International Journal of Innovative Research in Engineering Science and Technology

APRIL 2018 ISSN 2320 –981X

PARSING TEXT TO 3D SCENE CONVERSION USING LEARNED SPATIAL KNOWLEDGE

[1]Ms.M.Parameswari, [2]Ms.L.Sindhu, [3]Ms.R.Poovarasi, [4]Ms.M.Sasikala

[1] Assistant Professor
Department of Computer Science and Engineering,
Dhirajlal Gandhi College of Technology,
Salem-636309, India.
parameswariinfo@gmail.com

[2] Assistant Professor
Department of Computer Science and Engineering,
Dhirajlal Gandhi College of Technology,
Salem-636309, India.
Sindhu.cse@dgct.ac.in

[3]Assistant Professor
Department of Computer Science and Engineering,
Dhirajlal Gandhi College of Technology,
Salem-636309, India.
pooradhu@gmail.com

[4]Assistant Professor
Department of Computer Science and Engineering,
Dhirajlal Gandhi College of Technology,
Salem-636309, India.
Sasibtech91@gmail.com

ABSTRACT

Many things can be easily understood by people as an image rather than saying it as a text. The Text to Scene aims to explore how to automatically generate 3D scenes from a natural text description. The ability to map text to 3D geometric scene representations has many applications in areas such as art, education, and robotics. However, prior work on the text to 3D scene generation task has used manually specified object categories and language that identifies them. We introduce a dataset of 3D scenes annotated with natural language descriptions and learn from this data how to ground textual descriptions to physical objects. This paper highly focuses on lexical analysis (a branch of NLP) things by converting the text inputted into 3D images. The main innovation of this work is to show how to augment these explicit constraints with learned spatial knowledge to infer missing objects and likely layouts for the objects in the scene. We demonstrate that spatial knowledge is useful for interpreting natural language and show examples of learned knowledge and generated 3D scenes. Using a POS tagger, these results in the construction of parse tree, it checks the Spatial Dependencies between the words which are parsed and generates a 3D scene.

Keywords: NLP, postagger, POS, spatial, 3D

1. INTRODUCTION

The aim of this paper to create 3D scene generation i.e. image representation for the input data. This is a branch of natural language processing also known as linguistic analysis. In this modern world we don't have time to patiently stand and listen to the content, so it would be good to work by understanding with the help of pictures and it high time for us to implement this idea immediately. Thus we propose a Text to 3d Scene generation system that incorporates user interaction. A user provides a natural language text as an input to this system and the system then identifies spatial dependencies between the sentences and then maps the sentence with the corresponding image in the database available. It is also said that we remember images more than the text. So this paper is a very small step to make the world to help in progressing. Teachers need to be able to draw on a variety of representations as there is "no single most powerful forms of representation. The diagrams and animation should illustrate concepts in a way which is impossible in any other medium. There are many fields like graphics designing, education, arts, Robotics this conversion of text to 3D image will play a very important role. This 3D scene conversion also mainly expands its application area in fields like architecture where it would be easy for the person to figure out the output before it gets practically implemented. Here the data from the user is got as a input in the language and then mapped to the corresponding data model which then renders the 3D image of the input data.

Fig 1: Generated scene for "There is an ice cream on the table."

Support vector regression algorithm is mainly used to classify and map the sentence with the corresponding image. After checking for the spatial dependency the sentence will be tokenized. The complex work of parsing the given sentence will be carried out by the postagger.

2. RELATED WORK

2.1 LEARNING SPATIAL KNOWLEDGE FOR TEXT TO 3D SCENE GENERATION

Angel X. Chang et.al [2] address the grounding of natural language to concrete spatial constraints, Inference of implicit pragmatics in 3D environments. It applies the approach to the task of text-to-3D scene generation. Here it presents a representation for common sense spatial knowledge and an approach to extract it from 3D scene data. The spatial knowledge representation that can be learned from 3D scenes and captures the statistics of what objects occur in different scene types, and their spatial positions relative to each other. The model spatial relations (left, on top of, etc.) and learn a mapping between language and the geometric constraints that spatial terms imply. It shows that using the learned spatial knowledge representation, infer implicit constraints, and generate plausible scenes from concise natural text input.

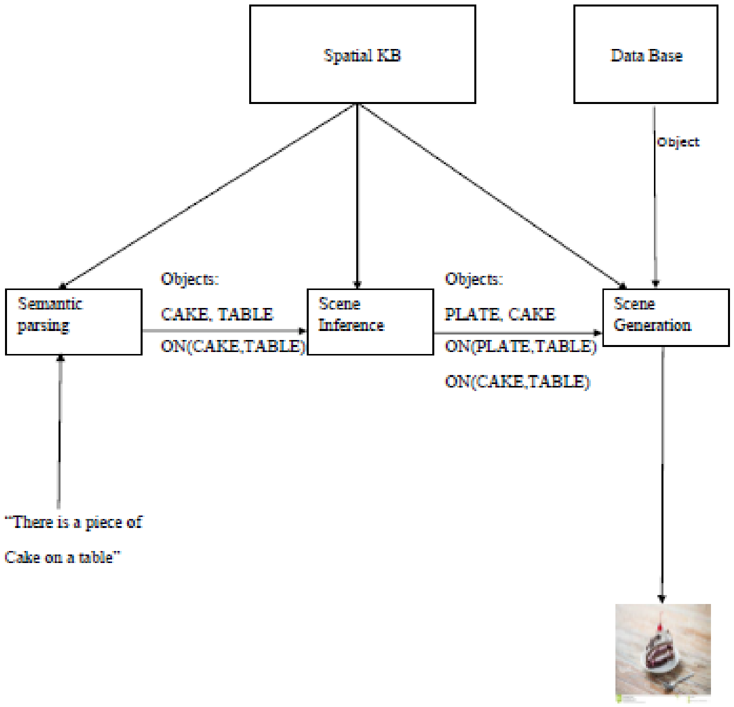
2.2 SEMANTIC PARSING FOR TEXT TO 3D SCENE GENERATION

Angel X. Chang et. al [3] define text to scene generation as the task of taking text describing a scene as input, and generating a plausible 3D scene described by that text as output. More concretely, it parse the input text into a scene template, which places constraints on what objects must be present and relationships between them. Next, using priors from a spatial knowledge base, the system expands the scene template by inferring additional implicit constraints. Based on the scene template, it select objects from a dataset of 3D models and arrange them to generate an output scene. After a scene is generated, the user can interact with the scene using both textual commands and mouse interactions. During interaction, semantic parsing can be used to parse the input text into a scene. It parses textual input provided by the user into a sequence of commands with relevant parts of the scene as arguments. Spatial knowledge can be helpful for resolving ambiguities during parsing. Here it describes a system prototype to motivate approaching text to scene generation as a semantic parsing application. While this prototype illustrates inference of implicit constraints using prior knowledge, it still relies on hand coded rules for mapping text to the scene representation.

3. WORKING METHODOLOGY

The process actually starts like we get the input from user as the natural language. The sentence is being parsed; it gets parsed like spots out semantics available in that sentence. It first checks for the noun in the sentence, then does it for the verb, then adjective, after which adverb and so on. Each and every particle is listed. Then finally after this the tokenized output, actually works as the input for the next phase. The output of the first phase is the input for our next process, which puts into the spatial domain, in others words it could be made simple by saying we perform the dependency checking operation. The spatial concept is foremost important because only with the accuracy of our paper mainly deals with; consider this in the real scenario. We mention there should a wooden table in the center of the room. If the system considers wood separately and table separately. Then it would display as if an ordinary table having wood on it. This would become more complex if dependency is not checked properly.
The single sentence is being parsed; it divides each sentence into noun part, verb part, adjective and all. They are abbreviated as NN which indicates noun, CC which says about conjunction part of the sentence inputted. It still distinguishes preposition, cardinal number and adjective. There are many new things still undiscovered or which is not yet familiar among people like pound sign, interjection, particle, symbol and so on.

After postagging stage the output is a sentence, this now acts as the input for the next phase. Here is where the real complexity lies because we have to check the spatial dependency, which is foremost important for the accuracy of our paper. Spatial dependency which can easily connect and match to the nearest scene, else we will find some kind of confusion. Others proposed ideas and papers have a drawback in finding the accurate spatial relationship between them. Here we make use of our algorithm namely SVR support vector regression which is helpful in getting accurate spatial relation. Then now we have to map the data i.e. mapping process takes place. In this we have assigned a key value for each sentence during tokenization and based on that it maps to the data. Then it checks for the nearest match with the image. Consider an example of any sentence say; Ravi is sitting on a wooden chair. In this example ravi is the subject and it will be abbreviated as NN and is would be the verb made s VV and while checking for wood it is the adjective. But while checking for the spatial relation we have a probability of showing table which is having wood upon it. But the real meaning is we should show a wooden table. This is where the real dependency lies. After which the output of our paper is to appear, it shows the generation of 3D scene .So after postagging the sentence, we have to map the sentence using SVR and finally we can generate 3D scene, which is our output.



3.1 DATASET

We introduce a dataset of 3D scenes annotated with natural language and learn from this data how to ground textual descriptions to physical objects. We examine the task of text to 3D scene generation. The ability to map descriptions of scenes to 3D geometric representations has a wide variety of applications; many creative industries use 3D scenes. Robotics applications need to interpret commands referring to real-world environments, and the ability to visualize scenarios given highlevel descriptions is of great practical use in educational tools.

text	category	text	category
chair	Chair	round	RoundTable
lamp	Lamp	laptop	Laptop
couch	Couch	fruit	Bowl
vase	Vase	round table	RoundTable
sofa	Couch	laptop	Computer
bed	Bed	bookshelf	Bookcase

Fig 2: Examples of data set for 3D scene

3.2 TAGGING APPROACHES

Nouns are traditionally grouped into proper nouns and common nouns. Proper nouns, like Regina, Colorado, and IBM, are names of specific persons or entities. Common nouns are divided into count nouns and mass nouns. Count nouns are those that allow grammatical enumeration; that is, they can occur in both the singular and plural (goat/goats, relationship/relationships) and they can be counted (one goat, two goats). Mass nouns are used when something is conceptualized as a homogeneous group. The verb class includes most of the words referring to actions and processes, including main verbs like draw, provide, differ, and go. English verbs have a number of morphological forms (non3rd-person-sg (eat), 3d-person-sg (eats), progressive (eating), past participle eaten). The third open class English form is adjectives; semantically this class includes many terms that describe properties or qualities. Most languages have adjectives for the concepts of color (white, black), age (old, young), and value (good, bad), but there are languages without adjectives.

- prepositions: on, under, over, near, by, at, from, to, with
- determiners: a, an, the
- pronouns: she, who, I, others
- conjunctions: and, but, or, as, if, when
- auxiliary verbs: can, may, should, are

3.3 WORD CLASSES

Part-of-speech tagging (or just tagging for short) is the process of assigning a part-of speech or other lexical class marker to each word in a corpus. Tags are also usually applied to punctuation markers; thus tagging for natural language is the same process as tokenization for computer languages, although tags for natural languages are much more ambiguous. Taggers play an increasingly important role in speech recognition, natural language parsing and information retrieval. The input to a tagging algorithm is a string of words and a specified tag set of the kind described in the previous section. Even in simple sentences, automatically assigning a tag to each word is not trivial. For example, book is ambiguous. That is, it has more than one possible usage and part of speech. It can be a verb (as in book that flight or to book the suspect) or a noun (as in hand me that book, or a book of matches). Similarly that can be a determiner (as in Does that flight serve dinner), or a complement (as in I thought that your flight was earlier)[4].

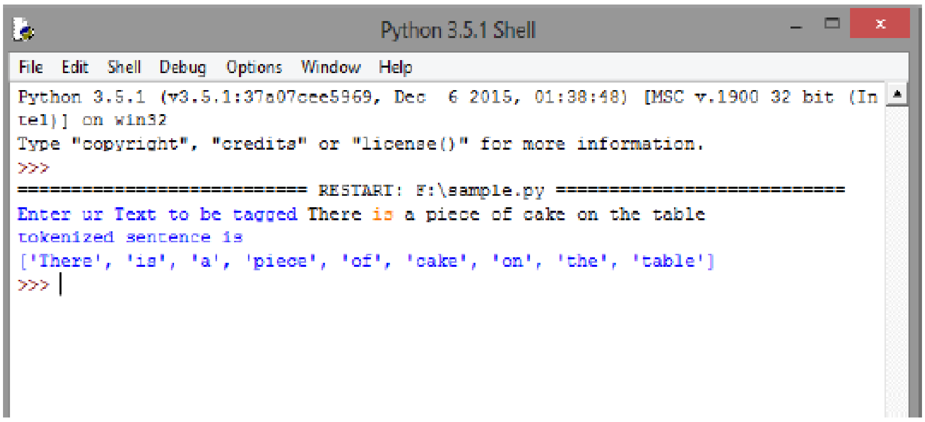
4. IMPLEMENTATION AND RESULTS

The first step is to get a sentence from the user for tokenization. In tokenization each word in the sentence is being given an unique identifier called tokens. After the tokenization of the sentence the next important module has to be executed that is Parts of speech Tagging (POS Tagging). It is done to segregate the nouns, verbs, adjectives etc. Here the given sentence is split into parts of speech. The next phase is to get the 3D image for the given sentence. This is done by the database which is connected using SVR algorithm. This algorithm helps in finding the spatial dependencies between the words in the sentence. SVR helps in one to one mapping of the tokenized and postagger sentence to the image which has its unique ID. The image is in the database as a binary format and it maps the sentence using the particular ID. Thus it creates the final output as 3D scene.

4.1 TOKENIZATION

A tokenizer divides text into a sequence of tokens, which roughly correspond to "words". We provide a class suitable for tokenization of English, called PTB Tokenizer. It was initially designed to largely mimic (PTB) tokenization, hence its name, though over time the tokenizer has added quite a few options and a fair amount of Unicode compatibility, so in general it will work well over text encoded in the Unicode Basic Multilingual Plane that does not require word segmentation or more exotic language-particular rules (such as writing systems that use : or ? as a character inside words, etc.).

An ancillary tool uses this tokenization to provide the ability to split text into sentences. PTB Tokenizer mainly targets formal English writing rather than SMS-speak.Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens* , perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization.[6]

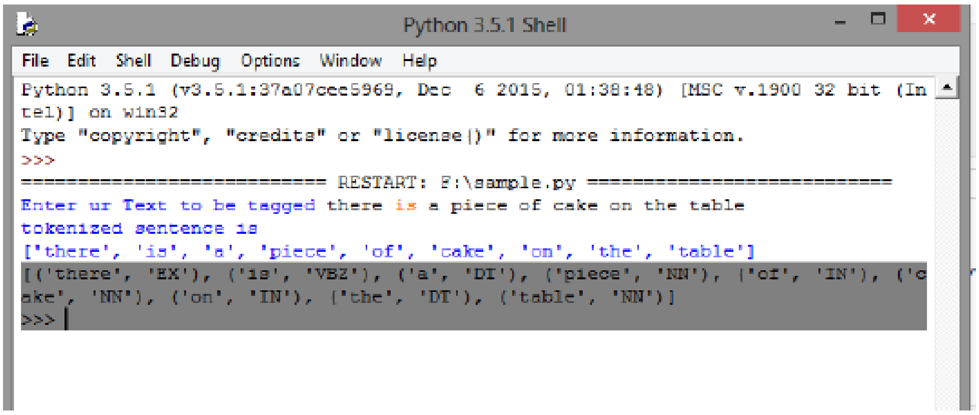


```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=========================== RESTART: F:\sample.py ===========================
Enter ur Text to be tagged There is a piece of cake on the table
tokenized sentence is
['There', 'is', 'a', 'piece', 'of', 'cake', 'on', 'the', 'table']
>>> 
```

Fig 3: Tokenizer

4.2 POSTAGGING

The process of assigning part of speech for every word in a given sentence according to the context is called as part of speech tagging. This is one of the useful tasks in Natural Language Processing (NLP). It plays an important role in Speech and NLP such as Speech Recognition, Speech Synthesis, Information Retrieval, word sense disambiguation and machine translation and also in 3d scene generation. Many algorithms have been developed for part of speech (POS) tagging. Support Vector Regression(SVR), Hidden Markov Model (HMM), Maximum Entropy Markov Model (MEMM), Neural Networks (NN), Decision Trees, Rule based and Transformation based techniques are to name a few. The paper is about performing Part of speech tagging to the Indian language using SVR Tool, which was implemented using support vector regressions. Words are divided into different classes called parts of speech (POS; Latin pars orations), word classes, morphological classes, or lexical tags. Part-of-speech tagging (POS tagging or POST), also called grammatical tagging, is the process of marking up the words in a text as corresponding to a particular part of speech, based on both its definition, as well as its context —i.e., relationship with adjacent and related words in a phrase, sentence, or paragraph. Parts of speech can be divided into two broad super categories: closed class types and open class types.[5] For example, prepositions are a closed class because there is a fixed set of them in English; new prepositions are rarely added. By contrast nouns and verbs are open classes because new nouns and verbs are continually added or borrowed from other languages. It is likely that any given speaker or corpus will have different open class words but all speakers of a language, and corpora that are large enough, will likely share the set of closed class words.



```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
=========================== RESTART: F:\sample.py ===========================
Enter ur Text to be tagged there is a piece of cake on the table
tokenized sentence is
['there', 'is', 'a', 'piece', 'of', 'cake', 'on', 'the', 'table']
[('there', 'EX'), ('is', 'VBZ'), ('a', 'DT'), ('piece', 'NN'), ('of', 'IN'), ('c
ake', 'NN'), ('on', 'IN'), ('the', 'DT'), ('table', 'NN')]
>>> 
```

Fig 4: POSTAGGING

4.3 TEXT TO SCENE GENERATION

In the text to 3D scene generation task, the input is a natural language description, and the output is a 3D representation of a plausible scene that fits the description and can be viewed and rendered from multiple perspectives. More precisely, given an utterance x as input, the output is a scene y: an arrangement of 3D models representing objects at specified positions and orientations in space. It defines text to scene generation as the task of taking text describing a scene as input, and generating a plausible 3D scene described by that text as output. More concretely, we parse the input text into a scene template, which places constraints on what objects must be present and relationships between them. Next, using priors from a spatial knowledge base, the system expands the scene template by inferring additional implicit constraints. Based on the scene template, we select objects from a dataset of 3D models and arrange them to generate an output scene.

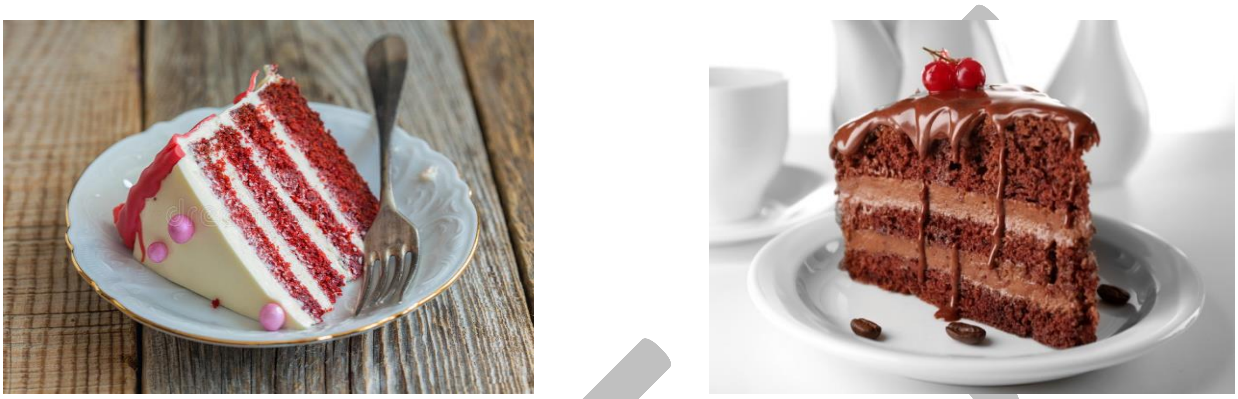


Fig 5: Results of 3D conversion for the text "a piece of cake on the table"

5. CONCLUSION

Prior work in 3D scene generation is based on rule-based technique; we now have improved the spatial dependency between the sentences. It really helps in checking the spatial relation which would help us come out of confusions. This paper presents an idea which helps to map the postagger sentence with the image; it just finds the nearest match and generates the formulated output. As mentioned we request the user to enter the sentence of his or her choice. The system takes this as the input and then decides to tokenize the sentence first and then to postage it i.e. the process of segregating every word in its parts of speech. The process semantic parsing takes place. It recognizes the noun part, verb, particle, adjective, and adverb and puts a tag set for all. After which spatial relation is verified. It takes ample of time to process a sentence internally, but produces the output immediately. This sentence is now taken for mapping phase, which is the last phase of the paper. Now this sentence is taken and SVR classifies considering which would be the nearest match to the sentence. It then maps that particular image from the database to this sentence and displays the output.

6. REFERENCES

1. Aker A and Gaizauskas R (2010), 'Generating image descriptions using dependency relational patterns', InPr. ACL, Pages1250–1258.

2. Angel X. Chang, Manolis Savva and Christopher D. Manning (2014), ‘Learning Spatial Knowledge for Text to 3D Generation’, In EMNLP.

3. Angel X. Chang, Manolis Savva and Christopher D. Manning (2014), ‘Semantic Parsing for Text to 3D Scene Generation’, In SCENEGEN.

4. BobCoyne and RichardSproat (2001), ‘WordsEye:An Automatic Text-to-Scene ConversionSystem’.

5. Brants T, Popat A. C., Xu P, Och F. J, and Dean J (2007), ‘Large language models in machine translation’, In EMNLP.

6. Duygulu P, Barnard K, de Freitas N and Forsyth D (2002), ‘Object recognition as machine translation’, In ECCV.

7. Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg and Tamara L Berg (2011), ‘Understanding and Generating Image Description’, In CVPR.

8. Yangyan Li, Angela Dai, Leonidas Guibas and Matthias Nießne (2015), ‘Database-Assisted Object Retrieval for Real-Time 3D Reconstruction’, Eurographics/O. Sorkine-Hornung and M. Wimmer, Volume 34, Number 2.