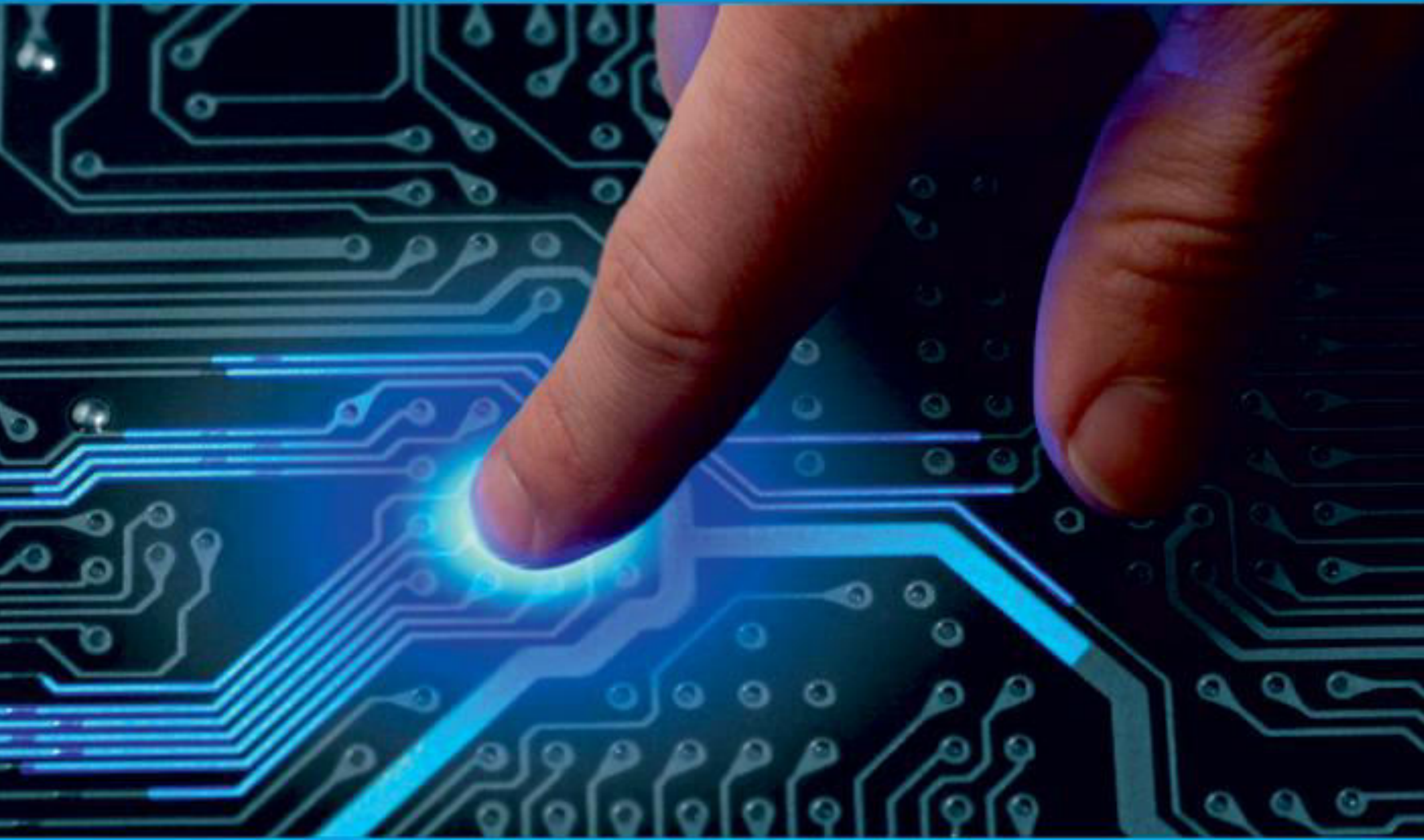
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 10, October 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Enhancing Text Insights through API-Driven Natural Language Processing

Mrs. S. Niveditha, R Yogesh Limbani, Adityakumar Patel, Sultan Mohammed Basit

Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India

ABSTRACT: With the exponential growth of text data across the globe, it becomes increasingly challenging to derive insights manually. This paper presents an API-based approach to Natural Language Processing (NLP) that performs paragraph summarization, sentiment analysis, and named entity recognition (NER). Our approach uses cloud-based NLP services like TextRazor, IBM Watson Natural Language Understanding, and Google Cloud Natural Language API, providing a scalable, efficient solution for text analysis on various devices. This paper highlights the architecture of the platform, the advantages of the API-driven model, and performance evaluations of each service.

KEYWORDS: API-based NLP, Text Summarization, Sentiment Analysis, Named Entity Recognition, Cloud NLP services

I. INTRODUCTION

Text data, originating from numerous sources like social media, articles, and reports, is proliferating. Traditional on-device NLP systems struggle to handle large datasets, prompting the need for more efficient approaches. By leveraging API-based NLP systems, we eliminate the need for resource-heavy local models, enabling real-time processing on various devices.

- Challenges with Traditional NLP: Traditional, device-dependent NLP solutions struggle to scale effectively with the growing volume of text data, making real-time insights difficult.
- Importance of API-Driven NLP: API-based solutions offer the advantage of processing large datasets in real-time by offloading tasks to cloud services, thus ensuring better scalability and efficiency.
- Objective of the Paper: To demonstrate the implementation and advantages of using external NLP APIs like TextRazor, IBM Watson NLU, and Google Cloud NLP for enhanced text analysis.

II. RELATED WORK

Existing systems for text processing often rely on local models which, though effective, are limited in scalability. Studies have explored cloud-based NLP systems, demonstrating their potential for improving efficiency. Previous works (e.g., machine learning for gambling addiction prevention) reveal how data-driven platforms can provide robust insights with minimal local computational resources.

- Existing NLP Systems: Many systems rely on on-device processing, which limits scalability and requires high computational resources.
- Cloud-Based NLP: Previous works in fields such as behavioural machine learning and sentiment analysis suggest that cloud-based solutions provide superior performance for large-scale data processing.
- Comparison with Similar Studies: While local models have their strengths in privacy, API-based solutions optimize for speed and real-time insights, making them suitable for large datasets and enterprise applications.

III. PROPOSED ALGORITHM

Our platform utilizes external NLP APIs to perform three main tasks:

- Text Summarization: The system extracts the most relevant information from a paragraph.
- Sentiment Analysis: Evaluates the sentiment (positive, negative, neutral) of a text.
- Named Entity Recognition (NER): Identifies key entities such as names, locations, and organizations.

System Architecture: The core system is built using React.js for the frontend and Node.js for backend processing. The user sends a text input to the backend, which then processes the request by invoking external APIs for NLP.

Frontend (React.js):

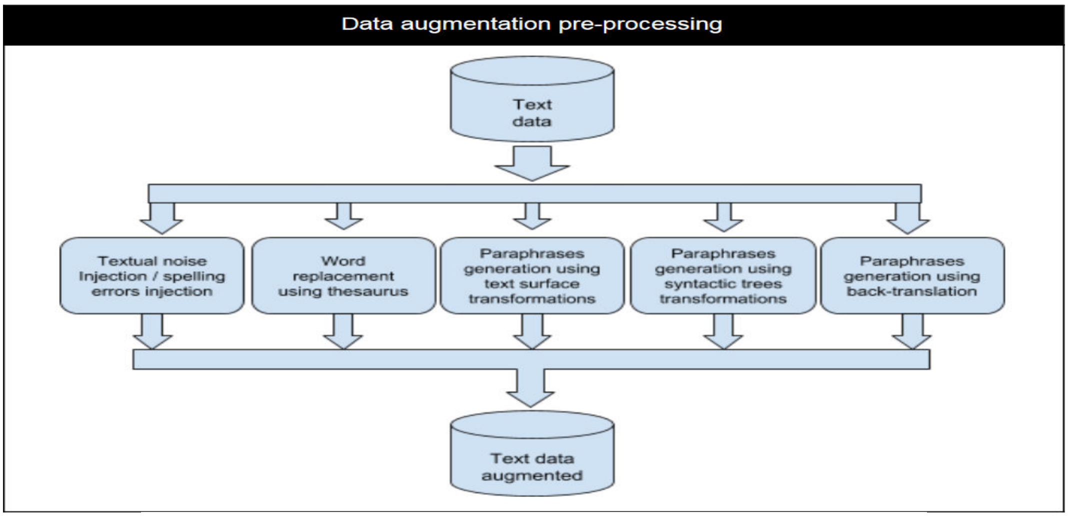
- Simple user interface where users input the text for analysis.
- Results are displayed in real-time as they are fetched from the backend.

Backend (Node.js):

- Responsible for routing user requests to the appropriate NLP API.
- Processes the API responses before sending them back to the frontend.

NLP APIs:

- Each API is invoked based on the task (e.g., TextRazor for entity recognition, IBM Watson for sentiment analysis).


Data augmentation pre-processing
Text data
Textual noise Injection / spelling errors injection
Word replacement using thesaurus
Paraphrases generation using text surface transformations
Paraphrases generation using syntactic trees transformations
Paraphrases generation using back-translation
Text data augmented

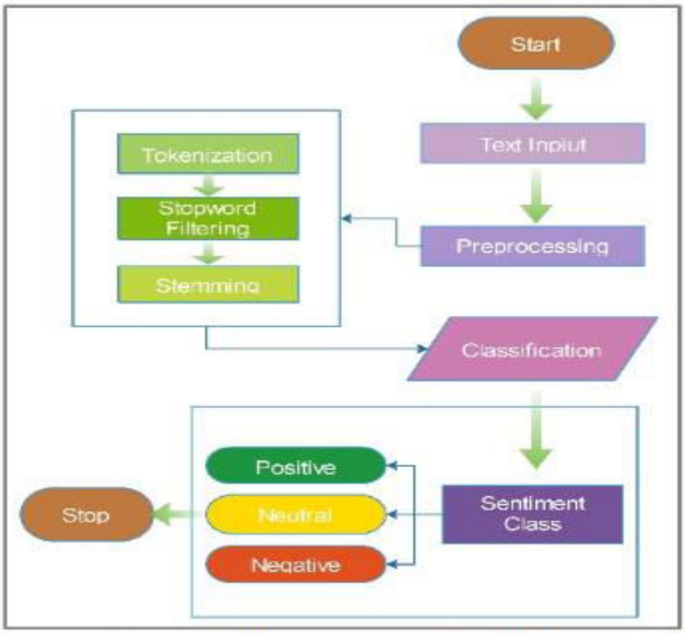


Figure 1: General framework of SA process.

IV. PSEUDO CODE

```
Step 1: Input the text data for processing (e.g., paragraphs, articles, or reviews).
Step 2: Call the Text Summarization API to get a concise summary.
Step 3: Perform Sentiment Analysis using the respective API to determine text polarity.
Step 4: Use the Named Entity Recognition (NER) API to extract entities like names, locations, and organizations.
Step 5: Process the results from each API:
    • If summarization result is too lengthy, re-process.
    • If sentiment result is neutral or unclear, refine the input for better clarity.
    • For NER, cross-check identified entities with a database of known names/places.
Step 6: Combine and present the results on the frontend.
Step 7: Repeat for new input data.
```

V. SIMULATION RESULTS

In our experiments, we evaluated the performance of TextRazor, IBM Watson, and Google Cloud NLP for text summarization, sentiment analysis, and named entity recognition (NER) on datasets of varying lengths and complexities. The results showcase differences in API performance across multiple metrics, such as processing time, accuracy, and scalability.

Experiment Setup:

- Datasets: Collected from various sources including news articles, product reviews, and social media posts.
- Tools: React.js frontend, Node.js backend, and APIs from TextRazor, IBM Watson NLU, and Google Cloud NLP.
- Metrics Evaluated:
  - ➢ Text Summarization Accuracy
  - ➢ Sentiment Analysis Accuracy
  - ➢ NER Precision
  - ➢ Response Time

Results:

- Text Summarization:
  - ➢ TextRazor provided detailed, structured summaries but took longer to process.
  - ➢ IBM Watson NLU gave concise summaries, ideal for short-form text.
  - ➢ Google Cloud NLP balanced accuracy and speed across all text lengths.
- Sentiment Analysis:
  - ➢ IBM Watson NLU showed the highest accuracy in sentiment detection, especially for complex or nuanced emotions.
  - ➢ Google Cloud NLP performed well on general sentiment analysis but struggled slightly with ambiguous inputs.
  - ➢ TextRazor had lower sentiment granularity, focusing more on subject extraction.
- Named Entity Recognition (NER):
  - ➢ TextRazor excelled in detecting a wide variety of entities, including niche or lesser-known names.
  - ➢ Google Cloud NLP performed consistently for standard entities (e.g., people, places).
  - ➢ IBM Watson NLU showed moderate performance, but it excelled in entity categorization.

VI. CONCLUSION AND FUTURE WORK

The API-driven NLP platform effectively handles various text analysis tasks, including summarization, sentiment analysis, and NER. Using cloud-based APIs offloads computational tasks from local machines, ensuring scalability and efficiency. Our results demonstrate that different APIs excel in specific tasks, with IBM Watson NLU being superior in sentiment analysis, TextRazor leading in NER, and Google Cloud NLP offering balanced performance.

Key Takeaways:

- API-driven NLP offers a scalable and efficient alternative to local processing, especially for handling large text datasets.

- Each API has distinct strengths, making it beneficial to combine services depending on the task at hand.
- TextRazor is ideal for entity recognition, IBM Watson excels in sentiment, and Google Cloud NLP performs well overall.

Future Work:

- Multilingual Support: Expanding the system to handle more languages using additional APIs.
- Real-Time Processing: Implementing improvements to ensure faster real-time text processing, especially for large-scale datasets.
- Topic Modeling: Adding topic modeling to analyze themes across larger texts such as news articles and reports.

REFERENCES

1. Bird, Steven, Edward Loper, and Ewan Klein. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc., 2009.
2. Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008.
3. Devlin, Jacob, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805 (2018).
4. Agarwal, Apoorv, et al. Sentiment Analysis of Twitter Data. Proceedings of the Workshop on Language in Social Media (LSM 2011), 2011.
5. Mikolov, Tomas, et al. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781 (2013).
6. Rathore, Vandana Singh, and Divya Srivastava. Sentiment Analysis Techniques on Big Data: A Comparative Study. International Journal of Computer Applications 89.7 (2014).
7. Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
8. McCallum, Andrew, and Wei Li. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction, and Web-Enhanced Lexicons. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, 2003.
9. Feldman, Ronen. Techniques and Applications for Sentiment Analysis. Communications of the ACM 56.4 (2013): 82-89.
10. Liu, Bing. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, 2020.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING