TOWARDS APPLYING NATURAL LANGUAGE APPROACH IN IMPROVING STUDENTS' WRITING SKILLS: LECTURER FEEDBACK TOOLS (LEFT)

[1]Hasniza Nordin, [2]Juhaida Abu Bakar, [3]Tengku Faekah Tengku Ariffin & [4]Nurbaya Ghazali

[1,2&3]Universiti Utara Malaysia, Malaysia
[4]Politeknik Sultan Abdul Halim Mu'adzam Shah, Malaysia

[2]Corresponding author: juhaida.ab@uum.edu.my

ABSTRACT

Automatic text summarisation involves creating a concise and coherent description while preserving essential information and context. Time constraints faced by educators prompted this study. Writing is a crucial skill for students, especially in English; however, many need more feedback on their mistakes. Writing, akin to scaffolding, necessitates the involvement of proficient essay writers to assess and review students' work. Scaffolding facilitates student engagement and improves learning outcomes, making it an effective instructional strategy. To address this issue, Lecturer Feedback Tools (LeFT), a Natural Language Processing web application system, is introduced to educators to provide feedback to enhance students' writing skills. It utilises four summarisations SpaCy, Gensim, Natural Language Toolkit, and Sumy. What sets this tool apart is its ability to record all operations in a database, ultimately boosting educators' productivity and students' work. The evaluation results show a consensus among participants, with 66.7% expressing the system's usefulness and 83.3% finding it easy to use, reflecting their satisfaction with the study's outcomes.

Keywords: Text summarisation, scaffolding, natural language processing, writing skills, web-based application.

INTRODUCTION

Natural Language Processing (NLP) is a specialised branch of artificial intelligence that enables computers to comprehend human language during interactions. Text summarisation, on the other hand, refers to creating a brief and coherent summary while preserving essential information and the broader context. Over recent

years, numerous methods for automating text summarisation have emerged and found extensive application across various domains (Allahyari et al., 2017).

Numerous websites and software applications utilising NLP have emerged to enhance user convenience. Particularly prominent are educational websites that aid students in skill improvement. Consequently, the development of Lecturer Feedback Tools (LeFT) has taken place, primarily focusing on enhancing students' writing capabilities. The outcomes of this study will be applied at Kolej Poly-Tech MARA Alor Setar, Kedah, to support lecturers in nurturing their students' writing proficiency.

Mastering writing is crucial for students due to its widespread application, particularly in English, where it serves various purposes like composing reports, crafting CVs, and building resumes. Despite this importance, many students still need help to attain proficiency in writing. One contributing factor is the need for correction when students make writing errors, leading to repeating the same mistakes. Scaffolding, also called writing, entails the involvement of skilled individuals in assessing and reviewing students' written work, facilitating engagement in learning, and improving learning outcomes. Researchers such as Belland et al. (2017) and Doo et al. (2020) recognise this approach as an effective instructional strategy. The guidance and support of lecturers play a pivotal role in students' success. Moreover, involving lecturers in examining their work allows students to gain new knowledge through practical experience.

Scaffolding, a technique for enhancing students' writing skills, involves one person assisting another, much like writers helping each other. However, this approach has become challenging due to time constraints and a large student population. Consequently, the idea of creating resources to address this challenge emerged. In response, the Lecturer Feedback Tool (LeFT) was developed to aid educators in reviewing and offering feedback on student essays. The primary goal of creating this tool is to support students in enhancing their English writing abilities. This approach aids students by summarising the essays they upload, employing summarisation techniques that extract relevant sentences from the original text. These summaries typically include the most pertinent information. With this review process, lecturers can quickly provide feedback, and students can utilise it to refine their English writing skills.

The utilisation of NLP technology played a pivotal role in the creation of LeFT, with the primary goal of enhancing students' writing skills. Several studies (Chary et al., 2019; Lende & Raghuwanshi, 2016; Yu et al., 2020) have highlighted the substantial benefits that students have derived from incorporating NLP into their educational experience. These advantages stem from NLP's effectiveness in facilitating scientific learning, particularly in language acquisition. NLP supports students and is a valuable tool for educators, enabling them to experiment with diverse teaching methodologies. NLP will foster teacher development and positively impact student outcomes.

RELATED WORKS

This section overviews Automatic Text Summarization (ATS) research and its evolution through web-based applications. ATS encompasses three distinct approaches: extractive, abstractive, and a hybrid blend of both. The extractive approach selects the most pertinent sentences from the input document(s) and assembles them to form the summary. On the other hand, the abstractive process transforms the input document(s) into an intermediate representation before generating a summary composed of sentences that may differ from the

originals. As El-Kassas et al. (2020) demonstrated, the hybrid approach combines elements from extractive and abstractive methodologies.

In a study conducted by Kandpal et al. (2020), a novel approach that combines Sentiment Analysis and Auto-Text Summarization is introduced to assist content writers in improving the quality of their manuscripts. This research involves various observations that analyse the summarised text's polarity and subjectivity using different summarisation techniques. The study also employs Sumy, spaCy, NLTK, and Gensim for experimentation.

In the work by Chen (2022), a combination of fuzzy logic and a neural network is applied to extract essential sentences, followed by an abstraction model to generate a summary. The evaluation of the abstraction model utilises ROUGE scores, revealing a 9.2% improvement over the traditional abstraction model. Additionally, an interactive web-based application is provided to facilitate a more user-friendly presentation of the model.

In another study, Sawicki et al. (2023) delve into a systematic automated review of NLP literature using NLP techniques. One of the approaches for extracting critical information discussed is text summarization. The study explores four extractive summarisation methods, including the Luhn method (Luhn, 1958), Latent Semantic Analysis (Steinberger & Jezek, 2004), LexRank (Erkan & Radev, 2004), and TextRank (Mihalcea & Tarau, 2004). Additionally, the study explores abstractive summarisation, a method that comprehends the context of the text and generates a summary. Abstractive summarisation methods include BERT-like models such as the distilbart-cnn-12-6 model (Lewis et al., 2019), part large-cnn (Lewis et al., 2019), bert-extractive-summarizer (Miller, 2019), and the Pegasus model (Zhang et al., 2020). Recent work also introduces a transformer model designed explicitly for long text summarisation called the "Longformer" (Beltagy et al., 2020).

SpaCy serves as an illustration of an extractive summarisation technique. These methods extract textual elements, such as phrases and sentences, and aggregate them to construct a summary. Consequently, specifying the requisite summarisation sentences is paramount in an extractive approach. This process involves token filtering through word frequency assessment, list normalisation, and assigning weights to each punishment based on the occurrence of tokens within them—all of which are integral steps in spaCy's summarisation process. The outcome is stored as a key-value pair in "sent weight," where sentence units from the document string represent the keys, and their respective weights serve as the values. The final step entails generating a summary using the "nlargest" function. This study also incorporates three leading summarisation tools: Gensim, NLTK, and Sumy. It is worth noting that while Gensim includes an inbuilt summarisation feature, its effectiveness falls short of spaCy's capabilities. This research is founded on the findings presented in Agbe (2019).

METHODOLOGY

The research was carried out by the Rapid Application Development (RAD) methodology, as put forth by Martin (1991). RAD is an agile software development approach that employs prototyping to gather application system requirements. Despite the evolution of software development methodologies, as Jnr et al. (2018) noted, RAD remains pertinent and continues to be extensively utilised by software developers. It comprises four primary phases: requirements planning, user design, construction, and cutover.

In preparing requirements for the Lecturer Feedback Tools (LeFT) web application, obtaining its specifications is crucial. Unified Modeling Language (UML) diagrams, including use case, activity, and class

diagrams, are employed to document and visualise these specifications. Adediran and Al-Bazi (2018) and Hussain et al. (2014) noted that UML diagrams are widely utilised for presenting device specifications.

The web application's user interface development occurs concurrently with the user concept and construction phases. Users actively participate in the design and construction stages, offering valuable input to enhance the user interface and information flow of LeFT. Finally, in the cutover phase, LeFT's usability is evaluated. This section will encompass the requirements planning, user design, and construction phases, while the Result section will elucidate the cutover phase.

Within this section are two distinct subsections: (1) outlining the web-based requirements for LeFT and (2) establishing a web-based prototype of LeFT intended to illustrate the gathered requirements.

One of the methods employed during the requirement-gathering phase was observation. Observations were conducted on the client and the existing method to compare comprehensively. By observing users, analysts could delineate process flows, metrics, pain points, and areas for potential enhancement. Furthermore, similar systems were also subjected to observation. This approach augments the likelihood of practical prototype improvements, as it allows for rectifying shortcomings in the previous framework as the implementation progresses.

All the tables, diagrams, and use cases observed are still in their early developmental stages. Given that the improvements will likely be integrated into the final product, the resultant system may not necessarily mirror its previous incarnation.

Table 1 displays the six functional requirements and their priorities identified during the initial requirements-gathering phase. These include website user registration and login, assignment management, summarisation management, marks management, and enhancing usability experience. Meanwhile, Table 2 enumerates two non-functional specifications, explicitly addressing issues related to reliability and usability.

Table 1

Functional Requirement

No.	Requirement ID	Requirement Description	Priority
1.	LEFT_01	*Registration*	
	LEFT_01_01	Students and lecture can fill in the register form	M
	LEFT_01_02	The system verifies the registration	M
2.	LEFT_02	*Login*	
	LEFT_02_01	Students, lecture, and admin can fill in their ID and password	M
	LEFT_02_02	The system shall display the homepage	M
	LEFT_02_03	The system will authenticate the lecturer and student's password	D
	LEFT_02_04	Students and lecturers can log in from the system	M

No.	Requirement ID	Requirement Description	Priority
3.	LEFT_03	*Manage assignment*	
	LEFT_03_01	Student can insert their assignment	M
	LEFT_03_02	Student and lecturer can review assignment	M
	LEFT_03_03	Student can view their marks	O
4.	LEFT_04	*Manage summarisation*	
	LEFT_04_01	Lecturer can create summarisation text	M
	LEFT_04_02	The lecturer can read summarisation text	M
	LEFT_04_03	The lecturer can update the summarisation text	M
5.	LEFT_05	*Manage marks*	
	LEFT_05_01	The lecturer can read marks	M
	LEFT_05_02	The lecturer can calculate marks	M
	LEFT_05_03	Lecturer view marks	M
	LEFT_05_04	The lecturer can update marks	M
6.	LEFT_06	*Manage usability experience*	
	LEFT_06_01	Users can read 'About' information	O
	LEFT_06_02	Users can fill 'Contact Us information	O

Table 2

Non-Functional Requirement

No.	Requirement ID	Requirement Description	Priority
1.	LEFT_01	*Reliability issues*	
	LEFT_01_01	For a single user, the system should crash no more than once per 10 hours	M
	LEFT_01_02	The system should be reloaded and work perfectly when it crashed	M
	LEFT_01_03	A single user cannot access the system by using more than one device	M
2.	LEFT_02	*Usability issues*	
	LEFT_02_01	All users can easily use this tool	M
	LEFT_02_02	It can only be used when it comes to internet coverage	M

The requirements outlined in Table 1 were transformed into the computer system's functionality. The subsequent step involves depicting and structuring the app's needs using suitable modelling techniques and tools. The Unified Modeling Language (UML) was employed in this project to visualise and model these requirements. The models utilised here include two behavioural diagrams, use case and activity diagrams, and a class diagram representing the app's structural components. These diagrams were created using the StarUML tool.

The class diagram depicted in Figure 1 portrays the structural elements outlined in LeFT. The attributes and operations associated with LeFT can be observed within this class diagram, clearly showing the interactions between the various classes.

Figure 1

LeFT Class Diagram
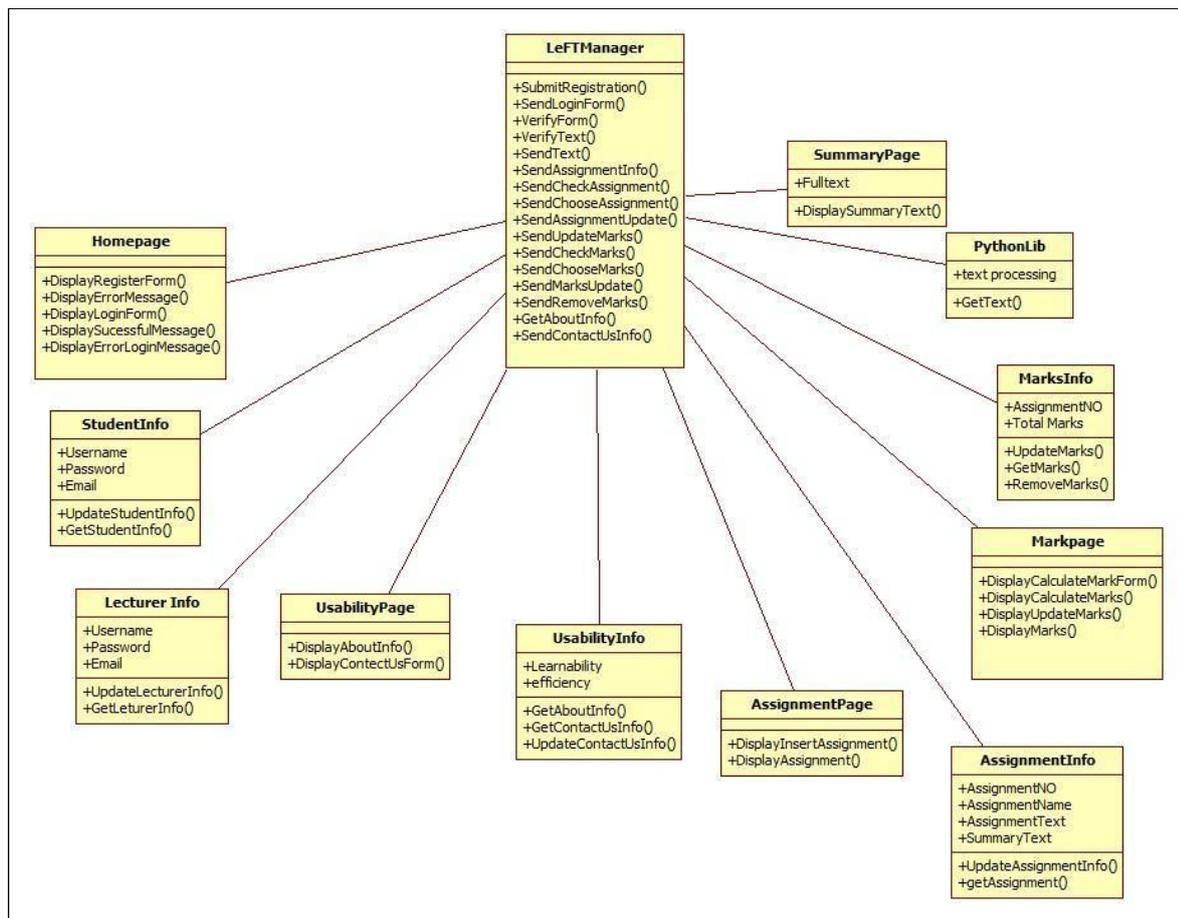
A LeFT prototype was created to embody the requirements elucidated in the preceding section. Software prototyping is a conventional approach for showcasing software requirements, allowing users to provide additional feedback and suggestions based on their interaction with the prototype. The primary integrated development environment (IDE) tool employed was Dreamweaver, while the Pycharm development platform facilitated essential functions. For data storage, SQLite was utilised as the database. Figures 2-8 present screenshots showcasing the chosen LeFT interfaces.

Figure 2

LeFT Homepage
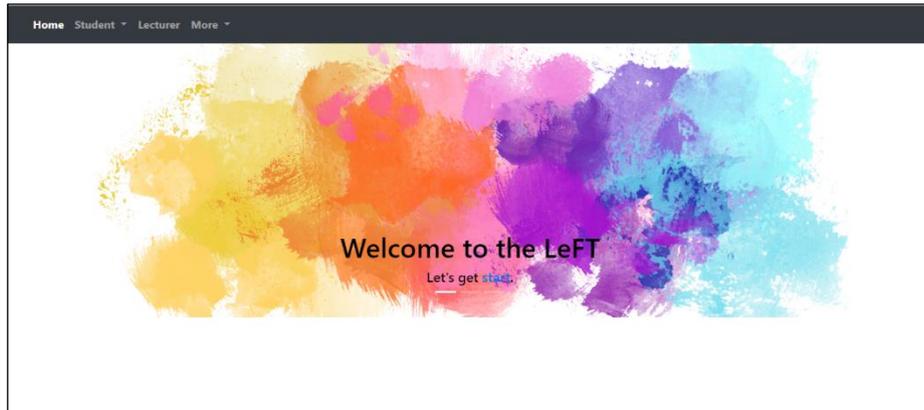
In Figure 2, a homepage showcasing four distinct menus can be observed. This homepage is accessible to all users, and each user has the option to select whether they wish to enter as a lecturer or a student from the available menu. There are two user categories: students and lecturers. All users can access the main menu options and the "Contact Us" feature.

Figure 3

Registration Page

In Figure 3, a registration page designed exclusively for students can be observed. The registration feature is intended solely for new students using this tool for the first time. On the other hand, lecturers are not required to go through the registration process as their usernames and passwords have already been predefined as defaults.

Figure 4

Login Page
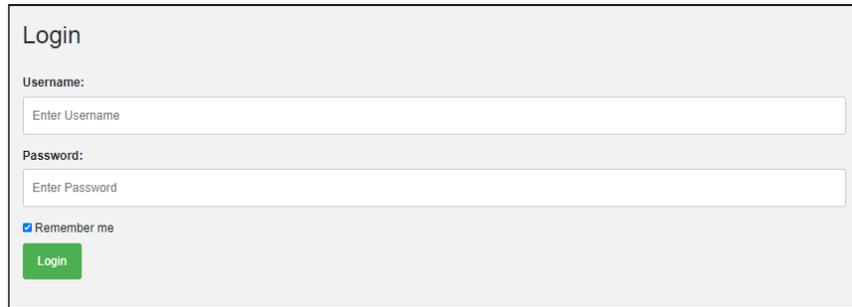
In Figure 4, the login function is illustrated, and it is a crucial step for both students and lecturers to input their accurate usernames and passwords. Provide correct information to ensure users are able to access these tools.

Figure 5

Summarisation based on SpaCy, Gensim, NLTK and Sumy page
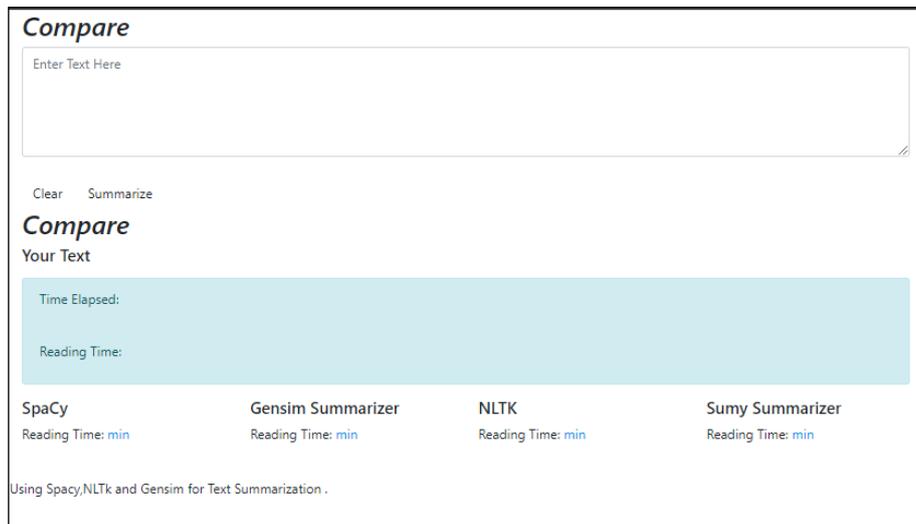
Figure 5 depicts the summarisation page within this tool. On this page, lecturers can generate summaries from assignments submitted by students. The area above serves as a space where lecturers can input text to be summarised. Subsequently, lecturers must click the 'Summarize' button, upon which the system will generate four distinct types of summaries. The outcomes of the text summarisation process will be displayed in Figure 6 below.

Figure 6

Summarisation Result Page

Figure 7

Manage Marks Page

Figure 7 shows a page for calculating student scores based on their submitted assignments. This page includes a rubric and designated areas for inputting the marks. Each row contains a description to facilitate the lecturer in assigning scores. Once the effects have been entered into the appropriate spaces, the lecturer must click the 'Calculate Score' button to obtain the total score, as demonstrated in Figure 8.

87

Figure 8

Calculate the Mark Page

In the evaluation setting, when conducting a system review, approximately 30 participants, consisting of UUM students, two UUM lecturers, and seven UUM staff members, will be involved in the testing process. Google Form links have been distributed via WhatsApp for participant access. Additionally, a tutorial video explaining how to use these tools was shared through a video link along with the Google Form link earlier. A Post-Study System Usability Questionnaire (PSSUQ) questionnaire adopted by Lewis (1992) has been provided to gather feedback on the system, consisting of two sections. Section A pertains to respondent demographics and background information, while Section B focuses on assessing the usefulness and ease of use of LeFT. All participants are expected to complete all the questions included in the questionnaire.

RESULTS

Out of the 30 respondents who completed the distributed questionnaire, the demographic and background information reveals that 53.3% are male, while the remaining 46.7% are female. The data also indicates that most respondents fall within the 21-25 age group, accounting for 73.3% of the total. Additionally, 13.3% belong to the 26-35 age bracket, 10% are in the 36-45 age group, and the remainder are aged 46 and above.

Regarding internet usage, 90% of respondents reported using the internet 'daily,' while the remaining 10% indicated 'weekly' usage. Approximately 80% of respondents had not heard about the Lecturer Feedback Tool. As a result, 53.3% of respondents confirmed they had never used the system, 3.4% were uncertain if they had used it before, and 43.3% reported that they had used it. The demographic information is depicted in Figure 9.

Figure 9

Demographic Information of the Respondents
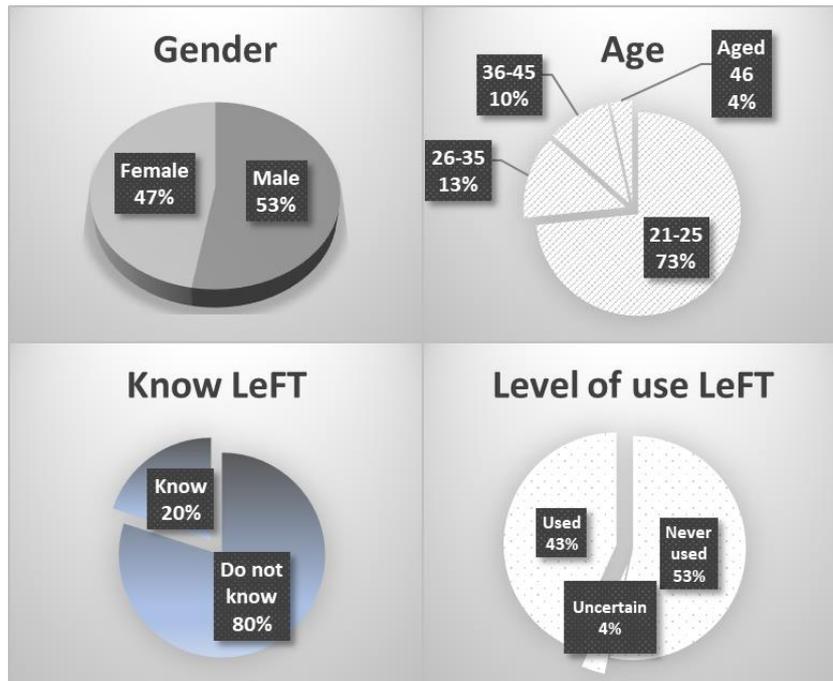
The Usability of LeFT

An assessment was conducted based on the input from the participants in Section B of the post-LeFT task questionnaire. This specific section aimed to gauge the participants' perspectives on the effectiveness, user-friendliness, and overall satisfaction with LeFT, using a Likert scale ranging from "Strongly Agree" (5) to "Strongly Disagree" (1). It is worth noting that this assessment draws inspiration from the PSSUQ (Post-Study System Usability Questionnaire).

Table 3

The Usefulness of LeFT

No.	Usefulness task	Categories	Count	Percentage (%)
1.	I can complete the task quickly using LeFT.	1	0	0
		2	0	0
		3	6	20
		4	11	36.7
		5	19	63.3
2.		1	0	0

I can recover quickly and easily whenever I make a mistake while using LeFT.	2	0	0
	3	0	0
	4	10	33.3
	5	20	66.7
3. LEFT has all the functions and capabilities I expect it to have.	1	0	0
	2	0	0
	3	1	3.3
	4	9	30
	5	20	66.7
4. All functions and capabilities work or behave precisely as anticipated.	1	1	3.3
	2	0	0
	3	1	3.3
	4	14	46.7
	5	14	26.7
5. The terminology is clear and precise.	1	0	0
	2	0	0
	3	1	3.3
	4	16	53.3
	5	13	43.3
6. There are too many inconsistencies in LEFT.	1	22	75.9
	2	5	17.2
	3	2	6.9
	4	0	0
	5	0	0
7. LEFT gives error messages that tell me how to fix the problem.	1	0	0
	2	1	3.3
	3	3	10
	4	11	36.7

		5	15	50
8.	Overall, I am satisfied with LEFT.	1	0	0
		2	0	0
		3	0	0
		4	10	33.3
		5	20	66.7

Table 4

The Ease of Use of LeFT

No.	Ease of Use task	Categories	Count	Percentage (%)
1.	It is easy to learn how to use LeFT.	1	0	0
		2	0	0
		3	6	20
		4	11	36.7
		5	19	63.3
2.	I feel comfortable using LeFT.	1	0	0
		2	0	0
		3	0	0
		4	13	43.3
		5	17	56.7
3.	I feel in control and have freedom when using LeFT.	1	0	0
		2	0	0
		3	1	3.4
		4	11	37.9
		5	17	58.6
4.	Navigation through LeFT is easy.	1	1	3.3
		2	0	0
		3	1	3.3

		4	13	43.3
		5	16	53.3
5.	I always knew where to get the information I wanted.	1	0	0
		2	0	0
		3	0	0
		4	13	44.8
		5	16	55.2
6.	The LeFT interface is pleasant.	1	0	0
		2	0	0
		3	0	0
		4	12	40
		5	18	60
7.	I like the LeFT interface.	1	0	0
		2	0	0
		3	0	0
		4	12	41.4
		5	17	58.6
8.	The help provided in LeFT is informative.	1	0	0
		2	0	0
		3	2	6.7
		4	9	30
		5	19	63.3
9.	Menus and screens are laid out logically.	1	0	0
		2	0	0
		3	1	3.3
		4	11	36.7
		5	18	60
10.		1	0	0

All content (information, dialogue, or graphics) in LeFT is relevant.	2	0	0
	3	1	3.3
	4	13	43.3
	5	16	53.3
11. Overall, I am satisfied with how easy it is to use LeFT.	1	0	0
	2	0	0
	3	0	0
	4	5	16.7
	5	25	83.3

Based on the data provided in Tables 3 and 4, it is evident that a significant proportion of respondents either "agree" or "strongly agree" with the statements presented. For instance, in the usability assessment, 67.7% of respondents strongly agreed with words such as "Whenever I make a mistake while using LeFT, I can recover quickly and easily" and "LeFT has all the functions and capabilities I expect it to have." When asked about inconsistencies in LeFT, 75.9% of respondents indicated a strong disagreement (rating of 1, meaning they strongly do not agree). Moreover, 66.7% of 20 respondents strongly agreed that this web-based system is sound.

In summary, respondents exhibited a high level of agreement (66.7%) with the system's usefulness. Similarly, in terms of ease of use, a substantial 83.3% of respondents strongly agreed that the overall design is user-friendly. These findings collectively indicate a high level of satisfaction among the respondents regarding the procedure.

Expert Review

LeFT engaged an experienced reviewer who conducted a comprehensive evaluation of this project. According to the reviewer's feedback, this tool has room for improvement. Specifically, she recommended incorporating a session feature to ensure only authorised users can access and view their assignments. The existing system lacked this session functionality, displaying all user assignments on a single page. This posed a security risk, allowing other users to edit or delete tasks that did not belong to them.

Additionally, the tool should include a search function based on students' matric numbers. This feature would streamline locating the specific assignments that lecturers seek, enhancing overall efficiency.

CONCLUSION

This paper has outlined the design and development process of LeFT, a system with several facets for potential study. The RAD methodology is utilised, employing prototyping to collect application system requirements. While preparing the conditions for LeFT, this study uses UML diagrams, including use case, activity, and class diagrams. This methodology encompasses the delineation of web-based requirements

for LeFT and the development of a LeFT web-based prototype. Six functional requirements have been identified: registration, login, assignment management, summarisation management, marks management, and improving the overall usability experience. Additionally, two non-functional requirements have been identified, focusing on the aspects of reliability and usability within LeFT.

In the future, additional features will be introduced to enhance the functionality of LeFT. For example, a save function for the summarised text will be incorporated, as well as a feature for the continuous storage of total marks, eliminating the need for users to navigate to another page for mark entry. Furthermore, security will be bolstered by implementing sessions to ensure that students can only access their work, preventing unauthorised access to other students' submissions.

In conclusion, LeFT offers notable advantages to its users, particularly students, as it was designed to enhance their writing skills. However, the precise impact of this project on students' performance remains to be determined. Nonetheless, it has proven effective in alleviating the workload of lecturers when providing feedback to students and saving valuable time, making it a helpful tool for educational purposes.

ACKNOWLEDGMENT

REFERENCES

Adediran, T. V., & Al Bazi, A. (2018). Developing an agent-based heuristic optimisation system for complex flow shops with customer-imposed production disruptions. *Journal of Information and Communication Technology, 18*(2), 291-322.

Agbe, J. (2019). *Machine learning Web Apps.* https://github.com/Jcharis/Machine-Learning-Web-Apps/tree/master/Summaryzer_Text_Summarization_App/

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarisation techniques: a brief survey. *arXiv* preprint arXiv:1707.02268.

Bhatia, N., Jaiswal, A.: Trends in extractive and abstractive techniques in text summarisation. International Journal of Computer Applications 117(6) (2015)

Belland, B. R., Walker, A. E., Kim, N. J., & Lefler, M. (2017). Synthesising results from empirical research on computer-based scaffolding in STEM education: A meta-analysis. *Review of Educational Research, 87*(2), 309-344.

Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. *arXiv preprint* arXiv:2004.05150 (2020)

Chary, M., Parikh, S., Manini, A. F., Boyer, E. W., & Radeos, M. (2019). A review of natural language processing in medical education. *Western Journal of Emergency Medicine, 20*(1), 78.

Chen, S. (2022). *Abstractive Text Summarisation for Tweets.* Master's Projects. San Jose State University. SJSU ScholarWorks.

Dalal, V., Malik, L.: A survey of extractive and abstractive text summarisation techniques. *In: 2013 6th International Conference on Emerging Trends in Engineering and Technology*, pp. 109–110. IEEE (2013)

Doo, M. Y., Bonk, C., & Heo, H. (2020). A meta-analysis of scaffolding effects in online learning in higher education. *International Review of Research in Open and Distributed Learning, 21*(3), 60-80.

Erkan, G., & Radev, D.R.(2004). Lexrank: Graph-based lexical centrality as salience in text summarisation. *Journal Of Artificial Intelligence Research, 22*, 457–479.

Hussain, A., Mutalib, N. A., & Yasin, A. (2014, June). jFakih: modelling mobile learning game. *In 2014 International Conference on Computer and Information Sciences (ICCOINS)* (pp. 1-6). IEEE.

Jnr, B. A., Majid, M. A., & Romli, A. (2018). An empirical study on predictors of green sustainable software practices in Malaysian electronic industries. *Journal of Information and Communication Technology, 17*(2), 347-391.

Kandpal, P., Wadkar, Y., Attri, H., & Bhorge, S. (2020, December). Comparison of Sentiment Analysis on Auto-Summarized Text and original Text using various Summarization Techniques. *In 2020 IEEE Pune Section International Conference (PuneCon)* (pp. 206-211). IEEE.

Lende, S. P., & Raghuwanshi, M. M. (2016, February). Question answering system on education acts using NLP techniques. *In 2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)* (pp. 1-6). IEEE.

Lewis, J. R. (1992). Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. *In Proceedings of the Human Factors Society annual meeting* (Vol. 36, No. 16, pp. 1259–1260). Sage CA: Los Angeles, CA: Sage Publications.

Lewis, M., Liu, Y., Goyal, N., et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* preprint arXiv:1910.13461

Luhn, H.P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development, 2*(2), 159–165.

Lund, A. M. (2001). Measuring usability with the use of a questionnaire12. *Usability interface, 8*(2), 3–6.

Martin, J. (1991). *Rapid application development.* Macmillan Publishing Co., Inc.

Mihalcea, R., Tarau, P. & Textrank (2004): Bringing order into text. *In Proceedings of The 2004 Conference on Empirical Methods in Natural Language Processing,* pp. 404–411.

Miller, D. (2019). Leveraging BERT for extractive text summarisation on lectures. a*rXiv* preprint arXiv:1906.04165

Sawicki, J., Ganzha, M., Paprzycki, M., et al. (2023). The state of the art of Natural Language Processing—a systematic automated review of NLP literature using NLP techniques. *Data Intelligence, 5*(3), 707-749.

Steinberger, J. & Jezek, K. (2004). Using latent semantic analysis in text summarisation and summary evaluation. *Proc. ISIM,* 4, 93–100.

Yu, J., Luo, G., Xiao, T., Zhong, Q., Wang, Y., Feng, W., ... & Tang, J. (2020, July). MOOCCube: a large-scale data repository for NLP applications in MOOCs. *In Proceedings of the 58th annual meeting of the Association for Computational Linguistics* (pp. 3135–3142).

Zhang, J., Zhao, Y., Saleh, M., et al. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarisation. *In International Conference on Machine Learning*, pp. 11328–11339.