

# FUN WRITER: A CONTEXT-BASED INTELLIGENT WRITING PLATFORM TO ASSIST AND MOTIVATE WRITING ACTIVITIES USING ARTIFICIAL INTELLIGENCE AND NATURAL LANGUAGE PROCESSING

Jay Pang<sup>1</sup>, Marisabel Chang<sup>2</sup>

<sup>1</sup>The Webb Schools, Front Entrance, 1175 W Baseline Rd,  
Claremont, CA 91711

<sup>2</sup>Computer Science Department, California State Polytechnic University,  
Pomona, CA 91768

## **ABSTRACT**

*The aim of this paper is to apply computer techniques to assist high school students' essay writing. A python program named Fun Writer is developed to assist users in paraphrasing, summarizing, and keywording. We applied it to real text sentences and paragraphs and conducted a qualitative evaluation of the approach. The results show that Fun Writer can automatically provide users with different versions of sentence paraphrases, Text or paragraph summaries, and keyword retrieval, thus being flexible to the user's needs. The program will be a website where the user can upload his own input text and freely select to either paraphrase, summarize, or extract keywords.*

## **KEYWORDS**

*Writing, application, paraphrase, summary, keyword*

## **1. INTRODUCTION**

Essay writing can be difficult, especially if students need to solve problems with redundancy and understanding [1]. High school students particularly find it difficult to properly paraphrase or analyze texts when writing. What many end up doing is switching a couple of words for synonyms or swapping and flipping the sentence around to paraphrase, which often sounds unnatural or obvious that it is taken from elsewhere. They may summarize a paragraph or a text in a shorter or simple version but fail to keep the key ideas of the original. They even find analyzing other texts containing many complex subjects and terminology can seem impossible [2]. Therefore, we wanted to create an online tool to deal with this situation.

Currently, many writing assistance programs already exist, including Grammarly and google spell check, but these programs generally focus on the grammar portions of an essay [3][4]. However, the changes that these programs provide appear as suggestions which in many cases is not what the user is trying to find. Our goal was to create a python program where the user can choose exactly a specific part of a text they want analyzed and how [5]. There are many libraries that, if implemented correctly, can be purposed into analyzing the content of a sentence or

paragraph. The three libraries we used for this program are Pegasus, Bart Transformer, and NLTK Rake for paraphrasing, summarizing, and finding keywords, respectively [6].

Our method took inspiration from prompt generators and programs such as Grammarly. Many useful features we noticed in Grammarly include the ability to spot redundancy within text and find paraphrased solutions to replace them. Unlike Grammarly which can be attached to a browser and be used in many websites, we decided to start off our program on its own website, which can be expanded upon in the future. Also unlike Grammarly, the user selects whether they want to paraphrase, summarize, or find keywords for the text.

To prove the effectiveness of the program, each function's results can be analyzed using different prompts. Paraphrasing solutions can be compared to the original text, checking if the subject remains consistent. Summary solutions are considered successful if they retain the same meaning but within much fewer words. Keyword solutions can be analyzed by checking whether they contain mainly subjects and actions rather than transition or conjunction words such as “because” or “and”.

The rest of the paper is organized as follows: Section 2 describes the challenges we faced throughout the development of this project. Section 3 explains our methods and solutions for our program and the challenges we met, while Section 4 presents two specific experiments we conducted to test our program. Section 5 compares our program to other text editing/analyzing programs. Section 6 contains our conclusion and plans for future development.

## **2. CHALLENGES**

In order to build the project, a few challenges have been identified as follows.

### **2.1. Finding a Way to Solve the Issues**

The first challenge we faced was finding a way to solve the issue regarding essay writing beyond grammar or word spelling level. I spent a while observing how student essays can be improved and eventually settled on paraphrasing, summarizing, and finding keywords. Actions on these three aspects can not only help students to understand it better but also enhance the quality.

### **2.2. Paraphrased Sentences in Different Ways**

The second challenge faced while developing this project was that sentences can be paraphrased in different ways and keywords can also differ depending on the context [7]. We noticed that the previous program could paraphrase sentences in ways the user may not prefer. To solve this issue, we designed the program to give multiple outputs for both paraphrasing and keywords. In the paraphrasing and keyword functions, several solutions are taken from the libraries and saved into variables, which are then displayed and chosen by the user themselves.

### **2.3. Unpredictability of the Libraries we Used**

One more challenge we encountered was the unpredictability of the libraries we used. The outputs they created would often vary even with the same inputs, and there would often be mistakes regarding punctuation and other grammar issues as well. Therefore, we needed to test several different libraries for each action with many inputs until we found the 3 most consistent ones to use.

### 3. SOLUTION

Our program must recognize certain elements within the language in order to assist in writing. Coding each behavior for the program is impossible because of the number of rules in English, so as mentioned above, we implemented three different libraries for paraphrasing, summarizing, and finding keywords. Shown in Figure 1, The user first inputs the text they want to analyze into a text box, which is saved into a variable when the program is activated. We avoided making the program to save after every change in the text to allow the user to write long texts in the box without constantly running the program. The user can now give the second input, which is to select their desired function for the program. There are three buttons located beneath the text box labeled “Paraphrase”, “Summarize”, and “Keyword”. The program will perform the corresponding function on the text within the box when a button is clicked. Each library can create multiple different solutions, which the algorithm will output in a column.

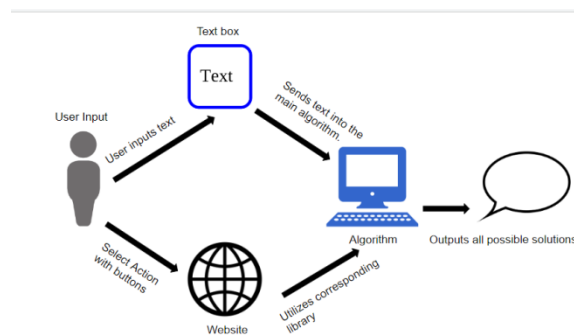


Figure 1. Project Layout

To implement this system, we imported all three libraries at the top of the program [8]. The program begins running after the user selects an action, in which the text within the text box are called into a function. The function outputs are saved within the variable “message” as seen above, which is returned and displayed on the website. The function for paraphrasing is defined above, while the summarizing and keyword actions are performed within this function.

## 4. EXPERIMENT

### 4.1. Experiment 1

The most effective way to experiment with this program is to simply test different inputs and compare them with the results. The Paraphrase function is designed to help students reword certain sentences, so the result maintains the same meaning and subjects as the original text. The Keywords function is meant to assist the user in understanding the text and to help them with summary writing, while the Summary function gives them an example of what it could look like. The keyword outputs should consist of mainly subject nouns and verbs from the text, as well as related words, but not function words such as “to”, “the”, or “for” etc. While the Summary outputs must be shorter than the original, it must also retain adequate amounts of important information. For the first experiment, we tested the program using the prompts as follows:

The input text: “A tractor is an engineering vehicle specifically designed to deliver a high tractive effort (or torque) at slow speeds, for the purposes of hauling a trailer or machinery such as that used in agriculture, mining or construction. Most commonly, the term is used to describe a farm

vehicle that provides the power and traction to mechanize agricultural tasks, especially (and originally) tillage, and now many more.”(Segment with 2 sentences, 66 words)

The results of the three functions of Fun Writer are as follows:

Paraphrase: The last sentence is paraphrased into three versions:

The term refers to a farm vehicle that provides the power and traction to mechanize agricultural tasks. (17w)

The term is used to describe a farm vehicle that provides the power and traction to mechanize agricultural tasks.(19w)

The term is often used to describe a farm vehicle that provides the power and traction to mechanize agricultural tasks.(20w)

Keywords: engineering vehicle specifically designed', 'mechanize agricultural tasks', 'high tractive effort', 'farm vehicle', 'slow speeds', 'used', 'used', 'trailer', 'tractor', 'traction', 'torque', 'tillage', 'term', 'purposes', 'provides', 'power', 'originally', 'mining', 'many', 'machinery', 'hauling', 'especially', 'describe', 'deliver', 'construction', 'commonly', 'agriculture'<sup>5(27 item)</sup>

'Summary: “A tractor is an engineering vehicle specifically designed to deliver a high tractive effort (or torque) at slow speeds. Most commonly, the term is used to describe a farm vehicle that provides the power and traction to mechanize agricultural tasks, especially (and originally) tillage. (Segment with 2 sentences but only 44 words)

## 4.2. Experiment 2

In the second experiment, we chose a random narrative segment rather than an informational text. After testing, we looked for the same conditions presented above. The input was:

“There were two things that were important to Tracey. The first was her dog. Anyone that had ever met Tracey knew how much she loved her dog. Most would say that she treated it as her child. The dog went everywhere with her, and it had been her best friend for the past five years. The second thing that was important to Tracey, however, would be a lot more surprising to most people” (segment with 6 sentences, 73 words in total)

The results of the three functions of Fun Writer are as follows:

Paraphrase: The last sentence is paraphrased as:

Most people would be surprised by the second thing that was important to Tracey.

Keyword: “ever met tracey knew', 'past five years', 'dog went everywhere', 'two things', 'second thing', 'best friend', 'would say', 'tracey', 'tracey', 'dog', 'dog', 'would', 'treated', 'surprising', 'people', 'much', 'loved', 'lot', 'important', 'important', 'however', 'first', 'child', 'anyone'

Summary: “There were two things that were important to Tracey. The first was her dog. Most would say that she treated it as her child. The dog went everywhere with her and it had been her best friend for the past five years. The second thing that was important to her would be a lot more surprising.”(Summary in 5 sentences, 56 words in total)

Results from both experiments show mostly accurate predictions. The paraphrase function works well at sentence level by providing 3 versions of paraphrasing for users to select, so that they can express the same idea in various grammatical and lexical ways. The summary function can shorten the original text but maintain its main ideas, so that users can understand or write about the topic more concentratedly. The Keywords function can display the content-type of key words that mainly consist of subject nouns and action verbs to help the user figure out the most useful information of the text or use them in writing.

## 5. RELATED WORK

Fitria, T. N. analyzes Quillbot, an online paraphrasing app that focuses on paraphrasing English texts [9]. Both Quillbot and our program use synonyms, passive/active voices, word forms, and word reorganization to paraphrase texts, and requires the user to input text into the website. However, unlike ours, Quillbot also performs grammar and sentence fixing functions that are not yet implemented in our program.

Litvak, M. et al. describes MUSEEC, a tool that uses three different methods of summarizing texts, MUSE, POLY, and WECOM, which creates multiple different summaries and compares them to find the best [10]. This is similar to the paraphrasing algorithm in our project, where we create a few solutions for users to choose.

Zhang, C. describes CRF, a method of finding keywords, and further explains the process of keyword extraction [11]. Unlike previous works, they focus on the use of functions and the processes behind keyword finding algorithms. Despite our use of libraries and functions for the project, I took note of many aspects used within these algorithms from this work.

## 6. CONCLUSIONS

In this paper, we report on how we use artificial intelligence, natural language processing and python coding to create a context-based intelligent writing platform to assist and motivate writing activities. The website platform utilizes the libraries Pegasus, Bart Transformer, and NLTK rake. The users can paste or type text into a text box on the website, then select the functions of paraphrase, summarize, or keywords by clicking three buttons below the box. To test the effectiveness, we gave several unique inputs varying in length and content density, and found that each function works well, as is illustrated in Experiment 1 and 2 in above.

One limitation of this program is that the accuracy is constantly varying, and there are many small grammar related issues within the library that occasionally appear. Another limitation is usage itself, as the website can be better optimized with more actions to create a better writing environment.

In the future, we can solve these issues by including with extra conditions and logic placed within the algorithm itself to check for mistakes in the solutions. The website can be reorganized into a more practical state with more precise and common actions such as grammar checks and passage tone.

**REFERENCES**

- [1] Hounsell, Dai. "Essay writing and the quality of feedback." *Student learning: Research in education and cognitive psychology* (1987): 109-119.
- [2] Maddela, Mounica, Fernando Alva-Manchego, and Wei Xu. "Controllable text simplification with explicit paraphrasing." *arXiv preprint arXiv:2010.11004* (2020).
- [3] Ghufron, M. Ali, and Fathia Rosyida. "The role of Grammarly in assessing English as a Foreign Language (EFL) writing." *Lingua Cultura* 12.4 (2018): 395-403.
- [4] De Amorim, Renato Cordeiro, and Marcos Zampieri. "Effective spell checking methods using clustering algorithms." *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*. 2013.
- [5] Sanner, Michel F. "Python: a programming language for software integration and development." *J Mol Graph Model* 17.1 (1999): 57-61.
- [6] Deelman, Ewa, et al. "Pegasus, a workflow management system for science automation." *Future Generation Computer Systems* 46 (2015): 17-35.
- [7] Bhagat, Rahul, and Eduard Hovy. "What is a paraphrase?." *Computational Linguistics* 39.3 (2013): 463-472.
- [8] Robey, Daniel, Jeanne W. Ross, and Marie-Claude Boudreau. "Learning to implement enterprise systems: An exploratory study of the dialectics of change." *Journal of management information systems* 19.1 (2002): 17-46.
- [9] Fitria, Tira Nur. "QuillBot as an online tool: Students' alternative in paraphrasing and rewriting of English writing." *Englisia: Journal of Language, Education, and Humanities* 9.1 (2021): 183-196.
- [10] Litvak, Marina, et al. "Museec: A multilingual text summarization tool." *Proceedings of ACL-2016 System Demonstrations*. 2016.
- [11] Bale, Tracy L., and Wylie W. Vale. "CRF and CRF receptors: role in stress responsivity and other behaviors." *Annu. Rev. Pharmacol. Toxicol.* 44 (2004): 525-557.