Orientational Aspects of Objects in Text to 3D Scene Generation for Interior Designing

Initial Submission for Review

Neelesh C. A.
Dept. of Computer Science Engineering,
PES University
Bengaluru
Karnataka, India
neeleshca26@gmail.com

Ninaad R. Rao
Dept. of Computer Science Engineering,
PES University
Bengaluru
Karnataka, India
ninaadrrao@gmail.com

Yashaswini S
Dept. of Computer Science Engineering,
PES University
Bengaluru
Karnataka, India
yash.sridhar8@gmail.com

Shylaja S S
Dept. of Computer Science Engineering,
PES University
Bengaluru
Karnataka, India
shylaja.sharath@pes.edu

ABSTRACT

In this paper, we propose the method of generating a 3D scene from text with respect to interior designing and consider the orientation of objects that are present in the scene. This paper focuses on interior designing and has considered not only objects whose placement is with respect to the floor but also objects whose placements are with respect to wall and ceiling of a room. Our approach uses Natural Language Processing to extract useful information from the user text, which will aid the rendering engine to generate a better 3D scene. Since manually placing the object could be cumbersome, experimental results show that the approach used in this paper generated the expected results in most cases.

KEYWORDS

Natural Language Processing, 3D Rendering Engine.

1 INTRODUCTION

The task of generation of a 3D scene can be interpreted mainly in two ways. First method is the simple drag and drop of individual 3D models to suit the requirement of the user. This will meet the entire orientational aspect of the user. But for the end user, the task of 3D scene design can be very complex using drag and drop method as there are many models to search for and it becomes a cumbersome task for the correct 3D scene. The second approach would be the skill to visualize and describe the 3D scene using English sentences which are then mapped to our 3D models. The ability to generate a 3D scene with rudimentary English sentences can be simpler as the user just has to describe his visualization and the task of placing and selecting the objects is done by our model. We consider the second approach and the task of the end user is minimized to a great extent.

Text to 3D scene generation has a wide number of applications. It can be used in the educational sector where a complex concept could be described and a 3D geometric model could explain the concept. 3D scene generation can be useful even in fields like art or forensics. In this paper, the main focus is on interior designing which has a lot of focus on the orientation of the objects.

The key components of the approach considered is capturing all the prepositions and mapping the preposition with the respective parent-child relationship. Considering the orientational aspects of every object based on the requirements given by the end user, a 3D model of the scene is generated and presented. An implied parent is considered if it is not explicitly stated by the user.

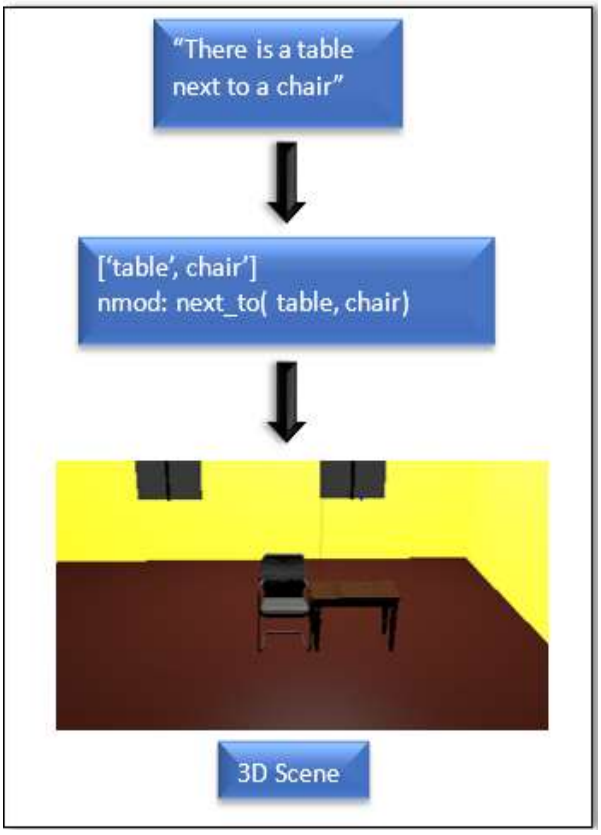

Figure 1: Flow chart of the task involved in generating a 3D scene from text

Our approach renders 3D scenes for interior designing but it does not consider the intricacies of interior designing which is a very important field on its own. In the technique considered, English sentences are processed using Natural Language Processing (NLP) and the 3D scene is generated using a 3D rendering engine, Blender which gives the final output.

Most recent work on Text to 3D scene Generation has been carried out by Chang et. al. [5,6,7,8] and considers objects and trains a classifier on a scene discrimination task and extract high-weight features that ground lexical terms to 3D models. They have integrated their learned lexical groundings with a rule-based scene generation approach and have shown through a human judgment evaluation that the combination outperforms the existing approaches in isolation.

The other work in this field is done by WordsEye specified by Coyne and Sprout in [3] which used manually chosen mappings between language and objects in scenes. WordsEye has used natural language as medium for describing visual ideas and images to acquire artistic skills on window-based interface, it automatically converts text into 3D-scenes by depicting entities and objects involved, their poses, grips, shapes and spatial tags and relations, color, kinematics, attributes like twisting, bending and tries to avoid conflicting constraints by specifying path, orientation and position. But when these approaches were scaled down to interior designing, it failed to give good results for objects whose reference is a wall or the ceiling, neither does it consider the orientational aspect of objects.

2 OVERVIEW OF THE PROCEDURE

The task of generating a 3D scene involved cleaning of the data that the text processor processes. This cleaned data is given to the text parser which gives all the parent-child relationship of the 3D models developed by Chang et. al. in [10]. The output generated is later used to render a 3D scene.

3 PREPROCESSING

There were several types of noise that were encountered during text processing and each had to be resolved differently for different purposes. There were some modifications that were significant.

3.1 Corrections in the Punctuation

For punctuations like full stop and comma, the text parser considered the word with the punctuation and took it as a noun, e.g.” There is a chair, a table and a sofa.”. Here, “chair,” was processed as a noun instead of considering it as “chair” “,” which is the required format.

3.2 Numerical Representations in the result

For numbers written as words, e.g. “three”, it had to be converted into numerical “3”. This is useful for the parser to know how many objects were considered in case of plural forms of the word.

3.3 Parts of Speech

A Parts of Speech (POS) Tagger developed by Toutanova et. al. in [2,4] was used for the given input sentence. The tags that were used were NN (Singular Noun), JJ(Adjective), IN(Preposition), CD (Cardinal Number), NNS (Plural Noun), PRP (Personal Pronoun), EX (Existential There), VBZ (Verb, 3rd Person Singular Present), DT(Determiner).

3.4 Modifications of tags

When the main objective is orientation of objects, some of the prepositions like ‘top’ and ‘front’ were tagged as noun(NN), which was not suited for interior designing.

Hence, the next task was to change some of the tags. e.g. “A lamp on top of a table.” Here, top is a preposition rather than the noun form. These tags had to be changed to their appropriate POS tag, i.e. preposition(IN). Some prepositions like “in front of” and “next to” were taken as separate words instead of a single preposition. This was revised in the text processor.

4 TASK DESCRIPTION

The preprocessed text was now fed to the Core NLP designed by Manning et. al. in [9] and the desired output was generated.

4.1 3D model characteristics

Since all the 3D models of Shapenet in [10] were singular in nature, presence of a plural word in the given input would not map the word to the corresponding model. To avoid this complication, stemming and lemmatizing of the words were done E.g. “chairs”, which is the plural form, is considered as “chair” in the final output of the processor.

4.2 Multiple child scenario

In the case that many objects are related to a single parent, e.g. “There is a lamp and a bottle on a table.”, both the children (lamp and bottle) are mapped to a single parent (table) and the appropriate relation (on((lamp, bottle),table) was obtained, accounting for a common parent for multiple children.

4.3 Multiple object scenario

s our models were singular in nature, when the input sentence contained multiple nouns of the same type, e.g. “Three chairs next to a table.”, separate relations were made for each parent-child relationship. The output for the example given is,

next_to(chair, table),
next_to(chair, table)
next_to(chair, table).

4.4 Text processor features

One of the outputs of the processor contained the word followed by the stemmed, lemmatized form of the word (which was used for getting all the nouns), the POS tag, the Named Entity Recognition (If present), and the parsed output.

The output of the text processor also contained tokens, expression parse trees, parent-child relation (if present). From this, the final

output of the text processor considered all the nouns that were present in the sentence and the parent-child relationship.

5 MAPPING OF 3D MODEL WITH THE OUTPUT OF THE TEXT PROCESSOR

From the text file, a suitable object had to be chosen from the list of models obtained from shapenet. The models were organized based on Wordnet's Synset(Synonym Set) ID ([1]).

The output of the processor contained a Synset (Synonym Set) ID of the models. A suitable object was chosen from this synset ID. All the 3D models containing the particular noun in the input were considered. Every model had tags associated with them. There were many models that could be considered to the particular noun. Some models had many tags. This reduces the probability of it being the model that the user expects. Hence, we chose the model with the least number of tags. E.g. Model with only the annotation 'table' was considered instead of a model with annotations 'table, study table, desk, bench'. The latter would be inappropriate for the 3D scene if the user expected a basic table.

If the object (the model) had no relation with respect to any other object specified in the user input, then the implied object will be related to the object i.e the implied object becomes the parent of the model (e.g. the implied object for a 'table' is the 'floor').

5.1 Relative Scaling of Objects

Every object required will be of different size with respect to every other object. E.g. a table and a lamp will be of different scales. The scale of the objects was considered when generating the 3D scene. distinguishes it from a numbered equation.

6 PLACEMENT OF OBJECTS

The output generated so far maps the text processor and the corresponding 3D model. The output from the processor is read so as to identify the objects needed. Preprocessing is done to the models to make them into a usable form and the output is generated in the 3D rendering engine, Blender .

The algorithm uses a rule-based approach. This approach generates a set of suitable points which satisfies the constraints specified by the user. The constraints include the parent object and the preposition, as well the constraints of the room and previously placed objects.

6.1 Collision detection

Collision detection is the process of detecting whether objects occupy each other's space. Collision detection is essential for proper object placement as without it, objects might overlap. This was done for all objects before placing. It was done by taking into account all the surfaces of the object to be placed and other objects that were placed before it.

6.2 Bounds checking

The objects had to be placed in a room. Hence, the bounds of the room were the floor and walls. Bounds checking was done for each object for the wall and floor. If either of the two failed, it would check for the next point in the set of points. After each object was placed, it was appended into a list so that its information was available for future collision checking.

7 RESULT

The results of our model for some sentences are as follows

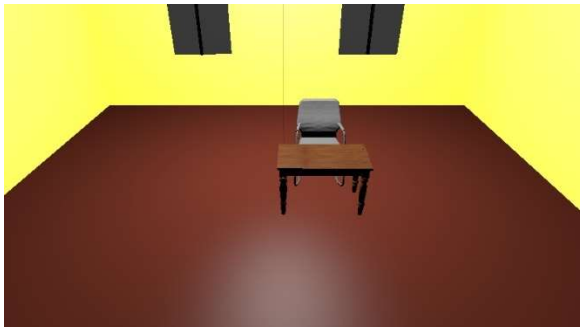

Figure 2: Result for the sentence "There is a table in front of a chair".

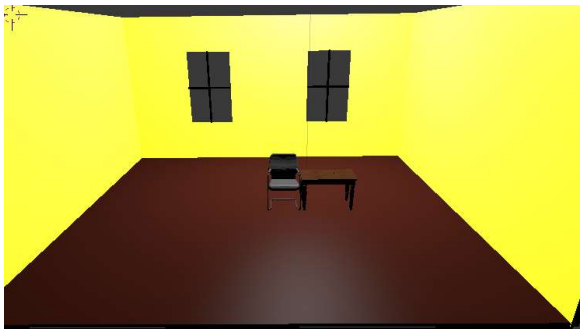

Figure 3: Result for the sentence "There is a table next to a chair".

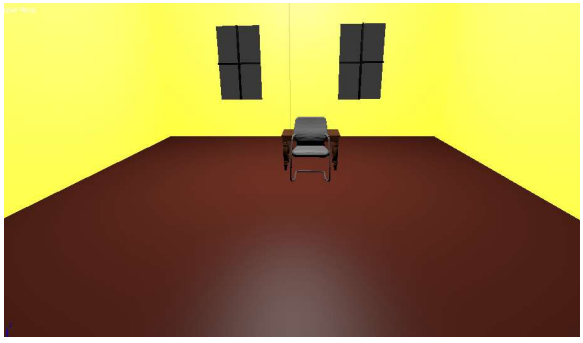

Figure 3: Result for the sentence "There is a table behind a chair".

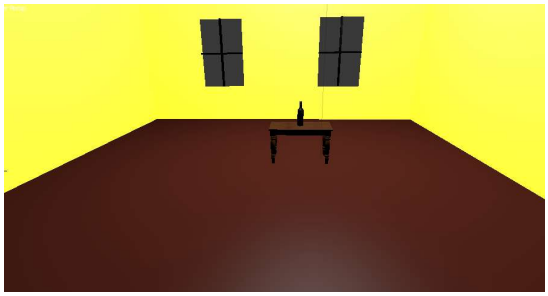

Figure 4: Result for the sentence "There is a bottle on a table".

Figure 5: Result for the sentence "There is a laptop, bottle and a lamp on a table".

Figure 6: Result for the sentence "There are four chairs to the left of a table".

Figure 7: Result for the sentence "There are three bottles on a table".

Figure 8: Result for the sentence "There is a clock on the left wall. There is a clock on the right wall. There is a clock on the front wall".

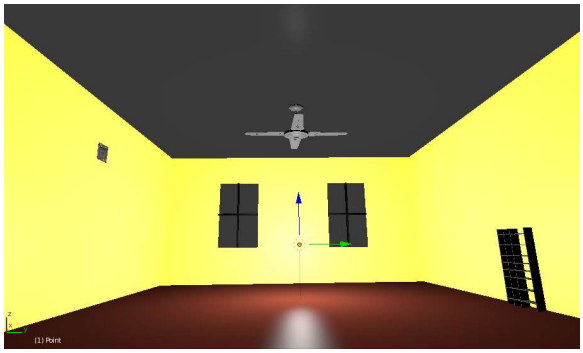

Figure 9: Result for the sentence "There is a fan on the ceiling. There is a bookshelf in front of the right wall. There is a clock on the left wall".

8 COMPARISONS

Although the results of both WordsEye and Stanford's text to 3D engine were accurate for most cases, there were some errors with respect to orientation of objects.

8.1 Comparative results for the sentence "There is a clock on the right wall"

Figure 10: Results from Stanford's Text to 3D Scene Generator

Figure 11: Results from WordsEye's Text to 3D Scene Generator

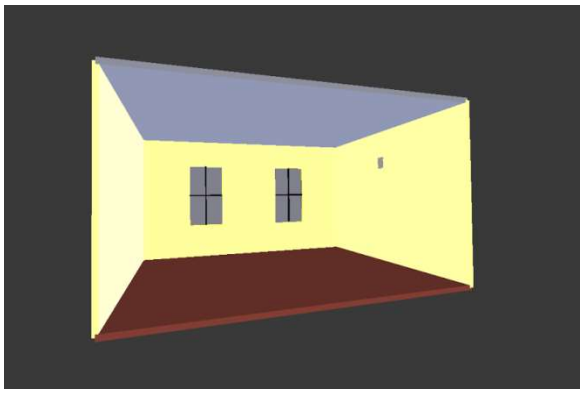

Figure 12: Result from our Text to 3D Scene Generator

Both the models interpreted the preposition "on" and the adjective "right" differently and generated the result. Although semantically correct, when the user considers interior designing, the expected result would differ.

8.2 Comparative results for the sentence "There is a laptop, bottle and a lamp on a table"

Figure 13: Results from Stanford's Text to 3D Scene Generator

Figure 14: Results from WordsEye's Text to 3D Scene Generator

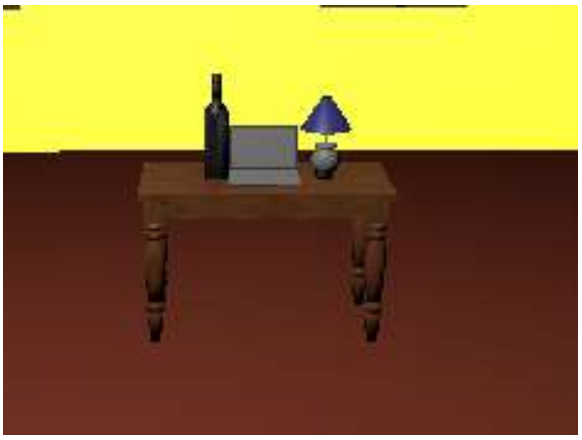

Figure 15: Result from our Text to 3D Scene Generator

Stanford's Text to 3D scene generator has considered multiple child for a single parent but the 3D models taken do not match the scenario.WordsEye on the other hand has not considered multiple children for a single parent and hence the output is not as expected.

9 DRAWBACKS

There were many erroneous results from the final text processor output. For sentences such as "There is a chair in front of a table. There is a sofa in front of the chair." There should have been two instances of "in_front_of" with different nouns, but there was only one instance of "in_front_of" which mapped to one of the sentences. There is a similar case for the preposition "next_to".

There were limitations with respect to the quantity of a particular object. The limit for the quantity of objects is 20. If the user wants an object more than twenty times, it is not considered. The text processor works for few limited sentences at a time.For objects that had to be placed on a table, if there was lack of space, it would not be placed in an

appropriate position. Bounds checking for individual object was not done in these cases.

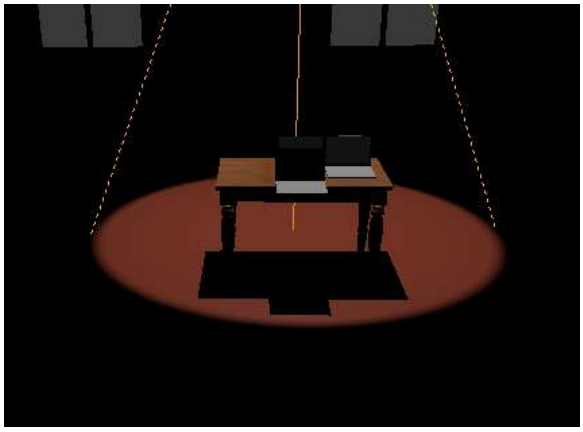

Figure 16: "There are three laptops on a table". Since for all the three laptops, parent object is table and since proper bounds checking was not done, one of the laptops was not placed on top of the table.

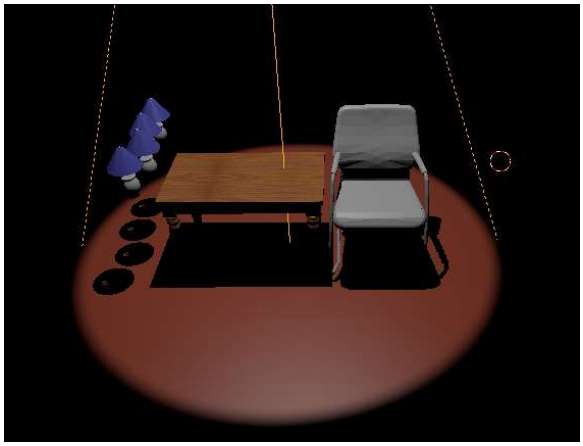

Figure 17: "There is a lamp on a table and there are four chairs to the left of the table.". Interpretation of the multiple object was incorrect and this resulted in a wrong scene generation.

10 CONCLUSION

In this paper, we present a technique to generate a 3D scene from text using Natural Language Processing and a 3D rendering engine. Our model yielded good accuracy with respect to user perspective of a scene. It also generated expected results with respect to the orientation of objects. For complex sentences which involved many interconnected parent-child relationships, the results were incorrect. At present, when the user enters more than three sentences, the result can be incorrect. We plan to make it work for a complex set of sentences. The same object is selected regardless of the context. If the object selection is randomized then the result may be incorrect. Hence, a list of appropriate objects should be made. With respect to the object placement, better object collision and bounds checking should be implemented. More research has to be done on this subject to get better results when the user enters a paragraph or when the sentence is ambiguous.

REFERENCES

[1] George A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM, v.38 n.11, p.39-41, Nov. 1995.*

[2] Kristina Toutanova and Christopher D. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *EMNLP '00 Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13, Page 63-70.*

[3] Bob Coyne and Richard Sproat. WordsEye: An Automatic Text-to-Scene Conversion System. *In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, p.487-496, August 2001*.

[4] Kristina Toutanova, Dan Klein, Christopher D. Manning and Yoram Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, Pages 173-180.*

[5] Angel X. Chang, Manolis Savva, and Christopher D. Manning. Learning Spatial Knowledge for Text to 3D Scene Generation, *EMNLP 2014.*

[6] Angel X. Chang, Manolis Savva, and Christopher D. Manning, Interactive Learning of Spatial Knowledge for Text to 3D Scene Generation, *In Proceedings of the ACL 2014 Workshop on Interactive Language Learning, Visualization, and Interfaces (ACL-ILLVI).*

[7] Angel X. Chang, Manolis Savva and Christopher D. Manning. Semantic Parsing for Text to 3D Scene Generation. *In Proceedings of the ACL 2014 Workshop on Semantic Parsing.*

[8] Angel Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D. Manning, "Text to 3D Scene Generation with Rich Lexical Grounding , *ACL 2015.*

[9] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014, pp. 55-60.*

[10] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi & Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. *CoRR, abs/1512.03012.*