Visualization of Mechanics Problems based on Natural Language Processing

Aditya Nayak
Dept. of Information Technology K.J.Somaiya College of Engineering

Narayan Acharya
Dept. of Information TechnologyK.J.Somaiya College of Engineering

Amanpreet Singh
Dept. of Information Technology
K.J.Somaiya College of Engineering

Avani Sakhapara
Dept. of Information Technology
K.J.Somaiya College of Engineering

Bharat Geleda
Dept. of Information Technology
K.J.Somaiya College of Engineering

ABSTRACT
In The ability to visualize our thoughts has always fascinated us. Even more intriguing is the subject of making a computer able to visualize those thoughts, just by understanding the human language. In this paper, a text to scene generation system is proposed for the educational domain where a basic Newtonian physics problem is conveyed to the system in natural language and the scene depicting the problem is generated and displayed to the user. The paper describes the implementation of the system as well as the results obtained. It is based on the integration of advances in NLP and computer graphics technology to generate a virtual environment. It makes it easier for students to visualize the problems and also helps teachers in explaining better.

General Terms
Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords
Natural language processing, Knowledge representation, scene generation, Physics education.

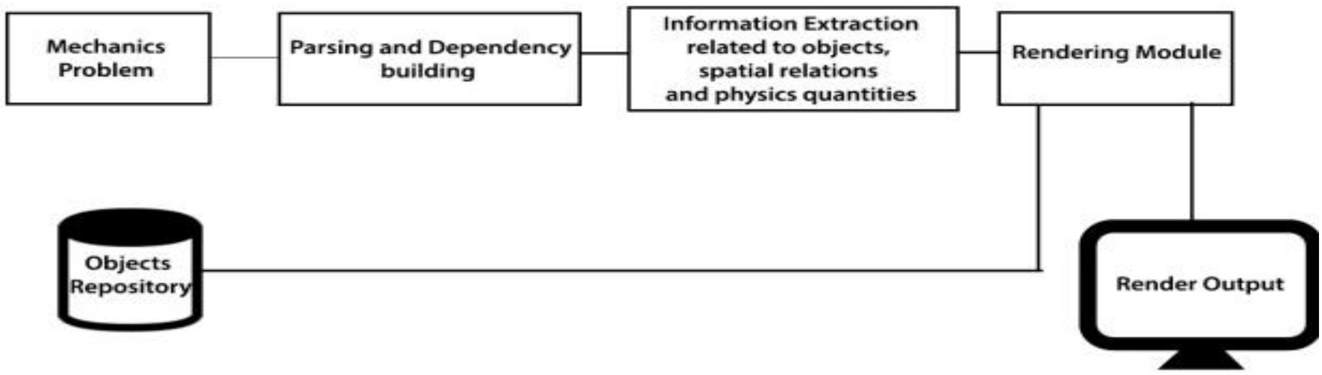
1. INTRODUCTION
Natural language is how humans communicate with each other. But the basic process of communication includes visualization of words and phrases in our minds. But, sometimes it is difficult to correctly perceive the spoken or written words. This is where graphical representation makes life a bit easier. Several software tools are available for such purposes. However, it is still difficult for many people to produce such graphics, since it is a tedious process and requires artistic skills. Moreover, getting fully acquainted with a software tool is difficult.

Instead the whole graphics generation process can besimplified from the user's perspective by giving input in the form of descriptions in natural language.

Sometimes, the problem statements that students come across are too complex to solve just by reading. A diagram or a figure is given or has to be imagined by the student and drawn in order to solve the problem.

The proposed text to scene generation system helps the user visualize Newtonian Mechanics problems in the form of simple images. The rest of the paper is organized as follows; in section 2 the related work is overviewed. In section 3 the methodology is discussed. Section 4 talks about the implementation details of the proposed system and in section 5 the conclusion is stated along with the results.

2. LITERATURE REVIEW
The conversion of natural language text into graphics has been investigated in a few projects over the years [6]. The Put system [5] is an early example based on spatial relationships between objects with input in the form of (X, Y, P) where X and Y were objects, and P was a spatial preposition. CarSim [4] is a domain specific system which converts a traffic accident report to 3D scene. More recently, WordsEye [3], a web based tool has been developed for depicting a scene by integrating natural language descriptions with libraries consisting of 3D objects. WordsEye relies on a large database of 3D models and poses to depict entities and actions. Every 3D model can have associated shape displacements, spatial tags, and functional properties to be used in the depiction process.



3. METHODOLOGY
The entire working of the system is divided into the following phases:

3.1 Mechanics Problem
The system, first of all accepts a Mechanics problem as the input from user in plain English. It is expected that the input is a valid question with correct grammatical structure.

3.2 Parsing and Dependency Building
Here the processing is divided into two parts: .

3.2.1 Information Extraction
The statement is passed through the NLP module. They are used to tag and recognize the various parts of speech in the sentence. Also, a dependency tree is obtained which is used to relate objects to verbs, adverbs, prepositions and other parts of speech which is then used for information extraction purposes.

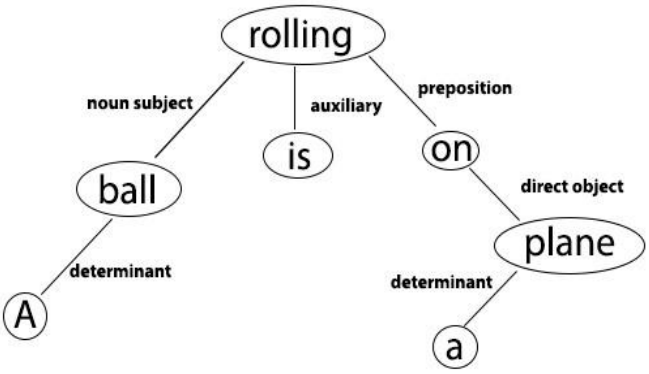


Figure 1: The Typed Dependencies obtained for our sample problem. These are used to obtain spatial relations among objects that are rendered

Dependencies and co-references are resolved among these and a dependency graph is made. Figure 2 shows the type dependency obtained after parsing the input.

The noun subjects are used along with their adjectives and prepositions. The results obtained above are analyzed to recognize the required attributes mentioned in the input problem viz. units, objects, physical, quantities etc. Table 1 displays the required attributes. This is done using TokensRegex [1] which is a custom annotator. Also, the various noun subjects are segregated into two classes based on whether they have to be rendered in the output or not. The objects are classified into custom tags which are then used to decide whether to render them.

Table 1. Recognizing the required attributes in the input

Question	SI Units	Physics Quantities	Physics Renderables
A block of mass 2 kg moving at a speed of 10 m/s accelerates at 30 m/s2 for 5.00 s.	kg,m/s, m/s2,s	Mass, Speed, Acceleration, Time	Block
A projectile is fired from the top of a 40 m high cliff with an initial speed of 50 m/s at an unknown angle	m,m/s	Speed	Projectile, Cliff
A block is projected along a rough horizontal road with a speed of 10 m/s.	m/s	Speed	Block, Road

3.2.2Objects and spatial representation
The most important part of scene depiction is the spatial relation between two or more objects. Spatial relation is most often denoted by prepositions like on, under, beyond and so on [2]. For example, in the sentence "The ball is on an inclined plane", it can be seen that the two objects, ball and inclined plane, are related by the preposition "on"[7]. This relation is obtained from the dependency tree and while rendering, the ball is placed on the inclined plane after performing appropriate transformations

3.3 3D Modeling
Please useFrom a set of problems, a finite set of objects are identified which are most commonly used in diagrams for representing the problems and their 3D models are created. A repository of such models is then used to find the approximate match of the physical objects mentioned in the input.

3.4 Scene Rendering
Here, the integration between NLP and computer graphics is done. The information extracted from the dependency structure is looked up in the repository and the appropriate transformations are applied on the 3D models before rendering the entire scene.

4. IMPLEMENTATION DETAILS

TABLE2 The sample problems taken as input

Question	**Physics quantities**	**Renderable Objects**	**Relationship Extracted**
A ball is rolling on a plane. What is the net force acting on it?	force	ball, plane	ball on plane
A block can slide on a frictionless incline.Its mass is 4 kg. Find the normal reaction.	-	block, incline	block on incline
A car starts from rest on 0.5km long bridge.	-	car, bridge	car on bridge

A body projected along a rough horizontal plane moves with a deceleration of 4.0 m/s2. What is the coefficient of kinetic friction between the block and the plane?	deceleration	block, plane	block along plane
A projectile is fired from a cliff of height. Find relation between height of cliff,velocity and distance covered.	height, velocity, distance	projectile, cliff	projectile from cliff

For the part of speech tagging in the NLP Module, the outputs of both Stanford NLP [9] pipeline as well as General Architecture for Text Engineering (GATE) [8] were analyzed on a corpus of sample problems. They are used to tag and recognize the various parts of speech in the sentence. Ultimately, Stanford NLP was chosen because of its simplicity simpler and suitability for our purpose as compared to GATE which has a steep learning curve.

Table 2 shows some of the sample problems taken as input and the information generated by the system for the output. If the first problem is considered, the ball and the plane are identified as the objects that have to be rendered in the scene. The quantity to be calculated, 'force' is recognized as a physics quantity. Also the preposition 'on' is used as to represent the spatial relation between the ball and plane. This spatial relation is given as the initial output.

Thereafter, the objects ball and plane are searched in the repository of models. The 3D models of the objects are created using Blender [10] modeling software and converted to an intermediary object file. When found, the objects are rendered as a complete scene described by the input problem as can be seen in Figure 3. For the rendering purpose Java 3D API and blender models are integrated with the Stanford NLP module.

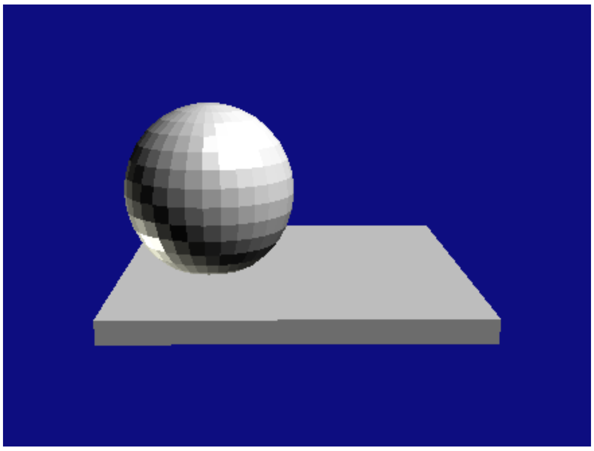


Figure 2: The objects ball and plane are rendered based on the obtained spatial relationship

The rendering times of the models from the repository and accuracy of depiction of the objects and their relations as per the input are used for evaluating the performance. The system was tested with a corpus of 20 mechanics problems, the output of which was then observed and inferences on accuracy of the system were drawn.

The criteria used for measuring the accuracy of depiction is as follows:

1. 1 point for result that extracts correct spatial relations and renders as per expectation.
2. 0.5 points for result that extracts correct spatial relations but does not render as per expectation.
3. No point for result that extracts incorrect spatial relations.

The reason for inaccurate rendering of scene may be due to unavailability of models in the repository or incorrect depiction.With these grading rules the system scored approximately 60% i.e. sum total of 45% questions that were accepted and 15%(i.e. 30%/2 due to criteria number 2) that were partially accepted. The results obtained are depicted in Figure 4.

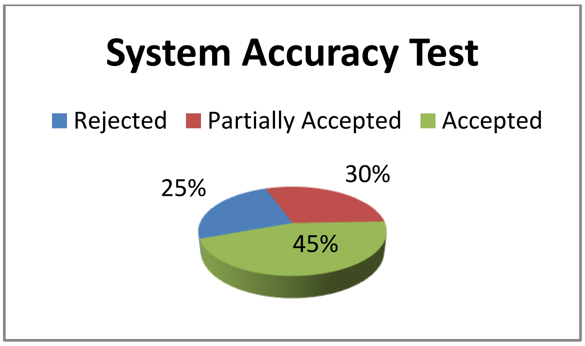


Figure 3: Results for system accuracy

Also, the entire process of rendering the models in Java 3D and loading of the Stanford NLP libraries is performed on PC running on an Intel core i5 processor with 2.3 GHz CPU and 4GB memory. Stanford NLP libraries require about 50 to 70 seconds to load during initialization. Also, the rendering procedure takes about 400 milliseconds once the input has been given.

5. CONCLUSION AND FUTURE WORK

The paper talks about an early prototype of a text to scene generation system in the educational domain which renders a scene representing a physics mechanical numerical problem

The proposed system aims to assist students and teachers in visualizing textual content as well as reduce the time taken to learn the intricacies of graphics software. It has elements of NLP and image processing and rendering and has varied applications in the broad range of domains.

The system in its current form can be used for educational purposes by students and teachers alike. Apart from that, on widening the scope, the opportunities are endless. The system functionalities can be extended to create automated story boards for movies, and visualizing written pieces of work such as books, newspapers, magazines, etc. Also, static scenes can be converted to animations to complete the visualization experience.

6. ACKNOWLEDGMENTS

We would like to thank K.J. Somaiya College of Engineering for letting us undertake this project and use their resources for our work. Also, we would like to appreciate the insightful inputs and efforts of Prof. AvaniSakhapara, without whom this uphill task would not have been possible.

7. REFERENCES

[1] Chang, Angel X., Savva, M. and Manning, Cristopher D. 2014 Interactive Learning of Spatial Knowledge for Text to 3D scene Generation. Computer Science Department, Stanford University.

[2] Zeng,Xin, Mehdi, Q. H. and Gough, N. E. 2005 3D Scene Creation using story based descriptions. School of Computing and IT, University of Wolverhampton, UK.

[3] Coyne, B. and Sproat, R. 2001 WordsEye: An automatic text-to-scene conversion system. Computer Graphics Proceedings pp. 487–496.

[4] Sylvain, D., Egges, A., Legendre, V., and Nugues, P., 2001 Generating a 3D simulation of a car accident from a written description in natural language : The Carsim system. In ACL2001: Workshop on Temporal and Spatial Information Processing, pages 1–8, Toulouse.

[5] Clay, S.R. and Wilhelms, J. 1996 Put: Language-Based Interactive Manipulation of Objects. IEEE Computer Graphics and Applications, pages 31–39.

[6] Adorni, G., Di Manzo,M., and Giunchiglia, F. 1984 Natural Language Driven Image Generation. In *COLING 84*, pages 495–500.

[7] Hawkins, B. 1984 The Semantics of English Spatial Prepositions. PhD thesis, University of California, San Diego, San Diego,CA.

[8] https://gate.ac.uk/ : General Architecture for Text Engineering (accessed on 18/7/2014)

[9] www.nlp.stanford.edu:The Stanford NLP Group (accessed on 20/2/2015)

[10] www.blender.org : Blender 3D modeling software (accessed on 1/3/2015)