Visualization of Free Body Diagrams

Vidhi Jain
Student,
Dept. of Computer Engineering,
K. J. Somaiya College of Engineering

Amola Hinge
Student,
Dept. of Computer Engineering,
K. J. Somaiya College of Engineering

Krishna Laddha
Student,
Dept. of Computer Engineering,
K. J. Somaiya College of Engineering

Saloni Buddhadeo
Student,
Dept. of Computer Engineering,
K. J. Somaiya College of Engineering

Deepak Sharma, PhD
Associate Professor,
Dept. of Computer Engineering,
K. J. Somaiya College of Engineering

ABSTRACT

When it comes to educational domain, Physics is one such subject which demands most imagination and the difficulty arises when we are unable to convey our imagination to others. Mechanographics is one such application which aides in visualization of the physics problem by generating a 2D structure depicting forces, movements and resulting reactions acting on the body. Such graphical illustrations are referred as Free Body Diagrams (FBD). In this paper we present a comprehensive system, Mechanographics, which processes a physics problem using Natural Language Processing (NLP) in Java, and extract the information in the form of dependencies to render a 2D diagram on the screen using JavaFX.

General Terms

Natural Language Processing, Graphics Representation.

Keywords

2D scene rendering, Free Body Diagrams, International Standard Units (S.I. Units), Typed Dependencies.

1. INTRODUCTION

An era where technology has put its hands into almost every teaching institutions, there is a need for a system which can generate scene from text to help students understand what their question is about - a system which can understand problems in natural language, by incorporating knowledge of human-to-human interaction and give its graphical illustrations. This will not only aid in imagination, but will most certainly remind them of the detailing which are easily forgotten by a rather confused human mind. Drawing of a free-body diagram helps to visualize all the forces acting on a single object along with the direction, which otherwise is a time consuming task. Mechanographics aims at solving such mechanics problem by giving the corresponding Free Body Diagram (FBD) and its equation. The system also gives summarized information about the problem by displaying all the physical and derived quantities with their units, which are extracted from the question and have an impact on the FBD.

The system is implemented in two modules: Text Processing and Graphics Rendering.

Initially, the system takes input in general English language, then parses and retrieves the information using Stanford CoreNLP [1] parser for semantic analysis. Information thus deposited by the NLP libraries are used to classify the mechanics problem, identify all the objects, forces and their related physical quantities. Later, the extracted data is worked upon to convert all the physical quantities in their corresponding S.I. unit. The processed and polished data is rendered on the screen as summarized data and graphical format, also along with its equation using JavaFX. An interface is developed to take the input and display the output as text, FBD and Equation. This paper provides a system overview, and in the subsequent sections then describes the implementation of the prototype system with an example followed by evaluation and discussion. The final section covers conclusions and directions for future work.

The related approach has been defined in Visualization of Mechanics Problem [2] but the originality of this paper lies in creating spatial arrangement for every type of object (plane or inclined surface), solve every possible way the sentence can be framed in common language using different conjunctions, for a given category of physics problem (active/passive voices) and be able to map the respective forces precisely by determining their magnitude, units and direction. The system also deals with human assistance, error handling with real time interaction.

2. RELATED WORK

Text-to-scene generation using NLP has been a topic of research for quite a while. One such web based system is WordsEye [3], which focuses on translating the semantic intent of the user, as expressed in language, into a graphic representation. It renders 3D image of any scene by extracting objects from the large database and poses to depict entities and actions. The 3D model can have associated attributes, displacement, spatial representation and functional properties of the objects.

Another such system is CarSim [4], which visualizes and animates 3D scenes from car accident reports. With an accuracy of 17%, it processes within the reports, gathers relevant information and converts it into a formal description. The formal description is thus used to get the corresponding 3D structure and animate it.

3. PROPOSED SYSTEM

The proposed system to convert text-to-scene is limited to the scope of physics problem based on Newton's Law of motion, where it assumes the question to be in appropriate grammar.

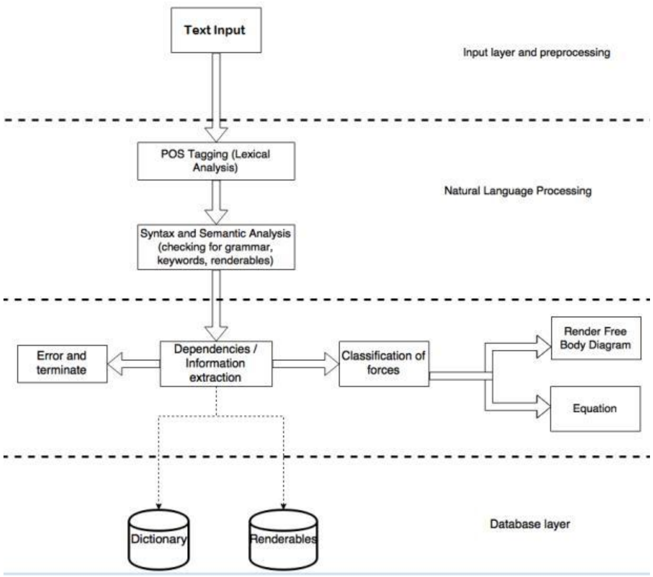

Text Input
Input layer and preprocessing
POS Tagging (Lexical Analysis)
Natural Language Processing
Syntax and Semantic Analysis (checking for grammar, keywords, renderables)
Error and terminate
Dependencies / Information extraction
Classification of forces
Render Free Body Diagram
Equation
Dictionary
Renderables
Database layer

Fig 1: Multi-tier system architecture

The application, Mechanographics, takes the input in common language; the user is provided with assistance in the form of auto-correction and is given spelling suggestions. The input question given by the user is given to the language component i.e. Stanford CoreNLP [1], to convert the syntactic structure of sentence into a semantic representation and extract the meanings about the category of the problem and relations between the physical quantities in the form of Universal Dependencies.

The dependencies are then worked upon using text processing algorithm to identify the objects and physical quantities associated with nouns and adjectives. This information is used to assemble the original virtual scene and to render the corresponding free body diagram using JavaFX.

Using Newton's Law of Motion, final equation is computed.

Example 3.1: A ball is resting on a table

Render-able: ball as an object, table as a plane surface

Example 3.2: A ball is resting on an inclined table

Render-able: ball as an object, table as an inclined surface

4. IMPLEMENTATION DETAILS

4.1 User input

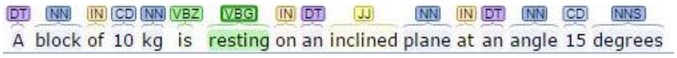
The user- interface, developed in JavaFX provides the user to enter the input in textual format. It is facilitated with human assistance in the form of auto-correction and auto-completion implementing data structures like HashSet and Tries [5] on words file provided with Mechanographics, in Java. It also carries forward the limitations of Stanford CoreNLP for sentences having the word 'respectively' as adverb.

Example: A block of 10 kg is resting on a plane inclined at an angle of 15 degrees

4.2 Semantic representation

From an architectural perspective, Stanford CoreNLP [1] poses to be simple, concrete and straightforward annotation based NLP pipeline, developed as a Java API. Unlike other text processing toolkits like GATE [1], the user does not have to necessarily learn the technology in order to use it. This is an open source API, robust and with rich documentation is widely used in many NLP approaches. Dependency is to be resolved, which means associating the nouns to different adjectives, prepositions and verbs. Dependency graph is thus obtained using PennTreebankLanguagePack [1] for English language. After tagging and tokenizing a sentence in lexical and semantic analysis, universal dependencies are extracted using LexicalizedParser and TreeBankLanguage to obtain the corresponding dependencies in relational format with main verb as the root of the sentence.

Example:



DT NN IN CD NN VBZ VBG IN DT JJ NN IN DT NN CD NNS
A block of 10 kg is resting on an inclined plane at an angle 15 degrees

Fig 2 : Tokenization of sentences- lexical analysis

4.3 Information extraction

Typed dependencies are later used to extract information as how different nouns and verbs are related to each other. Identification of to the user if it cannot relate any physical quantity to render-able objects and extraction of information related to each object, like mass, force, direction, plane/inclined, or separated by conjunctions, etc. is done in this stage by designing an algorithm with associates the relations from the Typed Dependencies [1].

- Each object/mass is defined by its physical quantity
- Meanings of adjective and prepositions are considered as constraints
- Accessing the numerical constraints among parameters and calculate values using inferences.

The system prompts error any noun, indicating error in typing or grammar. Thus, offers real time interactivity and extensibility for further development.

Example:

Typed Dependencies:

det(block-2, A-1)
nsubjpass(resting-7, block-2)
case(kg-5, of-3)
nummod(kg-5, 10-4)
nmod:of(block-2, kg-5)
auxpass(resting-7, is-6)
root(ROOT-0, resting-7)
case(plane-11, on-8)
det(plane-11, an-9)
amod(plane-11, inclined-10)
nmod:on(resting-7, plane-11)
case(degrees-16, at-12)
det(degrees-16, an-13)
amod(degrees-16, angle-14)
nummod(degrees-16, 15-15)
nmod:at(plane-11, degrees-16)

4.4 2D Image rendering

JavaFX uses data extracted from the previous stage as an object containing all the physical quantities and their association with the render-able and displays the corresponding FBD and its equation on the screen.

Example:

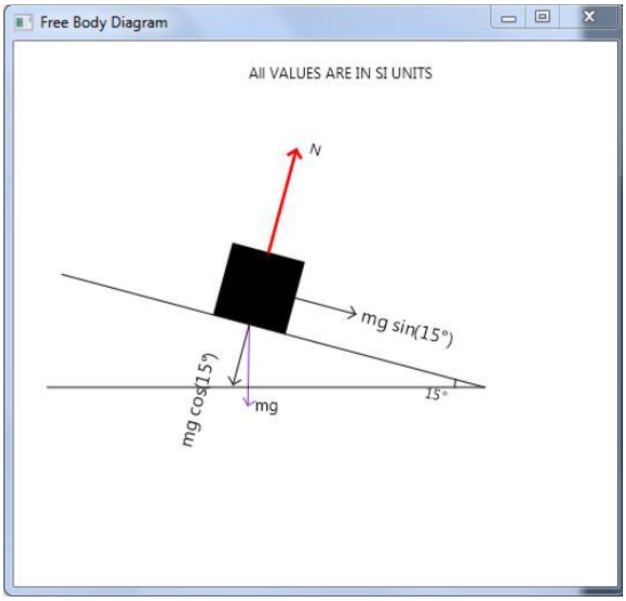
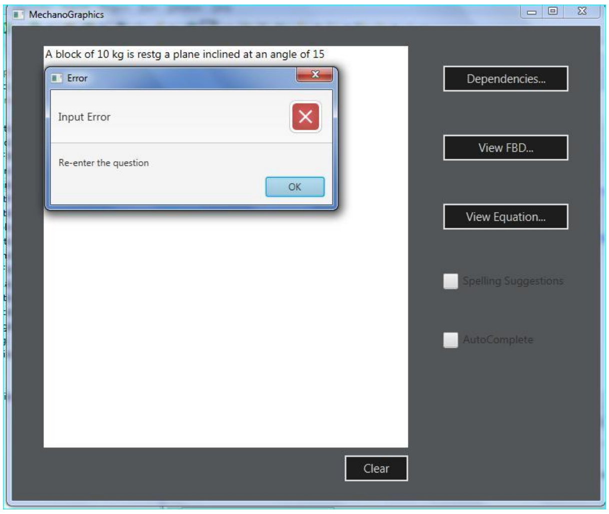


Fig 3: Free Body Diagram generated by the system for the mentioned example

4.5 Real time error handling

There may be times when the system might give imprecise or unexpected results caused by the semantic ambiguity in the given text. In order to get the correct results the user may be asked to enter the information again just by re-phrasing the sentence or providing adequate information required to generate the FBD.

Example:



A block of 10 kg is restg a plane inclined at an angle of 15

Input Error

Re-enter the question

Fig 4: Error message thrown by the system for incorrect input type

Also, when processing the problems based on inclined environment, the system might not be able to map the associative dependencies. Example: It might confuse in between - inclination of force or inclination of the surface. In such cases the system, concedes to not be able to process the text further. Thus, allowing the user to adjust the description without changing the former input. The system then chooses the last description while reconstructing the output.

5. ANALYSIS AND EVALUATION

The system is mainly designed for the non-professionals to use and be able to get the correct FBD for any question. Also, the learn-ability of the system to extend itself further by adding few more categories of Physics problem can be the basis of evaluation in terms of usability and ease.

The system does not maintain any database to identify the objects to be rendered on the screen. Also, on the other hand, system tries to be interactive with the user by providing assistance at possible stages - while entering the input and intermediary stage when it encounters non-matching input.

System converts the physical quantities to their corresponding SI unit, creates the respective FBD and gives its equation. The system also gives a summarized output detailing about all the information entered and thus extracted by the system for cross-checking.

6. EXPERIMENTAL SETUP

System, developed on Windows Platform, using Eclipse IDE and Stanford CoreNLP and JavaFX jar files. The user interface is designed to be sleekly simple. The rendering of the objects on the screen is evaluated for accuracy and efficiency, based on the time it takes to correctly render the diagram on the screen and correct output i.e. resolving of the forces correctly and proper mapping and conversion of Physical quantities in S.I. Units. The system was then tested on a corpus of 25 grammatically correct, physics problem belonging to the category of resting and free-falling type.

The corresponding rendered output for any particular test case is assessed and classified into three sections:

- **Completely correct:** Rendering is correct with proper associations.
- **Partially correct:** Object rendered, but the values are incorrectly mapped.
- **Incorrect:** Throws error or incorrect rendering

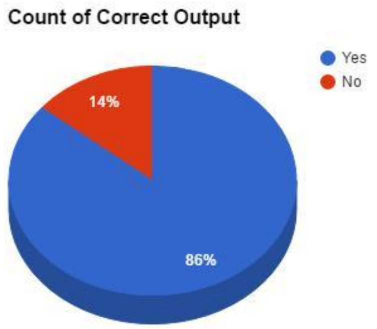


Fig 5: Chart showing the accuracy of Mechanographics for 25 test cases

7. CONCLUSION AND FUTURE SCOPE

This paper talks about Mechanographics, a fully implemented system, which successfully solves the time consuming task of drawing free body diagrams for most physics numerical. Also, integrating simple NLP with graphics can be flexible and easy way in providing visualization of any text, which acts as the primary source with giving the user a chance to modify its question real time thereby simplifying error handling quickly. Mechanographics provides a simplest tool for the non-professionals in using the system and to quickly get familiar with it. At the processing end, Mechanographics is able to productively process approx some 86% of different variations of sentences belonging to the same category.

Scenes generated from our system are promising. However we see our system as an initial framework which focuses on the depiction of natural language descriptions in real-time. In our current work we have implemented Resting, free-falling and friction as three categories of physics problem. Accomplished for a small domain, this can be further developed to cover a vast variety of subjects in the educational domain like, tension and truss in physics or linear and co-ordinate geometry. Future work will concern improving the language component and expanding the descriptions for more complex tasks. We hope this work serves as a foundation to much more encompassing and automated systems.

8. ACKNOWLEDGEMENT

We would like to thank HOD, faculty and the staff of Computer Department of K. J. Somaiya College of Engineering to support us throughout the project. Also thank Prof. Deepak Sharma for his guidance right from the beginning of the project, all the creative improvements and healthy discussions.

9. REFERENCES

[1] Manning, Christopher D., et al. "The stanford corenlp natural language processing toolkit." *ACL (System Demonstrations)*. 2014.

[2] Nayak, Aditya, et al. "Visualization of Mechanics Problems based on Natural Language Processing." *International Journal of Computer Applications* 116.14 (2015).

[3] Coyne, Bob, and Richard Sproat. "WordsEye: an automatic text-to-scene conversion system." *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001.

[4] Dupuy, Sylvain, et al. "Generating a 3D simulation of a car accident from a written description in natural language: The Carsim system." *Proceedings of the workshop on Temporal and spatial information processing-Volume 13*. Association for Computational Linguistics, 2001.

[5] Oommen, B. John, and Ghada Badr. "Dictionary-based syntactic pattern recognition using tries." *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer Berlin Heidelberg, 2004.

[6] Zeng, Xin, Qasim Mehdi, and Norman Gough. "3D scene creation using story-based descriptions." (2005).

[7] Srihari, Rohini K., and Debra T. Burhans. "Visual semantics: Extracting visual information from text accompanying pictures." *AAAI*. 1994.

[8] "Javafx 8". *Docs.oracle.com*. N.p., 2017. Web. 15 Mar. 2017. Weaver, James L et al. *Pro Javafx 2*. 1st ed. Berkeley, CA: Apress, 2012. Print.