NLP in Business Analytics: Generating Insights from Textual Data using NLP Models

Priyanshu Mishra
Department of AI
G H Raisoni College of Engineering
Nagpur, India
priyanshu19mishra@gmail.com

Ketki Ninawe
Department of AI
G H Raisoni College of Engineering
Nagpur, India
ketkininawe1@gmail.com

Krupali Dhawle
Department of AI
G H Raisoni College of Engineering
Nagpur, India
krupali.acem@gmail.com

Sumit Prasad
School of Management Studies and Commerce,
Uttarakhand Open University
Haldwani,
sprasad@uou.ac.in

Anil Band
Department of CSE
Schools of Engineering
G.H.Raisoni University
Amravati,India
anil7band@gmail.com

Parul Dubey
Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University),
Pune, India
dubeyparull29@gmail.com

***Abstract*— The following study aims to develop and finetune an unified platform that includes Natural Language Processing applications like Sentiment detection, Information extraction, and Text summarization to ease the analysis job in the business world which involves textual data. The motivation for this research is to emphasize the necessity of a platform that works on textual data using Natural Language Processing models, which helps in business analytics through insights gain and pictorial representation of data, enabling better decision-making and understanding of market trends. This platform will be utilized under academic institutions, allowing business students and educators to get valuable insights, such as pictorial representation, by seeing the limitations on the use of chatbots in many educational institutions developing a unified and task-oriented platform that works under the institutional and learning guidelines is the main motivation for this research. This platform helps in generating graphs and charts that makes it easy to understand the required patterns that are more understandable and that is easily accessible to draw conclusions from the textual data which increases the approach beyond numerical data for business analysis.**

Keywords— Sentiment detection, Information extraction, Text summarization, NLP- Natural language processing, Business analytics

I. Introduction

Business analytics refers to the analysis of data using statistical models and various quantitative methods to gain insights, make informed decisions, and improve business performance. It provides meaningful insights for better decision-making, correct identification of trends, and efficiency improvement, giving companies a competitive edge and enhancing customer understanding. It also supports risk management and performance measurement. Business analytics works on a huge amount of data which is mostly numerical data, which further gets converted into meaningful insights. Business analytics lags in analyzing unstructured data that is text format, which includes reviews from various customers on various products and services, posts on various social media platforms, emails, etc. Most of the recent methodology works well on numerical data in comparison to their performance over textual data. To handle this huge pile of unstructured textual data, NLP is used. It addresses this by applying algorithms to process and interpret unstructured textual data, extracting meaningful insights such as sentiment analysis, information extraction, and text summarization. This research based on business analytics, highlights the need of an unified, task-oriented platform that generates meaningful insights from the textual data using different NLP models and generating meaningful and specified graphs and charts to generate business reports.

This research aims to provide an integrated platform consisting of three NLP-fragmented tasks: Sentiment detection, Information extraction, and Text summarization. These models are seamlessly connected to visualization tools, offering a comprehensive solution for diverse users ranging from academics to professionals. All three models develop different dedicated graphs and charts on the basis of insight gained from a large amount of data. The user-friendly interface ensures that individuals with varying levels of technical expertise can efficiently navigate and utilize the platform's features. This adaptive and scalable platform is dedicated to continuous improvement, integrating the fine-tuned models to deal with the load of the users' data without compromising the complexities.

Through this implementation, textual data, i.e., raw data, will be converted into meaningful insights with the help of the models which gives insights by performing dedicated NLP operations. These insights are then seamlessly converted into pictorial representations, i.e., tables, graphs, and images, using sophisticated visualization tools, which will further help in business analytics which make it easier and more compatible.

This is a task-oriented platform, which is mainly focused on business students. This research is under the academic

979-8-3315-0526-4/24/$31.00 ©2024 IEEE
DOI: 10.1109/CICN.2024.244

institution guidelines where chatbots are strictly banned (or against academic norms). While working on big data, business analytics can be hampered by high storage requirements and slow processing speeds. This implementation works on both shortcomings, i.e., time complexity and space complexity. In this, we convert a large amount of data, i.e., big data, into meaningful insights, which automatically reduces the space complexity. Rather than big data, when these meaningful insights are converted into pictorial representation, the time taken to execute will be less.

II. LITERATURE SURVEY

This study takes many other research studies into consideration to align various applications. Mishra et al. [1] proposed the architecture of a uniform platform to perform various NLP tasks, which opened the idea of the same in business analytics. Zhang et al. [2] proposed the PEGASUS model, a transformer-based approach designed for abstractive summarization. By pretraining with extracted gap sentences, a BLEU score of 45.1 was achieved, demonstrating the effectiveness of this approach. Brown et al. [3] utilized the GPT-3 model for sentiment analysis, achieving 90% accuracy on benchmark datasets. Fine-tuned BERT-based models [4] were trained for aspect extraction which attained 85% as F1 score, highlighting the utility of transformers in this domain. Xu et al. [5] explored Graph Neural Networks (GNN) for information extraction, achieving 88% accuracy, explaining the potential of GNNs in understanding textual relationships. Dai et al. [6] introduced a sentiment-aware text summarization technique that integrates sentiment analysis with abstractive summarization. This achieved a ROUGE-L score of 39.5. This study explains the importance of sentiment in creating contextually accurate summaries. Kumar et al. [7] demonstrated improvements in contextual relevance through multi-modal embeddings, which combined textual and visual data for more comprehensive insights. This study strengthened the context of visualization for our proposed architecture. Lee et al. [8] employed reinforcement learning for text summarization. This approach achieved 92% naturalness in human evaluations, ensuring better coherence and readability in generated summaries. Rajput et al. [9] addressed the challenge of limited training data in low-resource languages by tailoring transformers, which resulted in an F1 score of 78%. Banerjee et al. [10] focused on disaster tweet detection using NLP models. This achieved 87% precision, and showcased the ability to extract meaningful information from social media platforms. Sun et al. [11] conducted a detailed analysis of pretrained models under "BERTology," identifying key limitations in multitasking environments, which provided insights into improving NLP systems. These studies have proved to be the foundational finding for this research. Various models are implemented for the same applications. The results used architecture along with the results from the various studies has been listed in the table 1 below. The following are used to compare the performance of the architectures used in this study with the exciting architectures.

TABLE I. RESULTS OF SIMILAR STUDIES OVER RESPECTIVE NLP APPLICATION

S. No.	Authors	Year	Model/Technique	Results
1	Zhang, J., Zhao, Y., Saleh, M., Liu, P. [2]	2020	PEGASUS: Transformer for Summarization	Achieved BLEU score of 45.1
2	Brown, T., Mann, B., Ryder, N., et al. [3]	2020	GPT-3 for Sentiment Analysis	Achieved 90% accuracy on benchmark data
3	Devlin, H., Chang, M.-W., Lee, K., Toutanova, J. [4]	2021	Fine-tuned BERT for Aspect Extraction	Achieved 85% F1 score
4	Xu, C., Li, J., Zhou, P. [5]	2021	Graph Neural Networks for Information Extraction	Achieved 88% accuracy
5	Dai, F., Wang, X., Zhang, R. [6]	2022	Sentiment-aware Text Summarization	Achieved ROUGE-L score of 39.5
6	Kumar, R., Gupta, S., Sharma, M. [7]	2022	Multi-modal Embeddings for NLP	Improved contextual relevance
7	Lee, S., Kim, J., Park, Y. [8]	2023	Reinforcement Learning for Summarization	Human evaluation: 92% naturalness
8	Rajput, P., Singh, K., Verma, A. [9]	2023	Transformers for Low-resource Languages	Achieved F1 score of 78%
9	Banerjee, A., Das, R., Iyer, N. [10]	2024	NLP models for Disaster Tweet Detection	Achieved 87% precision
10	Sun, Z., Gao, J., Chen, T. [11]	2024	BERTology: Pretrained Models Analysis	Identified limitations in multi-tasking

III. OBJECTIVE

The objective of this research is to transform all the raw data of textual form into meaningful visual representations by integrating various NLP tools, including Sentiment detection, Information extraction, and Text summarization, into a unified, task-oriented platform. This approach enhances efficiency and utility in both business and academic fields by offering a comprehensive solution for analyzing and visualizing textual data beyond numerical values. In the business domain, it facilitates better decision-making and data interpretation through graphical representations of business raw data, making analytics more compatible and accessible. In the academic domain, this implementation is under institutional guidelines, addressing the limitations of chatbots and providing useful insights that meet specific educational needs. By leveraging NLP this research aims to bridge the gap between complex data processing and user-friendly visualization, ultimately improving the overall analytical capabilities and usability for diverse users.

IV. System Architecture

The backend applies user interaction with the interface and gives the inputs for the models given as a choice. These models analyze textual data to generate insights such as sentiment classification, keyword extraction, and summary generation on the processed data. The insights are then visualized using visual representations, following the pipelining of the architecture, which are presented back to the user. This architecture in figure 1, elaborate the system architecture which ensures efficient text analysis with intuitive visual representations, to interpret large datasets easily.

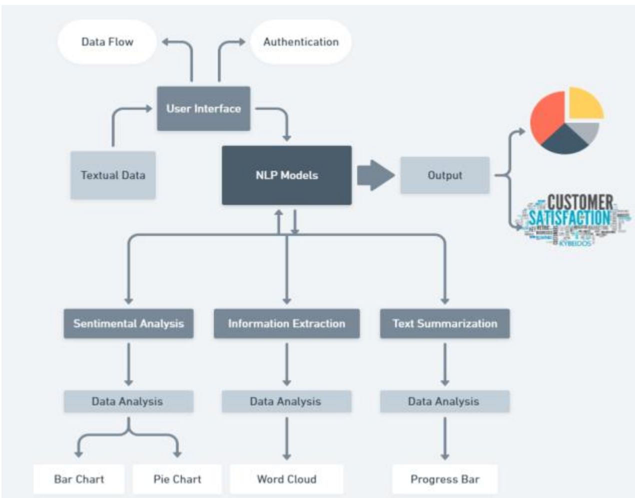

Fig. 1 System Architecture

V. Implementation

A. Sentiment Analysis

Sentiment analysis, also called "opinion mining", is a NLP technology or application that identifies the emotive undertone of a text, paragraph, or piece of writing. Sentiment analysis is carried out utilizing several NLP techniques, as described in the survey. Figure 1 demonstrates the model's application in the technological realm of business. The following is one method for performing sentiment analysis with improved outcomes and training scores.

a) Sentiment Analysis Model

Transformers is the architecture used to build the sentiment analysis model. Due to the self-attention mechanism of the encoder-based transformer, which efficiently processes the sequences in the parallel. This architecture makes it a good choice over other architectures to process the textual data. This architecture has yielded more accuracy, which counts to 89.32%. Importing libraries like Transformers and Pandas is required, as does preparing data from the Kaggle Twitter review dataset. This involves segregating the data for training, validation, and testing. To ensure flexibility among datasets with different sentiment indicators, sentiment labels are expressed as classes (0-neutral, 1-positive, 2-negative). The text data is then loaded and tokenized using a pre-trained tokenizer from Hugging Face's Transformers library. Finally, the model architecture is configured with the appropriate amount of emotion classes.

The model is the composition of:

DistilBertForSequenceClassification((distilbert): DistilBertModel() (transformer): Transformer() (sa_layer_norm): LayerNorm() (ffn): FFN() (output_layer_norm): LayerNorm() (classifier): Linear())

This is the basic model configuration, with parameters, layers, and bias defined at multiple model levels.

Training Configuration: Training arguments specify settings such as batch size, number of training epochs, and assessment procedure. The built model is trained on 35 epochs, with 128 as the size of the training batch, weight decay of 0.05, and warmup steps of 500. A Trainer is created using the model, training data, and these training arguments. The output logits are frequently treated to a SoftMax activation function in the model's last layer. SoftMax translates the raw model output to probability scores for each sentiment class.

Model Training: The model is trained using the Trainer with the specified training arguments. Metrics like F1-score, accuracy, precision, and recall are computed during training.

'test_loss':0.26389102935,
'test_f1':0.88503729426950286,
'test_accuracy':0.8932339955849,
'test_precision':0.835116140511,
'test_recall':0.80397183356136

The trained sentiment analysis model is evaluated on a test dataset using classification metrics like accuracy reports which are reflected in figure 2.

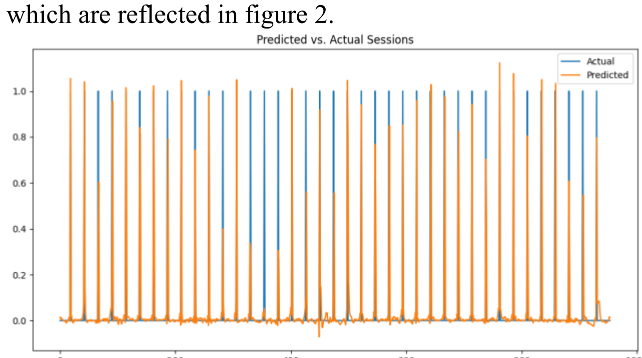

Fig. 2. Model's performance on dataset

The results from the model are then integrated with different visual representations using various Python libraries like Matplotlib, and Plotly. Various plots or representations can be generated using the results generated by the model. Some of the plots can be Pie chart, Bar chart, Line graph for time series analysis. One of the sample pie charts is generated on the customer reviews. The model is performing its analysis on several statements and generating the plot. The plot has been displayed to the user at the interface which is shown in figure 3. It can predict sentiment probabilities for sentences and could be further improved through hyperparameter tuning, using a larger dataset, and potentially combining models for even higher precision.

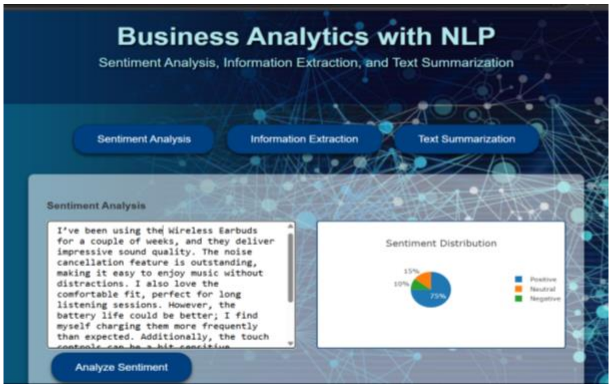

Fig. 3 Customer review's sentiment analysis

B. Information Extraction

Information extraction is an application in the domain of NLP that involves taking out the most relevant information of the text material. Identification and extraction of information from the text, such as facts or entities like names of people, organizations, places, and relationships between the entities, as well as the occurrences of events, are the goals of information extraction. Performing information extraction is necessary because it emphasizes on extracting valuable information out of the user data and this information can be used to improve the understanding of the insights in the business analysis. Various models can be used for information extraction as mentioned in the literature survey. Figure 1 depicts the performance of the model for this task under the tag of the NLP models shown in the diagram of architecture of the whole system. Following is one of the ways of performing the information extraction.

a) Information Extraction Model

The model performs information extraction and question-answering using the spaCy library and Transformer. Transformers have effectiveness in capturing long-range dependencies and handle sequential data efficiently and here it's used as its performance was better than the other architectures for the downstream tasks of question answering and important mathematical intuitions of transformer like:

$$Attention(Q, K, V) = softmax(Q * K^T / sqrt(d, k)) * V \quad (1)$$

where Q, K, and V are query, key, and value vectors, respectively. d, k is the dimension of the key vectors.

Multi-Head Attention, applies multiple scaled dot-product attention layers in parallel with different projections for Q, K, and V.

$$MultiHead(Q, K, V) = concat(h * Attention(W_{q^i}Q,$$
$$W\_k$$i*K,W_v^i*V))*W_o^i (2)

where h is the number of attention heads, W_q^i, W_k^i, W_v^i, W_o^i are weight matrices for each head. The encoding equation is:

$$Encoder(x) = LayerNorm(x + MultiHead(x, x, x) = LayerNorm(Encoder(x) + FFN(Encoder(x))) \quad (3)$$

$$PE(pos, i) = sin(pos/pow(10000{,}2 * i/d_{model}))\ if\ i\%2 == 0 = cos(pos/pow(10000, (2 * i + 1)/d_{model})) \quad (4)$$

where PE is the positional encoding, pos is the position, i represents the index for the embedding dimension, d_model is the model dimensions.
The equation for the decoder is:

$$Decoder(x, masked_{attn}) = LayerNorm(x + Multihead(masked_{attn}(x), x, x)) = LayerNorm(decoder(x, masked_{attn})) + FFN(Decoder(x, masked_{attn}))) \quad (5)$$

The key highlights and operations reflected in the implementation:
Library Setup, all the necessary libraries are installed and Cuda is selected as the working environment for the architecture. We have downloaded and loaded spaCy language models for text processing, including en_core_web_lg, en_core_web_sm, and en_core_web_trf. The data has loaded data from a JSON lines file named "cleaned_masdar.jsonl" using the jsonlines library. This data is expected to contain articles. Text Processing with spaCy, the loaded data is processed using spaCy models. For instance, the "body" of each article is processed using spaCy's language models. Named entities and their labels are visualized using a function. Information Extraction performs named entity recognition (NER) to extract organization (ORG) entities out of the text. The architecture performs text analysis to find organizations involved in negotiations based on certain keywords such as "negotiate", "ceasefire" or "talks". Dependency parsing is used to extract the relationship between tokens. In one example, the architecture demonstrates how to find verbs related to a specific location ("Aleppo") and their modifiers. We utilized the Hugging Face Transformers library to perform the extraction. It uses a pre-trained model ("deepset/roberta-base-squad2") to answer a question related to a context (in this case, a sentence from an article). The score of "0.826231185" was achieved with the correct answer for the question. The performance of the trained model is reflected from figure 4 below.

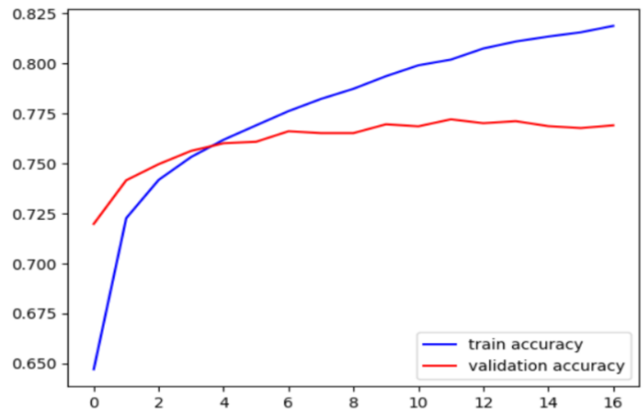

Fig. 4. Model performance for Information extraction.

The results from the model are taken as the input and used to generate the charts using libraries. Similar to the above model, the result of this model can be used to generate many meaningful plots. One such plot is Word Cloud that has been generated as a sample on the customer review. Important words are highlighted in the representation which makes it easier to analyze such textual data. The sample word cloud chart that has been generated by the model on the textual data is shown in figure 5.

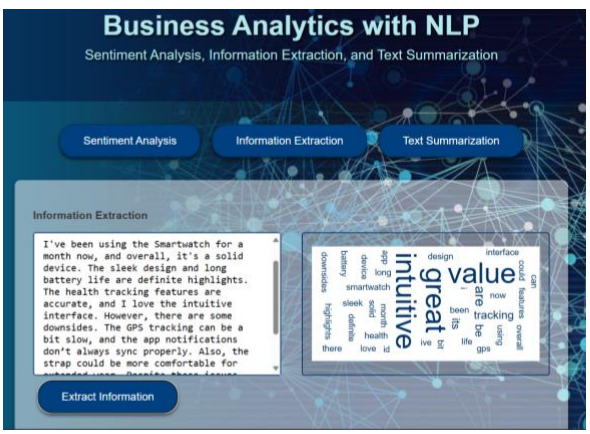

Fig. 5 Text extraction on the customer review.

C. *Text Summarization*

Making a summary of a longer text is known as text summarizing. It can be used to quickly get the essential information or the main ideas in the text by creating a shorter version of the larger text while still retaining its essence. Text summarizing aims to condense the main ideas of the original text into a concise and easily comprehensible summary. It can be used in various real-life applications in the business world like customer feedback analysis, notes on financial reports and forecasting, etc. Text summarization can be divided into two forms, “Extractive summarization” and “Abstractive summarization”.

a) Text summarizing Model

The next model, based for text summarization, is built using the GRU architecture (Gated Recurrent Units). GRUs are a simplified version of the LSTM (Long Short-Term Memory) architecture that prominently works on processing the sequence-to-sequence text data. The architecture preserves the semantic integrity of the input while processing large text data in this pipeline of the system architecture.

A bidirectional GRU serves as both an encoder and a decoder while processing these sequence-to-sequence structures. Input text is converted into fixed-length vectors by the encoder. These vectors are used by the decoder to output the summary for the input. The characteristics of GRU is that it has the capacity to manage long-term dependencies in sequences using the gating mechanism of its architecture which makes it a prominent choice for tasks like text summarization. During encoding and decoding, cells of the GRU went on for self learning which is very dynamic in nature which gives them the decision of deciding for which part of the text to keep and which to discard. Their performance for the same is much better in comparison to the working odd LSTMs. When generating summaries, the model also uses attention techniques to focus on key areas of the input text.

The weighted representation for the word is calculated as:

$$h'i = \Sigma a(ij) * hj \quad (6)$$

where a(ij) represents the attention score, ensuring that only the most relevant parts of the input are considered when generating each word in the summary.

The loss function used during training is the cross-entropy loss between the generated and reference summaries

$$Loss(i) = -y(i)Log(p(i)) \quad (7)$$

where y(i) is the true word and pi is the predicted probability of the word.

Key components of the GRU-based summarization model: The model begins by consuming the input text data. The configuration process involves obtaining, validating, and preparing the dataset. The model checks for data consistency, making sure that all essential input files are properly prepared and ready for training. The input text is tokenized with a tokenizer designed for sequence-based models. For indicating the start and the end of a particular sequence, tokens like <start>, <end> are used by the model. It takes the tokenized input, further processes it by the layers of the architecture, and generates the output which is the summary for the input data. Learning rate is set to 0.01, batch size is 128, and the model is trained over 35 epochs.The quality of the output is increased using an attention mechanism. This allows the model to provide the summary based on the most important tokens of the input sequence. During training, the model focuses on backpropagation to minimize the loss function that is on cross-entropy and optimizes the parameters of the model. After training, the performance of the model is checked using certain parameters which are listed below. These reflect the training of the model and show how well it will perform on the dedicated task.

Highlights:
To properly handle huge datasets, the model is trained with settings such as 35 epochs, 0.01 weight decay, and gradient accumulation. Attention methods are employed to increase the accuracy of generated summaries by concentrating on critical sections of the input text.
ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an evaluation metric that compares the quality of generated summaries to reference summaries. GPU acceleration is utilized in both model training and evaluation to speed up computations. The performance of the model has been shown in figure 6 below.

Mean Square Error: 0.0151001363
Mean Absolute Error: 0.031962894
R-squared: 0.6148836
Root Mean Squared Error: 0.122882608

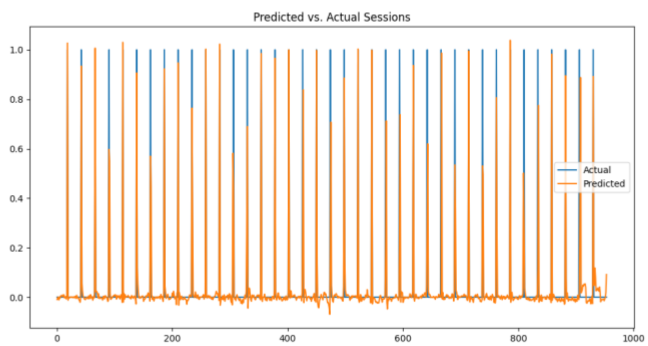

Fig. 6. Text summarization model performance

Models performance on the customer review on a product has been shown in figure 6. Unlike the above models, many different plots can be integrated to this model as well. One such prominent plot is Progress Bar, to note the progress. But mainly having such a summary, as shown in figure 7, are more useful.

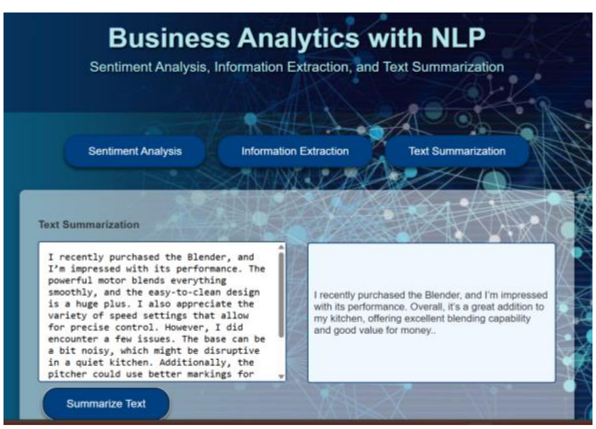

Fig. 7 Summarization of the customer review

VI. Results

The findings of this study show the effective construction of a unified NLP platform that combines sentiment analysis, information extraction, and text summarization to evaluate massive amounts of textual data for business analytics. The DistilBERT-based sentiment analysis model obtained 89.32% accuracy, with strong precision, recall, and F1 scores. Using spaCy and Transformer-based architectures, the information extraction model successfully recognized key entities and relationships, generating a word cloud and receiving an accuracy score of more that 82% in answering text questions. The approach for summarization is based on a GRU architecture which accurately generates the summary of the data passed to the trained model. The error performance of the model has been mentioned in the implementation portion. Each model is connected to libraries to generate the visualization in the form of graphs, charts, etc. This performance of the proposed model bridges the Gap in the current architectures of AI models like chatbots which mostly generate the textual output and lags in generating the visualizations of the processed data. Some visualizations like, Pie chart is the result from sentiment analysis, word cloud was the result from information extraction and summary text was generated from the text summarization. The proposed model's performance has been better in comparison to the results that are mentioned in table 1. This study achieves promising results in comparison to various models used for the respective NLP application. Therefore, the platform performs well in dedicated tasks for the various applications on the various user data.

VII. Conclusion

As a result of this study, this user-oriented platform for business analytics based upon the application of NLP is one of the prominent solutions for taking valuable insights from text data using various models like model for sentiment analysis, information extraction and text summarization. This solution works on text data that advances the analytical process as it adds text analysis along with numerical data in the real world. The integrated models prominently works in analyzing text from various sources and provide visual representation, making it convenient to draw insights from the raw data which makes it an adaptive tool for academic and business application in the real world.

The performance of the models, interface as always, a zone of improvement. Many other applications of NLP, like emotion detection, can also be integrated to evolve the approach. Fine tuning the integrated models on hyperparameters, making them more efficient to process on load of input data to be within time and space complexities are always the scope of improvement for this study.

Future study on this topic could also focus on increasing the platform's ability to numerical data as well. Models can be built and integrated with these models that will process the numerical data. One most prominent approach could be building a single large model that could process the text along with numerical data with the highest efficiency and accuracy. The enhancement would enable the platform to maintain its efficiency across both types of data, making it the most advanced and appropriate approach that would be very adaptive in the real world of business analytics. These developments and the results of the model will not only be used at the industry level but in the academic realm as well. This development opens up exciting possibilities for future research and development which involves fast processing of the big data in no time and giving the desired results to the user. Overall, the study opened a new horizon in finding solutions to process the data in text format in the world of business.

References

[1] P. Mishra, S. Bankar and M. Madankar, "Text Processing Hub: NLP Applications," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-5, doi: 10.1109/ICICET59348.2024.10616266.

[2] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," Proceedings of the 37th International Conference on Machine Learning (ICML), 2020.

[3] T. Brown, B. Mann, N. Ryder, et al., "Language models are few-shot learners," Advances in Neural Information Processing Systems (NeurIPS), 2020.

[4] H. Devlin, M.-W. Chang, K. Lee, and J. Toutanova, "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 (NAACL-HLT), 2021.

[5] C. Xu, J. Li, and P. Zhou, "Graph neural networks for information extraction," Proceedings of the 2021 Annual Conference of the Association for Computational Linguistics (ACL), 2021.

[6] F. Dai, X. Wang, and R. Zhang, "Sentiment-aware text summarization using transformer models," Proceedings of the 16th International Conference on Computational Linguistics (COLING), 2022.

[7] R. Kumar, S. Gupta, and M. Sharma, "Multi-modal embeddings for improving contextual relevance in NLP applications," Journal of Artificial Intelligence Research, vol. 65, pp. 452-468, 2022.

[8] S. Lee, J. Kim, and Y. Park, "Improving text summarization with reinforcement learning," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 35, no. 3, pp. 625-638, 2023.

[9] P. Rajput, K. Singh, and A. Verma, "Transformers for low-resource languages: Bridging the gap," Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023.

[10] A. Banerjee, R. Das, and N. Iyer, "NLP-based disaster tweet detection: Challenges and solutions," Proceedings of the 2024 International Conference on Emerging Trends in Engineering and Technology (ICETET), 2024.

[11] Z. Sun, J. Gao, and T. Chen, "BERTology: A study on the effectiveness and limitations of pretrained transformer models," Proceedings of the 2024 International Conference on Machine Learning and Applications (ICMLA), 2024.

[12] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A. M. (2020). "Transformers: State-of-the-art Natural Language Processing." Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

[13] Saiyyad, M. M., & Patil, N. N. (2024). Text Summarization Using Deep Learning Techniques: A Review. Proceedings of the International Conference on Recent Advances in Science and Engineering, 194-202.