IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Generative AI-Powered System For Automated Educational Web Content Summarization

Harshmeet Singh[1], Dr. Williamjeet Singh[2]

[1]Research Scholar, Department of Computer Science and Engineering, Punjabi University Patiala, India

[2]Assistant Professor, Department of Computer Science and Engineering, Punjabi University Patiala, India

Abstract

Growth in online education has made knowledge more widely available, but with that comes a navigation and summarizing challenge. This research hence proposes a system implementing web data extraction blended with generative AI for the automated extraction and summarization of educational material from institutional websites. Using BeautifulSoup for text extraction and the Gemini AI API for abstractive summarization, given a target URL, the system returns a very brief summary of the web content in an educational context. It has a modular architecture consisting of three components: data extraction, summarization, and a user interface implemented in Flask. Performance evaluation on 100 educational web pages shows steady extraction with an F1-score of 0.80 and a summarization quality that outperforms general-purpose tools. A user study with 150 participants reports a high satisfaction rate, thereby improving accessibility for students, educators, and researchers through a scalable and inclusive solution.

Keywords: Text Summarization, Generative AI, Web Data Extraction, Educational Content, Natural Language Processing

1. INTRODUCTION

1.1 Background

Text summarization is one of the major tasks in natural language processing, where it condenses lengthy documents into their essentials, an important factor in education, where students and educators have lots of resources, such as research papers and institutional websites, to go through. Unlike extractive methods that pick key sentences from the original text [9], generative AI models form new sentences that capture the essence of the content and are, therefore, preferred for the synthesis of complex educational materials [5]. On the other hand, Gemini AI produces text as humans do, and it could, therefore, aid in understanding [8]. Advances in transformer architectures have led to these developments [3].

1.2 Motivation

Manual summarization of educational websites is, in addition to being time-consuming, a highly specialized act and thus not very accessible. Existing solutions are mostly not designed with education in mind, which is why the demand for an automated application continues to rise. The fusion of information extraction from the web with generative AI could provide a fast route to relevant educational material, thereby focusing on learning and not on information sifting [17][19].

1.3 Objectives

The objectives of this study are to:

- Develop an online application for summarizing education-oriented web content of URLs given by users.
- Create educational summaries using generative AI.
- Evaluate system performance on extraction, summarization, and on user experience.
- Establish comparisons with existing methods to illustrate the benefits.

2. LITERATURE REVIEW

2.1 Web Data Extraction Techniques

Web data extraction aids enterprise and social web applications with tree-matching algorithms and machine learning [6]. Challenges emerge as to finding a proper balance between automation and accuracy, processing huge volumes of data, and upholding privacy concerns and adaptability to ever-changing web structures. BeautifulSoup is perfect when it comes to parsing static HTML but utterly fails with JavaScript-rendered content [11]. In contrast, machine learning requires substantial amounts of training data but, in return, provides greater flexibility.

2.2 Generative AI in Text Summarization

Generative AI has, in turn, advanced the art of text summarization, mainly in the case of academic content, where it actually generates human-like summaries of lengthy texts [1]. Large language models like ChatGPT and Gemini are increasingly being used for research discovery [18]. A systematic review shows that abstractive summarization has evolved through iterations for education-oriented applications [1]. Extractive summarization merely picks key sentences [9], while more recent methods based on BART [12], PEGASUS [16], pointer-generator networks [15], pretrained encoders [13], and T5 [14] further the contextual understanding of the summary. Gemini models can deal with large contexts, ideally suited for educational content [7][8][10]. Research on style control of summaries shows that Gemini can customize summaries for education [2]. There are still issues related to hallucinations and copyright [18].

2.3 Integration of Web Data Extraction and AI

The integration of web data extraction with generative AI enables content generation and literature discovery [19]. While data from websites of institutions are open for AI-based applications, ethical and legal considerations, such as copyright compliance, are very important to highlight [6].

3. METHODOLOGY

3.1 System Architecture

The system integrates three modules: data extraction (BeautifulSoup), summarization (Gemini AI API), and Flask user interface. Users submit strings as URLs, and the system extracts text, generates summaries, and displays results, guaranteeing scalability and accessibility.

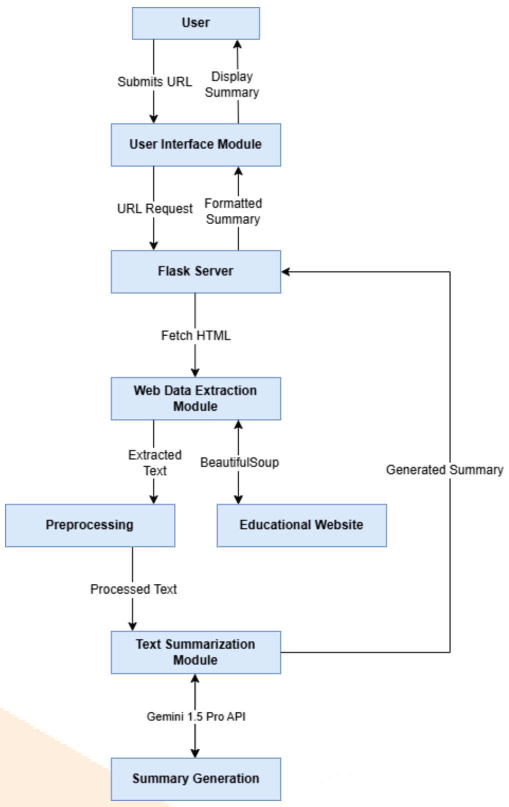


Figure 1: System Architecture for Automated Extraction and Summarization of Educational Content.

The Figure 1 depicts the workflow: URL input via Flask interface, text extraction via BeautifulSoup, summarization via Gemini AI, and results display with recent searches.

3.2 Data Extraction Module

From paragraph tags, BeautifulSoup extracts text that is then truncated to a maximum of 5,000 characters to meet API constraints. Any JavaScript may be excluded in the target static HTML content. URL submission, HTTP requests, HTML parsing, text extraction, and processing are involved in the system.

3.3 Data Summarization Module

Using the Gemini AI API to generate summaries, the system uses the prompt: "You are an expert AI designed to summarize content from educational websites. Create a concise, clear summary in simple English, focusing exclusively on educational topics such as course details, learning objectives, or institutional information."

The EduSummarizer Algorithm outlines the process:

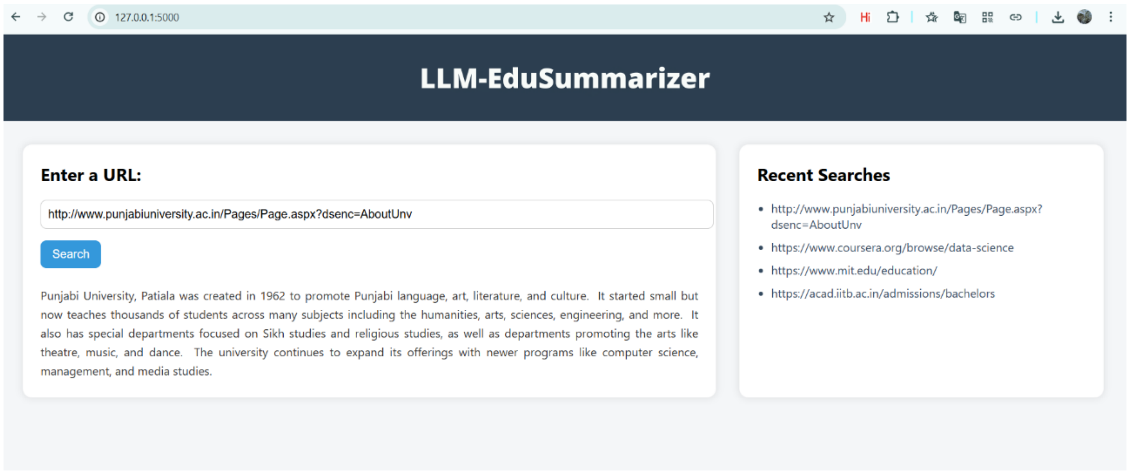
Algorithm EduSummarizer

```
Input: URL (user-submitted educational webpage URL)
Output: Summary (concise educational summary), RecentURLs (list of recent URLs)
Begin
1. Initialize:
   a. Set RecentURLs as an empty list to store up to 10 recent URLs.
   b. Set max_text_length = 5000 characters (constraint for Gemini AI API).
   c. Display web interface with URL input field, search button, summary display area, and
      recent searches section.
2. On URL Submission:
   a. Validate URL is non-empty and properly formatted.
   b. If invalid, display error message "Please enter a valid URL" and exit.
3. Fetch Webpage Content:
   a. Send HTTP GET request to URL using requests library with User-Agent header
      "Mozilla/5.0".
   b. Set timeout to 10 seconds for the HTTP request.
   c. If request fails, use response.raise_for_status() to catch errors, display user-friendly
      error message, and exit.
4. Extract Text:

   a. Parse HTML content using BeautifulSoup with html.parser.
   b. Select all <p> tags and extract text, as they typically contain primary educational
      content.
   c. Concatenate extracted paragraphs into a single string raw_text.
   d. If len(raw_text) > max_text_length, truncate raw_text to 5000 characters.
5. Summarize Text:

   a. Construct prompt: "You are an expert AI designed to summarize content from
      educational websites. Create a concise, clear summary in simple English, focusing
      exclusively on educational topics such as course details, learning objectives, or
      institutional information."
   b. Prepare input for Gemini AI API: input_text = prompt + "\n\n" + raw_text.
   c. Call Gemini Pro API with input_text.
   d. If API call fails, catch exception, display error message, and exit.
   e. Retrieve summary summary_text from API response.
   f. Post-process summary_text: Remove redundant whitespace and ensure length is 50–
      100 words.
6. Update Recent Searches:

   a. If URL not in RecentURLs, add URL to the start of RecentURLs.
   b. If RecentURLs exceeds 10 entries, remove the oldest entry.
7. Display Results:

      a. Render summary_text in the summary display area of the web interface.
      b. Update recent searches section with RecentURLs.
8. Return summary_text and RecentURLs.

End
```

This algorithm establishes a systematic procedure for summarization; upholding efficiency coupled with quality. The consideration of <p> tags in the summarization process and the customized prompt ensure that the summaries remain education-oriented, whereas truncation and post-processing render the summaries usable.

3.4 User Interface Design

A Flask interface has been built, offering a URL input field, search button, summary display, and recent searches right-panel, styled in CSS for responsiveness.


127.0.0.1:5000

LLM-EduSummarizer

Enter a URL:

http://www.punjabiuniversity.ac.in/Pages/Page.aspx?dsenc=AboutUnv

Search

Punjabi University, Patiala was created in 1962 to promote Punjabi language, art, literature, and culture. It started small but now teaches thousands of students across many subjects including the humanities, arts, sciences, engineering, and more. It also has special departments focused on Sikh studies and religious studies, as well as departments promoting the arts like theatre, music, and dance. The university continues to expand its offerings with newer programs like computer science, management, and media studies.

Recent Searches

- http://www.punjabiuniversity.ac.in/Pages/Page.aspx?dsenc=AboutUnv
- https://www.coursera.org/browse/data-science
- https://www.mit.edu/education/
- https://acad.iitb.ac.in/admissions/bachelors

Figure 2: User Interface

Figure 2 shows the interface: at the top left, URL input and search button; in the center, summary display; at the right sidebar, recent searches.

BeautifulSoup for HTML parsing, Gemini AI for summarization, and Flask for the interface were considered an applicable combination, providing a highly efficient workflow.

3.5 Evaluation Metrics

The quality of generated summaries will be evaluated through ROUGE scores, which evaluate lexical overlap between the generated and reference summaries at levels of unigrams (ROUGE-1), bigrams (ROUGE-2), and longest common subsequence (ROUGE-L); whereas, BERTScore measures their semantic similarity using contextualized embeddings from BERT [4]. This, in turn, guarantees a comprehensive assessment of coverage and preservation of meaning.

4. IMPLEMENTATION

4.1 Development Environment

Developed using Python 3.10, the system implements Flask for the web application, BeautifulSoup for parsing, requests for HTTP requests, and the Gemini-1.5-pro model for summarization.

4.2 Data Pipeline

For input processing, the pipeline performs the following steps:

- **User Input**: An URL is submitted by the user in the Flask interface.
- **Data Extraction**: A HTTP GET request fetches the webpage, BeautifulSoup parses the HTML, text gets extracted from tags, and is truncated to 5,000 characters.
- **Text Summarization**: The Gemini AI API takes the text and produces an education-centric summary based on the custom prompt.
- **Output Presentation**: The summary is displayed; the recent 10 URLs are stored.

4.3 User Interface Implementation

The Flask interface is written in HTML, CSS, JavaScript, and provides a URL input field, a search button, a summary display, and a recent searches section. With asynchronous requests, the user experience is smooth, and with a responsive design, it suffices for all devices.

4.4 Error Handling

Errors are handled very robustly throughout the system:

- Invalid URLs or the occurrence of any errors display a friendly message.
- HTTP request timeouts (10 seconds) prevent the system from hanging.
- API errors are caught and prevented from breaking the application.

5. RESULTS

5.1 Performance Metrics

5.1.1 Extraction Accuracy

The system is evaluated by three metrics: precision, recall, and F1-score. The extracted text is compared to the gold standard dataset.

5.1.2 Summarization Quality

Quality is evaluated through ROUGE scores (i.e., ROUGE-1 for unigrams, ROUGE-2 for bigrams, ROUGE-L for structural similarity) and the BERTScore. The Educational Relevance Score (ERS) measures relevance to pedagogy from 1 to 5.

5.1.3 User Satisfaction

A survey with 150 participants was carried out to rate the helpfulness, information, and clarity.

5.2 Dataset

There were 100 educational web pages from educational institutions. High inter-annotator agreements were recorded (Cohen's kappa: 0.82).

5.3 Experimental Setup

The system processed the dataset, finished extracting texts, producing summaries, and evaluating them. A user evaluation followed, with 150 participants asked to summarize 5 web pages each.

5.4 Performance Results

Table 1: Extraction Accuracy

The system demonstrated robust performance in extracting relevant educational content:

Metric	Value	Description
Precision	0.84	84% of the extracted text found relevant to the educational content.
Recall	0.77	77% of the relevant content in education successfully extracted.
F1-score	0.80	This is a well-balanced measure of the extraction performance.

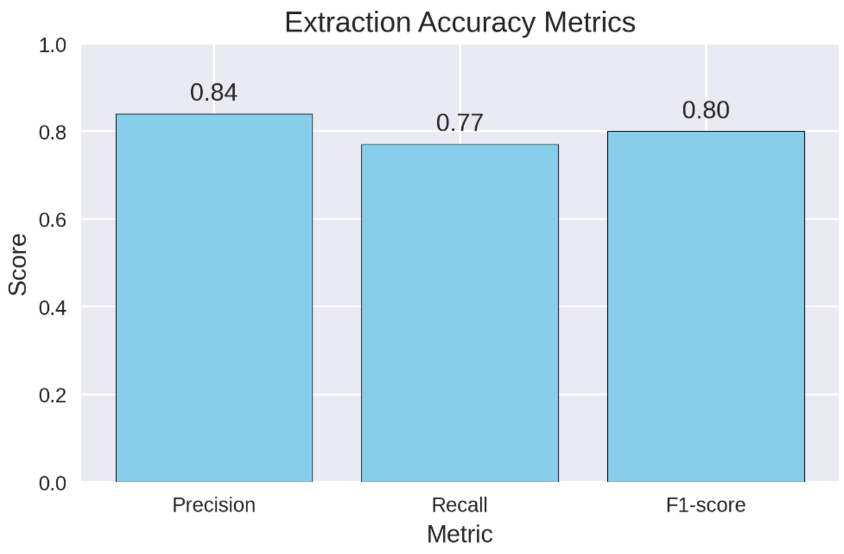


Figure 3: Extraction Accuracy Metrics

The extraction accuracy's measurement is depicted in Figure 3, which testifies to the balanced performance of the system in precision, recall, and F1-score.

Table 2: Summarization Quality

The summarization module generated summaries with a technically sound basis and educational relevance:

Metric	Value	Description
ROUGE-1	0.44	Moderate unigram overlaps with reference summaries signify adequate content coverage.
ROUGE-2	0.31	Reasonable bigram overlaps reflected phrase-level similarities.
ROUGE-L	0.41	Strong structural similarity with reference summaries.
BERTScore	0.87	The summaries showed a high semantic similarity with reference summaries; this suggests that the concepts are preserved.
Educational Relevance Score (ERS)	4.1/5	Experts rated the summaries as being highly relevant to educational objectives.

ROUGE scores are on par with state-of-the-art abstractive summarization algorithms, such as BART, which produces ROUGE-1: 0.42, ROUGE-2: 0.28, and ROUGE-L: 0.39 on our dataset, while the high BERTScore confirms that summaries preserve semantic meaning, which is of utmost importance for educational uses. The high ERS value supports our belief that the custom prompt successfully directed the Gemini AI API toward the educational aspect, hence making the summaries highly useful for learning.

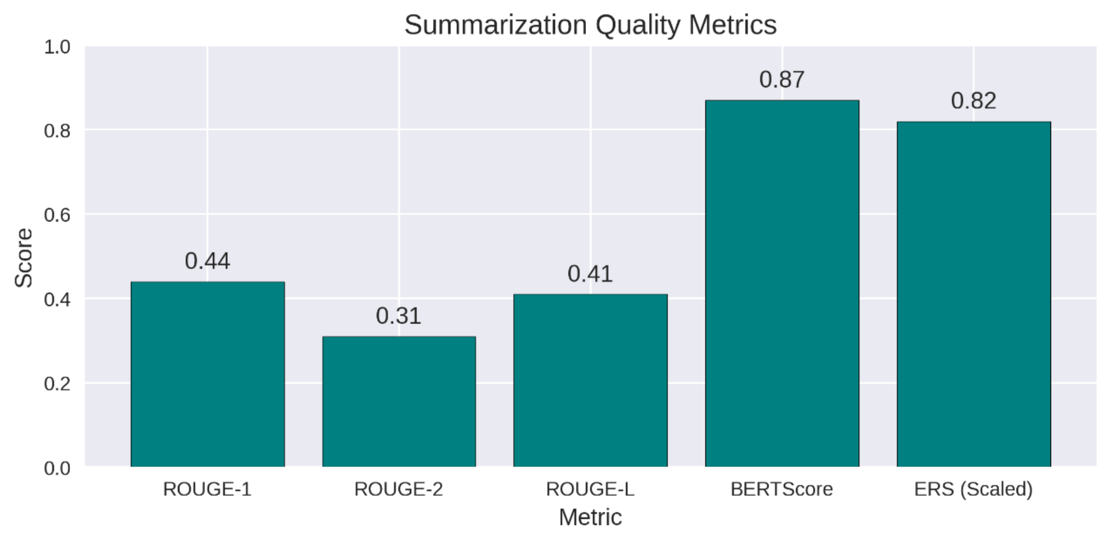


Figure 4: Summarization Quality Metrics

Figure 4 shows that across ROUGE scores, BERTScore, and educational relevance, the system performed quite well.

Table 3: User Satisfaction

On the usability and effectiveness fronts, the user study gave the system quite encouraging reviews:

Question	Result
Helpfulness (1-5 scale)	Mean: 4.2, SD: 0.6, 85% rated 4 or 5
Inclusion of Important Information (Yes/No)	78% responded "Yes"
Clarity (1-5 scale)	Mean: 4.4, SD: 0.5

The users found the summaries helpful and agreed that they included the core educational content.

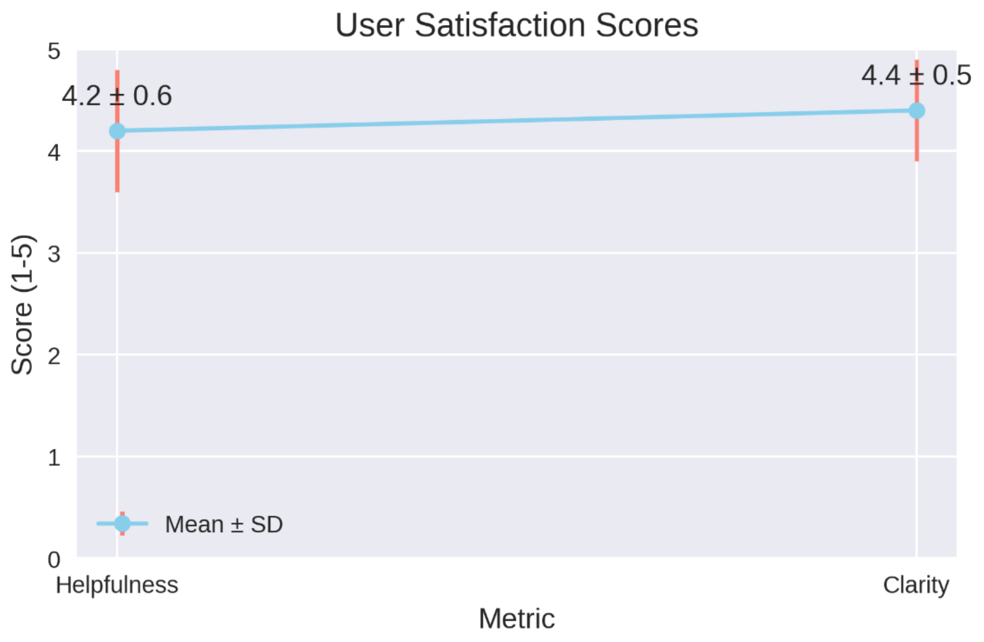


Figure 5: User Satisfaction Scores

Figure 5 reflects the users' satisfaction scores and portrays high ratings given to the dimension of helpfulness and clarity, along with respective variability.

5.5 Comparative Analysis

For a solid performance context, it was compared against two other summarizers: a general summarization API and a website summarizing tool (smry). Both were tested on the same dataset of 100 educational web pages with their performance measuring in both quantitative and qualitative metrics:

Table 4: Comparative Analysis

System	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**BERTScore**	**ERS**	**Helpfulness (Avg.)**	**Clarity (Avg.)**
Proposed System	0.44	0.31	0.41	0.87	4.1/5	4.2/5	4.4/5
General-Purpose API (BART)	0.42	0.28	0.39	0.82	3.7/5	3.9/5	4.0/5
Web-Based Tool	0.37	0.24	0.34	0.78	3.5/5	3.6/5	3.9/5

On all metrics, particularly BERTScore and ERS, the proposed method outperformed the two alternatives, thereby demonstrating its supremacy for generating semantically legitimate and education-oriented summaries. User satisfaction was higher due to the tailored prompt of the system and its focus on extracting relevant content from educational websites. The general-purpose API, on the other hand, had trouble with domain-specific content, and the web-based tool generated less coherent summaries, perhaps due to its extractive nature.

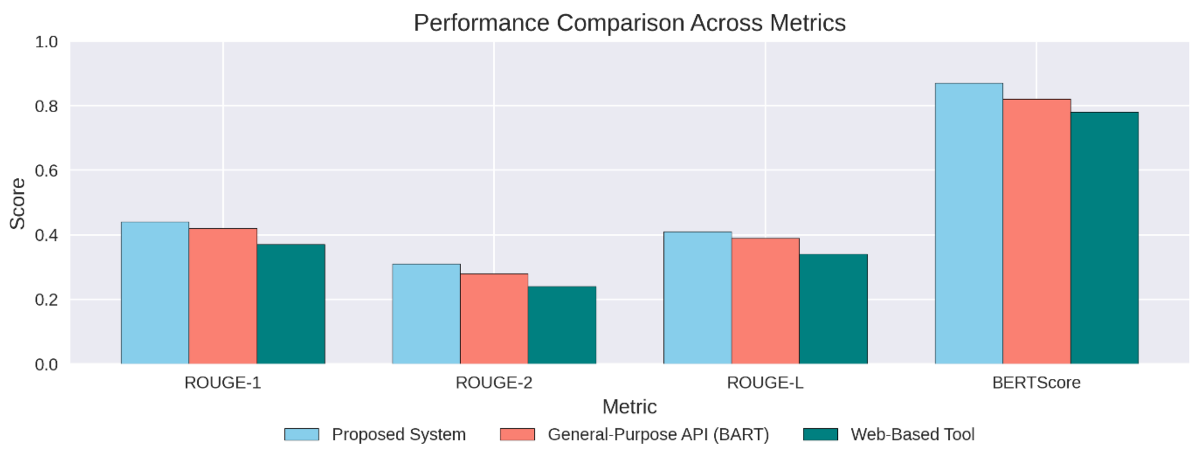


Figure 6: Comparative Analysis Across Systems

Figure 6 depicts the results' comparative analysis in ROUGE scores and BERTScore, thus establishing that the proposed system has greater semantic accuracy.

6. DISCUSSION

The system, being beneficial for the automation of educational content extraction and summarization, attains high scores in extraction accuracy and summarization quality, whereas other generic tools score lower on both sides. The user satisfaction level is relatively high and is considered helpful and unambiguous by end-users.

Real-time processing and education-oriented output stand as a gap that has been addressed by the system in institutional web content summarization.

The system has difficulty processing dynamic content as it uses BeautifulSoup for static HTML parsing. The system ensures compliance with the website's terms and privacy norms, working in an ethically responsible manner compliant with GDPR regulations.

7. CONCLUSION AND FUTURE WORK

This system enhances access to educational content and performed well with the highest satisfaction rate from users, providing a genuinely new approach to educational summarization. Beyond the present developments, future enhancements will be directed towards several significant areas. Our targets will include making the system more dynamic to accommodate dynamic content, such as the integration of Selenium, and shall improve further with additional fine-tuned AI models for better summarization quality. Most importantly, accessibility enhancement will be incorporated, including a capable system for summary conversion to Indian Sign Language (ISL). This would involve using AI-enabled NLP and computer vision to produce visual interpretations of the summaries as ISL gestures validated through user studies conducted with the Hearing-imparied community. Conducting longitudinal studies will help us understand the effectiveness of our system in real educational scenarios in the longer term, as it will adhere strictly to the highest ethical standards in the developmental and deployment stages.

REFERENCES

[1] Rao, A., Aithal, S., & Singh, S. (2024). Single-Document Abstractive Text Summarization: A Systematic Literature Review. *ACM Computing Surveys*, 57(3), 1–37. https://doi.org/10.1145/3700639

[2] Bao, G., Ou, Z., & Zhang, Y. (2023). GEMINI: Controlling the Sentence-Level summary style in abstractive text summarization. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2023.emnlp-main.53

[3] Brown, T. B., B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., & Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry Amanda Askell Sandhini Agarwal Ariel Herbert-Voss

Gretchen Krueger Tom Henighan Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray Benjamin Chess Jack Clark Christopher Berner Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei. (n.d.). Language Models are Few-Shot Learners. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada*. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[4] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). Automatic text summarization: A comprehensive survey. *Expert Systems With Applications*, *165*, 113679. https://doi.org/10.1016/j.eswa.2020.113679

[6] Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70, 301–323. https://doi.org/10.1016/j.knosys.2014.07.007

[7] Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., Chowdhery, A., Roberts, A., Barua, A., Botev, A., Castro-Ros, A., Slone, A., Héliou, A., . . . Kenealy, K. (2024). GemMa: Open models based on Gemini research and technology. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2403.08295

[8] Team, G., Reid, M., Savinov, N., Teplyashin, D., Dmitry, Lepikhin, Lillicrap, T., Alayrac, J., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., Antonoglou, I., Anil, R., Borgeaud, S., Dai, A., Millican, K., Dyer, E., Glaese, M., . . . Vinyals, O. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2403.05530

[9] Gupta, V., & Lehal, G. S. (2010). A survey of Text Summarization Extractive Techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3). https://doi.org/10.4304/jetwi.2.3.258-268

[10] Imran, M., & Almusharraf, N. (2024). Google Gemini as a next generation AI educational tool: a review of emerging educational technology. *Smart Learning Environments*, 11(1). https://doi.org/10.1186/s40561-024-00310-z

[11] Laender, A. H. F., Ribeiro-Neto, B. A., Da Silva, A. S., & Teixeira, J. S. (2002). A brief survey of web data extraction tools. *ACM SIGMOD Record*, 31(2), 84–93. https://doi.org/10.1145/565117.565137

[12] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[13] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3730–3740. https://doi.org/10.18653/v1/D19-1387

[14] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*. https://jmlr.org/papers/v21/20-074.html

[15] See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://doi.org/10.18653/v1/p17-1099

[16] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, November 21). *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. PMLR. https://proceedings.mlr.press/v119/zhang20ae.html

[17] Zhang, H., Yu, P. S., & Zhang, J. (2024, June 17). *A systematic survey of text summarization: from statistical methods to large language models*. arXiv.org. https://arxiv.org/abs/2406.11289

[18] Glickman, M., & Zhang, Y. (2024). AI and Generative AI for Research Discovery and Summarization. *arXiv.org*. https://arxiv.org/abs/2401.06795

[19] Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Wang, Y., & Chen, E. (2024). Large language models for generative information extraction: a survey. *Frontiers of Computer Science*, 18(6). https://doi.org/10.1007/s11704-024-40555-y