

User Interactivity for Text Visualization System

Shashak Samaiya^{1*}, Yastika Jain², Priyanka Jain³

¹Bansal Institute of Science and Technology, Bhopal, India

²Jaypee University of Engineering and Technology, Guna, India

³Centre for Development of Advanced Computing, Pune, India

*Corresponding Author: ssamaiya97@gmail.com, Tel: 9039573722

Available online at: www.ijcseonline.org

Accepted: 15/Oct/2018, Published: 31/Oct/2018

Abstract— This paper outlines the long-term vision to automate a language-less visualization system for communication with minimum language dependency. It proposes an interactive approach for Preksha, as Hindi text visualizer. Preksha is the process of understanding a natural language, transform the knowledge into a language independent abstract form and rendering a 3D Scene for visualization. Considering linguistic nature as morphologically rich and free-word-order for ‘Hindi’ language, the proposed work offers an approach for user interaction with system, in case of requirement.

Keywords—Human Computer Interface (HCI), text visualization, natural language understanding, natural language processing, 3D scene generation, VRML, computer graphics.

I. INTRODUCTION

The Automatic Text Visualization (ATV), i.e. scene generation based on natural language understanding, is a new interdisciplinary research area based on the work in natural language processing and computer graphics. Natural language processing aims to acquire and understand the natural languages used by humans. It extracts the knowledge hidden in natural language that is in text or speech form. Natural language visualization transforms a text/speech to visual representations. It uses knowledge from many disciplines including linguistics, computational linguistics, psycholinguistics, philosophy, and computational graphics. Representation of common-sense knowledge capture, context disambiguation, lexicons and ontology preparation, spatial relation resolutions and animation generation are the vital issues of natural language visualization.

ATV is one of the important applications in literacy development by understanding and learning language. It can help understanding and comprehension while reading a text in secondary languages and foreign languages. It may help in learning a language especially for rehabilitation of persons with delayed development. Preksha being an automatic text to scene visualizer needs to be highly interactive. User interaction is both natural and useful for text to scene generation since the process is fundamentally under-constrained. Interaction and user feedback are a good way to improve coverage and to refine spatial knowledge. The vastness and constraints of Indian languages make it very difficult for each and every thing to be stored in the software

repository. Hence, it has the provision for amendment of new attributes in case it encounters a text which has some unknown content. After presenting Introduction in Section I and Related work in Section II, this Paper is structured to present background of Preksha in Section III. Section IV presents the proposed approach for user interactivity and related implementation. The paper is concluded in Section V.

II. RELATED WORK

As far as we know, no web-based application provides automatic text visualization of Hindi or Indian languages. However, natural language understanding with syntactic and semantic aspects has been attempted since early 1970s [2] and [4]. It has incrementally stepped forward from picture to scene, animation and video. We have studied selected interesting applications in this section to explore the wide spectrum of applications of the text visualization that covers classification on input, output, utility, procedure and domain of work.

The work WordsEye automatic text-to-scene conversion [5] is the most talked about system which was developed in 2001. It works on parts-of-speech (POS) tagger and parser. It uses WorldNet semantic hierarchy for syntactic and semantic analysis on the input text and then generates dependency representation to produce a description of the arrangement of objects in a scene. Same year, an event recreation system CarSim [6] is developed, which is an automatic 3D scene generator of a car accident description. It determines the grammatical relationships between the words of a sentence

and creates a pre-defined XML template. Then it generates the scene using detected dynamic objects, static objects and accidents out of XML templates. After major work on natural language understanding and natural language processing in this area, an intelligent storytelling system named CONFUCIUS with multimedia support is presented in [19]. While incorporating media allocation, knowledge base and 3D animation generation, it uses three central features (multimodality, animation, and intelligence) to create 3D animation from a natural language text. It provides a story narrator for object modeling, human animation, collision detection and application of narrative montage in virtual environments. The newer work [3] presents the concept of providing general knowledge for background support in generated scene. [20], [21] provided A study on the relationships between the Virtual Reality with learning and Brain Computer Interface (BCI).

III. PREKSHA: A HINDI TEXT VISUALIZER

Preksha architecture comprises a consolidated framework with resource repositories, processing engines, computational tools, and with a mechanism of data and control flow. This framework supports tools for resource management for linguistic and scene repositories. Three engines a) natural language processing engine b) knowledge engine and c) scene engine carry out the computational processing. The Architecture of Preksha is shown in Figure 1.

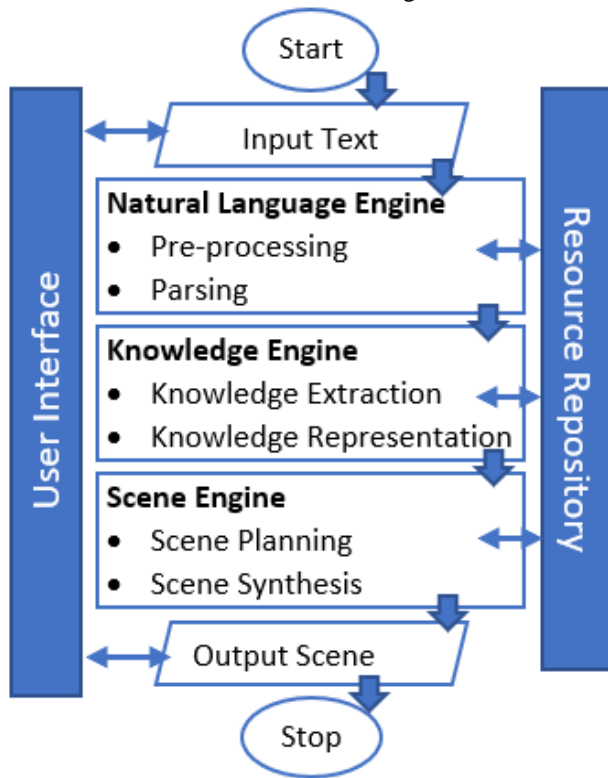


Figure 1. Architecture of Preksha Framework

The architecture is enriched by user-management tool with new user registration, login authentication, history, and feedback maintenance. The framework is interactive to communicate with user as per the requirements. The design of Preksha system is modular and it maintains a well balance for cohesion and coupling between the modules. The input of Preksha system is a Hindi language text through user interface. With the support repository, the processing carried out through three of the engines. The system provides output in form of 3D scene rendered in Preksha environment.

Considering a sample text “कमरे में गोल टेबल के साथ गमला है”, The Figure 2 and Figure 3 presented the Preksha output a) TAG parsed output and b) Final Scene engine output.

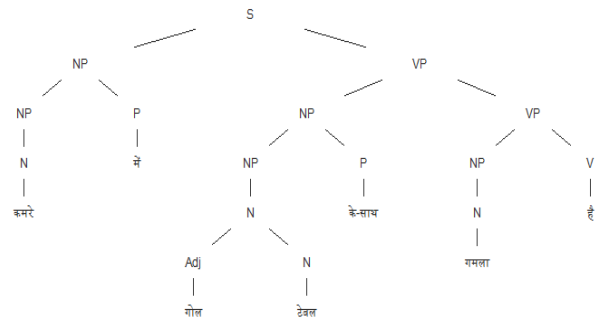


Figure 2. Parsed output for sample sentence “कमरे में गोल टेबल के साथ गमला है”



Figure 3. Scene generated for the sample sentence “कमरे में गोल टेबल के साथ गमला है”

The presented work in this paper is an extension of our previously reported works. The basic architecture for Hindi text visualization is presented in [17]. The work in [19] and [14] respectively provides details on spatial intelligence and knowledge acquisition portion for Hindi language. The ambiguity and challenges in working with pre-positions and postpositions are explained in [12] as the prepositions in English language are post-positions in Hindi and other Indian languages. The language parsing in Preksha is based on the TAG formalism [12]. Applying Earley algorithm [8], a new algorithm for Hindi TAG parsing is introduced in [13]. The

initial evaluation strategy for Preksha research is discussed in [15]. The Scene planning work using VRML file format (*.wrl) is presented in [16].

A reverse work, knowledge acquisition from an input image and generate language Hindi is explained in [15]. As mentioned in [12] and [13], the utility of this approach may be useful in cognitive support and language learning. Relevant details should be given including experimental design and the technique (s) used along with appropriate statistical methods used clearly along with the year of experimentation (field and laboratory).

IV. PREKSHA:INTERACTIVITY

The Preksha framework is consists of three major components namely, Preksha engines (Preksha-E), Preksha Repository (Preksha-R) and Preksha Interface (Preksha-T). This paper focuses on component Preksha-T for a specific reason to have fallback strategy. The Application provides a communication interface between user and the system. User can enter input text in Hindi language and system produces the 3D scene along with intermediate module outputs. This interface is enriched with multiple functionalities to utilize the various features of the system, like – Input planner, intermediate output, error log, tree view of parsed output and version log. It is also supported by the mode of where user can guide system to proceed further in a specific case. The Preksha-T is facilitated with login authentication functionality. User authentication supports in generating user log, history maintenance, report preparation, feedback provision and result analysis. Here, user can log-in or go to ‘new user’ registration by clicking appropriate button. A user can enter into system as a guest user without login and authentication process also. After user authentication, the user gets a processing window as shown in Figure 4.

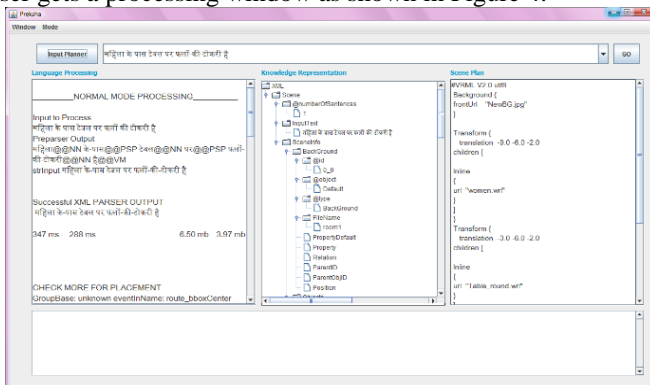


Figure 4. User interface

The user is provided with a text area for submitting input text. On clicking ‘GO’ button, Preksha fetches the input from text area and provides it for further operations. The interface provides a logical distribution of three intermediate components result in three panes. The outcome of language

processing engine is presented in left pane of interface, which is an input for knowledge processing engine. The output of knowledge processing engine is presented in middle pane, which is the input for last scene engine. The output of scene engine is presented in last pane. A bottom pane is provided in Preksha interface, this is an error log window. This window is responsible to show any errors or warnings aroused during the program execution.

System interacts with the user in case of any unseen data is present in the input text. System has a scope to have unknown or new words in input text that are not part of Preksha stored repository. ‘AddModel’ functionality is implemented where an unknown word/model can be added runtime into system. In advanced mode of operation, system prompts the user with an indication of unknown entity. Here, user can opt from a list of choices to proceed further. The options to proceed are shown in Figure 5 and discussed further.

- Continue by ignoring new words.
- Continue by taking default models.
- Continue by matching models with one of the existing models.
- Continue by uploading new model.
- Do not continue and exit.

A sample Input text- टेबल पर अमरुद रखा है {table par amarood rakha hai} [A guava is kept on the table] is used for explaining these options. Here, ‘अमरुद’ is a new entity, which is not in repository and prompts user to add model. System takes input from user and precedes according to his/her selection.

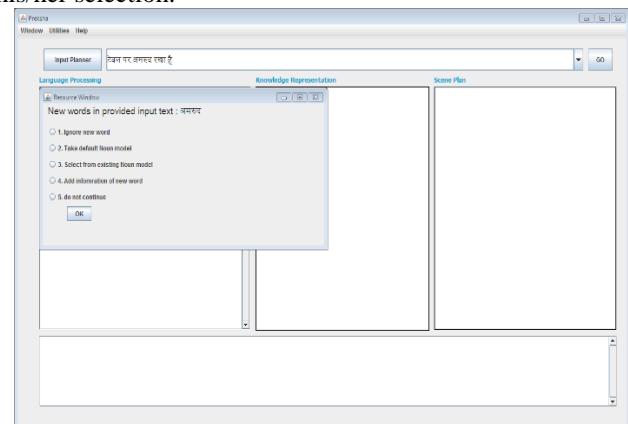


Figure 5. ‘AddModel’ Interface

Option 1: In case, user does not want to add information of new unknown word into processing, he/she may proceed by ignoring the word directly. In this case, the model corresponding to new word will be absent from final scene generation. Figure 6 explains this scenario. Figure 7 is the result representation is case shown in Figure 6.

- New entity: अमरुद{amarood}[guava]
- User selection: Option 1 - Continue by ignoring new words.
- System provides an environment with model of 'Table' only

Figure 6. Option 1 of AddModel functionality



Figure 7. Result from Option 1 of AddModel functionality

Option 2: By this option, user allows system to proceed with using a default model for corresponding new unknown word. At this time, an image of default model is presented so that its visualization can be mapped with word while perceiving the complete scene, as shown in Figure 8. This also gives an indication in scene that this default model is a representation of unknown entity. Figure 9 shows the result based on case discussed in Figure 8.

- New entity: अमरुद{amarood}[guava]
- User selection: Option 2 - Continue by taking default models.
- System provides an environment with model of 'Table' and a default object model placed on the Table.

Figure 8. Option 2 of AddModel functionality



Figure 9. Result from Option 2 of AddModel functionality

Option 3: This option facilitates user to select any existing model to be mapped with new model to be appeared. This case is explained in Figure 10.

- New entity: अमरुद{amarood}[guava]
- User selection: Option 3 - Continue by matching models with one of the existing model.
- Users will be provided with the list of available resources. Here, user may be able to select any similar resource from the available repository like 'सेब{seb}[apple]' or 'नाशपाती{nashpati}[pear]'.
- System provides an environment with model of 'Table' and a user selected object model placed on the Table.

Figure 10. Option 3 of AddModel functionality

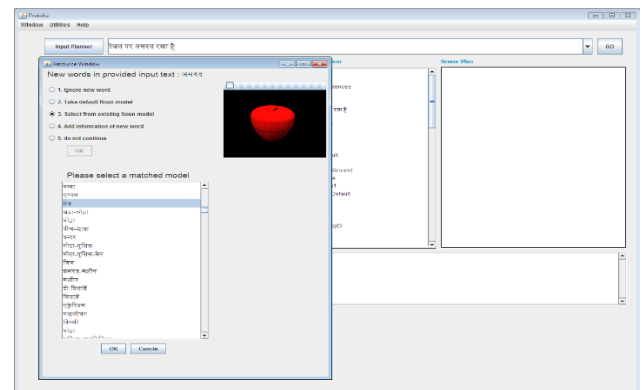


Figure 11. Object match from Option 3 of AddModel functionality

Figure 11 represents the interface for object selection and Figure 12 presents the final scene generated using option 3 of AddModel functionality.



Figure 12. Result from Option 3 of AddModel functionality

Option 4: This is most advanced option where user can update the model along with making a resource entry through an interface. This interface guides the user to feed

grammatical category and visual properties. It also gives the facility of file to be uploaded in WRL format as a VRML model. Figure 13 presents the interface to upload model and Figure 14 presents the final scene generated using option 4 of AddModel functionality.

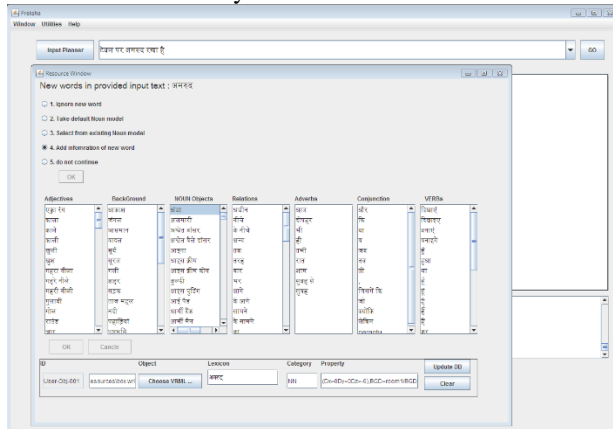


Figure 13. Information update for Option 4 of AddModel functionality

This model can be stored temporarily in Preksha repository and later on after proper verification; it can be included in main repository. In case of “add model” option selection, the uploaded model by user is able to be stored in system database and can be used in further processing also.

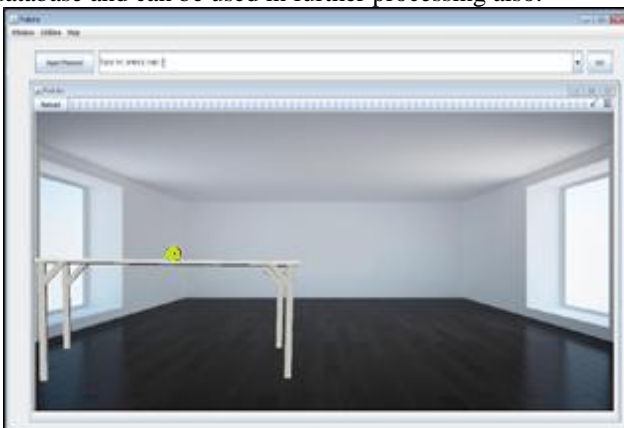


Figure 14 Result from Option 4 of AddModel functionality

Option 5: Using this option, user can terminate the process. This is the possible case where user has provided an erroneous input or he/she wants to change the input text. It brings the user back to Input interface for further more input. Preksha-T is implemented in swing (graphical user interface) GUI toolkit (Java Swing - Oracle Corporation, 2015) of Java using Netbeans (netbeans - Oracle Corporation, 2017) (Integrated Development Environment) IDE. MySQL (mysql - Oracle Corporation, 2017) database is considered in experiment for its scalability, security, reliability, and uptime.

V. CONCLUSION AND FUTURE SCOPE

Preksha is targeted to work with morphologically rich free order language Hindi for the purpose of automatic text visualization (ATV). ATV research requires language processing, knowledge processing, and scene processing. ATV is needed or is desirable in many applications such as summarization of news articles, storytelling, event/accident visualization, rehabilitation of people with cerebral injuries, rehabilitation of children with delayed development, gaming, security alerts and summarization of long text documents for rapid browsing.

Preksha system is enriched with communication interface for user to interact with the system. In this paper, we have proposed an approach to have functionality of interaction between a user and system. A possibility is explored to have various options which can occur for a better Human Computer Interaction (HCI). The proposed design is explained with implementation and results. The work is extendible with other HCI systems and technologies.

REFERENCES

- [1] Jain, P., Bhavsar, R., Kumar, A., Pawar, B. V., Darbari, H. and Bhavsar, V. C.: “Tree Adjoining Grammar based Parser for a Hindi text-to-scene conversion system” in 4th International Conference for Convergence in Technology (I2CT), 2018
- [2] Winograd, T.: “Understanding Natural Language”. pp. 1-191. New York: Academic Press. pp. 191. New York: Academic Press, also published in Cognitive Psychology, 3:1, 1970.
- [3] Chang, A. X., Savva, M. and Manning, C. D.: “Learning spatial knowledge for text to 3D scene generation” pp. 1-11. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014.
- [4] Clay, S.R., Wilhelms, J.: “Put: Language-based interactive manipulation of objects”. pp. 31-39. In IEEE Computer Graphics and Applications. 1996.
- [5] Coyne, B., Sproat, R.: “WordsEye: An automatic text-to-scene conversion system”. pp. 487-496. SIGGRAPH, Computer Graphics Proceedings. 2001.
- [6] Dupuy, S., Egges, A., Legendre, V., Nugues, P.: “Generating a 3d simulation of a car accident from a written description in natural language: The carsim system”. pp. 1-8. In Proceedings of ACL Workshop on Temporal and Spatial Information Processing. 2001.
- [7] Earley, J.: “An Efficient Context-Free Parsing Algorithm”. pp. 94-102. In Commun. ACM 13(2). 1970.
- [8] Hassani, K., and Lee, W. S.: “Visualizing Natural Language Descriptions: A Survey”, In ACM Computing Surveys (CSUR) Surveys Homepage archive, Article No. 17, Volume 49, Issue 1, July 2016.
- [9] Jain, P. and Pawar, P.: "From 'Pre-Position' to 'Post-position'", pp. 66-71. In International Journal of Modern Computer Science (IJMCS) ISSN: 2320-7868 (Online), Volume 4, Issue Oct, 2016.
- [10] Jain, P., Bhavsar, R. P., Lele, A. Kumar, A., Pawar, B. P., Darbari, H.: “Knowledge acquisition for automatic text visualization”. In National Conference on Advances in Computing (NCAC-2017). 2017
- [11] Jain, P., Darbari, H., and Bhavsar, V. C.: ‘Vishit: A Visualizer for Hindi Text’. pp. 886-890. In Fourth International Conference on

- Communication Systems and Network Technologies (CSNT), IEEE Xplore. 2014.
- [12] Jain, P., Darbari, H., and Bhavsar, V. C.: "Text Visualization as an Aid to Language Learning Disability", pp. 88. In ELELTECH 2013 National Conference on e-Learning and e-Learning Technologies, India. 2013.
- [13] Jain, P., Darbari, H., and Bhavsar, V. C.: "Cognitive support by Language Visualization: A case study with Hindi Language". pp. 110-115. In 2nd International Conference for Convergence in Technology (I2CT), IEEE Xplore. 2017.
- [14] Jain, P., Darbari, H., and Bhavsar, V. C.: "Spatial Intelligence from Hindi Language Text for Scene Generation". pp. 132-138. In 2nd International Conference for Convergence in Technology (I2CT), IEEE Xplore. 2017.
- [15] Jain, P., Pawar, P., Koriya, G., Lele, A., Kumar, A., Darbari, H.: "Knowledge acquisition for Language description from Scene understanding". In IEEE International Conference on Computer, Communication and Control (IC4-2015) Conference. IEEE explore. 2015.
- [16] Jain, P., Bhavsar, R. P., Pawar, B. V. and Darbari, H.: "VRML for automatic generation of 3D Scene". In International Journal of Computer Application (2250-1797) Issue 8 Volume 2, March-April 2018.
- [17] Joshi, A. K., Levy, L. S., and Takahashi, M.: "Tree Adjunct Grammars". In J. Comput. Syst. Sci. 10(1). 1975.
- [18] Jain, P., Bhavsar, R. P., Pawar, B. V. and Darbari, H.: "Empirical Evaluation for Hindi text-to-scene generation system" in International Journal of Creative Research Thought © 2018 IJCRT | ISSN: 2320-2882 | Volume 6, Issue 1 February 2018
- [19] Ma, M.: (2006), 'Automatic conversion of natural language to 3D animation'. Ph.D. thesis, University of Ulster, Derry, Ireland, pp. 1-250.
- [20] B. Senthil Kumar, Jaya Prakash D, "A Study on The Relationships Between The Virtual Reality & Learning", International Journal of Computer Sciences and Engineering, Vol.6, Issue.7, pp.966-969, 2018.
- [21] Y. Galphat, M. Gangwani, A. Bhave, B.S. Chadha, S. Adnani, "Integrating BCI with Virtual Reality", International Journal of Computer Sciences and Engineering, Vol.6, Issue.1, pp.129-131, 2018.

which covered aspects of IoT, machine learning and web services. yastika1308@gmail.com

Priyanka Jain is working as Joint Director, AAI Group, C-DAC, Pune. She has 17 years of experience in technologies related to Natural Language Processing (NLP), Machine assisted Translation (MT), Distributed Architectures, Cognitive Science and Computer Vision. She has published various national & international papers. priyankaj@cdac.in



Authors Profile

Shashak Samaiya is pursuing B.E in Computer Science and Engineering from Bansal Institute of Science and Technology, Bhopal. He is currently in 7 th semester of his B.E. course. He did summer internship project of Intelligent Classroom Management and Monitoring system using Python programming language which covered aspects of IOT, machine learning and web services. ssamaiya97@gmail.com



Yastika Jain is pursuing B.tech in Computer Science from Jaypee University of engineering and technology, Madhya Pradesh. She is currently in 6th semester of her B.tech course. She has participated in various university level workshops and projects. She did a summer internship project of Intelligent classroom management and monitoring system using Python programming language

