

Text2Simulate: A Scientific Knowledge Visualization Technique for Generating Visual Simulations from Textual Knowledge

Ifeoluwatayo A. Ige¹, Bolanle F. Oladejo²

Computer Science Department, Rochester Institute of Technology, Rochester, USA¹
Computer Science Department, University of Ibadan, Ibadan, Nigeria^{1,2}

Abstract—Recent research has developed knowledge visualization techniques for generating interactive visualizations from textual knowledge. However, when applied, these techniques do not generate precise semantic visual representations, which is imperative for domains that require an accurate visual representation of spatial attributes and relationships between objects of discourse in explicit knowledge. Therefore, this work presents a Text-to-Simulation Knowledge Visualization (TSKV) technique for generating visual simulations from domain knowledge by developing a rule-based classifier to improve natural language processing, and a Spatial Ordering (SO) algorithm to solve the identified challenge. A system architecture was developed to structurally model the components of the TSKV technique and implemented using a Knowledge Visualization application called ‘Text2Simulate’. A quantitative evaluation of the application was carried out to test for accuracy using modified existing information visualization evaluation criteria. Object Inclusion (OI), Object-Attribute Visibility (OAV), Relative Positioning (RP), and Exact Visual Representation (EVR) criteria were modified to include Object’s Motion (OM) metric for quantitative evaluation of generated visual simulations. Evaluation for accuracy on generated simulation results were 90.1, 84.0, 90.1, 90.0, and 96.0% for OI, OAV, OM, RP, and EVR criteria respectively. User evaluation was conducted to measure system effectiveness and user satisfaction which showed that all the participants were satisfied well above average. These generated results showed an improved semantic quality of visualized knowledge due to the improved classification of spatial attributes and relationships from textual knowledge. This technique could be adopted during the development of electronic learning applications for improved understanding and desirable actions.

Keywords—Knowledge visualization; visual simulation; text-to-simulation knowledge visualization technique; natural language processing; electronic learning

I. INTRODUCTION

Knowledge visualization is the application of visualization techniques to disseminate knowledge among individuals [1], [2]. The main purpose of knowledge visualization is to improve the communication of knowledge through visual means. Although most available knowledge sources are in either numeric or textual formats, it is highly necessary to provide a visual representation of such knowledge for easier assimilation and retention in human minds and for fostering required action [3]. Visual knowledge could be represented

with charts, maps, images, scenes, simulations, and videos. In this article, a textual knowledge visualization technique that visually simulates textual knowledge was developed. A visual simulation is a visual representation containing visual objects that dynamically move based on predefined spatial attributes or collisions. Although visual metaphors, scenes, and animations could be used for the visualization of textual knowledge, these techniques do not emphasize precision in the visual representation of spatial attributes. Such techniques may not be appropriate for related domains that require accuracy in visual object positioning and movement. Furthermore, expressing textual knowledge in visualizations is of utmost importance for effective assimilation to be achieved [4]. This paper, therefore, presents a Text-to-Simulation Knowledge Visualization (TSKV) technique for generating visual simulations from domain knowledge. A review of related works is described in the next section of this article which is followed by an implementation of the TSKV technique. A quantitative evaluation is finally presented to test for accuracy using modified existing information visualization evaluation criteria.

II. REVIEW OF RELATED WORK

Knowledge visualization, according to [5] and [6] can be summarized into four major formats. Sketches, diagrams, images, and interactive visualizations. Fig. 1 show a word cloud of knowledge visualization techniques. A sketch is a visual representation that shows an abstract drawing or prototype of an idea or concept. Diagrams consist of boxes and circles that represent concepts and entities; and lines, and edges that depict the relationship between the entities. Diagrams are used to illustrate the classification and clustering of related concepts in a domain. Examples are Knowledge Graphs used for visualizing search histories [7]; concept maps used to illustrate concepts and relationships between them in specific domains such as medicine [8]; teaching and learning among others [9]. An image (visual metaphor) is a pictorial representation of humans and events. It is usually generated through rendering, photography, or paintings. Images are used to express emotions and give an idea about a concept such as comic images, icons, and emojis for storytelling [10]. Interactive visualization involves visually representing knowledge with animated objects and shapes. It allows users to interact and make decisions while viewing the visualization. Visualizations are usually shown in an ordered sequence of images. Examples are models such as archetypes [11];

semantic visualization tools [12]; virtual reality [13]; simulations, and animation [14].



Fig. 1. Knowledge visualization techniques

Existing literature research has shown that explicit knowledge could be visually represented with knowledge graphs, knowledge maps, and concept maps. Hao [15] used knowledge graphs to develop a surveying and remote-sensing application. Li [16] employed a concept map to develop adaptive learning systems. Visual metaphors replace key terms and concepts found in textual documents which visual characters. Huron [17] used data feeds to represent sediments. Chau [18] also used flowers to represent search results from the web. Hiniker [19] applied visual metaphors for clustering and viewing citations of literature from a large database. Keith [20] used visual metaphors to develop narrative maps for visualizing online narratives.

More recent KV techniques such as text-to-scenes, text-to-videos, and text-to-simulations generate more realistic visual presentations in form of images [21]; animations [22]; scenes [23], [24]; videos [25] and visual simulations [14]. A brief discussion of the major techniques related to our work is presented thus:

A. Text-to-Scene

Text-to-scene conversion generates static scenes from natural language text. Several authors have worked on this research area by applying different AI approaches. The first conversion system was WordEye [26]. Other authors applied machine learning techniques for text-to-scene conversion [27], [28]. Deep learning techniques such as Variational Auto Encoders (VAE) and Generative Adversarial Networks (GAN) are recently being applied for text-to-scene conversion [29], [30]. Our technique does not produce static scenes but dynamic simulations showing object-to-object collision behavior and movement.

B. Text-to-Simulation

Visual simulations are generated from textual inputs. Very limited research in this area exists in [14], [31], [32], and [33]. VoxSim architecture was presented in their work. They

applied a rule-based approach specifically predicate logic to set rules for each motion verb which was applied during conversion. However, precise geometric information of objects and their relative positions with other objects were not specified. Also, behavioral attributes of objects such as motion were not specified. The above issues were reported to have led to ambiguities in some simulation results. In our work, precise spatial grounding information of visual objects was included. Exact geometric information and their behavior attributes for all visual objects was also defined.

C. Text-to-Video

This involves converting natural language text to Videos that semantically depict the textual input. Recent research on text-to-video conversion exist [25], [34], [35]. A.I. techniques have been applied. Specifically, machine learning [36]; deep learning and neural networks such as convolutional neural networks [34]; recurrent neural networks [37], long short-term memory networks [38], and generative adversarial networks [25], [35] Although a rule-based approach was adopted, there is a slight resemblance between our work and most text-to-video conversion systems in that our prototype tool coupled with TTS and an existing screen capturing tool can generate a video of all simulations performed.

D. Natural Language Processing (NLP)

This involves NLP which involves several tasks such as sentence segmentation, tokenization, Part-of-Speech (POS), Named Entity Recognition Dependency parsing among others [39]. SpaCy [40], Natural Language Processing Tool Kit [41] and Stanford CoreNLP tool [42] are some NLP tools that perform NLP tasks. Although these tools perform these tasks, classification and extraction of domain-related keywords and attributes remain a research challenge due to the semantic interpretation of such words. A rule-based classifier was developed to classify and extract domain-related keywords, keyword attributes, and relations between keywords.

E. Spatial Relations and Arrangement

This involves defining the specific mapping of spatial keywords to predefined spatial information. Chang [28] and Ma [24] developed spatial relations for prepositions. Fisher [43] presented an arrangement model for determining the order of object placement and position. Our work also defines specific relation information for spatial keywords and prepositions by developing an algorithm for spatial positioning and rendering of objects.

F. CAD Models and Scenes

Computer-Aided Design is employed for creating models which are visual replicas of real-world objects. Most KV applications make use of existing CAD model datasets such as ShapeNet [23]; ScanNet [44] and Scene datasets such as SceneNet [45] for developing visual representations of textual knowledge. This work did not make use of existing datasets but developed new models. This is due to the scarce availability of models in the selected domain used for application validation.

III. METHODOLOGY

We present a mathematical model for generating visual simulations from textual knowledge and a system architecture in the subsequent subsections.

A. Mathematical Model Formulation

We mathematically model the task of generating visual simulations from textual knowledge. This is carried out using linear functions given a set of textual inputs $I_f (i = 1, 2, \dots, h, n)$. The following definitions show each milestone required from textual input to visual simulation output.

Definition 1: Suppose there exist a Semantic Classifier C ;

$$C(I_f) = [\{E\}, \{A\}, \{e_f R e_h\}] \quad (1)$$

Where the set of entities is $E_f (i = 1, 2, \dots, h, n)$, A is the attribute(s) set; R is the relationship between entities, e_f and e_h are entities.

Definition 2: Suppose there exist a model repository O and entity set E ;

$$o_f = f(e_f) \quad (2)$$

Where o and e are some models and entities in O and E and all entities in E have a one-to-one function in O

Definition 3: Suppose there are a finite set of shapes $O_f (i = 1, 2, \dots, h, n)$ with scaling factor s_f ; centroid coordinate $p(b_f, c_f)$, spatial attributes A_f and relation r . Let m be some numeric value; α be the length of o_f and $\alpha > A_i$. Let there exist Scaling and relative positioning Transforms S and R respectively. Let there exist a rendering Engine K . The visual simulation is given by V .

1) For single object placement o_f ;

$$f(o_f) = S + p(b_f, c_f) \quad (3)$$

$$V = K(f(o_f)) \quad (4)$$

2) For two relatively positioned objects o_f and o_h ;

$$f(o_f) = S_i + p(b_f, c_f)$$

$$f(o_h) = S_h + p(b_h, c_h)$$

$$V = K(f(o_f \cap o_h)) \quad (6)$$

Where:

$$p(b_f, c_f) = p[(b_h), (m + c_h)] \quad \text{and}$$

$$f(o_f \cap o_h) = S_f + p(b_f, c_f) + S_h + p(b_h, c_h) + R(r)$$

3) For two relatively positioned objects o_f and o_h with spatial attribute A_i ;

$$f(o_f) = S_f + p(b_f, c_f); f(o_h) = S_h + p(b_h, c_h)$$

$$V = K(f(o_f \cap o_h)) \quad (7)$$

Where:

$$p(b_f, c_f) = p[(b_h - (\alpha/2 - A)), (m + c_h)] \quad \text{and}$$

$$f(o_f \cap o_h) = S_f + p(b_f, c_f) + S_h + p(b_h, c_h) + R(r)$$

Given textual input, the semantic classifier extracts all entities, attributes, and relationships between entities from textual input as described in Eq. (1). Next, a set of entities are extracted from the model repository (Eq. (2)). Each model has a unique centroid coordinate $p(b_h, c_h)$ and a scaling vector. Eq. (3) and (4) describe how single models are rendered while taking into consideration its centroid coordinate and its scaling vector. To render two objects relative to each other (Eq. (6)) the centroid coordinate of an object is recalculated using the centroid coordinate of its relative object b_h and some numeric value which could be the height or thickness of the relative object added to c_h to get the c_h . For object relative positioned to each other with spatial attributes (Equation 7), the b_f value of an object is recalculated by; first subtracting the value of the spatial attribute from the mid-length value of its relative object. This result is further subtracted from the b_h value of its relative object. Some numeric value which could be the height or thickness of the relative object added to c_h to get the c_f .

B. System Architecture

Fig. 2 shows the Text-to-Simulation Knowledge Visualization architecture. It comprises four major modules: Natural Language Processing module, 2D Graphic Models Knowledge Base, Spatial Ordering module, and the user interface module.

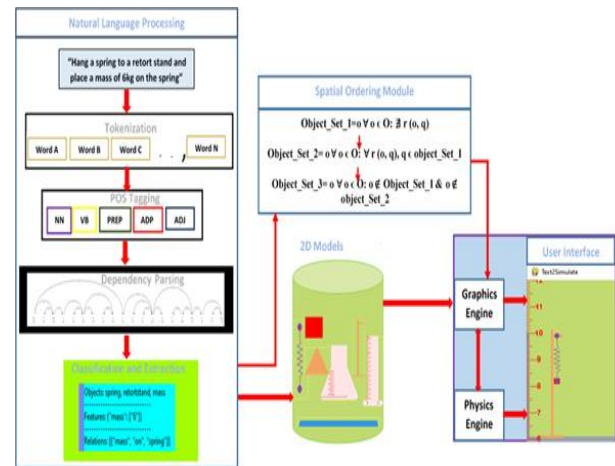


Fig. 2. Text-to-simulation knowledge visualization (TSKV) architecture

The Natural Language Processing module performs natural language processing tasks which are tokenization (breaking down the user's input sentence into words, numbers, punctuation marks, full stops, and discrete items); Part of Speech Tagging (allocating a POS tag per word); Dependency Parsing (assigning dependency labels to show relationship patterns between object and subject tokens); Classification and Extraction of domain-specific words, attributes, and relationships. 2D models can be found in the image repository. The Spatial Ordering Module determines the order of model rendering. The Graphic and Physics Engines are built into the application developed to cater for collision and interaction among models and also for rendering.

1) Natural language processing and dependency parsing:

We make use of SpaCy toolkit for natural language processing and dependency parsing. It is an existing industrial natural language processing library written in Python programming language [40]. SpaCy is known for a higher level of speed and accuracy in major NLP tasks such as POS tagging [46] and dependency parsing [47] when compared to other NLP applications. Tokenization of text, POS tagging, and parsing sub-libraries are used in our work. The NLP pipeline selected for this research supports English Language since it is the commonest medium of communication where this architecture was implemented and evaluated.

Each word in the user’s input is passed through a text corpus for relevance. If the sentence is not domain-related, the user is prompted to input domain-related sentences. Tokenization is done by converting each word in the sentence to tokens. Dependency parsing is done by assigning POS tags to each word, assigning dependency labels that show relationship patterns between object and subject tokens. Fig. 3 shows an example of a dependency graph given the sentence: *place a ruler*”

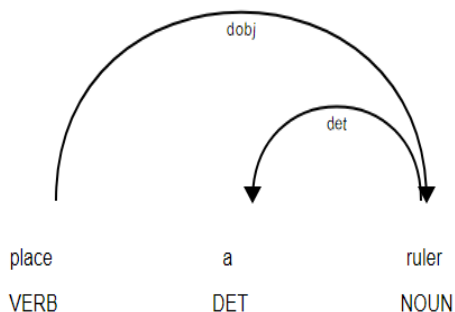


Fig. 3. Dependency graph for ‘place a ruler’

2) 2D models: A repository was created to store 2D models. Major apparatus objects used for performing High school experiments in Mechanics (a subtopic in Physics) are modeled in 2D for this research. The intended and purposely selected users for evaluation of the implemented architecture informed the choice of 2-Dimensional modelling. 2D graphic models provide necessary visual knowledge without much distraction and complexity.

The apparatus image library consists of 14 classes as shown in Fig. 4.

Each 2D apparatus model was created using modified existing objects from the Pymunk library. An apparatus model comprises of one or more objects. Fig. 4 is a class diagram showing each apparatus class, the attributes, and the methods. It also shows the relationship among the apparatus classes. The classes are CircularWeight(), BlockWeight(), RetortStand(), SpiralSpring(), KnifeEdge() and MetreRule(), Table(), Cup(), InclinedPlane(), TestTube(), Beaker(), GraduatedCylinder() and ErlenmeyerFlask().

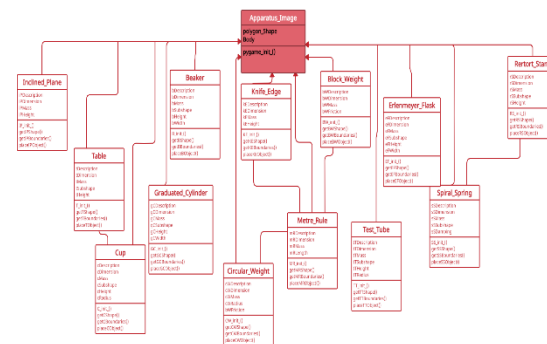


Fig. 4. Class diagram for apparatus model repository

Thirteen sub-classes inherit attributes ‘polygon_shape’ and ‘body’ from the main class ‘Apparatus_Image’ as shown in Fig. 4. Each object class contains the following methods:

- a) *_init_()*: This is a constructor which is called when a model is instantiated. It is used to initialize each class’s attributes.
- b) *getShape ()*: This method returns the model when called
- c) *getBoundaries ()*: This returns the dimensions of each model (left, right, top, bottom, centroidCordinate(x,y))
- d) *placeObject ()*: This places the model in space when called.

3) Rule-based classification and extraction algorithm: We present a novel rule-based classification algorithm that accepts a predefined objects-of-interest list and tokens (Fig. 5). Extracted noun tokens from the POS tagging must exist in the predefined list of objects of interest. The algorithm returns an object list, objects-attribute list, relations list, and object-rel-object list which shows the relationship between objects. In this work, the relationship between two objects is only considered.

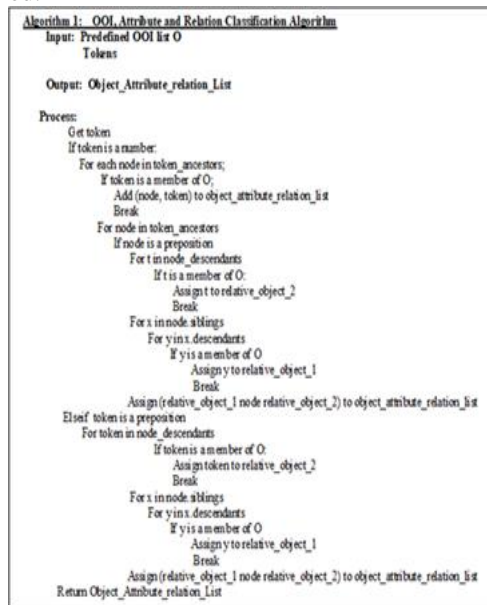


Fig. 5. Rule-based classification and extraction algorithm

4) *Attributes and relations grounding*: We consider spatial attributes such as the weight, size, and unique position of objects which are extracted during NLP. The object's color and other characteristics are not emphasized as much work has been done on this area in literature [24], [28]. The spatial attributes are very essential for precision in visual simulation output. Three predefined sets of relations: *on*, *under*, and *inside* are considered. The bounding box approach was employed to determine close distance and positions appropriately. The (x, y) axis coordinates, the height, and the width of the bounding box of the model were used to determine relative positions.

5) *Spatial Ordering*: Spatial ordering is the task of sequentially rendering models and determining relative positioning for related models. Given a set of object list O, object-attribute list P, relations list Q, and object-rel-object list R, a visual simulation is sequentially rendered based on the following procedure:

$$\forall o \in O, p \in P, q \in Q, r \in R;$$

Render Object_Set_1 (s);

Render Object_Set_2 (s)

Render Object_Set_3 (s)

Where:

a) Object_Set_1_list = o $\forall o \in O: \nexists r(o, q) // o$ is independent of q

b) Object_Set_2_List = o $\forall o \in O: \forall r(o, q), q \in$ Object_Set_1

c) Object_Set_3_List = o $\forall o \in O: o \notin$ Object_Set_1 & o \notin Object_Set_2.

The next section presents an implementation of the methodology.

IV. "TEXT2SIMULATE" KNOWLEDGE VISUALIZATION SYSTEM

We develop a knowledge visualization system called *Text2Simulate* based on the architecture shown in Fig. 2.

A. Software Requirement Gathering and Elicitation

Domain experts (three teachers) who taught physics in High schools located in a remote area were interviewed to retrieve system requirements since the targeted users of the application are high school physics subject tutors and senior high school students from remote areas. It is expected that the users have an elementary level of proficiency in English language. The users should also be familiar with basic concepts and terms used in high school physics. The *Text2Simulate* Knowledge visualization system can be used by students with little or no supervision of teachers.

B. Dataset

The dataset used for this application is the models in the apparatus image library and physics-related sentences in natural language text. The apparatus object image library contains apparatus object images. Existing objects in the Pymunk library are modified and used to create apparatuses models for Ruler, knife-edge and other apparatuses shown in Fig. 2.

C. Graphic User Interface

The user interface is divided into three major sections as shown in Fig. 6. The visual simulation is viewed in the right section. On the upper left, the textbox is used for accepting textual input from users. Apparatus models can also be viewed by clicking on objects from the Toolbar. Selected models are then viewed on the right viewing pane. When the button 'analyze' is clicked, the extracted objects, object-attribute list, and the object-rel-object list are shown on the lower-left pane. The user then clicks the 'simulate' button to generate visual simulations which are shown on the right pane.

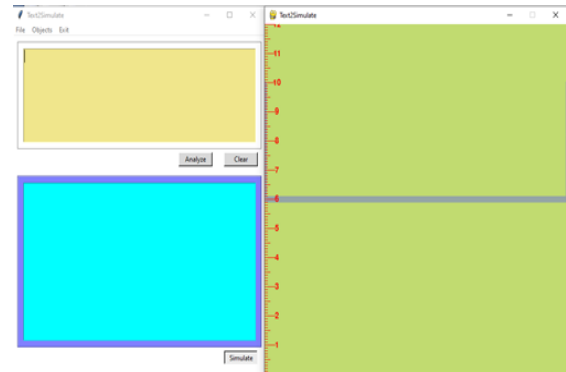


Fig. 6. Text2Simulate knowledge visualization system interface

D. Implementation Results

The models can be viewed by clicking each model as seen in Fig. 7. A table model is shown when clicked. Fig. 8 to 10 are screenshots of sentence inputs, classification results and visual simulation. The classification results show the objects, attributes and relations extracted from the sentence input. The figures show the screenshot of the visual simulation of the sentence. Fig. 8 shows a visual simulation result for 'Hang a spring to a retort stand and place a mass of 6kg on the spring'. During visual simulation, the motion can be viewed when the mass is attached to the spring. The spring continues to oscillate till it gets the equilibrium position. The screenshot was taken when the spring got to its equilibrium position.

Fig. 9 and 10 show visual simulation for sentence inputs describing principle of moments using a balanced ruler experiment [48].

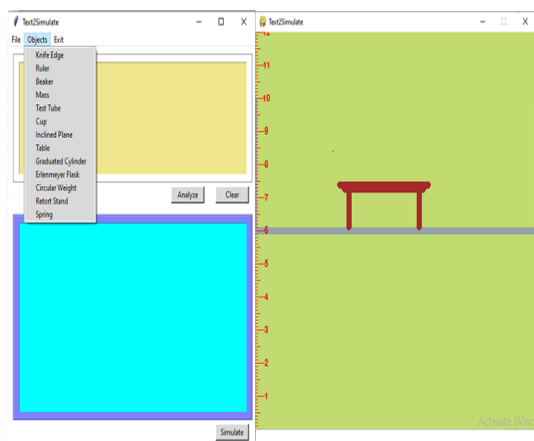


Fig. 7. A visual representation for 'Table' model

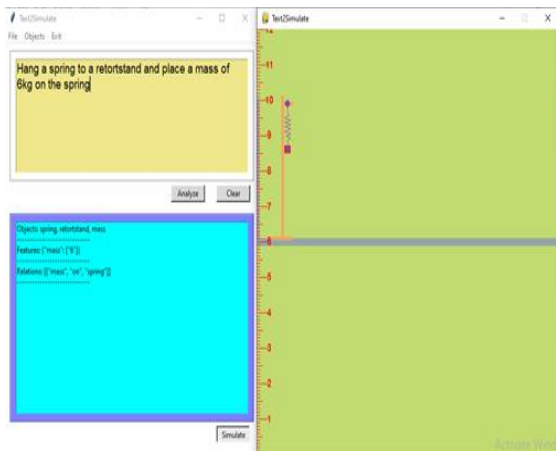


Fig. 8. Visual simulation screenshot of 'Hang a spring to a retort stand and place a mass of 6kg on the spring' sentence when spring is at equilibrium position

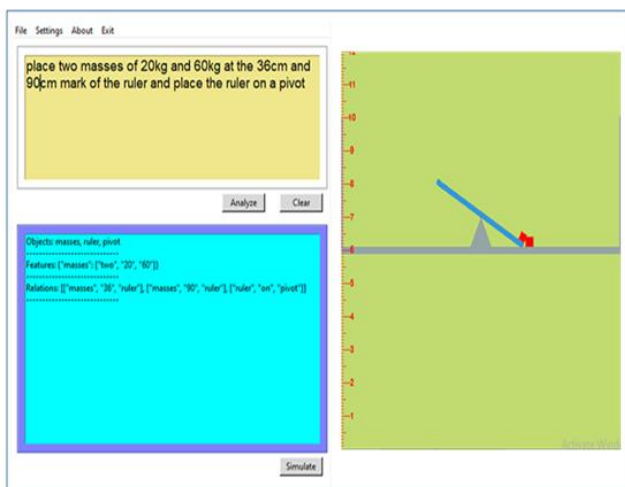


Fig. 9. Visual simulation screenshot of 'place two masses of 20kg and 60kg at the 36cm and 90cm mark of the ruler and place the ruler on a pivot'

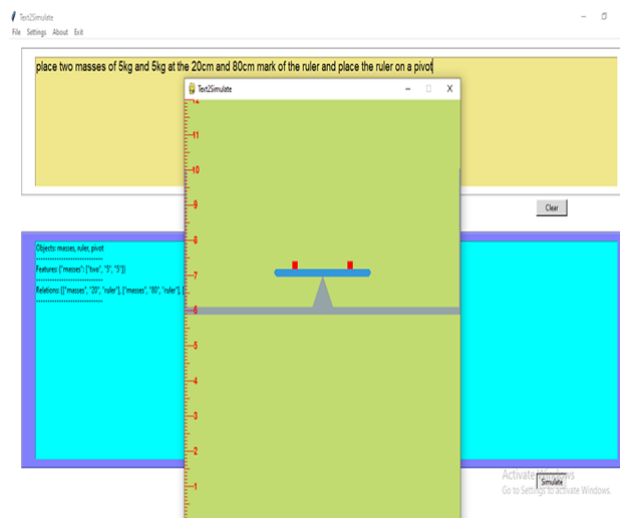


Fig. 10. Visual simulation screenshot of 'place two masses of 5kg and 5kg at the 20cm and 80cm mark of the ruler and place the ruler on a pivot'

V. QUANTITATIVE EVALUATION

We first perform a quantitative evaluation on the rule-based semantic classifier. Then, the accuracy of converting textual knowledge to visual simulation representation is evaluated. Finally, a user evaluation on the knowledge visualization tool is conducted.

A. Rule-Based Classification Evaluation

A total of 110 sentences were purposively selected and used for evaluation. A total of 60 sentences were extracted from the general domain; while the remaining 50 were domain-specific sentences. *Objects_of_Interest*, *Object_Attribute* and *Object_Relation_Object* classification results are compared with human-generated classification. Standard performance evaluation metrics (recall, accuracy, precision, and F1 score) based on the confusion matrix are employed. Each outcome is assigned True Positive (TP) if it correlates with human-generated classification; assigned False Positive (FP) if it is extracted as a member of a list but false with human-generated classification and assigned False Negative (FN) if human-generated classification assigns it to a list but is not included in the extracted list. Eftimov [49] and Popovski [50] reported that the True Negative metric is not required for the evaluation of rule-based entity classification methods. Hence, True Negative values are not reported. Table I presents the three classification categories; Objects of Interest list; Object-Attribute list and Object-Relation-object list.

TABLE I. THE CONFUSION MATRIX FOR EVALUATION OF OBJECTS, ATTRIBUTE AND RELATIONS CLASSIFICATION

	LIST	TP	FP	FN
1	Objects of Interest	242	0	11
2	Object-Attribute	32	0	4
3	Object-Relation-Object	52	0	4

Fig. 11 displays the graph showing recall, accuracy, and F1 score values for *Objects_of_Interest*, *Object_Attribute*, and *Object_Relation_Object* classification.

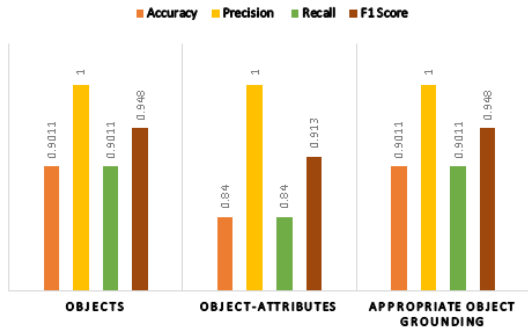


Fig. 11. A graph showing performance evaluation metrics for *Objects_of_Interest*, *Object_Attribute* and *Object_Relation_Object* classification

The accuracy for *Objects_of_Interest* classification is 0.9565 as presented in Fig. 11; precision is 1; recall is 0.9565 and F1 score of 0.9777. For *object_Attribute* classification and extraction, the accuracy is 0.8888; precision is 1; recall is 0.8888 and F1 score is 0.9411. Evaluation for the *Object_Relation_Object* classification produces; 0.9286 (recall); 1 (precision); 0.9286 (accuracy); and F1 score of 0.9630. The precision value of 1 for all the classifications is achieved because objects belonging to the list from the classifier actually belong according to human judgment. It can also be attributed to the advantage of employing a rule-based approach for the classifier modeling as reported in Al-Moslmi [51]. Accuracy and recall have the same value for all the categories. This shows that the classifier can correctly classify and extract all three categories to separate lists. This is also because True Negative outcome is not computed for this evaluation. Summarily, it can be concluded that evaluation results show that the classifier performed well above average. During error analysis of semantic classifier results, the dependency parser did not identify a few relations existing between objects as a preposition. They were identified as modifiers. However, human judgment categorized them as prepositions. Some attributes were also not identified since they did not have numeric values. The dependency parser did not identify a few compound words as nouns (objects). The words were split into modifier and noun ('inclined plane). However, human classification categorized them as Nouns.

B. Evaluation of Knowledge Visualization System based on Visual Simulation Results

Existing evaluation criteria used for evaluating the visual representation of information visualization techniques in [52] and [53] were adopted and modified to include Object's Motion criteria. Two domain experts were selected to evaluate *Text2Simulate* Knowledge Visualization application based on visual representation results. The simulation generated from the *Text2Simulate* application were evaluated using 50 purposively selected domain-specific sentences and compared with human judgment based on the following modified criteria: Objects Inclusion (OI); Object-Attributes Visibility (OAV); Object's Motion (OM); Relative Positioning (RP) and

Exact Visual Representation of Text (EVR). Each criterion was evaluated using performance metrics based on a confusion matrix. Each result is Assigned True Positive (TP) if the visual simulation matches with human judgment; Assigned False Positive (FP) if the visual simulation is not semantically correct based on human judgment and Assigned False Negative (FN) if there is no visual simulation (object is static) but there should be simulation-based on human judgment. The confusion matrix for visual simulation tasks is presented in Table II.

TABLE II. CONFUSION MATRIX OUTCOMES FOR VISUAL SIMULATION

	CRITERIA	TP	FP	FN
1	Objects Inclusion (OI)	82	0	9
2	Object-Attributes Visibility (OAV)	21	0	4
3	Object's Motion (OM)	82	0	9
4	Relative Positioning (RP)	27	0	3
5	Exact Visual Representation of Text (EVR)	48	1	1

Fig. 12 shows the evaluation results for precision, accuracy, recall, and F1 score metrics of Objects Inclusion, Object-Attributes, Object's Motion, and Relative Positioning.

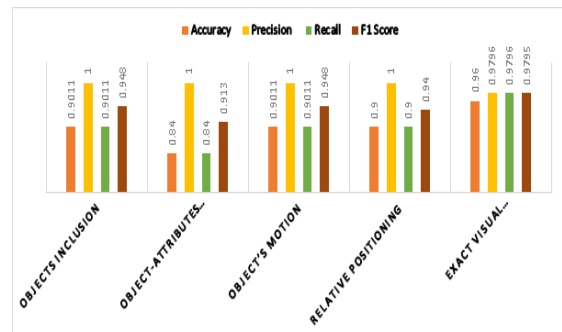


Fig. 12. A graph showing performance evaluation metrics for visual simulation

The visual presentation results showed almost a hundred percentage positivity for the knowledge presented as shown in Fig. 12. Spatial attributes were significantly visualized as well as the motion of objects which was dependent on their unique attributes, sizes, and intersection with other relative objects. During error analysis, a statement such as 'place two masses of 5kg and 20kg on a table' produced a visual representation showing only one mass and a table. Although the second mass was also rendered, it was hidden as both masses were placed at the same default location. Also, it was observed that some objects were not rendered even though the classifier identified them as objects as their models were not found in the image library.

C. User Study Evaluation

A total of 10 physics teacher participants and 20 student participants who have little or no prior graphic knowledge and reside in developing areas were purposively selected for the user study. Both set of participants were trained on how to simulate physics experiments on the knowledge visualization tool. They were then asked to perform two experiments

namely: Principle of Moments Using a Balanced Ruler and Hooke's Law as shown in Fig. 8, 9 and 10. The participants were finally asked to fill a user feedback questionnaire which was based on a two-point Likert scale (Yes/No) after the performing both experiments. The questions in the survey was drafted to indicate the level of system effectiveness and their overall satisfaction with the application developed based on the following 10 purposively selected metrics: Graphic design (T1), User-Friendliness (T2), Meaningful Arrangement (T3), Meaningful Size (T4), Object-Attribute visible (T5), Semantic Correlation of Text and Simulation (T6), Ability to Understand (T7), Ease of Use (T8), Reading Robustness (T9), Reusability (T10). *Text2Simulate* knowledge Visualization application was chosen as the independent variable while T1 to T10 were the dependent variables. Results of the analysis of mean is in Fig. 13.



Fig. 13. Graph of mean for teachers and students evaluation of *Text2Simulate* based on T1-T10 metrics

Fig. 13 showed that the teachers were fully satisfied as T2, T3, T4, T5, T6, T8, T9, T10, and T11 metrics have an average value of 1. The metrics are User-Friendliness, Meaningful Arrangement, Meaningful Size, Object-Attribute visible, Semantic Correlation of Text and Simulation, Ease of Use, Reading Robustness, Reusability, and Knowledge Sharing. This can be attributed to the essential need for *Text2Simulate* application as a teaching aid and for electronic learning due to the recent pandemic occurrence as stated by one of the teachers. The high performance on these metrics is also due to the current non-availability of the laboratory apparatuses in schools of the participants and their preference for *Text2Simulate* application over conducting the experiment for students in their laboratories if available. The mean scores for student's overall satisfaction based on the metrics are well above 0.5. It is shown that the scores range from 0.7 to 0.95. This reflects that most of the students were satisfied with the application.

VI. CONCLUSIONS AND FUTURE WORK

A Text-to-Simulation Knowledge Visualization (TSKV) technique for generating visual simulations from domain knowledge has been presented and implemented using a newly developed Knowledge Visualization application called '*Text2Simulate*'. The generated results have shown that precise semantic visual representations of spatial attributes and relationships between objects of discourse can be generated from natural language text using the above

technique. The developed rule-based semantic classifier can be used for domain-related classification of text which requires classification and extraction of objects, object properties, and the relationship between objects. The text-to-simulation technique for knowledge visualization produced a better visual representation of textual knowledge than existing knowledge visualization techniques due to its ability to visualize spatial object attributes retrieved from the text. This technique could be employed when developing electronic learning applications.

REFERENCES

- [1] R.A. Burkhard. Towards a Framework and a Model for Knowledge Visualization: Synergies between Information and Knowledge Visualization. *Knowledge and Information Visualization. S.-O. Tergan and T. Keller. Berlin, Springer Verlag, 2005: 226-243.*
- [2] X. Bai, L. Li and S. Zhang. Software for 3D model retrieval using local shape distributions. <http://code.google.com/p/shape-retrieval>. 2012.
- [3] J. Sleight, M. Schneider, J. Amann and E. Vayena. Visualizing an Ethics Framework: A Method to Create Interactive Knowledge Visualizations from Health Policy Documents. *J Med Internet Res* 2020: 22(1).
- [4] M.J.Eppler. What is an Effective Knowledge Visualization? Insights from a Review of Seminal Concepts. *2011 15th International Conference on Information Visualization, 349-354.*
- [5] M. Eppler and B. Burkhard. Knowledge Visualization. In *Schwartz, David G. (ed.): Encyclopedia of Knowledge Management. Hershey, PA: Idea Group Reference, 2005: 551-560.*
- [6] R. Meyer. Knowledge Visualization. *Trends in Information Visualization, 2010: 23.*
- [7] L. Xu, Z. T. Fernando, X. Zhou and W. Nejd. LogCanvas: Visualizing Search History Using Knowledge Graphs. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018.*
- [8] R. A. Burkhard. Learning from Architects: Complementary Concept Mapping Approaches. *Information Visualization. 2006; 5(3):225-234. doi:10.1057/palgrave.ivs.9500128*
- [9] K. Shatri and K. A. Buza. The Use of Visualization in Teaching and Learning Process for Developing Critical Thinking of Students. *Eur J Soc Sci Edu Res* 2017; 9: 71-74.
- [10] C. Tong, R. C. Roberts, R. Borgo, S. Walton, R. Laramée, K. Wegba, A. Lu, Y. Wang, H. Qu, Q. Luo, and X. Ma. Storytelling and Visualization: An Extended Survey. *Information. 2018; 9 (65).*
- [11] H.A.M. Sasieta, F. D. Beppler and R. C. Pacheco. A Model for Knowledge Visualization Based on Visual Archetypes. *Acta Scientiarum Tech, 2012; 34(4): 381-389.*
- [12] G. Molcho and R. Schneor. MIND - Semantic based Knowledge Visualization. In *Procedia CIRP* 2015; 36: 89-94.
- [13] Z. Qu, S. Hou, L. Zhu, J. Yan and S. Xu. The Study of Smart Grid Knowledge Visualization Key Technologies. *Telk Indo J Elec Eng* 2014; 12 (1): 323-333.
- [14] J. Pustejovsky and N. Krishnaswamy. Building Multimodal Simulations for Natural Language. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts. Association for Computational Linguistics, Spain, 2017.*
- [15] X. Hao, Z. Ji, X. Li, L. Yin, L. Liu, M. Sun, Q. Liu and R. Yang. Construction and Application of a Knowledge Graph. *Remote Sens* 2021; 13: 2511. <https://doi.org/10.3390/rs13132511>
- [16] Y. Li, Z. Shao, X. Wang, X. Zhao, and Y. A. Guo. Concept Map-Based Learning Paths Automatic Generation Algorithm for Adaptive Learning Systems. *IEEE Access* 2019; 7: 245-255.
- [17] S. Huron, R. Vuillemot and J. D. Fekete. Visual sedimentation. *IEEE Trans. Vis Comp Graphics* 2013; 19: 2446-2455
- [18] M. Chau. Visualizing web search results using glyphs: Design and evaluation of a flower metaphor. *ACM Trans Mgt Info Sys* 2011; 2(1):2.

- [19] A. Hiniker, S. Hong, K. Kim, N. Chen, J. D. West, and C. R. Aragon. Toward the operationalization of visual metaphor. *J Assoc Info Sci Tech*, 2017: 68.
- [20] B. Keith and T. Mitra. Narrative Maps: An Algorithmic Approach to Represent and Extract Information Narratives. 2020. *ArXiv*, abs/2009.04508.
- [21] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang and X. He. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017:1316-1324.
- [22] H. Sarma, R. Porzel, J.D., Smeddinck, R. Malaka and A. B. Samaddar. A Text to Animation System for Physical Exercises. *The Comp J*, 2018; 61: 589-1604.
- [23] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. In *Proceedings of CVPR*, 2015. *arXiv*: 1512.03012..
- [24] R. Ma, A. Patil, M. Fisher, M. Li, S. Pirk, B. Hua, S. Yeung, X. Tong, L. Guibas and H. Zhang. Language-driven synthesis of 3D scenes from scene databases. *ACM Trans Graph (TOG)* 2018; 37:1 – 16.
- [25] Y. Li, M. R. Min, D. Shen, D. E. Carlson and L. Carin. Video generation from text. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, February 2-7, 2017.
- [26] B. Coyne and R. Sproat. WordsEye: An Automatic Text-to-Scene Conversion System" In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, August 12, 2001, pp. 487-496.
- [27] C. L. Zitnick, D. Parikh and L. Vanderwende. Learning the Visual Interpretation of sentences. In *Proc. ICCV*, 2013, pp 1681–1688.
- [28] A. X. Chang, A. Savva and C. D. Manning. Learning Spatial Knowledge for Text to 3D Scene Generation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- [29] F. Tan, S. Feng, and V. Ordonez. Text2Scene: Generating Compositional Scenes from Textual Descriptions. *Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp 6703-6712.
- [30] F. Huq, N. Ahmed, A. Iqbal. Static and Animated 3D Scene Generation from Free-form Text Descriptions. *arXiv: Computer Vision and Pattern Recognition*, 2020.
- [31] J. Pustejovsky N. Krishnaswamy. Generating Simulations of Motion Events from Verbal Descriptions. *Proceedings of the Third Joint Conference on Lexical and Computational Semantics*, Dublin, Ireland, August 23-24, 2014, pp 99–109.
- [32] N. Krishnaswamy and J. Pustejovsky. VoxSim: A Visual Platform for Modeling Motion Language. *COLING*, 2016a
- [33] N. Krishnaswamy and J. Pustejovsky. Multimodal Semantic Simulations of Linguistically Underspecified Motion Events. *Spatial Cognition*, 2016b.
- [34] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem and A. Kembhavi. Imagine This! Scripts to Composition to Videos. In *Proceedings of 15th European Conference on Computer Vision (ECCV)*, Munich, Germany. 8-14 September, 2018, pp 610-626.
- [35] Y. Balaji, M. R. Min, B. Bai, R. Chellappa, R. and H. P. Graf. Conditional GAN with Discriminative Filter Generation for Text-to-Video Synthesis. *Proceedings of 28th International*, 2019.
- [36] J. Zhang, Y. Hao, L. Li, D. Sun and L. Yuan. StoryWizard: a framework for fast stylized story illustration. *The Visl Compr* 2012; 877-887.
- [37] K. Schwarz, P. Rojtborg, J. Caspar, I. Gurevych, M. Goesele and H. P. A. Lensch. Text-to-Video: Story Illustration from Online Photo Collections. In: *Setchi R., Jordanov I., Howlett R.J., Jain L.C. (eds) Knowledge-Based and Intelligent Information and Engineering Systems. KES 2010. Lecture Notes in Computer Science*, 6279. Springer, Berlin, Heidelberg, 2010.
- [38] T. Marwah, G. Mittal and V. N. Balasubramanian. Attentive semantic video generation using captions. *CoRR* abs/1708.05980, 2017.
- [39] D. Khurana, A. Koli, K. Khatter and S. Singh. Natural Language Processing: State of The Art, Current Trends and Challenges, *Computation and Language*, 2017. *arXiv*: 1708.05148.
- [40] M. Honnibal, I. Montani, S. V. Landeghem and A. Boyd. *SpaCy: Industrial-strength Natural Language Processing in Python*. Zenodo, 2020. doi:10.5281/zenodo.1212303.
- [41] S. Bird, L. Edward and K. Ewan. *Natural Language Processing with Python*. O'Reilly Media Inc. 2009.
- [42] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard and D. 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.
- [43] M. Fisher, D. Ritchie, M. Savva, T. A. Funkhouser and P. Hanrahan. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics (TOG)*. 2012; 31: 1 - 11.
- [44] A. Dai, A. X. Chang, M. Savva, M. Halber and T. A. Funkhouser. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. *CVPR*, 2017; 5828-5839.
- [45] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent and R. Cipolla. SceneNet: Understanding real-world indoor scenes with synthetic data. *arXiv preprint* 2015. *arXiv*:1511.07041.
- [46] J. D. Choi, J. Tetreault and A. Stent. It depends: Dependency parser comparison using a web-based evaluation tool, In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. 2015.
- [47] K. Ortman, A. Roussel and S. Dipper. Evaluating Off-the-Shelf NLP Tools for German. *KONVENS*. 2019.
- [48] G. N. Santos. Compiled Experiments for Modern Physics. Physics Department, De La Salle University, Manila, 2015:1-71.
- [49] T. Eftimov, B. K. Seljak and P. Korošec. A Rule-Based Named-Entity Recognition Method for Knowledge Extraction of Evidence-Based Dietary Recommendations. *PLoS ONE* 2017:12.
- [50] G. Popovski, S. Kochev, B. Korousic-Seljak and T. Eftimov. FoodIE: A Rule-based Named-entity Recognition Method for Food Information Extraction. *ICPRAM*. 2019.
- [51] T. Al-Moslmi, M. G. Ocaña, A. L. Opdahl, and C. Veres. Named Entity Extraction for Knowledge Graphs: A Literature Overview. *IEEE Access*, 8, 2020: 32862-32881.
- [52] C. M. Dal CM, S. Freitas, P. R. Luzzardi, R. A. Cava, M. A. Winckler, M. S. Pimenta and L. P. Nedel. Evaluating Usability of Information Visualization Techniques. In *Proceedings of Brazilian Symposium on Human Factors in Computing Systems*, 2012: 40-51.
- [53] S. Dahlgren. Design and Evaluation of a Visualization Method in a GIS for Complex Electronic Warfare Assessment with Regards to Usability. *A Master's thesis*. Department of Computer and Information Science Linköping University. 2019.