**A comprehensive visual and interactive assistant tool to simplify exploration of rhymes and synonyms for poets (Rhyme-Viz)**

**I-Introduction:** Writing a poem is a very rewarding and fun experience, and one of the steps to write one is finding words that rhyme together. This is one of the features that makes a poem different from daily conversation [1]. However, finding such words is not always easy. It can be time consuming and sometimes impedes the writers to articulate the true message that they had in mind. Of course, thesauri and rhyming dictionaries can be used. Nevertheless, these resources are fraught with words. There is usually no categorization based on concept or part of speech. Therefore, the task of finding rhyming words remains laborious. Hence, we were encouraged to develop a program to help with the task.

This paper presents a progress report on the Rhyme-Viz project, which offers two solutions for finding rhymes. One is simply presenting rhyming words with the input word. The other is showing synonyms of the input word, accompanied by the rhyming words of each synonym. This helps the writer to replace the input word with a word that has more rhymes or is just more suitable for their poem. This feature can also be used as a simple thesaurus. In addition, clustering based on the meaning, part of speech and other characteristics are implemented in Rhyme-Viz by taking advantage of visual elements like color, location and size.

To the best knowledge of authors, there is no software currently developed with similar capabilities. The closest thing is [2], which is basically a visual dictionary exclusive for iPads and not useful for finding rhymes.

To offer more relevant rhymes, Rhyme-Viz takes advantage of machine learning and natural language processing (NLP) techniques in the literature. For instance, [3] implements a system to determine “topic” and “focus” of each query. Reference [4] suggests a technique to measure semantic similarity. In [5] a crowd-sourcing method to classify words and language is presented. References [6] and [7] use Markov-Chain and an all-inclusive supervised learning algorithm, respectively, for tagging parts of speech and text analysis. A broader overview of NLP is presented in [8]. A method for detecting rhyming word is suggested in [9]; however, it does not have visualization. In Rhyme-Viz visualization is inspired by graph presentation techniques introduced in the class and literature [10], [11]. For development Python is used because of its available libraries and built in functions in language processing [12], and D3 is used for visualization because of its customizability and flexibility [13].

The rest of paper is as follows. Section II explains the proposed approach. It includes the data resources, cloud architecture and the visualization. Section III suggests evaluation techniques and activity plans. Conclusion is presented in section IV.

**II- Proposed approach**

1. **Data resources**
2. **Cloud Architecture, API Layer, and Database**
3. **Visualization**

**III- Evaluation**

**IV- Conclusion**

**References:**

[1] Strachan, John, and Terry, Richard. Poetry. Edinburgh: Edinburgh University Press, 2011, Chapter 3, pp. 47-71.

[2] Scheiner, Robert, “Wordflex touch dictionary.” Wordflex.com, http://www.wordflex.com, (Accessed October 1, 2017).

[3] Andersson, Marta, Adnan Ozturel, and Silvia Pareti. "Annotating Topic Development in Information Seeking Queries." In LREC. 2016.

[4] Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. "A study on similarity and relatedness using distributional and wordnet-based approaches." In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 19-27. Association for Computational Linguistics, 2009.

[5] Chang, Nancy, Russell Lee-Goldman, and Michael Tseng. "Linguistic Wisdom from the Crowd." In Third AAAI Conference on Human Computation and Crowdsourcing. 2016.

[6] Church, Kenneth Ward. "A stochastic parts program and noun phrase parser for unrestricted text." In Proceedings of the second conference on Applied natural language processing, pp. 136-143. Association for Computational Linguistics, 1988.

[7] Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." Journal of Machine Learning Research 12, no. Aug (2011): 2493-2537.

[8] Chowdhury, Gobinda G. "Natural language processing." Annual review of information science and technology 37, no. 1 (2003): 51-89.

[9] Byrd, Roy J., and Martin S. Chodorow. "Using an on-line dictionary to find rhyming words and pronunciations for unknown words." In Proceedings of the 23rd annual meeting on Association for Computational Linguistics, pp. 277-283. Association for Computational Linguistics, 1985.

[10] Herman, Ivan, Guy Melançon, and M. Scott Marshall. "Graph visualization and navigation in information visualization: A survey." IEEE Transactions on visualization and computer graphics 6, no. 1 (2000): 24-43.

[11] Borup, Rick. “Data Visualization for the Database Developer.” Paper presented at the Southwest Fox conference, Gilbert, AZ, October 2015.

[12] Bird, Steven, Ewan Klein, and Edward Loper. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.", 2009. pp.1-21

[13] Murray, Scott. Interactive Data Visualization for the Web: An Introduction to Designing with. " O'Reilly Media, Inc.", 2017.