs can be linear or parabolic based on the literature survey. Rotation can happen in any direction and to any degree based on its interaction with the subject. Scaling options are either shrinking or expanding. This is shown in figure 5. The structure of storing a verb is as given below:

Word # mask # ...values

- Word: This contains the verb.
- Mask: This is a three bits of binary value, which tells out of the three transformation which all have been included.

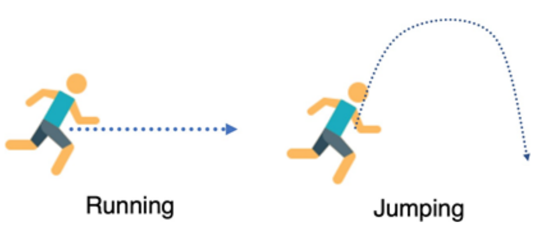


Fig. 5*: Verbs can be represented in the form of different transformation*

- Values: Values of pre-defined options gets stored on the order of mask.

When there is verb acting on an object, it mostly interacts with the subject also. For instance, if the text says, "The boy jumped", there is a simple parabolic motion which happened. But if the text says, "The boy jumped into the well", then the parabolic path should end with the subject, i.e. the 'well'. So this understanding of verb-interactions can be dealt by using preposition.

4.6 Preposition-Verb Interaction
Prepositions were identified for depicting real world actor-subject and actor-action interactions. For instance, if the text says, "..went through the mirror.." the system should understand what 'through' means and how base verb 'go' can interact with the subject 'mirror'. This understanding can defined by three parameters. The file structure is shown below. Based on the options selected the value for each preposition changes. Table 1 shows common prepositions and identified attribute code.

Word # interaction_site # degree_of_movement # Z-index

- Interaction sites: These sites are points on the subject which will guide the verb vector to align. The points can be center, left, right, top or bottom. [Figure 6]
- Degree of movement: This is defined by two options, namely, to mid and through mid. To mid means the vector path ends at the mentioned interaction site. Through mid means the vector path passes through the interaction site. [Figure 7]
- Z-index: It can have the three values (-1, 0 , +1). -1 means the object passes behind the subject. 0 means the object passes through the interaction site. +1 means the object passes in front of the subject. [Figure 8]

Table 1*:*
Representation of different prepositions

S No	Word	Attribute Code
1	Through	CMO . .
2	In	CN+1 . .
3	Over	TM+1 . .
4	Below	BM+1 . .
5	Into	CN+1 . .
6	under	BN+1 . .
7	beside	LN+1 . .
8	across	CN+1 . .
9	behind	CN-1 . .
10	out	RN+1 . .

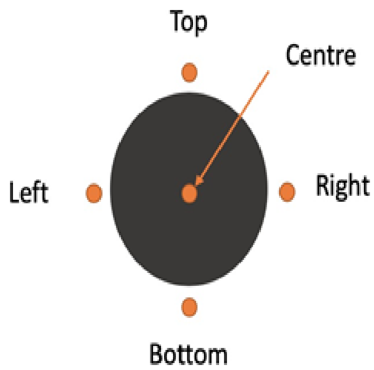


Fig. 6: *Different interaction sites*

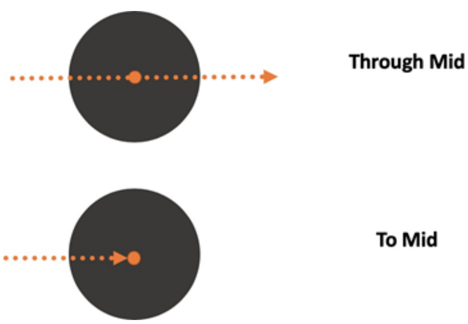


Fig. 7: *To Mid and Through Mid*

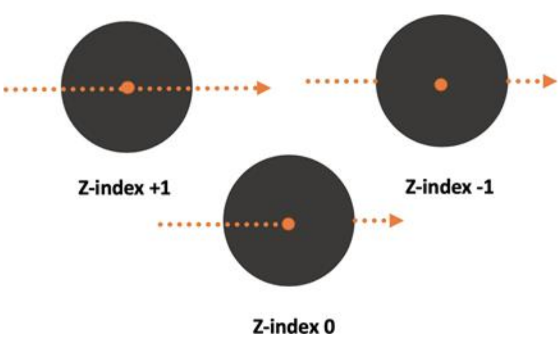


Fig. 8: *Z-indexes*

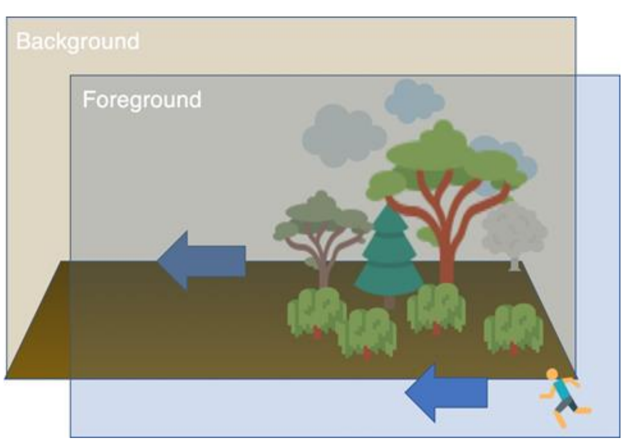


Fig. 9: *Background Generation using Perlin Noise*

4.7 Background Generation

Consider the text "A man is running in the woods". It is clear that the animation will continue till a new sentence arrives. So, as the object moves, the background should also move accordingly. Background contains a lot of graphical elements. Therefore, the entire background need not be generated at first. The generation and deallocation of background graphics can happen as the viewport moves with respect to the subject. Perlin noise is used here for continuous temporal random distribution. This noise filter will tell when to generate an element and when not to. The figure 9 gives an overview on how the interaction happens.

4.8 Probability Based Discourse Knowledge

Discourse knowledge in simple words means keeping track of all the relations and interactions which is already known. Here, as the story develops sentence by sentence, the knowledge of past actions is stored, so that whatever are the changes, it is reflected on the previous model of world. Moreover, the objects are likely to be called using pronouns such as 'She', 'He', 'They' etc. So for such sentences, the system should keep track of the previous objects and its properties. For instance, take 'She' as the object from new sentence. As given in table 2, it is known that Princess and Queen are more likely to be the 'She', since it refers to a female candidate. Moreover, the chance for it be Queen is more since it is the last used object. So it can be said, SHE("KING") is much less than SHE("PRINCESS") less than SHE("QUEEN").[Table 2]

Table 2
Temporal Based Entity Table

Tag	Meaning
Princess	F
Queen	F
King	M

4.9 Graphics and Scene Generator

In this study the graphics generator generates temporary object based on the attributes provided. This object is fed to the scene generator which knows where to place the object. Moreover, graphics generator only works when a new sentence is inputted. Scene generator is responsible for the animation of these object and controls the transformation and frame refresh rate.

4.10 NLP Engine, XML Parser

Natural Language Processing (NLP) Engine analyzes the basic semantic structure of a sentence like who, what action, to what and where. For example, consider user input as "The boy drank a beer on the bar." Here the system knows: OBJECT = boy; ACTION = drink; INTERACTION = beer; LOCATION = bar. So the NLP engine will generates a tree as shown is Figure 10.

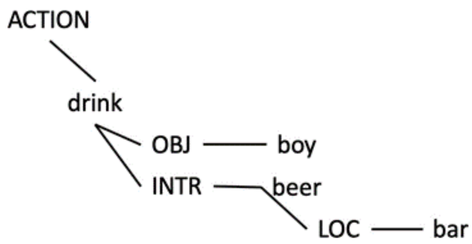


Fig. 10: *Analysis of natural language*

Natural language input is taken and fed to the NLP Engine sentence by sentence. Each sentence undergoes basic NLP operations like tokenization, Part-Of-Speech (POS) tagging, feature extraction and finally entity detection. The output from the NLP engine is in xml format given the easiness in its parsing. Following are the internal functions which is involved in generating output for graphics engine

- Tokenization: This is a simple operation which is performed on the raw text. This operation will segregate each word into tokens excluding all the unwanted characters.
- POS Tagging: This operations takes the tokens as the input and will create a label for each token. The labels being the type of parts of speech. This operation is done using pre-defined database of all the POS with commonly used words.
- Feature Extraction: Here, the feature being extracted is to segregate the subject and the predicate. This gives the details about the actor, action, to what it is interacting with and what is the background setting.
- Entity Detection: This helps in classifying the actors based on the discourse and generates final set of entities like actors, actions, attributes, interaction and background.

The above detected entities are given to the graphics engine via an xml formatted file structure. For above example, the generated xml will be as shown below:

<OBJECT>boy</OBJECT>
<ACTION PAST>drink</object>
<INTERACTION>beer</INTERACTION>
<BACKGROUND>bar</BACKGROUND>

The graphics engine receives xml formatted input which is parsed using the XML Parser. The retrieved information is checked against knowledge base and graphics generates the temporary object.

4.11 Validators

There are two validators, namely, scene and story validator which validates the generated scene. Scene validator checks whether the subject is focused or not. On that basis, it will tell the generator to shift the viewpoint. Story validator checks if all the objects are visible which is likely to be focused in future and moves the camera in z-axis to bring all the necessary actor for a particular story.

5 RESULTS

In this work, a basic working of all the components are explained as in the actual work. A java application was developed along with processing graphical library to showcase graphical engine. Using Jython, along with NLTK library, NLP engine was developed.

5.1 Entity Extractor

As part of NLP engine, entity extraction is programmed in Jython. Using NER (Named Entity Recognition) chunk checker, it was able to tokenize and tag each word in the sentence. Filtering this in post processing, using Inside-Outside- Beginning (IOB) tagging helped in improving the output by increasing addition parsed level. Simple sentences were fed and POS tagging was performed. The figure 11 shows entity classification based on the given natural language sentence.

Following are few NER mode of classification performed on the given text. These classifications are done based on past training of the model on English paragraphs.

geo = Geographical Entity org = Organization
per = Person
gpe = Geopolitical Entity tim = Time indicator
art = Artifact eve = Event
nat = Natural Phenomenon

The universal parts of speech table is shown in table 3. All the words has mapped parts of speech property which makes the graphics generation easier to fetch elements from the knowledge base.

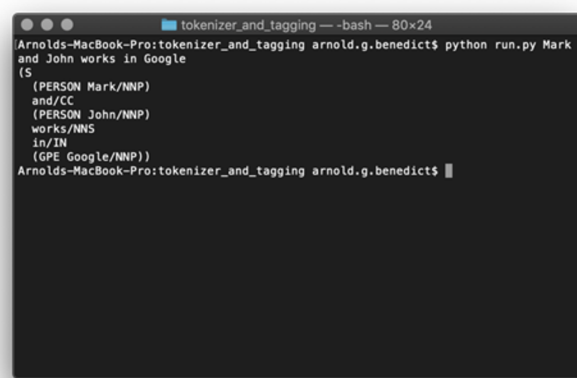


Fig. 11: *Natural Language input and its entity classification*

TABLE 3: *UNIVERSAL PARTS OF SPEECH*

Tag	Meaning	English
ADJ	adjective	old, bad, short, ordinary, huge, global
ADP	adposition	in, on, by, with, at, into, below
ADV	adverb	really, already, still, early, now
CONJ	conjunction	and, or, but, if, while, although
DET	determine	article the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !

X	other

5.2 Creating New Word
The process of adding new words to the knowledge base is as shown in figure 12. As new word is inserted, it is categorized on the basis of its parts of speech. The user has the ability to add one word to more than one single kind.

5.3 Object Noun Creator File
This java file was created to demonstrate the adding of noun elements into the knowledge base. This figure 13 shows the interface for adding the word KING:object/Noun

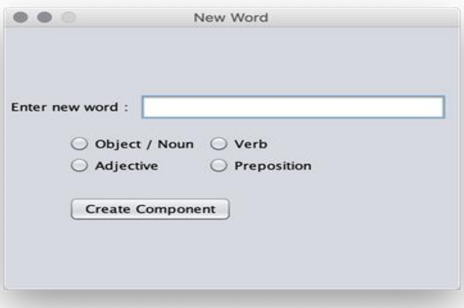
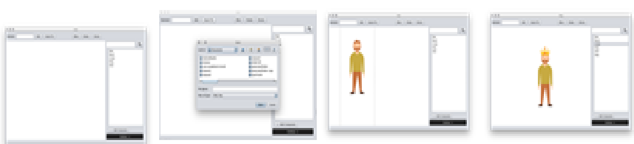


Fig. 12: *Adding new word to the Knowledge base*

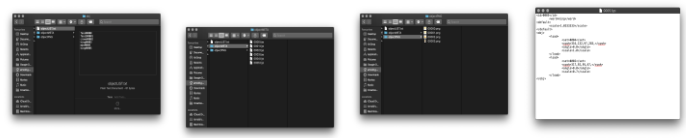


Blank Screen (b) Importing PNG (c) Usage of Move, Ro- (d) Importing existing graphics

(e) Performing Universal scale (f) Preview Screen

Fig. 13: *Steps involved in creating Object Noun*

and editing graphical elements and steps involved to do so. It has capabilities of importing PNG graphic images to the canvas. It has transformation capabilities such as dragging, rotation or scaling a component which is inserted. It can also use the existing word component which is previously created. On the editing of the elements and final placements, the user has the ability to set a global scale to entire component with respect to the world reference which it is working with. The figure 14 shows the generated files and directories upon the creation of the word KING.



(a) Updates Object List file (b) Generated Meta file
(b) Imported new PNG files (d) Generated Meta file XML

Fig. 14*: Generated Files for Object-Noun*

(a) Texture Fill (b) Color fill

Fig. 15: *Options for fill to show Adjective*

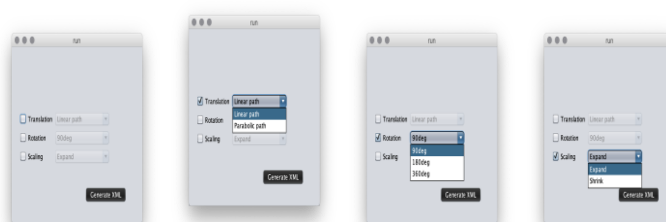
5.4 Background Noun Creator File
This java file was created to demonstrate the generation of background on the basis components added. It is similar to object noun creation.

5.5 Adjective Creator File
This demonstrates the mapping of attributes of almost all adjectives to graphical properties such as colour, transparency, texture, size, and frame refresh rate. The figure 15 shows the steps involved in the creation of adjective words. And figure 16 shows the inserted tuple based on the given mappings.

(a) Updated adjLIST file (b) Importing texture overlays

Fig. 16: *Final generated files*

\



(a) New verb (b) Options for translation (c) Options for rotation (d) Options for Scaling

Fig. 17: *Steps Involving in verb creation*

5.6 Verb Creator File
This demonstrates the different parameters required for showing basic verbs and its storage. The figure 17 shows the various options available to define a verb. The figure 18 shows the updated verbLIST file on the creation of new verb word.

5. 7 Preposition Creator File

This demonstrates the different identified attributes for representing preposition-verb interaction. The figure 19, 20 and 21 depicts the various attributes for determining interaction sites, degrees of movement and Z-indexes respectively. The figure 22 shows the updated prepLIST file on the creation of a preposition based on its interaction with the object.

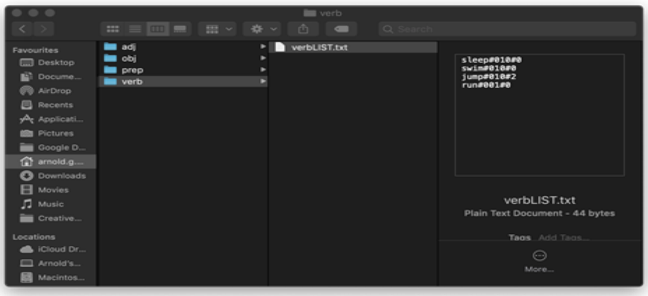
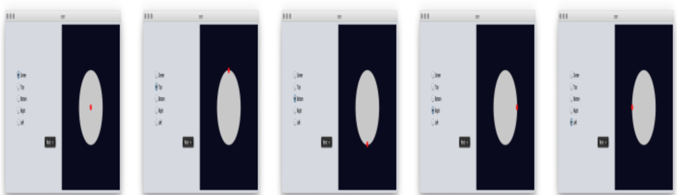
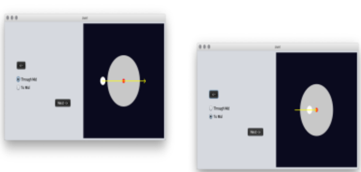


Fig. 18: *Updated verbLIST file*



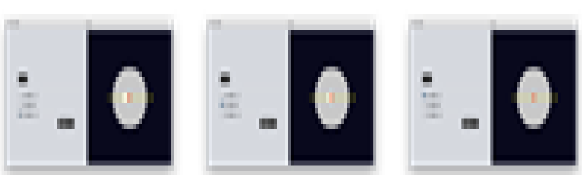
(a) Center (b) Top (c) Bottom (d) Right (e) Left

Fig. 19: *Different Interaction sites*



(a) Through Mid (b) To Mid

Fig. 20: *Different degrees of movement*



(a) Z-index 1 (b) Z-index 0 (c) Z-index +1

Fig. 21: *Different Z-indexes*

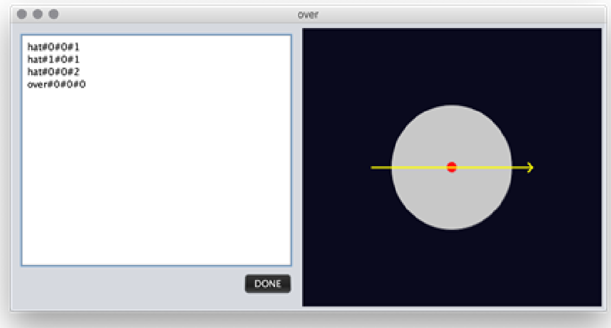


Fig. 22: *Updated prepLiST file*

5.8 Scene Generation

The figure 23 shows the generated scene for the below natural language input.

"A man is playing football in the forest"

The generated scene shows how the input text was used to display the given result. The knowledge base contained the definition of MAN:actor, FOOTBALL:object, FOREST:background. Accordingly it fetched the basic graphics elements and presented through the graphics generator.

Fig. 23: *Generated Scene based on the given natural language input*

6 DISCUSSION

Apart from the challenges, which the structuring and unavailability of similar works of the paper posed, this section deals with applications of this system and then later with the limitations of proposed system. There is no doubt to fact that the development of this framework is at its beginning stage. There is some scope for further analysis of the work which can be carried in future. Applications of such visualization system is it can change the way users perceive narrative data. For instance, in crime reconstruction, it can help the concerned officials to view an incident from a single point of view from witness reports, reducing the time of discussions. Other application that is identified is generation of image based on complex sentences. The image repositories are limited. Millions of contents are added every day, even though searching using keywords doesn't help always. With the power of emerging AI, it is possible to generate images on the input with complex sentences. The applications of this work is limitless, nevertheless the usable product is far from its first deployment. The current system has its own limitations. First of all is, its unintelligibility in understanding complex conjugated sentences. Moreover, the various scene generations technique for 2D graphics cannot be used for creating 3D graphics. More complex real world interactions can be designed in future to achieve the state of art system.

7 CONCLUSION

As this is the era of emerging AI and any breakthrough in its sub-domain are significant. This research work helps a readers make animations based of given unrestricted natural language. This visualisation framework have many use cases and development of such application could revolutionise the human-computer interaction technique. The originality of this concept is better motive to continue the work in the same direction. The interface between its language and graphics unit and the fact that those goes hand in hand are the key features. This method of building semantic layout can help in synthesize of realistic video using GANs.

ACKNOWLEDGMENT

The authors wish to thank Dr Balachandran K, HoD, Computer Science and Engineering, CHRIST (Deemed to be University), Bangalore and Dr Iven Jose, Dean, Faculty of Engineering, CHRIST (Deemed to be University), Bangalore.

REFERENCES

[1] Nicholas L. Cassimatis: Chapter 2 - Artificial Intelligence and cognitive Modelling have the same problem, 2012, pp. 11-24.
[2] Takako Aikawa, Lee Schwartz, Michel Pahud: NLP Story Maker, Microsoft Research One Microsoft Way, Redmond, 2002, pp. 1-4.
[3] Ting-Hao (Kenneth) Huang , Francis Ferraro , Nasrin Mostafazadeh, Ishan Misra , Aishwarya Agrawal , Jacob Devlin , Ross Girshick, Xiaodong He , Pushmeet Kohli , Dhruv Batra , C. Lawrence Zitnick , Devi Parikh , Lucy Vanderwende , Michel Galley , Margaret Mitchell : Vi- sual Storytelling, Microsoft Research Carnegie Mellon University, Johns Hopkins University, University of Rochester, Virginia Tech, Facebook AI Research, 2016, pp. 1-7.
[4] Amit Dubey, Frank Keller, Patrick Sturt: Probabilistic Modeling of Discourseaware Sentence Processing, University of Edinburgh, 2013, pp. 425-451.
[5] David Bau, Hendrik Strobelt, IWilliam Peebles, Jonas Wulff, Bolei Zhou, JunYan Zhu, Antonio Torralba : Semantic Photo Manipulation with a Generative Image Prior, Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH 2019), pp. 59:1-59:11.
[6] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, Jun-Yan Zhu: Semantic Image Synthesis with Spatially Adaptive Normalization, Computer Vision and Pattern Recognition, 2019, pp. 1-19.
[7] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dim- itris Metaxas : StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1-14.
[8] Aidan Clark, Jeff, Karen Simonyan : Efficient Video Generation on Complex Datasets, 2019, pp. 1-19.
[9] A. Jahanian, L. Chai, and P. Isola. : On the "steerability" of generative adversarial networks, 2019, pp. 1-25.
[10] Guojun Yin , Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, Jing Shao : Semantics Dis- entangling for Text-to-Image Generation, Computer Vision and Pattern Recognition [cs.CV], 2019, pp. 2327-2336.
[11] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, Hongsheng Li : Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis, Neural Information Processing Systems (NeurIPS 2019), pp. 1-14.
[12] Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, Aaron Courville : Modulating early visual processing by language, Neural Information Processing Systems 30 (NIPS 2017), pp. 1-14
[13] Seunghoon Hong, Dingdong Yang, Jongwook Choi, Honglak Lee : Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis, Computer Vision and Pattern Recognition (CVPR 2018), pp. 7986-7994.
[14] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, Hongsheng Li : Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis, Neural Information Processing Systems (NeurIPS 2019), pp. 1-14