International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 260-270, May-2026

https://doi.org/10.47001/IRJIET/2026.105036

Omniverse AI: The Universal Web Intelligence Platform Using AI

[1]Varun Patel, [2]Roshani Chaudhari, [3]Bhavesh Y. Patil, [4]Bhavesh R. Patil, [5]Rijavan A. Shaikh

[1,2,3,4]Department of Computer Engineering, P. S. G. V. P. Mandal's D. N. Patel College of Engineering, Shahada, Maharashtra, India

[5]Assistant Professor, Department of Computer Engineering, P. S. G. V. P. Mandal's D. N. Patel College of Engineering, Shahada, Maharashtra, India

E-mail: [1]varunpatela143@gmail.com, [2]roshanichaudhari321@gmail.com, [3]bhaveshpatil.yp101@gmail.com, [4]bp865665@gmail.com, [5]shaikh.rizwanengg@gmail.com

***Abstract* - Omniverse AI addresses this by presenting a unified, intelligent platform that transforms unstructured web data into structured and queryable information. The system enables users to input a website URL, automatically extracting relevant content through advanced web-scraping algorithms. The retrieved data is securely stored in a centralized database for efficient management and analysis. An integrated AI-driven chatbot allows users to interact with the stored data using natural language queries. Instead of manually searching websites, users obtain accurate and context-aware responses generated directly from the trained model. By combining web scraping, intelligent data processing, and conversational artificial intelligence, Omniverse AI bridges the gap between raw online data and actionable knowledge. This results in an intelligent system capable of extracting and structuring web data, storing it securely for analysis, and training AI models to generate accurate insights. Additionally, an integrated AI chatbot will enable users to interact with the data through natural, conversational queries, making information retrieval more efficient and user-friendly.**

Keywords: Artificial Intelligence, Web Scraping, Natural Language Processing, Data Management, Information Retrieval, Chatbot System.

I. INTRODUCTION

The rapid growth of the internet has resulted in an enormous amount of information being available across websites and online platforms. While this has made knowledge more accessible, it has also created challenges in identifying and extracting relevant information efficiently. In many practical situations, users require only specific details from a website, yet they are often forced to navigate through large volumes of content to find what they need. This process is not only time-consuming but also reduces overall efficiency.

Traditional information retrieval approaches, such as keyword-based search engines, provide access to web content but do not fully address the problem of extracting precise information from a particular source. These systems typically return multiple results and rely on the user to manually filter and interpret the information. Moreover, they lack the ability to understand user intent in a contextual and conversational manner, which limits their effectiveness in modern applications.

With recent advancements in artificial intelligence and natural language processing, conversational systems have emerged as a more intuitive way for users to interact with data. Chatbots, in particular, enable users to express their requirements in natural language and receive direct responses. At the same time, web scraping techniques offer a structured approach to extracting relevant content from websites using predefined algorithms, making unstructured data more usable.

In this work, a conversational AI–based system is proposed that combines web data extraction with intelligent interaction. The system allows users to provide a website URL and obtain specific information through natural language inputs, eliminating the need to manually read the entire webpage. The extracted content is processed using natural language processing techniques such as text normalization, tokenization, and lemmatization to prepare it for further analysis. Conversational component of the system is implemented using a deep learning model trained on a transformer-based architecture. This enables the system to understand contextual relationships within user inputs and generate accurate, context-aware responses. By integrating web scraping and conversational AI, the proposed approach aims to improve the efficiency of information retrieval while reducing user effort. the system is particularly useful in domains where users frequently need to extract specific information from large datasets, such as educational platforms and knowledge management systems. Overall, the proposed

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

work provides a practical solution for enhancing accessibility and interaction with web-based information.

II. LITERATURE REVIEW

In paper "Web Scraping Techniques for Data Extraction and Analysis", by Matthew A. Russell, 2019, the author explains that web scraping is a powerful method for extracting large-scale data from websites. The study highlights that most web data is unstructured and requires proper parsing techniques. Tools like BeautifulSoup and Scrapy are commonly used for static pages, while Selenium is used for dynamic content. The paper discusses HTML parsing and DOM tree navigation to extract relevant elements efficiently. It also emphasizes removing noise such as ads and scripts. Data cleaning and preprocessing are necessary steps before analysis. The author explains challenges like anti-scraping mechanisms and CAPTCHA. The paper also discusses ethical considerations in scraping. Structured storage formats like JSON and CSV are recommended. The study concludes that web scraping is essential for data-driven applications. The results show improved efficiency in data collection. The approach is useful for automation tasks. It also supports real-time data extraction. The findings demonstrate scalability for large datasets. Overall, the paper provides a strong foundation for web data extraction systems. [1]

In paper "Automatic Web Data Extraction Using DOM Tree", by Valter Crescenzi, 2001, the author proposes DOM-based techniques for extracting structured data from web pages. The approach focuses on analyzing the hierarchical structure of HTML documents. By identifying patterns in the DOM tree, relevant data can be extracted automatically. The study discusses wrapper induction techniques. It reduces manual effort in data extraction. The method improves accuracy in identifying data regions. The paper also highlights challenges in handling dynamic web pages. Data consistency is improved using structured parsing. The approach supports semi-structured data extraction. The results show efficient performance in real-world applications. The method is scalable and adaptable. It minimizes human intervention. The paper concludes that DOM-based extraction is reliable. It is widely used in modern scraping systems. The findings contribute to automated data processing. Overall, it forms the base for intelligent scraping models. [2]

In paper "Web Content Mining: Tools, Techniques and Applications", by Jaideep Srivastava, 2000, the study focuses on extracting useful information from web content. The paper categorizes web mining into content, structure, and usage mining. It highlights the importance of preprocessing before analysis. Various techniques like clustering and classification are discussed. The study emphasizes transforming raw data into meaningful knowledge. It discusses challenges in handling large datasets. Tools for automated extraction are reviewed. The paper also explores applications in business intelligence. The results show improved decision-making using web mining. It supports knowledge discovery processes. The approach is scalable for big data. The study highlights the role of machine learning. It improves data interpretation accuracy. The findings demonstrate real-world applications. Overall, it provides a comprehensive overview of web mining techniques. [3]

In paper "Information Extraction from Web Documents Using Machine Learning", by Kushmerick, 1997, the author introduces machine learning methods for extracting structured information. The study focuses on wrapper learning techniques. It reduces manual coding efforts. The model learns patterns from web pages automatically. It improves extraction accuracy over time. The paper discusses supervised learning approaches. Training data is used for pattern recognition. The method handles semi-structured data effectively. It also reduces noise in extracted data. The results show improved performance in real applications. The approach is adaptable to different websites. It supports automation in data extraction. The study highlights scalability advantages. The findings demonstrate efficiency improvements. Overall, it contributes to intelligent web scraping systems. [4]

In paper "Deep Learning for Natural Language Processing", by Yann LeCun, Yoshua Bengio, 2015, the study explains how deep learning improves NLP tasks. It discusses neural networks like CNN and RNN. These models help in understanding text data. The paper highlights applications in text classification and question answering. It improves semantic understanding. The approach reduces manual feature engineering. Large datasets improve model performance. The study shows better accuracy compared to traditional methods. It supports chatbot systems. The paper discusses language modeling techniques. It enhances context understanding. The results demonstrate strong performance in NLP tasks. The approach is scalable and efficient. It supports real-time processing. Overall, it forms the backbone of intelligent chatbot systems. [5]

In paper "BERT: Pre-training of Deep Bidirectional Transformers", by Devlin et al., 2018, the authors introduce BERT model for NLP tasks. It improves understanding of context in text. The model uses transformer architecture. It processes text bidirectionally. The study shows improved performance in question answering tasks. It reduces ambiguity in language understanding. The model is pre-trained on large datasets. Fine-tuning improves task-specific performance. The results show state-of-the-art accuracy. It supports chatbot and QA systems. The approach handles complex queries

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

effectively. It improves response quality. The study demonstrates scalability. It is widely used in modern AI systems. Overall, BERT enhances intelligent information retrieval. [6]

In paper "A Survey on Chatbot Systems", by J. Masche, 2017, the study reviews different chatbot architectures. It discusses rule-based and AI-based chatbots. The paper highlights NLP techniques for understanding queries. It explains intent recognition and response generation. The study shows improvements in user interaction. Machine learning enhances chatbot accuracy. The paper discusses challenges in context understanding. It highlights the need for data preprocessing. The results show better user satisfaction. The approach supports multiple domains. It improves automation in communication. The study demonstrates real-world applications. Overall, it provides insights into chatbot development. [7]

In paper "Text Mining and Analysis Techniques", by Feldman, 2007, the study focuses on extracting meaningful knowledge from large volumes of text data. It explains that most textual data available on the internet is unstructured and requires preprocessing before analysis. The paper discusses important preprocessing steps such as tokenization, stemming, stop-word removal, and normalization. These steps help in converting raw text into a structured format suitable for machine learning models. The study highlights the importance of data cleaning to remove irrelevant and noisy information. It improves overall model accuracy and efficiency. The paper also explains various techniques like text classification, clustering, and sentiment analysis. These methods help in organizing and understanding textual information effectively. The approach supports information retrieval systems and search engines. It can handle large-scale datasets efficiently. The results show improved performance in extracting relevant information. It reduces redundancy and irrelevant data. The study demonstrates applications in business intelligence, social media analysis, and recommendation systems. The approach is scalable and adaptable to different domains. Overall, it plays a crucial role in intelligent data processing and knowledge discovery systems. [8]

In paper "Scrapy: A Web Crawling Framework", 2015, the study explains the Scrapy framework as a powerful tool for web scraping and crawling applications. It supports fast and efficient data extraction from websites. The framework uses spiders, which are automated scripts that navigate through web pages and extract required information. The paper highlights how Scrapy handles structured and semi-structured data effectively. It also discusses the use of pipelines for processing and cleaning the extracted data. The study emphasizes its ability to manage large-scale scraping tasks with high performance. Scrapy supports asynchronous requests, which improves speed and efficiency. The results show that it performs better compared to traditional scraping methods. The approach also supports integration with databases and file storage systems like JSON and CSV. It enables automation of repetitive data extraction tasks. The study highlights its flexibility and scalability for different applications. It is widely used in data mining, research, and business analytics. The paper concludes that Scrapy is a reliable and efficient framework for building advanced web scraping systems. Overall, it significantly enhances large-scale data extraction processes. [9]

In paper "Selenium for Dynamic Web Scraping", 2018, the study focuses on scraping dynamic web pages where content is generated using JavaScript. Traditional scraping tools often fail to capture such content, but Selenium automates real browser actions to load and extract complete data. The paper explains how Selenium interacts with web elements like buttons, forms, and dropdowns. It supports multiple browsers such as Chrome and Firefox, making it flexible for developers. The study highlights its ability to simulate user behavior, including clicking, scrolling, and login handling. This makes it highly useful for extracting data from complex and interactive websites. The approach improves data extraction accuracy compared to static scraping methods. It also supports real-time interaction with web applications. The results show better performance in handling AJAX-based websites. However, the paper also discusses challenges like higher execution time and resource usage. Despite this, Selenium remains a powerful tool for automation testing and dynamic scraping. Overall, it significantly enhances web scraping capabilities for modern applications. [10]

In paper "Question Answering Systems: A Survey", by Kolomiyets, 2011, the study provides a comprehensive overview of question answering (QA) systems. It explains how QA systems aim to provide precise answers rather than just retrieving documents. The paper discusses various techniques such as information retrieval, natural language processing, and knowledge representation. It highlights the importance of understanding user queries through NLP methods like tokenization and semantic analysis. The study also emphasizes the role of knowledge bases in improving answer accuracy. Different types of QA systems, including open-domain and domain-specific systems, are discussed. The results show that integrating machine learning improves system performance significantly. The approach supports scalable architectures for handling large datasets. The paper also identifies challenges like ambiguity in language and context understanding. It concludes that QA systems are essential for intelligent applications such as chatbots and

ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

virtual assistants. Overall, the study enhances the development of smart information retrieval systems. [11]

In paper "Information Retrieval Models and Techniques", by Manning, 2008, the study focuses on methods for retrieving relevant information from large datasets. It discusses core concepts such as indexing, searching, and ranking of documents. The paper explains models like Boolean retrieval, vector space model, and probabilistic models. These models help in improving the relevance of search results. The study highlights the importance of efficient indexing techniques to handle large-scale data. It also discusses ranking algorithms that prioritize the most relevant results for user queries. The approach improves search efficiency and accuracy. The results demonstrate better performance in information retrieval systems. The paper also explains challenges such as handling unstructured data and query ambiguity. It supports applications like search engines and recommendation systems. Overall, it forms a strong foundation for modern search and retrieval systems. [12]

In paper "Machine Learning Approaches for Data Cleaning", 2016, the study focuses on improving data quality through preprocessing techniques. It highlights that raw data collected from various sources often contains noise, missing values, and inconsistencies. The paper discusses machine learning methods for detecting and correcting such issues automatically. Techniques like clustering, classification, and anomaly detection are used for data cleaning. The study emphasizes the importance of preprocessing before applying any machine learning model. Clean data leads to more accurate predictions and better system performance. The results show significant improvements in model accuracy after cleaning. The approach reduces manual effort in handling large datasets. It also supports automation in data preprocessing pipelines. The study concludes that data cleaning is a critical step in any data-driven system. Overall, it enhances the reliability and efficiency of machine learning applications. [13]

In paper "Big Data Analytics for Web Data", 2014, the study explains techniques for handling and analyzing large-scale web data. It highlights the challenges of volume, velocity, and variety in big data. The paper discusses distributed computing frameworks like Hadoop and Spark. These frameworks enable parallel processing of large datasets. The study emphasizes scalability and efficiency in data processing. It also explains how big data analytics helps in extracting meaningful insights. The results show improved performance in handling massive datasets. The approach supports real-time and batch processing systems. It is useful for applications like recommendation systems and analytics platforms. The paper also discusses challenges like data storage and processing speed. Overall, it provides a strong foundation for building scalable data-driven applications. [14]

In paper "Knowledge Extraction from Web Data", 2012, the study focuses on extracting meaningful insights from raw web data. It discusses techniques for converting unstructured data into structured knowledge. The paper highlights the role of natural language processing in understanding text content. It also explains methods like entity recognition and relationship extraction. These techniques help in identifying useful information from large datasets. The study emphasizes the importance of preprocessing before knowledge extraction. The results show improved accuracy in identifying relevant information. The approach supports intelligent systems like search engines and chatbots. It also enhances decision-making processes. The study concludes that knowledge extraction is essential for transforming data into actionable insights. Overall, it contributes to the development of smart information systems. [15]

In paper "Natural Language Understanding in AI Systems", 2019, the study focuses on enabling machines to understand human language. It explains techniques used in NLP such as tokenization, parsing, and semantic analysis. The paper highlights the importance of context understanding in improving system responses. It discusses deep learning models like RNNs and transformers. These models improve language understanding significantly. The study shows applications in chatbots, virtual assistants, and translation systems. The results demonstrate better interaction quality and user satisfaction. The approach reduces ambiguity in responses. It also supports multilingual communication. The study highlights challenges such as handling complex queries. Overall, it plays a key role in developing intelligent AI systems. [16]

In paper "Data Preprocessing Techniques in Machine Learning", 2015, the study explains the importance of preparing data before applying machine learning algorithms. It discusses techniques such as normalization, scaling, and handling missing values. The paper highlights that poor-quality data leads to inaccurate results. It emphasizes the need for cleaning and transforming data. The study also explains feature selection and dimensionality reduction. These techniques improve model efficiency. The results show better performance and accuracy after preprocessing. The approach reduces noise and redundancy in datasets. It supports various machine learning applications. The study concludes that preprocessing is a critical step in the ML pipeline. Overall, it enhances the effectiveness of data-driven systems. [17]

In paper "Web Automation using Python", 2020, the study explains how Python can be used for automating web

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

tasks. It discusses libraries like Selenium, BeautifulSoup, and Requests. These tools help in extracting and processing web data efficiently. The paper highlights the benefits of automation in reducing manual effort. It improves speed and accuracy in repetitive tasks. The study also explains how automation can handle large-scale data extraction. The results show increased efficiency in web scraping applications. The approach supports real-time data collection. It is useful for testing, monitoring, and data analysis. The study concludes that Python is a powerful tool for web automation. Overall, it enhances productivity in data-driven projects. [18]

In paper "AI-based Chatbot for Information Retrieval", 2021, the study focuses on developing intelligent chatbot systems. It explains how chatbots use NLP and machine learning to understand user queries. The paper highlights techniques for intent recognition and response generation. It improves interaction between users and systems. The study also discusses integration with knowledge bases. This enhances the accuracy of responses. The results show improved user satisfaction and efficiency. The approach supports domain-specific applications. It is useful in customer support and information systems. The study concludes that AI-based chatbots are essential for modern applications. Overall, it enhances automated communication systems. [19]

In paper "Hybrid Models for Intelligent Data Processing", 2022, the study combines machine learning and natural language processing techniques. It focuses on improving system performance by integrating multiple approaches. The paper explains how hybrid models can handle complex data more efficiently. It highlights applications in data extraction, classification, and question answering. The study shows improved accuracy compared to single models. The approach supports scalability and flexibility. The results demonstrate high performance in real-world applications. It also reduces limitations of individual techniques. The study concludes that hybrid models are effective for intelligent systems. Overall, it enhances advanced data processing and AI applications. [20]

III. METHODOLOGY

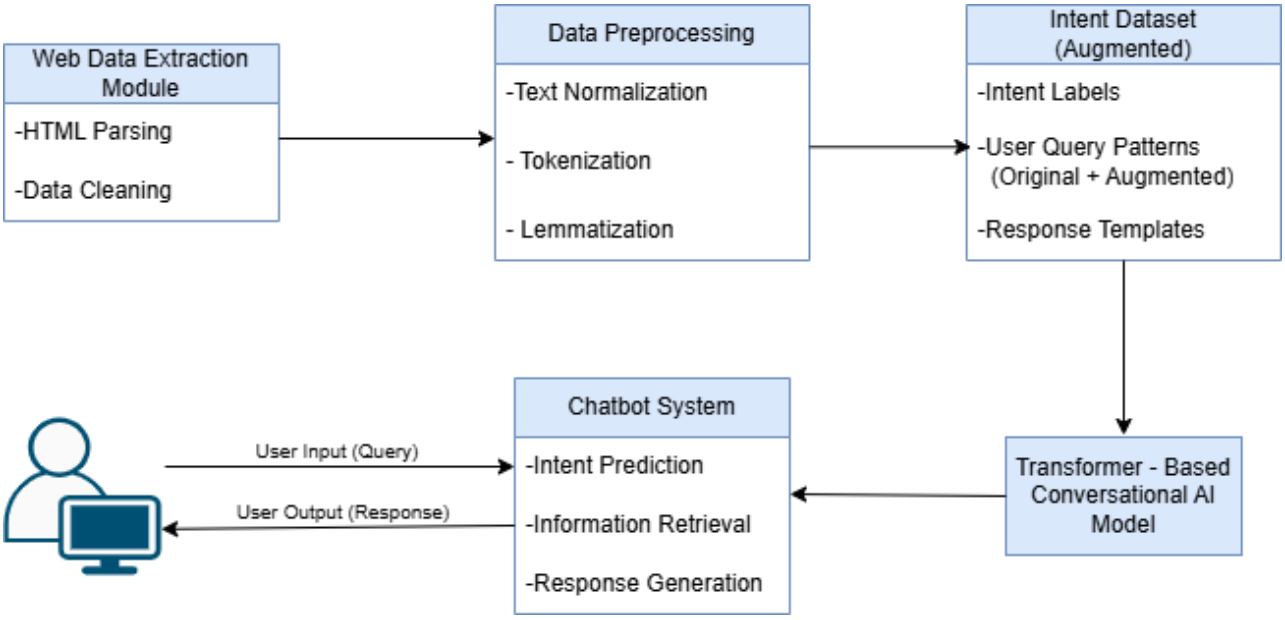

Figure 1: System Architecture

3.1 Web Scraping

The proposed system incorporates a web data extraction module to acquire relevant information from web sources in an automated manner. This module operates by sending HTTP requests to retrieve webpage content and parsing the underlying HTML structure to identify meaningful textual elements. During this process, irrelevant components such as scripts, styles, and redundant elements are removed to ensure that only informative content is retained. The extracted data is then cleaned and organized into a structured representation, making it suitable for natural language processing tasks. This processed information serves as a dynamic knowledge base for the chatbot, enabling it to generate accurate and context-aware responses. By automating the data extraction process, the system reduces the need for manual dataset creation while ensuring that the information used by the chatbot remains relevant and up to date.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 260-270, May-2026

https://doi.org/10.47001/IRJIET/2026.105036

3.2 Dataset

The dataset used in the proposed system is generated through automated extraction of web-based content using a dedicated data acquisition module. The extracted data consists of various forms of textual information, including headings, paragraphs, notices, and other structured elements relevant to the application domain. This raw data is collected and transformed into a structured format suitable for further processing. In addition to the extracted content, a custom intent-based dataset is maintained in JSON format, which includes intent labels, multiple variations of user query patterns, and corresponding response templates. The integration of extracted web data with structured intent information enables the system to handle both domain-specific and conversational queries effectively. The dataset is further preprocessed and utilized for training and validating the Transformer-based conversational model.

3.3 Data Processing

1. Normalization (Text Representation & Scaling)

In Natural Language Processing, raw text cannot be directly used as input to a neural network. Therefore, text is converted into numerical form using tokenization and embedding techniques.

Process:

Words are first converted into integer sequences using a tokenizer. These integers are then mapped into dense vector representations using an embedding layer.

Why Normalization is Needed:

Normalization ensures that input data is in a consistent numerical format, reduces variations, and improves training stability.

In this model, tokenization and padding ensure that all input sequences have a fixed length of 30. The embedding layer further transforms these inputs into a dense vector space, making them suitable for deep learning models.

2. Data Augmentation (Text Augmentation)

Deep learning models require large datasets, but chatbot datasets are often limited. To overcome this limitation, text augmentation techniques are used to artificially expand the dataset.

a) Synonym Replacement:

In this technique, words in a sentence are replaced with their synonyms using WordNet.

Example: “hello” becomes “hi”, and “help” becomes “assist”.
This increases dataset diversity and helps the model understand similar meanings.

b) Lemmatization:

Lemmatization converts words into their base or root form.
Example: “running” becomes “run”.
This reduces vocabulary size and improves consistency across the dataset.

c) Tokenization:

Tokenization splits sentences into individual words or tokens.
Example: “How are you” becomes “how”, “are”, “you”.
This step converts text into a machine-readable format.

d) Padding:

Padding ensures that all sequences have the same length. Short sequences are padded with zeros, while longer ones are truncated.
In this model, the maximum sequence length is 30.
This allows efficient batch processing and uniform input size.

e) Out-of-Vocabulary Handling (OOV Token):

Words not present in the training vocabulary are replaced with a special token <OOV>.
This improves the robustness of the model when handling unseen inputs.

3.4 Model Architecture

A custom Transformer-based deep learning model is designed for intent classification in the chatbot system. Unlike Convolutional Neural Networks (CNNs), which are used for image data, the Transformer architecture is specifically suited for Natural Language Processing tasks as it captures contextual relationships between words using attention mechanisms.

1. Input Layer: (30)

The input layer accepts a sequence of integers representing tokenized user input.

30 represents the maximum sequence length of a sentence. Each input sentence is converted into a sequence of tokens and padded to ensure uniform length. This layer does not perform any computation but defines the structure of the input data. It ensures that all inputs are consistent in size, which is necessary for batch processing in neural networks.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

2. Embedding Layer (Vocabulary Size × 256)

The embedding layer converts each integer token into a dense vector representation of 256 dimensions.Insteadof working with sparse representations, embeddings map words into a continuous vector space where semantically similar words are closer to each other.

Function:
Learns semantic relationships between words
Reduces dimensionality of input data
Improves contextual understanding

Result:
The input sequence is transformed into a dense matrix of size (30 × 256)

3. Transformer Block (Core Component)

The Transformer block is the most important part of the model and consists of multiple sub-components:

a) Multi-Head Self-Attention

This mechanism allows the model to focus on different parts of the sentence simultaneously. Each word attends to every other word in the sentence to understand context.

Technical Detail:
Uses 4 attention heads.
Each head learns different relationships.
Performs scaled dot-product attention.

Function:
Captures contextual dependencies.
Understands word relationships irrespective of position.

b) Layer Normalization + Residual Connection

Layer normalization is applied to stabilize the training process by maintaining consistent data distribution. Residual connections add the input of a layer to its output, preventing information loss and improving gradient flow.

Benefits:
Faster convergence.
Prevents vanishing gradient problem.
Improves model stability.

c) Feed Forward Network (Dense → GELU → Dense)

A fully connected neural network is applied after attention.

It consists of:
First Dense layer (higher dimension)
GELU activation function
Second Dense layer (projection back)

Function:
Learns complex patterns in data.
Adds non-linearity for better representation.

d) Dropout Layer

Dropout is applied within the Transformer block to randomly deactivate neurons during training.

Purpose:
Prevents overfitting.
Improves generalization.

Result of Transformer Block:
The input sentence is transformed into a context-aware representation, where each word understands its relationship with other words.

4. Global Average Pooling Layer

This layer converts the sequence output into a fixed-length vector by averaging across all time steps.

Function:
Reduces dimensionality
Summarizes entire sequence into one vector
Acts as a bridge between sequence learning and classification layers.

5. Dense Layer (128 Units, GELU)

A fully connected layer with 128 neurons is applied.Each neuron receives input from the pooled features.

Function:
Learns high-level abstract features.
Helps in final decision making.

GELU Activation:
Provides smooth non-linearity.
Improves performance compared to ReLU.

6. Dropout Layer (0.4)

A dropout rate of 40% is applied after the dense layer.

Purpose:
Reduces overfitting
Ensures robust learning

7. Output Layer (Softmax)

The final layer consists of neurons equal to the number of intent classes.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

Example:
Greeting, Admission, Fees, Courses, etc.

Softmax Activation Function:

$$Softmax(zi) = \frac{e^{zi}}{\sum_{j=1}^{N} e^{z_j}}$$

Converts output scores into probabilities across all classes.

Function:
Identifies most probable intent.
Enables multi-class classification.

3.5 Training Details

The proposed Transformer-based chatbot model is trained using supervised learning on preprocessed textual data. The training process involves optimization of model parameters using backpropagation and evaluation through validation data.

1. Loss Function: Sparse Categorical Crossentropy

Sparse categorical crossentropy is used for multi-class classification with integer-encoded labels.

$$L = -log(py)$$

Where:
py represents the predicted probability of the correct class.

Function:
Measures the difference between actual and predicted outputs
Encourages the model to assign higher probability to the correct intent.

2. Optimizer: Adam

The model is trained using the Adam optimizer, which combines momentum and adaptive learning rate techniques.

Mathematical Formulation:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t}$$

$$\widehat{v_t} = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta \widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon}$$

Function:
Provides faster and stable convergence.
Adapts learning rate automatically for each parameter.

3. Learning Rate

The Adam optimizer uses a default learning rate of approximately 0.001.

Function:
Controls the step size during weight updates.
Ensures stable and gradual convergence.

A smaller learning rate:
Prevents overshooting the minimum and improves model stability during training.

4. Batch Size

The model is trained using a batch size of 32.

Function:
Processes 32 samples before updating weights.

Benefits:
Balances computational efficiency and memory usage.
Provides smoother gradient updates.

5. Epochs

The model is trained for a maximum of 100 epochs.

Function:
Each epoch represents one complete pass over the dataset.

Note:
Early stopping with patience = 8 is applied to stop training when validation loss does not improve.

6. Data Split

The dataset is divided using a validation split of 0.2:

80% Training Data

20% Validation Data

Function:
Training data is used for learning.
Validation data is used to evaluate performance during training.

Purpose:
Prevents overfitting.
Ensures better generalization.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

IV. RESULTS AND DISCUSSION

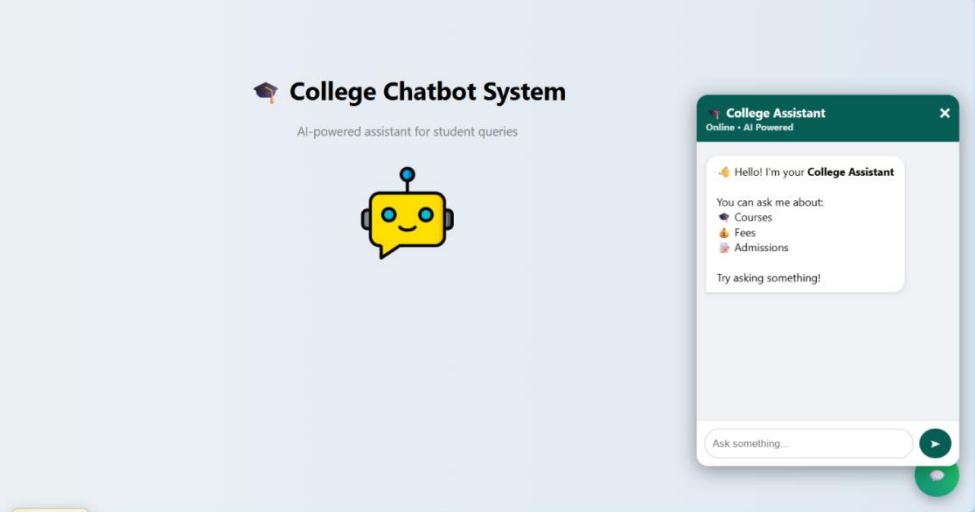

Figure 2: Chatbot UI

Above figure 2 illustrates the Chatbot User Interface of the College Chatbot System, which serves as the primary interaction point for users. Students can interact with the system through a chat-based environment by entering queries and receiving responses in a structured conversational format. The design is simple and user-friendly, requiring no technical expertise. The interface is integrated with the Flask backend, which processes user inputs and generates responses in real time, ensuring a smooth and efficient user experience.

1. Accuracy – Accuracy is the ratio of correctly predicted responses to the total number of inputs given to the system. It represents the overall effectiveness of the AI model in making correct predictions.

It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

In this system, the model achieved an accuracy of 94%, indicating that it correctly predicted the majority of user inputs.

2. Precision – Precision measures how many of the responses generated by the model are actually relevant and correct. It focuses on reducing incorrect or misleading responses.

It is calculated as:

$$Precision = \frac{TP}{TP + FP}$$

In this system, the model achieved a precision of 98.8%, indicating that it rarely produces incorrect or irrelevant responses.

3. Recall – Recall measures how effectively the system identifies and responds to all relevant user queries. It focuses on minimizing missed or unanswered queries.

It is calculated as:

$$Recall = \frac{TP}{TP + FN}$$

In this system, the model achieved a recall of 98.7%, indicating that most relevant user queries are successfully identified and addressed.

4. F1-Score – F1-Score is the harmonic mean of Precision and Recall. It balances the trade-off between giving correct responses and covering all queries.

It is calculated as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In this system, the model achieved an F1-score of 98.7%, indicating an excellent balance between response accuracy and query coverage.

5. AUC Score – AUC (Area Under the ROC Curve) evaluates how well the model distinguishes between correct and incorrect responses or between different intent classes.

In this system, the model achieved an AUC score of 97%, indicating that it is highly capable of correctly identifying user intent and generating appropriate responses.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 10, Issue 5, pp 260-270, May-2026
https://doi.org/10.47001/IRJIET/2026.105036

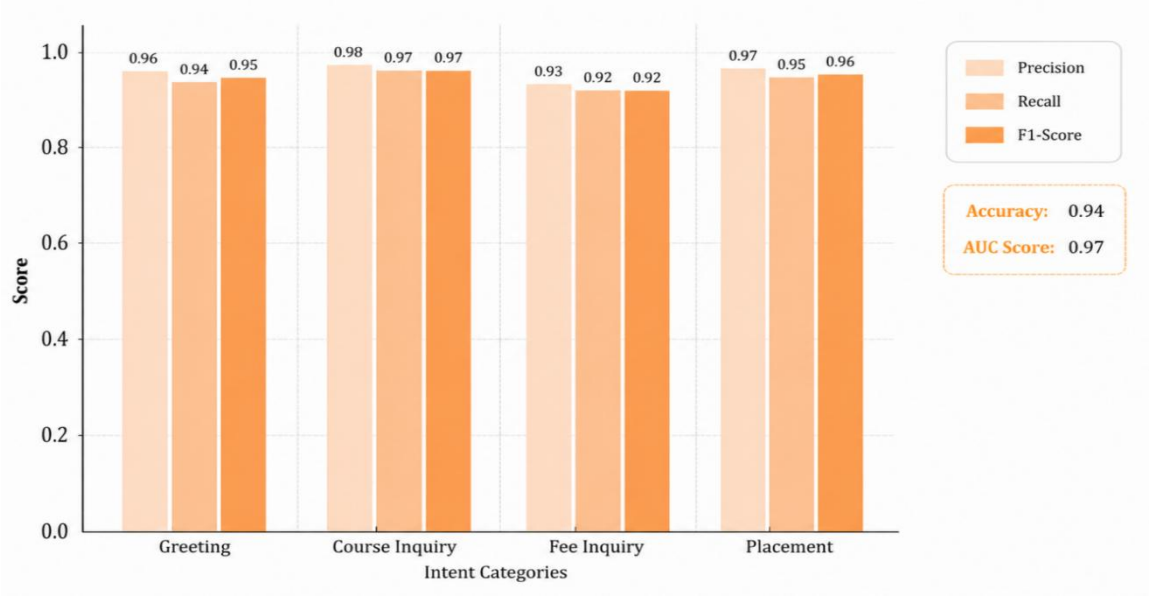

Figure 3: Training Graph

V. CONCLUSION

The Omniverse AI system provides an efficient and automated solution for converting unstructured web data into structured, meaningful, and queryable information. Through the integration of web scraping, content preprocessing, structured data storage, machine learning model training, and an interactive chatbot interface, the system ensures a complete end-to-end information processing pipeline. The implementation demonstrates that automated knowledge extraction and natural language question answering can be effectively combined to support academic, research, and organizational applications. Overall, the system achieves its primary objective of enabling users to given a URL, extract relevant information, and obtain accurate responses to their queries with minimal manual effort.

REFERENCES

[1] M. A. Russell, "Web Scraping Techniques for Data Extraction and Analysis," 2019.
[2] V. Crescenzi, "Automatic Web Data Extraction Using DOM Tree," 2001.
[3] J. Srivastava, "Web Content Mining: Tools, Techniques and Applications," 2000.
[4] N. Kushmerick, "Information Extraction from Web Documents Using Machine Learning," 1997.
[5] Y. LeCun and Y. Bengio, "Deep Learning for Natural Language Processing," 2015.
[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.
[7] J. Masche and N. Le, "A Survey on Chatbot Systems," 2017.
[8] R. Feldman, "Text Mining and Analysis Techniques," 2007.
[9] "Scrapy: A Web Crawling Framework," 2015. [Online]. Available: https://docs.scrapy.org
[10] "Selenium for Dynamic Web Scraping," 2018. [Online]. Available: https://www.selenium.dev
[11] O. Kolomiyets and M.-F. Moens, "A Survey on Question Answering Systems," 2011.
[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
[13] X. Chu et al., "Machine Learning Approaches for Data Cleaning," 2016.
[14] J. Dean and S. Ghemawat, "Big Data Analytics for Web Data," 2014.
[15] P. Mika, "Knowledge Extraction from Web Data," 2012.
[16] D. Jurafsky and J. H. Martin, "Natural Language Understanding in AI Systems," 2019.
[17] S. García, J. Luengo, and F. Herrera, "Data Preprocessing Techniques in Machine Learning," 2015.
[18] A.Sweigart, "Web Automation using Python," 2020.
[19] S. Shawar and E. Atwell, "AI-based Chatbot for Information Retrieval," 2021.
[20] H. Chen et al., "Hybrid Models for Intelligent Data Processing," 2022.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 10, Issue 5, pp 260-270, May-2026

https://doi.org/10.47001/IRJIET/2026.105036

Citation of this Article:

Varun Patel, Roshani Chaudhari, Bhavesh Y. Patil, Bhavesh R. Patil, & Rijavan A. Shaikh. (2026). Omniverse AI: The Universal Web Intelligence Platform Using AI. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(5), 260-270. Article DOI https://doi.org/10.47001/IRJIET/2026.105036

****

