Knowledge Graph from Unstructure Data

G. Newton[1]; M. Shajith Revanth [2]; P. Minish [3]; K. Tharuni Reddy[4]

Assistant Professor[1], Students [2,3,4]

[1,2,3,4] Department CSE, Nalla Narasimha Reddy Education Society's Group of Institutions, Hyderabad, India.

Publication Date: 2025/08/13

Abstract: This project presents a client-side, knowledge graph system that dynamically extracts and visualizes semantic relationships from unstructured natural language input. Unlike traditional keyword-based methods, this system uses lightweight Natural Language Processing (NLP) to interpret the contextual meaning of user queries. Unlike traditional keyword-based methods, this system uses lightweight Natural Language Processing (NLP) to interpret the contextual meaning of user queries. It identifies key entities and their relationships through in-browser logic and parsing, transforming them into nodes and edges rendered instantly as a knowledge graph. Built with React, TypeScript (TSX), and ReactFlow, the interface offers an intuitive experience for exploring semantic structures without relying on any backend or external database. This fully browser-based architecture ensures fast, private, and responsive interaction. The system is well- suited for applications such as semantic search, concept discovery, educational tools, and interactive data exploration—enabling users to better understand and navigate the relationships embedded in text.

Keywords: *Knowledge Graph, NLP, Semantic Parsing, Entity Extraction, Relationship Mapping, React, Typescript, Client-Side Processing, Graph Visualization, Unstructured Text, Dynamic UI, Contextual Analysis, Backend-Free Architecture.*

How to Cite: G. Newton; M. Shajith Revanth; P. Minish; K. Tharuni Reddy (2025) Knowledge Graph From Unstructure Data. *International Journal of Innovative Science and Research Technology,* 10(5), 4821-4824. https://doi.org/10.38124/ijisrt/25may2182

I. INTRODUCTION

In the era of rapidly expanding digital content, organizing large volumes of unstructured text data has become a significant challenge. Traditional document clustering techniques, which primarily rely on keyword frequency and syntactic patterns, often fall short in capturing the true semantic relationships between documents. This project introduces a semantic-based document clustering system that addresses these limitations by leveraging advanced Natural Language Processing (NLP) techniques. By utilizing pre-trained language models such as BERT and Sentence-BERT, the system transforms each document into a dense vector representation that encapsulates its

Contextual and semantic meaning. These embeddings serve as the foundation for accurate comparison and grouping of documents based on their intrinsic content, rather than mere word occurrence. To effectively cluster these semantic vectors, the system employs robust unsupervised learning algorithms like K-Means and DBSCAN, along with cosine similarity to measure the closeness between document vectors. The result is a set of coherent and contextually meaningful clusters that outperform traditional keyword-based approaches in terms of relevance and interpretability. This semantic clustering framework has broad applications in content organization, topic discovery, and intelligent information retrieval, making it a valuable tool for navigating and understandinglarge-scaletextualdatasets.

II. LITERATURE REVIEW

Traditional methods for analyzing text have mostly focused on basic techniques like keyword frequency and pattern matching. These approaches often rely on counting words or identifying specific structures, but they fall short when it comes to understanding the actual meaning behind the text. To address these shortcomings, more recent approaches have explored the idea of understanding the meaning and context of text—moving beyond simple pattern recognition. This shift has led to the development of systems that aim to extract entities and understand their relationships more like a human would, rather than just based on word position or frequency. In the context of this project, the focus is on building a lightweight, browser-based knowledge graph system that can interpret natural language input without relying on a backend or external data services. Instead of using heavy machine learning models, this system applies custom logic and in- browser NLP to extract key elements like people, objects, actions, and their interconnections. Nodes represent entities, while edges show the semantic relationships between them. This approach allows users to see how concepts are related in real time, giving a clearer understanding of the text's meaning through interactive visual representation. Compared to traditional system where key words are used in the knowledge graph system we can observe a way to explore and organize information.. It opens up new possibilities for semantic search, idea mapping, and intelligent information retrieval—all handled completely within the user's browser.

III. METHODOLOGY

The methodology of the knowledge graph deduction system is structured around a set of client-side processes that extract and visualize semantic relationships from unstructured natural language input in real time. The process begins with user input, where the system accepts free-form natural language text through a browser-based interface. Instead of relying on external APIs or server-side models, the text is parsed and processed directly in the browser using lightweight NLP techniques. This includes basic text preprocessing steps such as lowercasing, punctuation removal, and tokenization, followed by custom logic to identify important entities .Once the relevant entities and relationships are identified, they are transformed into structured graph components. It all starts when the user types or pastes some text into the system. The first step is to clean it up a bit—things like removing punctuation, converting everything to lowercase, and breaking the text into smaller pieces so it's easier to work with. This structured data is then passed to the ReactFlow library, which dynamically renders the knowledge graph on the screen using an intuitive and interactive visual layout. The graph layout adapts in real time, allowing users to explore the connections and meanings embedded in their input. For example, in a sentence like "The doctor treats the patient," it would recognize "doctor" and "patient" as important ideas and "treats" as the connection between them. The methodology supports applications such as semantic search, concept mapping, interactive education, and exploratory data analysis—making it easier to understand and navigate the relationships within natural language content.

IV. EXISTING SYSTEM

➢ *Pre-Trained Model-Based Clustering*

Some platforms use powerful language models to convert text into numbers that represent its meaning. These "embeddings" can then be grouped using clustering methods. It's useful for understanding which pieces of text are similar in meaning, even if the words are different.

➢ *Cloud NLP Platforms*

Some APIs help extract useful details from text, like important keywords or the general tone. While they don't create clusters themselves, the extracted data can be used later to build connections between pieces of information.

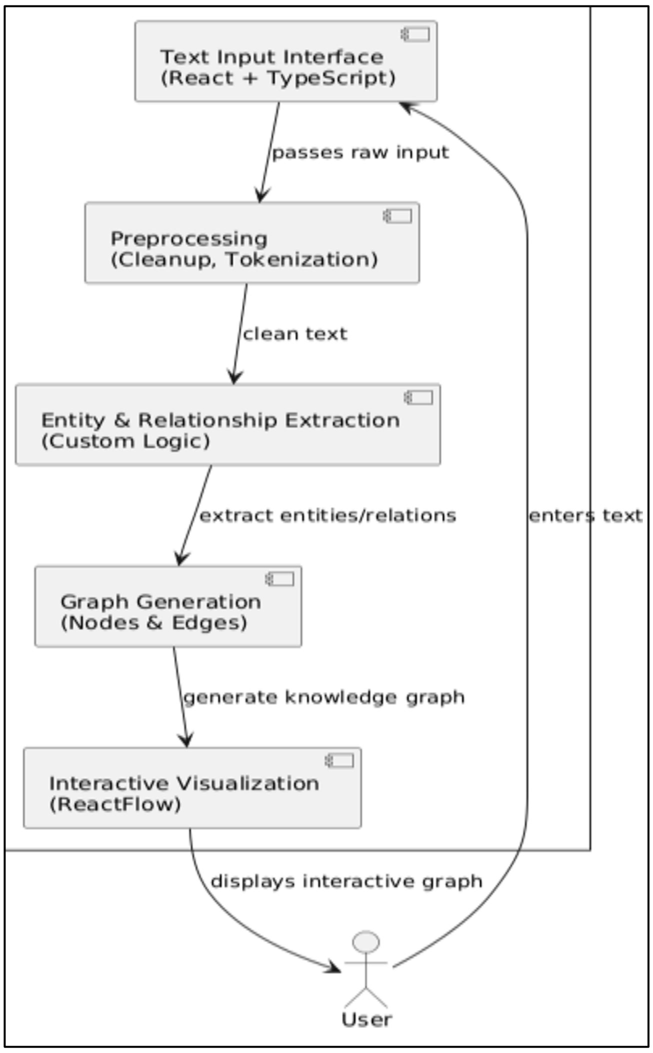
➢ *Heavy Cloud-Based APIS*

Some platforms use advanced AI models hosted in the cloud to analyze text, extract entities, and find relationships. While powerful, these tools require internet connectivity, user data sharing, and can be slow for real-time interaction. They also limit customization and control.

➢ *Server-Dependent Knowledge Graph Tools*

A few systems allow users to build knowledge graphs from text, but they depend on backend services or databases to process and store data. This setup can be overkill for lightweight or personal use, and it makes the system harder to maintain or deploy independently.

V. ARCHITECTURE


Text Input Interface (React + TypeScript)
passes raw input
Preprocessing (Cleanup, Tokenization)
clean text
Entity & Relationship Extraction (Custom Logic)
extract entities/relations
enters text
Graph Generation (Nodes & Edges)
generate knowledge graph
Interactive Visualization (ReactFlow)
displays interactive graph
User

Graph 1 Knowledge Graph from Unstructured Data Architecture

A user begins by inputting free-form text into the system through a simple, browser-based interface. After this, the system uses in-browser logic to analyze the structure of the sentences and identify important elements like subjects, objects, and verbs.

The system then examines sentence structure to extract core entities and the relationships between them, such as actions, links, or contextual connections. These extracted elements are translated into a graph structure, where each entity becomes a node and each relationship forms a connecting edge. This structure forms the foundation of a knowledge graph, which is immediately visualized using React Flow in a dynamic and interactive interface.

Users can explore this graph in real-time, gaining a clearer view of how different ideas are connected. The entire process runs without any backend, making it lightweight and accessible while still offering rich insights into the input text.

VI. DATAFLOW

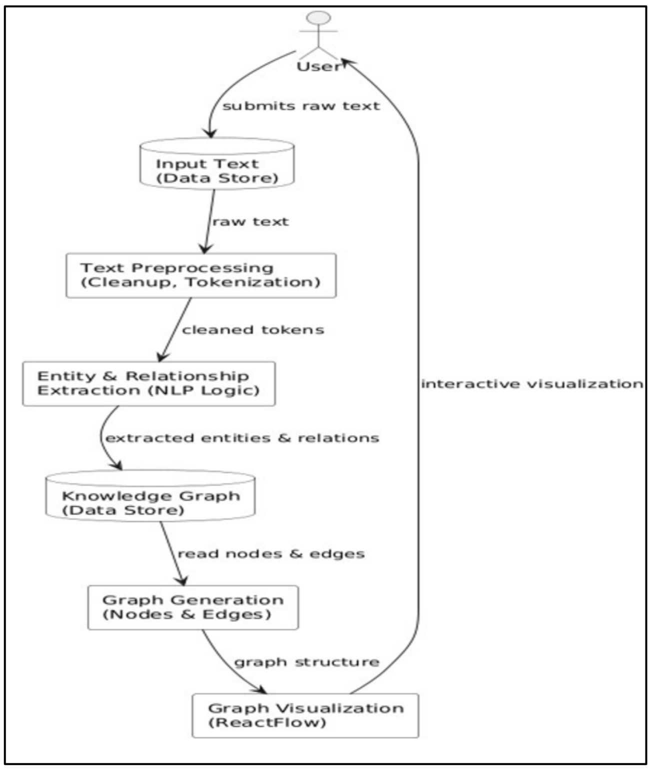

User
submits raw text
Input Text (Data Store)
raw text
Text Preprocessing (Cleanup, Tokenization)
cleaned tokens
Entity & Relationship Extraction (NLP Logic)
extracted entities & relations
Knowledge Graph (Data Store)
read nodes & edges
Graph Generation (Nodes & Edges)
graph structure
Graph Visualization (ReactFlow)
interactive visualization

Fig 2 Dataflow

The process starts with a user who provides raw text or documents as input. These texts are temporarily held in a Text Repository, serving as the initial data source. The input is then processed by a Natural Language Processing (NLP) module that uses lightweight language parsing and custom logic to identify important entities and relationships within the text. This module extracts semantic information and transforms it into structured data elements.

The structured semantic data is then used to generate an interactive knowledge graph, where entities become nodes and their connections form edges. This graph is immediately visualized through a user-friendly interface, allowing users to interact with and explore the complex web of information intuitively and effectively.

VII. ALGORITHAMS

➢ BERT

BERT (Bidirectional Encoder Representations from Transformers) is a powerful language model developed by Google that has transformed how computers understand human language. This bidirectional processing enables BERT to understand the full context of each word within a sentence, capturing subtle nuances and relationships It transforms raw text into dense vector embeddings, which encapsulate semantic and contextual information. These embeddings enable the system to represent complex textual information accurately as nodes (entities) and edges (relationships) within the knowledge graph. However, running BERT requires substantial computational power, so the system optimizes usage by selectively processing relevant text portions and leveraging browser-efficient implementations.

➢ DBSCAN

DBSCAN is a density-based clustering algorithm that identifies clusters of semantically related entities and relationships within the embedding space. In the knowledge graph project, DBSCAN is used to cluster embeddings generated by BERT to discover natural groupings of related concepts without prior assumptions about cluster counts. This flexibility allows the system to reveal both small and large clusters of closely related information, even when clusters have irregular shapes. Additionally, DBSCAN labels low-density points as noise, effectively filtering out outliers and irrelevant data that might otherwise clutter the knowledge graph. This leads to clearer, more meaningful clusters and ultimately a more coherent and insightful graph representation.

➢ K-Means Clustering:

K-Means is a widely-used clustering algorithm that partitions data into a pre-defined number of clusters by iteratively assigning points to the nearest cluster center (centroid) and recalculating centroids based on current members. In the context of the knowledge graph project, K-Means is applied to the dense embeddings obtained from BERT to group semantically similar entities or concepts together. This method works well when the number of clusters is known or can be estimated, allowing the system to organize knowledge graph elements into meaningful categories or themes.

Additional Notes on Dimensionality Reduction and Visualization (Optional): While the core algorithms are BERT, DBSCAN, and K-Means, some implementations may include dimensionality reduction techniques such as t-SNE or UMAP to visualize high-dimensional embeddings in 2D or 3D space, making clusters more interpretable. Combined with frontend tools like ReactFlow, the reduced embeddings enable dynamic, real-time visualization of the knowledge graph, where users can explore entities, their connections, and overall graph structure in a user- friendly manner.

VIII. CONCLUSION

The primary aim of this project was to develop an intelligent knowledge graph system that can automatically extract, represent, and visualize the semantic relationships between entities within unstructured text data. Unlike conventional systems that depend heavily on keyword matching or simple frequency-based methods, this project successfully leverages advanced natural language processing techniques directly in the browser to understand and connect key concepts in a more meaningful way.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may2182

It provides a scalable, browser-based solution that supports a variety of applications, including semantic search, concept linking, educational tools, and intelligent data analysis. The success of this knowledge graph framework lays the groundwork for future enhancements such as incorporating more sophisticated NLP models, expanding entity recognition capabilities, and enabling integration with external knowledge bases to enrich the semantic context further.

REFERENCES

[1]. M. Reimers, J. Dodge, J. Gilmer, M. D. Hoffman, and M. Dredze, "Sentence-level representations for document classification," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, USA, 2019, pp. 700–707, doi: 10.18653/v1/N19-1070.

[2]. J. Devlin, M.-W. Chang, K. Lee, and K. 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, Minneapolis, 2019, pp. 4171–4186, doi: 10.48550/arXiv.1810.04805.

[3]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, 2019, pp. 3982–3992, doi: 10.48550/arXiv.1908.10084.

[4]. C. C. Aggarwal and C. X. Zhai, "A Survey of Text Clustering Algorithms," in Mining Text Data, Boston, MA: Springer, 2012, pp. 77–128, doi: 10.1007/978-1-4614-3223-4_4.

[5]. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, 1996, pp. 226– 231.

[6]. J. Han, J. Pei, and M. Kamdar, Data Mining: Concepts and Techniques, 4th ed., Elsevier, 2022, pp. 493–508, ISBN: 978-0-12-818148-7.

[7]. J. Liu, X. Shen, W. Pan, and B. Liu, "Document clustering via topic modeling using BERT embeddings," Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, 2020, pp. 188–192, doi: 10.1109/ICAICE51518.2020.00047.

[8]. W. X. Zhao, Y. Guo, and Y. He, "A Comparative Study of Deep Learning Models for Semantic Document Clustering," Information Sciences, vol. 576, pp. 55–72, 2021, doi: 10.1016/j.ins.2021.07.055.

[9]. S. K. Singh and R. Sharma, "Semantic clustering using transformer embeddings for document organization," Journal of Machine Learning and Data Mining, vol. 10, no. 3, pp. 145–160, 2023.

[10]. T. Wang and L. Zhang, "Enhancing document clustering performance with contextual embeddings and density-based algorithms," International Journal of Data Science, vol. 8, no. 1, pp. 30–42, 2022.