# TextSter: An all in one text processing website

Aviroop Paul
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar - 751024, India
apavirooppaul10@gmail.com

Pratyusa Mukherjee
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar - 751024, India
pratyusa.mukherjee@gmail.com

Swati Das
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar - 751024, India
dasswati155@gmail.com

Anushka Dash
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar - 751024, India
addash.012@gmail.com

Meghna
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar - 751024, India
imeghna2403@gmail.com

2024 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC) | 979-8-3503-7018-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/ASSIC60049.2024.10507995

*Abstract*—**Comprehensive text analysis websites have revolutionized text data management by providing users with powerful tools such as text summarizers, generators, emotion analyzers, and dictionaries. These tools employ sophisticated algorithms to process text input and extract vital details and insights. Text generators produce original content based on user-defined criteria, while text summarizers provide concise summaries of lengthy publications. Dictionaries assist in defining difficult terminology and determining emotional tones. We present TextSter, a user-friendly website, that combines these natural language processing features. Users can generate contextually relevant content, perform sentiment analysis, search word definitions, and quickly summarize large documents. By comparing various models and approaches for each task, we integrated the best-performing techniques into our comprehensive Natural Language Processing (NLP) tool. Future research on this aims to develop advanced algorithms for capturing linguistic subtleties, integrating Artificial Intelligence (AI) and Machine Learning (ML) systems, and creating specialized NLP tools for specific industries. Our study demonstrates the versatility of integrating the four NLP technologies, enabling effective text data analysis and management.**

***Keywords—NLP (Natural Language Processing) Tools, Text Summarisation, Text Generation, Sentiment Analysis, Dictionary***

## I. Introduction

We are living in the age of the internet, where massive amounts of data are being generated every day. As a result, the need for Natural Language Processing (NLP) tools has become increasingly important. NLP is a sub field of Artificial Intelligence that focuses on human interaction with computers using natural langauge.

A text summarizer is a tool that reduces long texts into shorter ones by picking out the most crucial details and essential points. Based on input criteria, text generators use algorithms to produce new, original content. Tools for sentiment analysis assist in identifying a text's emotional tone, such as whether it is favourable, negative, or neutral. Providing definitions, synonyms, antonyms, and other pertinent information about words and phrases, dictionaries are comprehensive reference tools. These four NLP technologies can effectively and efficiently manage text data, and their use in a variety of sectors and areas is becoming more and more significant.

A website that offers all four NLP features—text generator, sentiment analysis, dictionary, and summarizer—offers a complete set of tools for managing and studying text data. Thus, our website allows users to quickly and easily enter any text and access a variety of capabilities like creating new text based on predetermined criteria, analyzing the sentiment of the text, retrieving synonyms and dictionary definitions, and summarizing large texts. Users can digest text data in fresh and imaginative ways, which improves their insights and decision-making.

## II. Technical Definitions

Our website uses the following tools to build the features:

- Sumy : Sumy is a library that allows automatic summarization of text.
- NLTK : Python library for natural language processing called NLTK stands for Natural Language Toolkit.
- VADER : VADER is a Python package for sentiment analysis. It stands for Valence Aware Dictionary and Sentiment Reasoner (VADER).
- PyTorch : PyTorch is an open-source machine learning package for Python.
- Wordnet : WordNet is a lexical database for English language words.
- Flask : Flask is a Python micro-web framework.
- Transformers (GPT-2) : Transformers is a deep learning package for processing natural language (GPT-2).
- HTML and CSS : The two fundamental technologies of web development are HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets).

## III. Related Work

Bird, S., Klein, E., & Loper, E. et al. [1] introduced Natural Language Processing (NLP) using Python with a particular emphasis placed on text analysis using the Natural Language Toolkit (NLTK) that contains NLP techniques. It covers a range of text analysis topics, including part-of-speech tagging, stemming, and tokenization.

Al-Taani et al. [2] discussed methods for condensing lengthy texts into shorter summaries while discussing different approaches to automatic text summarizing. It focuses on automatic text summarization, which entails sifting through lengthy texts for the most important information and producing a succinct summary. Different strategies for completing this task, including as extractive and abstractive summarization techniques, have been investigated along with their uses and difficulties.

Mager, M., Astudillo, R. F., Naseem, T., Sultan, M. A., Lee, Y. S., Florian, R., & Roukos, S. et al. [3] proposed GPT-too, a language-model-first approach for Abstract Meaning Representation (AMR)-to-text generation, leveraging advanced language modeling techniques. It introduces Abstract Meaning Representation (AMR), a formalism used to express the meaning of natural language sentences, and GPT-too, a technique for producing human-readable text from AMR. Based on AMR input, GPT-too uses sophisticated language modeling approaches, such as the transformer architecture, to produce text that is logical and contextually relevant.

Liu et al. [4] highlighted subjectivity and sentiment analysis as a crucial topic of NLP, investigating methods to categorize text based on its sentiment and subjective content. It discusses sentiment analysis, which is figuring out the sentiment or opinion included in a specific text. It examines the difficulties with sentiment analysis, such as subjectivity and the existence of sarcasm or irony, and approaches for categorizing text into positive, negative, or neutral attitudes.

Hutto and Gilbert et al. [5] introduced VADER, a rule-based sentiment analysis model made especially for social media language that uses a lexicon-based method to establish sentiment polarity. It introduces VADER (Valence Aware Dictionary and Sentiment Reasoner), a model of sentiment analysis created especially for examining sentiment in content from social media. When determining the sentiment polarity of certain words and phrases, VADER uses a rule-based methodology and depends on a pre-defined lexicon while taking context and intensity into account.

Miller et al. [6] introduced WordNet, a thorough lexical database for the English language that offers a semantic network of words and their connections. It introduces WordNet, a well-known lexical database that groups words according to their relationships and meanings. With the use of associations such as synonyms, antonyms, hypernyms (broader terms), and hyponyms (narrower terms), WordNet offers a semantic network that connects words. It is an important tool for NLP applications like word sense disambiguation and semantic similarity analysis.

## IV. Methodology

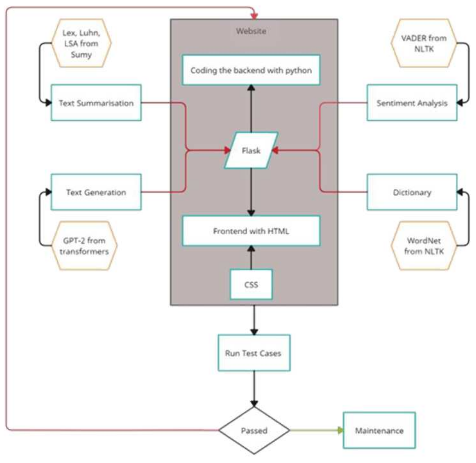


Fig. 1. Block Diagram of website methodology.

Fig. 1. shows the website methodology, and the integration of the different sub-sections to build the final product.

The sub-sections of the website are as follows:

### A) DEVELOPING THE BACKEND

This was by far the most critical part of the website as all the algorithms for the website including every function for Text Generation, Text Summarisation, Text Sentiment Analysis, and Dictionary were integrated with Flask [7]. Every function was modified to return the correct value as per Flask's in-built functions and send it back to the front end for a seamless connection between the front end and the back end. Routes were built for each of the 10 HTML pages, making sure the routes worked properly and no error was thrown, using effective error-handling techniques.

### B) BUILDING EACH NLP FEATURE

#### Text Generation

There were four approaches that were taken to implement the text generator:

1. Rule-based systems
2. Markov models
3. Recurrent neural networks (RNNs)
4. Transformers

However, Transformers (GPT-2) is what was chosen to develop the text generator.

The problems we had come across while using the other three approaches were as follows:

- Rule-based systems: These systems used predefined templates or grammar rules to generate text, but they lacked creativity and flexibility. The fixed rules made the code lengthy and hard to modify or adapt. Moreover, rule-based systems struggled with ambiguity and couldn't produce truly unique and diverse output.
- Recursive Neural Networks (RNNs): We employed Long Short-Term Memory (LSTM) networks, a type of Recursive Neural Networks (RNNs), to implement text generation. The LSTM networks captured long-term dependencies in the text and produced sensible outputs [8]. However, the generated text was limited to the specific dataset, resulting in a lack of creativity and coherence when generating longer texts compared to Markov models.

Reasons for choosing GPT-2 were that GPT-2 is a highly advanced language model trained on extensive text data. Its deep neural network architecture enables it to generate coherent and contextually relevant text with a wide range of variations. GPT-2's adaptability in generating text in multiple languages and styles makes it a reliable and powerful choice for our text generator.

After deciding on using GPT-2, we implemented sentiment based text generation so that texts of positive, negative and neutral sentiments can be generated for a given prompt.

A prompt to specify the length of the generated text was implemented such that the user has further control on the text.

Technical challenges faced:

- Finding out the Top-p (nucleus sampling) and Top-k values for the GPT-2 model that resulted in an optimized output.
- Improving the quality of outputs and selecting the appropriate model to address the issue.

Solutions found:

- Trial-and-error method for changing both top-p and top-k values that generated desired results.
- Top-p value was set to 0.95.
- Top-k value was set to 50.

## Text Summarisation

Building the summarise function was tricky as the choice of the algorithms was critical in the result generated by it and its accuracy [9]. After a lot of deliberation, we decided to go with extractive summarisation as abstractive summarisation, although more advanced, is a challenging task to implement and is still under research.

The following three summarization functions were implemented:

- Lex Summariser is a graph-based technique utilizing the PageRank algorithm to identify key sentences in documents. It treats each sentence as a graph node, calculating their similarity using cosine similarity. Notably, LexRank is advantageous as it requires no training data or human input and can be applied to various text types.
- Luhn Summariser is a heuristic algorithm that identifies important sentences in a document by analyzing the frequency of key words [10-12]. It assigns scores to sentences based on the occurrence of significant words, with higher scores indicating more important sentences. Luhn's simplicity and effectiveness in generating coherent and relevant summaries make it a valuable tool. However, it is not as powerful as LSA or Lex Summarizer.
- LSA Summariser uses Singular Value Decomposition (SVD) to identify key concepts in a document. It creates a word-frequency matrix, decomposes it with SVD to uncover underlying topics. LSA captures synonyms and related words, reducing redundancy in the summary and improving accuracy.

Technical challenges faced: □ □

- Using Flask to integrate the backend with the frontend was a difficult task. Often, the algorithm was working but returning the result threw errors.
- Dealing with null value errors, bad routes, and return type errors.

Solutions found:

- Writing code to handle null value errors and correct routing issues.
- Using error handling techniques identify and fix return type errors.
- Using Flask documentation and tutorials to understand how to integrate the front end with the back end.

## Sentiment Analysis

VADER, short for Valence Aware Dictionary for Sentiment Reasoning, utilizes the SentimentIntensityAnalyzer class within the nltk.sentiment library. To generate sentiment scores, VADER relies on a comprehensive lexicon containing 7500 features, employing advanced linguistic rules and considering both the polarity and intensity of textual content. This lexicon assigns predefined scores on a scale from -4 (extremely negative) to 4 (extremely positive) to words, acronyms, and emoticons, with 0 denoting neutrality.

The compound score is computed by summing the valence ratings of each feature, applying adjustments based on specific rules, and then normalizing the score within the range of -1 (highly negative) to +1 (highly positive) as outlined in Equation (1).

$$x = \frac{x}{(x^2 + a\ )^{1/2}} \tag{1}$$

where x = summation of valence scores of constituent words, and α = constant of normalisation (default value is 15).

For example:

Input text: 'Today is a good day.'

Output: {'neg': 0.0, 'neu': 0.58, 'pos': 0.42, 'compound': 0.4404}

TABLE I. shows the sentiment score calculation for the above example:

TABLE I. CALCULATION OF SENTIMENT SCORE

| | pos | neg | neu | total |
|---|---|---|---|---|
| today | | | 0 + ***1*** | normalizing function: $\frac{x}{\sqrt{x^2 + \alpha}}$ |
| is | | | 0 + ***1*** | |
| a | | | 0 + ***1*** | |
| good | 1.9 + ***1*** | | | |
| day | | | 0 + ***1*** | |
| | 2.9 | | 4 | 6.9 |
| | 2.9 / 6.9 | 0 / 6.9 | 4 / 6.9 | 1.9 / ((1.9^2) + ***15*** )^0.5 |
| | 0.42 | 0 | 0.58 | 0.44 |
| +1 to compensate for neutral words | | | | |
| 15 is the approximate max sentiment score | | | | |

## Dictionary

Challenges were faced while implementing the NLTK and WordNet modules. When dealing with the various word senses and meanings in WordNet, context and disambiguation were crucial factors to take into account. WordNet produced part-of-speech (POS) tags, but they were in shortened form, making it difficult to read and comprehend them. Pos_full_form() and if-else loops were used to accomplish this translation [13].

WordNet's compatibility with Flask led to its selection over the original API key technique. It provides a vast database of English words and their definitions, making it simple to locate synonyms and definitions. Advanced information retrieval and NLP tasks are made possible by WordNet's hierarchical structure and hyponymy linkages. It offers a dependable and well-structured interface for gaining access to lexical and semantic data about words. Additionally, its extensive use within the NLP community and open-source nature offer simple access to knowledge and assistance. Fig. 2. demonstrates the process of POS Tagging.

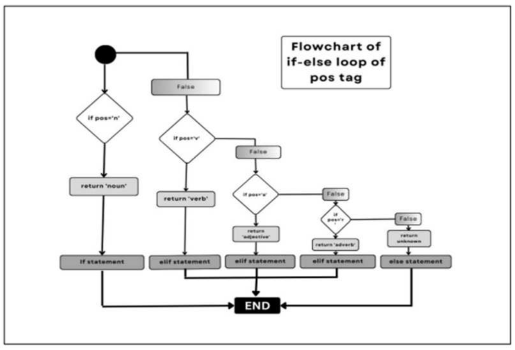


Fig. 2. Flowchart of parts of speech (POS) Tagging

## C) DEVELOPING THE FRONTEND

The front-end of the website was developed using HTML and CSS. Each page was designed for Text Summarisation, Text Generation, Text Sentiment Analysis, and Dictionary in a user-friendly way to help users easily understand the purpose of each function and how to use it effectively. Along with the pages for the features, the landing page of our website was designed using modern CSS to blend it with all the feature pages. Every feature page contained their respective result pages to display the result obtained after running the algorithms in the back end.

## D) PERFORMANCE TESTING

### TEXT GENERATION

The analysis was done with the text generator based on GPT-2. To analyse the relationship between text length and generation time, a mix of factual questions, open ended phrases and abstract topics were chosen [14-15]. The following four prompts were selected:

- P1: 'What is F1?'
- P2: 'What is machine learning?'
- P3: 'Once upon a time'
- P4: 'What would be the consequences of discovering a parallel universe?'

A table was made based on the time the text generator took to generate the input length of text. The average time was calculated and a graph was plotted between the length of generated text and average time taken. TABLE II. shows the time taken for various lengths of text.

TABLE II. GENERATED TEXT LENGTH AND TIME TAKEN

| LENGTH OF GENERATED TEXT (in words) | TIME TAKEN (seconds) | | | | |
|---|---|---|---|---|---|
| | P1 | P2 | P3 | P4 | AVERAGE |
| 50 | 2.2 | 1.92 | 2.13 | 1.72 | 1.20 |
| 100 | 4.86 | 4.28 | 5.12 | 4.02 | 4.57 |
| 150 | 7.68 | 7.42 | 8.03 | 6.18 | 7.33 |
| 200 | 12.43 | 13.22 | 13.47 | 9.8 | 12.23 |
| 250 | 17.69 | 15.27 | 17.49 | 11.75 | 15.55 |
| 300 | 21.05 | 24.14 | 20.81 | 18.02 | 21.01 |

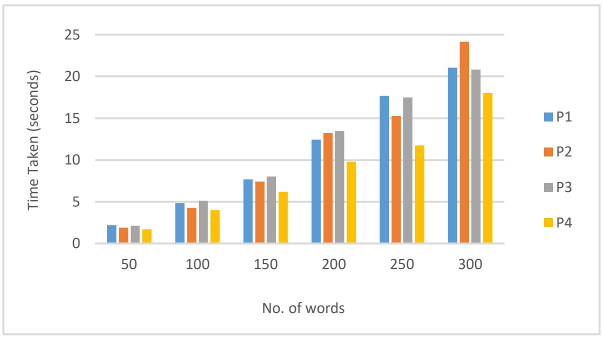


Fig. 3. Graph of no. of words vs. time taken in seconds.

We can clearly see from the graph in Fig. 3. that when the prompt given by the user is more detailed, it takes lesser time to generate the text of the required word limit as given by the user.

The average time of time to generate the text based on 50 words increment was also found out to be an average of 4 seconds, however, it depends on the complexity, and length of the prompt given by the user.

### TEXT SUMMARISATION

Text summarisation is a crucial task in natural language processing that aims to reduce the length of a given text while preserving its most important information. In this study, we tested the performance of three text summarisation approaches - LexRank, LSA, and Luhn - on ten different paragraphs. The paragraphs were selected to be diverse in terms of length, content, and writing style. We started with a paragraph of length 50 words and increased it by 25 words increment each time, resulting in the final paragraph's length being 275 words.

The summarisation process was carried out using the Python programming language and its related libraries. For each summariser, we evaluated its performance in terms of recall, precision, and F1 score. Among the three approaches, LSA consistently outperformed the others, achieving the highest F1 score across all the paragraphs. This result is consistent with previous studies that have shown LSA to be a highly accurate text summarisation approach. LexRank and Luhn, on the other hand, showed lower but still reasonable performance, with varying levels of success across the different paragraphs.

F-1 score calculated by:

F1 score = 2 *((precision * recall)/ (precision + recall))

where precision is the ratio of true positive predictions to the total predicted positives, and recall is the ratio of true positive predictions to the total actual positives. Table III. shows the time taken for various lengths of text.

TABLE III. GENERATED TEXT LENGTH AND TIME TAKEN

| Text | LSA | Lex | Luhn |
|---|---|---|---|
| 1 | 0.83 | 0.72 | 0.64 |
| 2 | 0.79 | 0.68 | 0.52 |
| 3 | 0.69 | 0.66 | 0.49 |
| 4 | 0.82 | 0.71 | 0.55 |
| 5 | 0.84 | 0.73 | 0.61 |
| 6 | 0.6 | 0.69 | 0.48 |
| 7 | 0.85 | 0.75 | 0.53 |
| 8 | 0.83 | 0.7 | 0.51 |
| 9 | 0.86 | 0.76 | 0.59 |
| 10 | 0.8 | 0.67 | 0.46 |

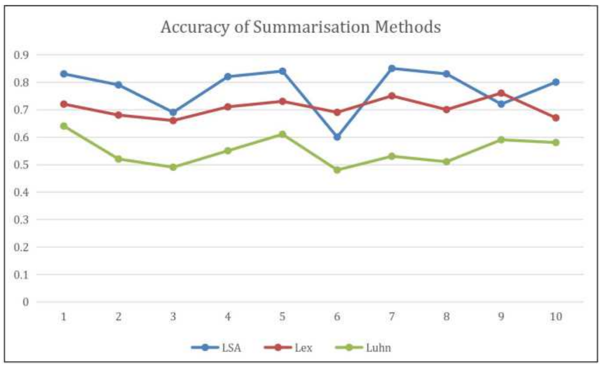


Fig. 4. Performance of different summarisation algorithms.

Fig. 4. shows the accuracy of the three summarisation methods. The scores in the dataset were calculated based on a human summarised text of the sample inputs, where the most important parts of the sentence were considered. F1 score was used as the evaluation metric, which is a measure of the precision and recall of a summarization model. The higher the F1 score, the better the model is at generating accurate and concise summaries of the input text.

Overall, the results of this study demonstrate the importance of choosing an appropriate text summarisation approach that aligns with the specific goals and requirements of a given task. In the case of our NLP text processing website, LSA would be the preferred summarisation approach due to its high accuracy in summarising diverse paragraphs. However, LexRank and Luhn could still be useful in certain scenarios where speed and simplicity are more important than accuracy. Future studies could investigate the performance of other summarisation approaches, as well as explore the potential of combining multiple approaches to improve summarisation performance even further.

### Sentiment Analysis

For testing the sentiment analysis of our website, we used texts of varying length, complexity and tones. We started our testing with a text of length 5 and gradually increased it in 5 word increments until the size of 25. Table IV. was made with the time taken for each length and a graph was plotted as shown in Fig. 5. to analyse the performance of the sentiment analysis feature.

TABLE IV. NO. OF WORDS VS. TIME TAKEN IN SECONDS

| Words | Time TAKEN (seconds) |
|---|---|
| 5 | 1 |
| 10 | 1.5 |
| 15 | 2 |
| 20 | 3 |
| 25 | 5 |

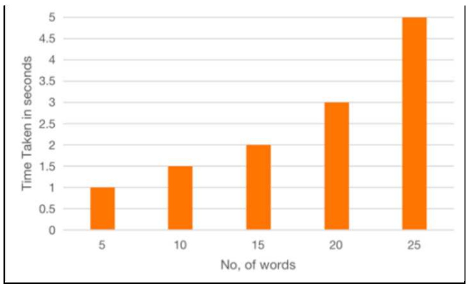


Fig. 5. Performance of sentiment analysis.

We can clearly infer that the sentiment analysis performs faster with lesser no. of words as compared to longer sentences. This is due to the fact that the algorithm is doing more checks for the correct sentiment in case of more no. of words to deduce the accurate tone of the sentence.

To further test the sentiment analysis feature, we took 5 sentences, consiting of words which are double negatives, positives, negatives and neutral. TABLE V illustrates the following data in a tabular format.

TABLE V. COMPLEXITY TESTING OF SENTIMENT ANALYSIS.

| Sentence | Compound Score | Sentiment Generated |
|---|---|---|
| He is a good student | 0.8808 | Positive |
| He is a bad student | -0.10846 | Negative |
| He is a good student but a bad singer | -0.01132 | Negative |
| He is a good student and great dancer but bad singer | 0.04754 | Positive |
| He is a good student but awful dancer and bad singer | -0.050972 | Negative |

We can infer from the above table that when the algorithm finds two negatives over one positive, it outputs a negative sentiment. Similarly, when it finds two positives over one negative, it outputs a positive sentiment. We can also observe that when a negative word is used to end the sentence, it deems the sentence to be negative.

When we observe the scores of the algorithms, we see that for sentence 3, it outputs a score closer to 0 but on the negative side, which means it finds the sentence to be almost neutral. Similarly, the scores which are closer to -1 indicate that they are negative, scores which are closer to 0 indicate they are of neutral sentiment and scores which are closer to +1 indicate that they are of positive sentiment.

### Dictionary

For testing the Dictionary feature of our website, we made sure that every word's Part-of-speech (POS) tagging is handled correctly. We removed ambiguities in the POS tags of certain words and tested our dictionary with close to 500 words through a random word generator.

We also tested the Dictionary feature by giving null values, incorrect words, spaces and emojis as input. We handled these test cases such that the website did not break upon receiving invalid input from the user.

## V. Implementation and Results

Fig. 6 shows the landing page of our website, followed by Fig. 7. and Fig. 8. which show the Text Summariser functionality. Fig 9. and Fig. 10. show the Text Generation feature. Next, Fig. 11, and Fig. 12 show the Sentiment Analyser feature, and finally Fig. 13, and Fig. 14 show the dictionary functionality. Lastly, the about page of the website is shown in Fig. 15.

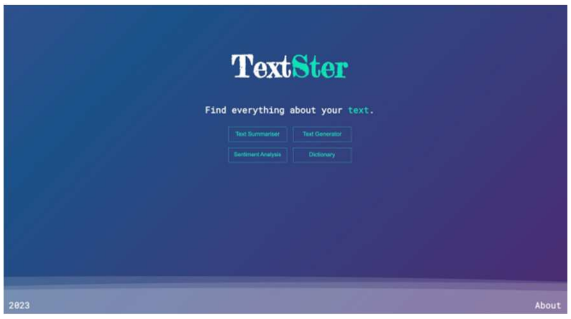
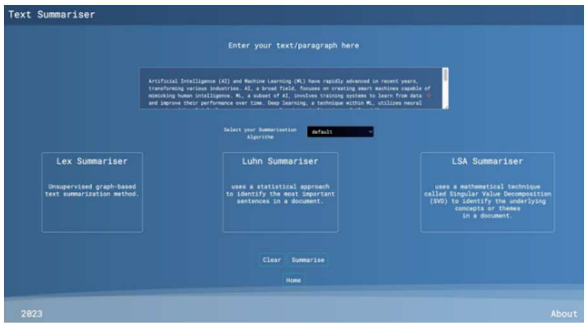


Fig. 6. Landing page of TextSter.


Text Summariser
Enter your text/paragraph here
Select your Summarisation Algorithm
default
Clear
Summarise
Home
2023
About

Artificial Intelligence (AI) and Machine Learning (ML) have rapidly advanced in recent years, transforming various industries. AI, a broad field, focuses on creating smart machines capable of mimicking human intelligence. ML, a subset of AI, involves training systems to learn from data and improve their performance over time. Deep learning, a technique within ML, utilizes neural

Lex Summariser

Unsupervised graph-based text summarization method.

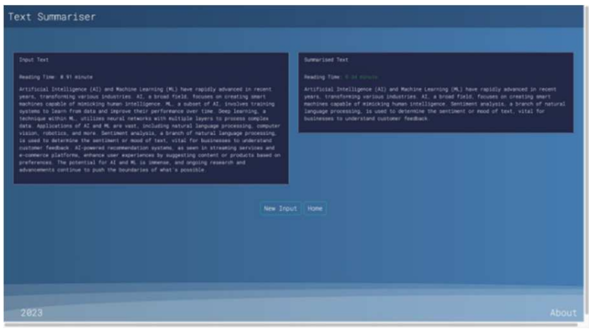
Luhn Summariser

uses a statistical approach to identify the most important sentences in a document.

LSA Summariser

uses a mathematical technique called Singular Value Decomposition (SVD) to identify the underlying concepts or themes in a document.

Fig. 7. Text Summariser home page.



Input Text

Reading Time: 0.91 minute

Artificial Intelligence (AI) and Machine Learning (ML) have rapidly advanced in recent years, transforming various industries. AI, a broad field, focuses on creating smart machines capable of mimicking human intelligence. ML, a subset of AI, involves training systems to learn from data and improve their performance over time. Deep learning, a technique within ML, utilizes neural networks with multiple layers to process complex data. Applications of AI and ML are vast, including natural language processing, computer vision, robotics, and more. Sentiment analysis, a branch of natural language processing, is used to determine the sentiment or mood of text, vital for businesses to understand customer feedback. AI-powered recommendation systems, as seen in streaming services and e-commerce platforms, enhance user experiences by suggesting content or products based on preferences. The potential for AI and ML is immense, and ongoing research and advancements continue to push the boundaries of what's possible.

Summarised Text

Reading Time: 0.34 minute

Artificial Intelligence (AI) and Machine Learning (ML) have rapidly advanced in recent years, transforming various industries. AI, a broad field, focuses on creating smart machines capable of mimicking human intelligence. Sentiment analysis, a branch of natural language processing, is used to determine the sentiment or mood of text, vital for businesses to understand customer feedback.

Fig. 8. Text Summariser result page.

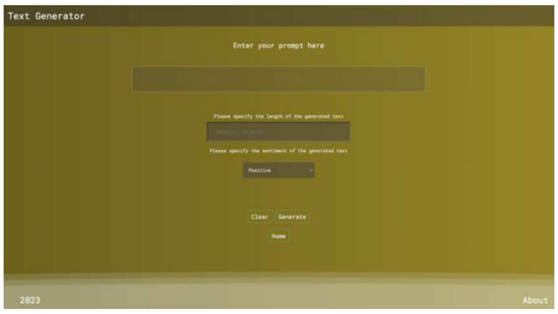
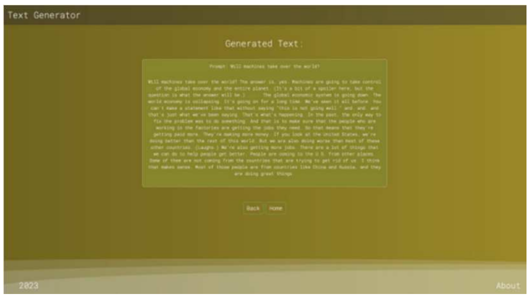

Text Generator
Enter your prompt here
Please specify the length of the generated text
Please specify the sentiment of the generated text
Positive
Clear
Generate
Home
2023
About

Fig. 9. Text Generator home page.



Prompt: Will machines take over the world?

Will machines take over the world? The answer is, yes. Machines are going to take control of the global economy and the entire planet. (It's a bit of a spoiler here, but the question is what the answer will be.) The global economic system is going down. The world economy is collapsing. It's going on for a long time. We've seen it all before. You can't make a statement like that without saying "this is not going well," and, and, and that's just what we've been saying. That's what's happening. In the past, the only way to fix the problem was to do something. And that is to make sure that the people who are working in the factories are getting the jobs they need. So that means that they're getting paid more. They're making more money. If you look at the United States, we're doing better than the rest of this world. But we are also doing worse than most of these other countries. (Laughs.) We're also getting more jobs. There are a lot of things that we can do to help people get better. People are coming to the U.S. from other places. Some of them are not coming from the countries that are trying to get rid of us. I think that makes sense. Most of those people are from countries like China and Russia, and they are doing great things.

Fig. 10. Text Generator result page.

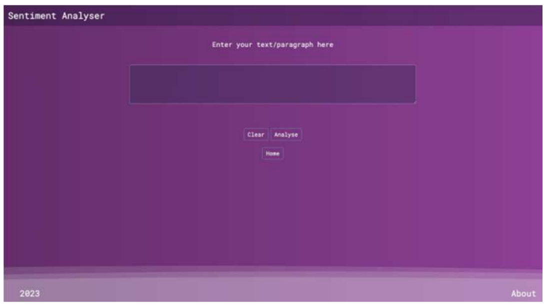
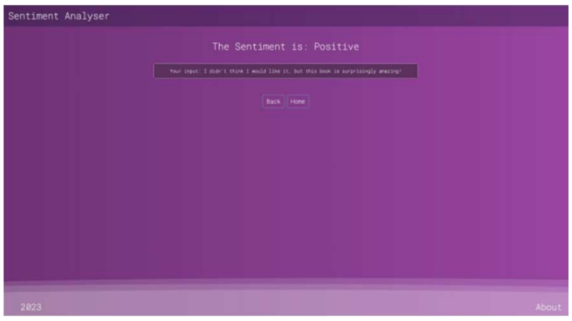


Fig. 11. Sentiment Analyzer home page.



Your input: I didn't think I would like it, but this book is surprisingly amazing!

Fig. 12. Sentiment Analyzer result page.

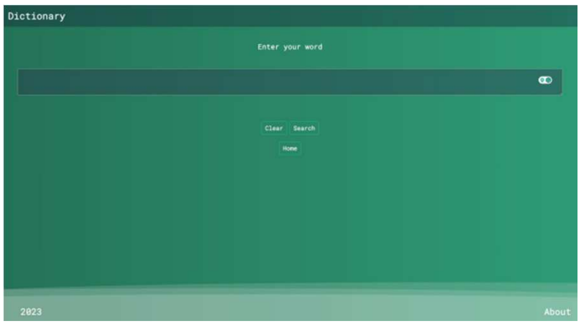
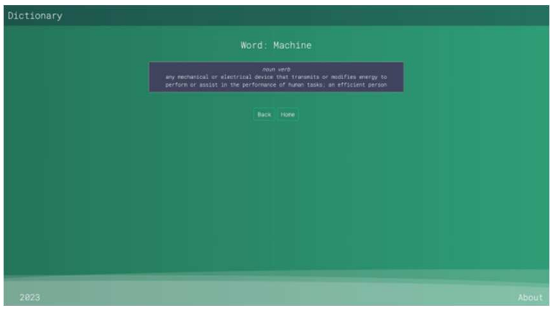
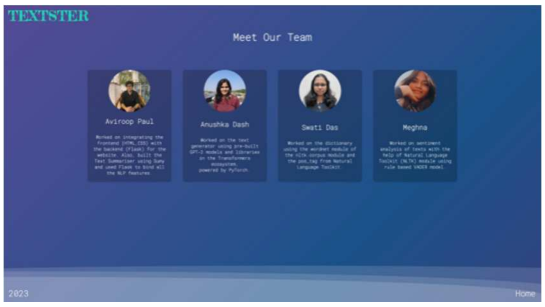


Fig. 13. Dictionary home page.



noun verb
any mechanical or electrical device that transmits or modifies energy to perform or assist in the performance of human tasks; an efficient person

Fig. 14. Dictionary result page.



Aviroop Paul

Worked on integrating the Frontend (HTML,CSS) with the backend (Flask) for the website. Also, built the Text Summariser using Sumy and used Flask to bind all the NLP features.

Anushka Dash

Worked on the text generator using pre-built GPT-2 models and libraries in the Transformers ecosystem, powered by PyTorch.

Swati Das

Worked on the Dictionary using the wordnet module of the nltk.corpus module and the pos_tag from Natural Language Toolkit.

Meghna

Worked on sentiment analysis of texts with the help of Natural Language Toolkit (NLTK) module using rule based VADER model.

Fig. 15. TextSter About page.

## VI. Conclusion and Future Work

This research paper delves into fundamental NLP technologies, encompassing text summarization, text generation, sentiment analysis, and dictionary search, tackling the formidable task of managing and comprehending extensive textual data. The paper underscores the prowess of GPT-2 as a robust text generator and emphasizes the incorporation of WordNet for efficient lexical information retrieval. The application of text generation translates into notable time and cost savings, while sentiment analysis provides indispensable insights derived from customer feedback. The amalgamation of these technologies presents a versatile solution for proficient text analysis, holding the potential to transform language and information interaction.

Furthermore, the research paper sheds light on the future prospects of NLP tools such as text summarization, text generation, sentiment analysis, and dictionary lookup. GPT-2 is anticipated to undergo advancements in industry-specific adaptations and multilingual text generation, affording greater control over the generated content. Sentiment analysis, coupled with tools like VADER, shows promise across diverse domains including marketing, customer service, politics, law, and healthcare. WordNet is poised to broaden language support and seamlessly integrate with advanced NLP techniques. The fusion of NLP with AI and ML paves the way for specialized tools tailored to specific industries. The field of NLP holds exciting and transformative opportunities on the horizon.

## VII. References

[1] Bird, S., Klein, E., & Loper, E. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009).

[2] Al-Taani, A. T. Automatic text summarization approaches (2017, December).

[3] Mager, M., Astudillo, R. F., Naseem, T., Sultan, M. A., Lee, Y. S., Florian, R., & Roukos, S. GPT-too: A language-model-first approach for AMR-to-text generation. arXiv preprint arXiv:2005.09123 (2020).

[4] Liu, B. Sentiment analysis and subjectivity. Handbook of natural language processing, 2(2010), 627-666 (2010).

[5] Hutto, C., & Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the international AAAI conference on web and social media (Vol. 8, No. 1, pp. 216-225) (2014).

[6] Miller, G. A. WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41 (1995).

[7] Abhishek, Tripathy, H. K., & Mishra, S. (2022). A Succinct Analytical Study of the Usability of Encryption Methods in Healthcare Data Security. In Next Generation Healthcare Informatics (pp. 105-120). Singapore: Springer Nature Singapore.

[8] Suman, S., Mishra, S., & Tripathy, H. K. (2021). A Support Vector Machine Approach for Effective Bicycle Sharing in Urban Zones. In Cognitive Informatics and Soft Computing: Proceeding of CISC 2020 (pp. 73-83). Springer Singapore.

[9] Dutta, S., Choudhury, S., Chakraborty, A., Mishra, S., & Chaudhary, V. (2023, February). Parkinson Risks Determination Using SVM Coupled Stacking. In International Conference On Innovative Computing And Communication (pp. 283-291). Singapore: Springer Nature Singapore.

[10] Explaining Text Generation with LSTM: https://www.analyticsvidhya.com/blog/2022/02/explaining-text-generation-with-lstm/, last accessed 2023/07/13.

[11] Gaber, T., Tharwat, A., Snasel, V., Hassanien, A.E. (2015). Plant Identification: Two Dimensional-Based Vs. One Dimensional-Based Feature Extraction Methods. In: Herrero, Á., Sedano, J., Baruque, B., Quintián, H., Corchado, E. (eds) 10th International Conference on Soft Computing Models in Industrial and Environmental Applications. Advances in Intelligent Systems and Computing, vol 368. Springer, Cham. https://doi.org/10.1007/978-3-319-19719-7_33.

[12] Tharwat, A., Gaber, T., Fouad, M.M., Snášel, V., & Hassanien, A.E. (2015). Towards an Automated Zebrafish-based Toxicity Test Model Using Machine Learning. Procedia Computer Science, 65, 643-651.

[13] Gaber, T., El Jazouli, Y., Eldesouky, E., & Ali, A. (2021). Autonomous Haulage Systems in the Mining Industry: Cybersecurity, Communication and Safety Issues and Challenges. Electronics.

[14] G. I. Sayed, M. A. Ali, T. Gaber, A. E. Hassanien and V. Snasel, "A hybrid segmentation approach based on Neutrosophic sets and modified watershed: A case of abdominal CT Liver parenchyma," 2015 11th International Computer Engineering Conference (ICENCO), Cairo, 2015, pp. 144-149, doi: 10.1109/ICENCO.2015.7416339.

[15] Applebaum, S., Gaber, T., & Ahmed, A. (2021). Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey. International Conference on Arabic Computational Linguistics.