WebPedia 2.0: LLMs to Revolutionize Wikipedia Content Creation, Automating Knowledge Curation with Generative AI

Dr. M.K. Jayanthi Kannan[1], Anurodh Pancholi[2], Priyaanshu Singh[3]

[1]Professor, School of Computing Science, Engineering and Artificial Intelligence, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114
[2,3]Student, School of Computing Science, Engineering and Artificial Intelligence, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114

Abstract:
The WEBPEDIA (Synthesis of Topic-Oriented Role-based Multi-document Summarization) system is designed to produce structured summaries from multiple documents by organizing content according to topic and the roles of various stakeholders (e.g., government, media, public) involved in an event. The implementation begins with a Document Collection module that gathers relevant texts from different sources discussing the same event, ensuring a diverse, multi-perspective input. Next, the Topic Segmentation module uses techniques like TextRank and semantic clustering to break the content into coherent topic segments. The Role Identification module employs Named Entity Recognition (NER), dependency parsing, and rule-based methods to map entities to corresponding roles, such as identifying a political figure as part of the government perspective. Once roles are identified, the Role-based Content Selection module uses TF-IDF scores, semantic similarity, and topic-role alignment techniques to filter and select content that is most relevant to each identified role, removing redundant or off-topic text. For summary generation, WEBPEDIA combines template-based methods with neural text generation models such as BART or T5, which helps maintain fluency and coherence in the output while adhering to factual and logical flow. The final summary combines each role's perspective into a comprehensive multi-document, multi-role narrative. For evaluation, the authors used datasets derived from real-world news coverage of complex events such as elections, pandemics, and protests. WEBPEDIA's performance was assessed using ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L), as well as qualitative metrics like coverage and diversity to evaluate how well the summaries reflected all relevant roles and topics. Human evaluations further measured the fluency, coherence, and role accuracy of the summaries. The results showed that WEBPEDIA significantly outperforms traditional summarization baselines by generating summaries that are not only informative and factually consistent but also nuanced with multiple stakeholder perspectives, showcasing its effectiveness in complex information environment.

Keywords: WebPedia 2.0, LLMs to Revolutionize Wikipedia, Content Creation Wikipedia, Automating Knowledge Curation, Generative AI, WEBpedia, Redefining Wikipedia, Crafting Wikipedia Articles with AI, Large Language Models.

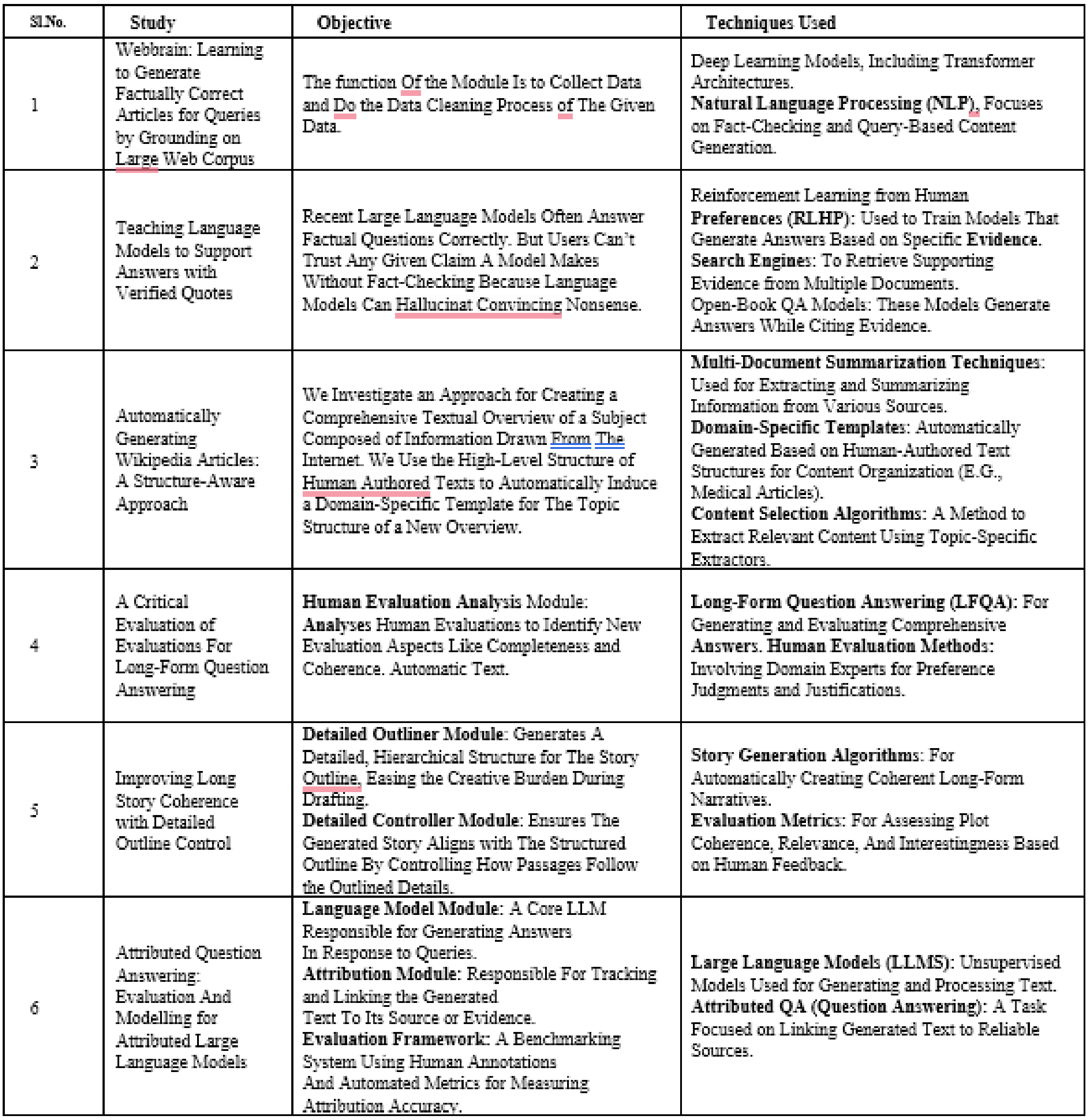
INTRODUCTION

Large language models (LLMs) have shown remarkable writing abilities, yet their potential for crafting grounded, long-form articles, such as comprehensive Wikipedia pages, remains uncertain. Document Collection and Preprocessing: Implement a web crawler or use APIs to collect multiple news articles related to the same event from diverse sources. Normalize text by cleaning, tokenizing, and performing sentence segmentation. Store preprocessed documents in a structured format for downstream processing. Topic Segmentation: Apply Text Rank or semantic clustering algorithms to segment documents into coherent topic blocks. Group similar segments across documents to identify dominant discussion themes. Use topic boundaries to guide content extraction aligned with user or system-defined event structure. Role Identification, Use Named Entity Recognition (NER) to detect key entities like organizations, people, and locations. Employ dependency parsing and role-based heuristics to classify each entity into a stakeholder role (e.g., government, media, public). Maintain a mapping of roles to entity mentions for selective content extraction. Role-based Content Selection: Apply TF-IDF and semantic similarity measures to select sentences most relevant to each identified role. Use redundancy removal techniques (e.g., Maximal Marginal Relevance) to prevent duplicate information. Ensure extracted content represents a balanced view of each stakeholder. Develop template-based and neural summarization pipelines using models like BART or T5. Generate summaries that align with topic segments and role-based perspectives. Use language modeling to enhance fluency, coherence, and factual accuracy of generated content. Evaluation and Analysis: Perform automated evaluation using ROUGE-1, ROUGE-2, and ROUGE-L metrics. Conduct human evaluations to assess fluency, coherence, factual consistency, and stakeholder coverage. Analyze trade-offs between extractive and abstractive performance in different real-world scenarios. We address these challenges by focusing on generating Wikipedia-like articles from scratch, breaking the task into two key phases. The first phase involves researching to create an outline—a structured list of multi-level sections—and gathering a set of reference documents. The second phase uses this outline and references to craft the full-length article. This task decomposition reflects the human writing process, which typically includes pre-writing, drafting, and revising stages (Rohman, 1965; Munoz-Luna, 2015).

THE DOMAIN ANALYSIS AND LITERATURE REVIEW OF PROPOSED RESEARCH

Generating high-quality, Wikipedia-style articles from scratch remains a significant challenge for large language models (LLMs). Although these models contain vast amounts of parametric knowledge, relying solely on this internal information often results in content that is shallow, occasionally inaccurate, and particularly weak when addressing long-tail or niche topics. Human writers, by contrast, engage in a thoughtful, multi-stage process that includes comprehensive research, the development of structured outlines, iterative drafting, and meticulous refinement. This process allows them to surface nuanced insights, synthesize information from diverse perspectives, and produce articles that are both informative and contextually rich—capabilities that LLMs, even the most advanced, continue to struggle with. Attempts to bridge this gap through retrieval-augmented generation (RAG) have shown promise by introducing external data into the generation pipeline, but these methods often fall short during the critical pre-writing phase. They typically surface generic facts rather than deep, multifaceted context, and they rarely facilitate the kind of conceptual planning or source triangulation that good writing requires. Moreover, instruction-tuned models that can be prompted to generate research questions tend to focus on low-level, fact-seeking queries that fail to stimulate thorough exploration or guide a structured research process.

Table 1: Literature Review of WEBpedia



Sl.No.	Study	Objective	Techniques Used
1	Webbrain: Learning to Generate Factually Correct Articles for Queries by Grounding on Large Web Corpus	The function Of the Module Is to Collect Data and Do the Data Cleaning Process of The Given Data.	Deep Learning Models, Including Transformer Architectures. **Natural Language Processing (NLP)**, Focuses on Fact-Checking and Query-Based Content Generation.
2	Teaching Language Models to Support Answers with Verified Quotes	Recent Large Language Models Often Answer Factual Questions Correctly. But Users Can't Trust Any Given Claim A Model Makes Without Fact-Checking Because Language Models Can Hallucinat Convincing Nonsense.	Reinforcement Learning from Human **Preferences (RLHP)**: Used to Train Models That Generate Answers Based on Specific **Evidence**. **Search Engines**: To Retrieve Supporting Evidence from Multiple Documents. Open-Book QA Models: These Models Generate Answers While Citing Evidence.
3	Automatically Generating Wikipedia Articles: A Structure-Aware Approach	We Investigate an Approach for Creating a Comprehensive Textual Overview of a Subject Composed of Information Drawn From The Internet. We Use the High-Level Structure of Human Authored Texts to Automatically Induce a Domain-Specific Template for The Topic Structure of a New Overview.	**Multi-Document Summarization Techniques**: Used for Extracting and Summarizing Information from Various Sources. **Domain-Specific Templates**: Automatically Generated Based on Human-Authored Text Structures for Content Organization (E.G., Medical Articles). **Content Selection Algorithms**: A Method to Extract Relevant Content Using Topic-Specific Extractors.
4	A Critical Evaluation of Evaluations For Long-Form Question Answering	**Human Evaluation Analys**is Module: **Analy**ses Human Evaluations to Identify New Evaluation Aspects Like Completeness and Coherence. Automatic Text.	**Long-Form Question Answering (LFQA)**: For Generating and Evaluating Comprehensive **Answers**. **Human Evaluation Methods**: Involving Domain Experts for Preference Judgments and Justifications.
5	Improving Long Story Coherence with Detailed Outline Control	**Detailed Outliner Module**: Generates A Detailed, Hierarchical Structure for The Story Outline, Easing the Creative Burden During Drafting. **Detailed Controller Module**: Ensures The Generated Story Aligns with The Structured Outline By Controlling How Passages Follow the Outlined Details.	**Story Generation Algorithms**: For Automatically Creating Coherent Long-Form Narratives. **Evaluation Metrics**: For Assessing Plot Coherence, Relevance, And Interestingness Based on Human Feedback.
6	Attributed Question Answering: Evaluation And Modelling for Attributed Large Language Models	**Language Model Module**: A Core LLM Responsible for Generating Answers In Response to Queries. **Attribution Module**: Responsible For Tracking and Linking the Generated Text To Its Source or Evidence. **Evaluation Framework**: A Benchmarking System Using Human Annotations And Automated Metrics for Measuring Attribution Accuracy.	**Large Language Models (LLMS)**: Unsupervised Models Used for Generating and Processing Text. **Attributed QA (Question Answering)**: A Task Focused on Linking Generated Text to Reliable Sources.

MOTIVATION AND OBJECTIVE OF WebPedia REDEFINING Wikipedia

The objective of this work is to develop a systematic approach for generating high-quality, Wikipedia-like articles from scratch by enhancing the pre-writing capabilities of large language models (LLMs). Specifically, the work aims to, automate the Pre-Writing Process: Design a framework that enables LLMs to conduct structured research, ask meaningful and diverse questions, and generate comprehensive outlines grounded in external sources. Introduce the WEBPEDIA Paradigm: Propose a novel multi-stage system—Synthesis of Topic Outlines through Retrieval and Multi-perspective Question Asking (WEBPEDIA)—that simulates human-like research by retrieving relevant content, generating perspective-driven questions, and iteratively refining topic understanding. Leverage External Information Sources: Overcome the limitations of parametric knowledge by integrating reliable external information through retrieval-based methods. Create and Utilize a New Benchmark: Develop the UtliedWIKI dataset comprising recent Wikipedia articles to fairly evaluate outline and article generation, minimizing the risk of training data leakage. Establish Evaluation Criteria: Define both automatic and human-centered metrics to assess the quality, breadth, and coherence of generated outlines and articles. Identify and Analyse Limitations: Highlight current challenges such as handling internet bias and avoiding hallucinated connections, providing insights for future research in grounded language generation. **Building the Future of Knowledge with Large Language Models,** WEBpedia seeks to fill this gap by focusing on pre-writing tasks that go beyond basic prompting techniques, allowing LLMs to perform more effectively in research-intensive and organizational roles. The project also aims to improve the depth and breadth of knowledge captured in generated content. Wikipedia articles are distinguished by their comprehensive coverage and evidence-backed synthesis, which requires integrating information from diverse perspectives. WEBpedia addresses the challenge of moving beyond surface-level generation to create articles that are both thorough and grounded in reliable references. Beyond its research goals, the WEBpedia framework has significant real-world applications in areas like education, journalism, and content marketing. These fields often require users to produce factual, organized, and comprehensive long-form content, even when they lack extensive expertise. By providing tools to streamline this process, the project has the potential to transform how individuals and organizations approach content creation.

WEBpedia Redefining Wikipedia, A central goal is the introduction of the WEBpedia framework, which includes perspective-guided question generation and outline-driven content creation. By leveraging large language models (LLMs), the framework simulates multi-perspective conversations and iterative research processes to uncover diverse viewpoints on a given topic. Additionally, it facilitates the systematic creation of detailed, multi-level outlines to guide the writing process effectively. Another key focus is the development and utilization of the UtliedWiki dataset, a curated collection of recent, high-quality Wikipedia articles. This dataset is designed to mitigate data leakage while ensuring relevance in evaluation. It also serves as a foundation for establishing metrics and benchmarks that assess the quality of generated outlines and final articles, allowing for meaningful comparisons with human-written content. WEBpedia also addresses real-world challenges in content generation by emphasizing grounded and factually accurate article creation, a critical requirement for Wikipedia-like entries. **Crafting Wikipedia Articles with AI-based Large Language Model,** The contributes valuable tools and insights to the field of automated writing systems, fostering advancements in both research and practical applications. **The main Objectives of WEBpedia include:** Evaluating the capacity of **LLM systems to generate** long-form grounded articles from scratch, and the pre-writing challenge in particular, we curate the UtliedWIKI dataset and establish evaluation criteria for both outline and final article quality. We **propose WEBPEDIA**, a novel system that automates the pre-writing stage. WIKIPEDIA researches the topic and creates an outline by using LLMs to ask incisive questions and retrieve trusted information from the Internet. Both automatic and human evaluation demonstrate the effectiveness of our approach. Expert feedback further reveals new challenges in generating grounded long-form articles.

THE FUNCTIONAL MODULES AND PROTOTYPE OF WebPedia

To address the challenge of generating grounded, high-quality Wikipedia-style articles, the authors propose WEBPEDIA—Synthesis of Topic Outlines through Retrieval and Multi-perspective Question Asking—a multi-stage system designed to simulate human-like research and writing workflows. The process begins with Topic Analysis and Perspective Discovery, where the system retrieves related

Wikipedia articles to identify and extract diverse perspectives surrounding a given topic. These perspectives serve as the foundation for guiding the subsequent question-generation phase, ensuring that the research is multi-dimensional and contextually rich. In the Perspective-Guided Question Asking stage, the system simulates dialogues between a Wikipedia writer (powered by an LLM) and a domain expert. The writer, informed by a particular perspective, poses targeted questions, while the answers are grounded in online sources that are retrieved and filtered to ensure reliability. This leads into the Conversation-Guided Research phase, where multi-turn exchanges allow the LLM to iteratively refine its understanding of the topic. During these exchanges, the system performs search-query decomposition, document retrieval, filtering based on Wikipedia's sourcing standards, and synthesis of trusted responses to ensure depth and factual accuracy.

Outline Generation begins with a draft produced using the LLM's internal knowledge. This initial outline is then refined and expanded using insights gathered from the multi-perspective conversations, resulting in a comprehensive, well-structured, and high-coverage outline. In the Full-Length Article Writing phase, the system uses the final outline along with the collected references to generate the article section by section. Semantic retrieval techniques such as Sentence-BERT are employed to gather documents relevant to each section, enabling the generation of text with proper in-text citations, content deduplication, and a synthesized summary or lead paragraph. Finally, Evaluation is conducted using a newly created benchmark dataset called UtliedWIKI, which comprises recent Wikipedia articles to avoid issues of training data overlap. The system's outputs are assessed using a combination of automatic metrics—such as heading recall based on semantic similarity and entity coverage—and expert human evaluations provided by experienced Wikipedia editors. Through this comprehensive workflow, WEBPEDIA aims to replicate the depth, accuracy, and structure characteristic of high-quality human-authored encyclopedia entries.

The primary goal of this work is to automate the generation of grounded, long-form Wikipedia-like articles from scratch, with a particular focus on the pre-writing stage. The system aims to enable large language models (LLMs) to effectively research topics, create comprehensive outlines, and compose full-length articles based on reliable references, rather than relying solely on their parametric knowledge. The primary goal of this work is to develop a comprehensive, automated system capable of generating grounded, long-form Wikipedia-style articles entirely from scratch, with a strong emphasis on enhancing the often-overlooked but critically important pre-writing stage of the content creation process. Rather than depending solely on the internal, parametric knowledge stored within large language models (LLMs)—which often leads to shallow, generic, or even hallucinated outputs—this approach focuses on enabling LLMs to engage in structured and intelligent topic research, the generation of rich, multi-perspective questions, and the construction of detailed, hierarchical outlines. These outlines act as a scaffold to guide the content generation process, ensuring that the resulting articles are not only coherent and well-organized but also factually accurate and deeply informative. The system integrates external knowledge sources through retrieval-augmented techniques, filters them according to trustworthiness criteria, and uses them to ground every stage of the article's development. By simulating research and writing process that closely mirrors how expert human contributors' approach encyclopedic writing—starting from topic analysis, followed by exploration through guided questioning, retrieval and synthesis of relevant information, outline refinement, and finally, structured article drafting—the proposed solution seeks to significantly raise the quality, depth, and factual rigor of machine-generated content.

IMPLEMENTATION AND PERFORMANCE ANALYSIS

Functional Modules of WEBPEDIA: The WEBPEDIA system architecture is composed of a series of interconnected modules, each responsible for a specific stage in the process of generating grounded, Wikipedia-style articles. The pipeline begins with the Topic Input & Initialization Module, which serves as the entry point for the system. It takes a concise topic phrase, such as “2022 Winter Olympics Opening Ceremony,” and validates the input through a simple user interface or API. Upon successful validation, it initializes the data structure needed to guide the writing task across subsequent modules. Next, the Related Articles & Perspective Mining Module retrieves a set of related Wikipedia articles to extract diverse viewpoints on the topic. This involves an Article Retriever that queries the Wikipedia API, a Table of Contents (TOC) Extractor that parses article structures, and a Perspective Generator that uses a large language model (LLM) to infer distinct perspectives from the TOCs. The result is a set of guiding perspectives $\{p_0, p_1, ..., p_n\}$ that form the backbone for in-depth exploration of the topic.

Building on these, the Perspective-Guided Question Generator creates targeted research questions for each identified perspective. By simulating a role-based agent for each viewpoint, this module formulates probing, multi-dimensional questions that guide deeper inquiry. These questions then feed into the Conversational Simulation Module, which mimics a research conversation between a Wikipedia writer and a grounded expert. The conversation is facilitated through several subcomponents: a Question Splitter that decomposes complex queries into simpler search terms, a Search Query Engine that interfaces with the web or a local knowledge base, a Source Evaluator that filters results using Wikipedia’s trustworthiness guidelines, and an Answer Synthesizer that composes grounded responses. The resulting exchange, $\{q_1, a_1, q_2, a_2, ..., q_m, a_m\}$, forms a rich dialogue per perspective, with all validated sources stored in a reference pool R.

PROPOSED SYSTEM DESIGN OF WebPedia

The research conversations in place, the Outline Drafting and Refinement Module takes over. It begins by generating a coarse initial outline using the LLM’s intrinsic knowledge of the topic. This outline is then significantly enhanced by integrating insights extracted from all the simulated conversations, producing a comprehensive and multi-level outline O. Each section and subsection in this outline is then mapped to relevant documents from the reference pool through the Document Retrieval for Section Writing Module. Here, section headings are encoded using Sentence-BERT embeddings and matched against reference content using cosine similarity, resulting in a section-specific document bundle for content generation. The Full-Length Article Generator composes the actual article. It uses the refined outline and retrieved documents to generate well-cited section text. This module includes a Section Writer that handles content creation, a De-Duplicator that ensures consistency and eliminates redundancy across sections, and a Summary Generator that constructs the lead section of the article. The final output is a full-length, structured, Wikipedia-style article S, complete with in-text citations and references. Finally, the Evaluation and Metrics Module assesses the quality of the generated content by comparing the system output against human-written gold standards. Using metrics like Heading Soft Recall and Heading Entity Recall, and leveraging tools such as Sentence-BERT and FLAIR NER, this module quantifies structural, semantic, and factual alignment. Together, these modules form a robust pipeline for producing reliable, high-coverage, and well-grounded articles that mirror the depth and accuracy of human-authored Wikipedia entries.

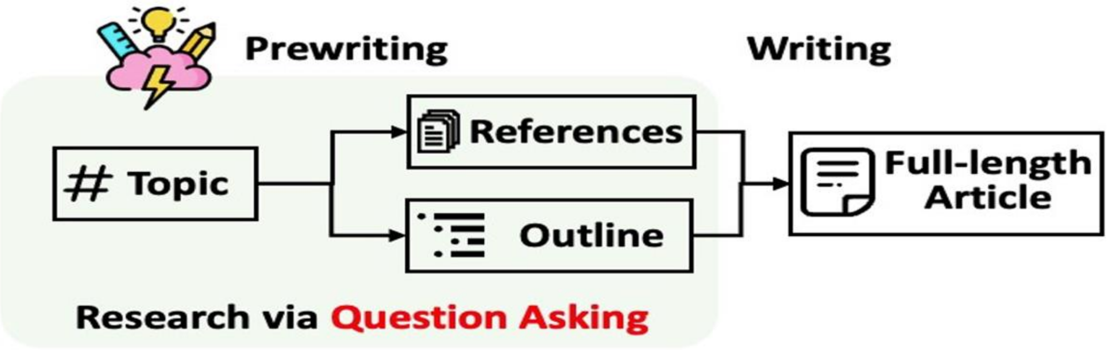


Figure 1: The Architectural Diagram of WebPedia to Redefine Wikipedia

ARCHITECTURE DIAGRAM OF WEBPpedia REDEFINING Wikipedia

The architecture diagram of the STORM system illustrates how it automates the pre-writing and writing stages of generating grounded, long-form articles. Here's an explanation of each component and its function within the system:

Input Topic: The process begins with a given topic. This input is the foundation for the subsequent research and content generation.

Perspective-Guided Question Asking: The system first discovers multiple perspectives by surveying Wikipedia articles on related topics. These perspectives guide the question-asking process by prompting the system to generate more focused, in-depth questions about the topic. Each perspective contributes to a more comprehensive understanding of the topic by directing the question-asking from various angles (e.g., basic facts, in-depth technical details, historical context).

Simulated Conversations: The system simulates conversations between a Wikipedia writer and an expert. In each conversation round, the LLM generates a question based on its perspective and conversation history, with the expert providing grounded answers based on trusted online sources. This conversation process is dynamic, as each answer can generate follow-up questions, iterating to collect more detailed and comprehensive information. These conversations help the system gather diverse and detailed information on the topic, which is essential for creating an accurate and well-rounded outline.

Curating References: During the simulated conversations, the system retrieves relevant online sources based on the queries and evaluates their trustworthiness. These sources form a reference pool that will later be used to generate the full-length article.

Outline Creation: The system creates an initial draft outline ($\mathcal{O}$D) based only on the topic. The outline is then refined using the perspectives and the simulated conversations, resulting in a comprehensive and organized outline ($\mathcal{O}$). The outline serves as the roadmap for the final article and ensures that all necessary sections are covered.

Writing the Full-Length Article: Based on the refined outline and gathered references, the system writes the article section by section. For each section, relevant documents from the reference pool are retrieved, and the LLM generates the section content with citations from these sources. The article is generated in parallel for different sections and then concatenated together. The system ensures coherence by eliminating repeated information and synthesizing a lead section summarizing the entire article, in line with Wikipedia's writing style.

Final Output: The result is a full-length, grounded Wikipedia-like article that adheres to Wikipedia's structure and writing norms, with a comprehensive coverage of the topic. The diagram effectively captures the flow of processes from the input topic to the final article, showcasing how STORM automates research, question-asking, and article generation while maintaining high standards of quality and organization.

METHODOLOGY AND ALGORITHMS PROPOSED FOR WEBPpedia

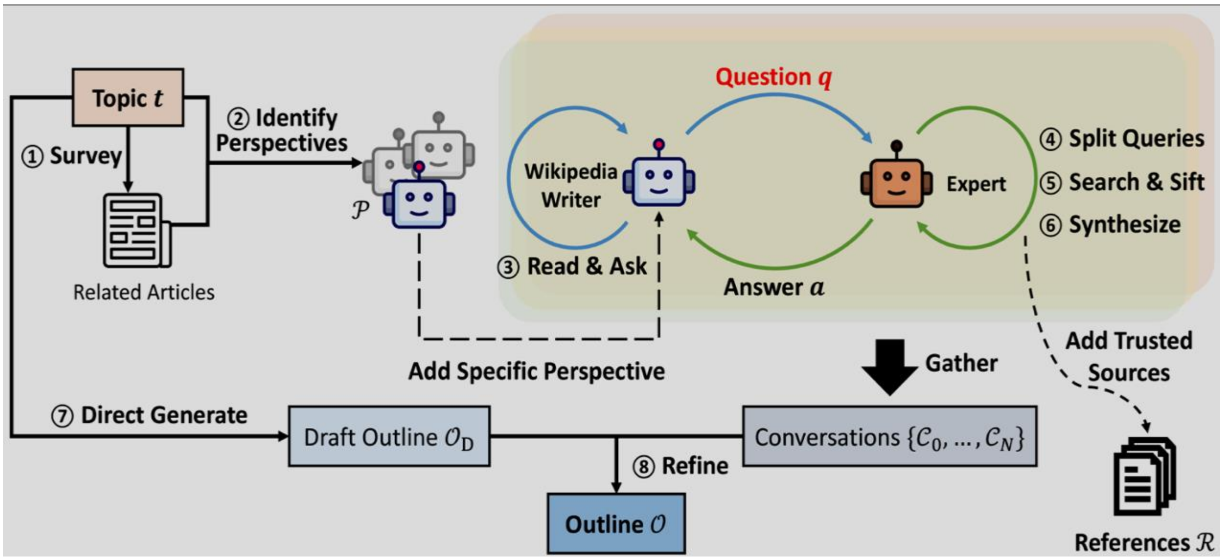


Figure 2: WebPedia to automate the Pre-Writing

Figure 3 illustrates the WEBpedia that automated the prewriting, starting with a given topic, WEBPEDIA identifies various perspectives on covering the topic by surveying related Wikipedia articles. It then simulates conversations between a Wikipedia writer who asks questions guided by the given perspective and an expert grounded on trustworthy online sources. The final outline is curated based on the LLM's intrinsic knowledge and the gathered conversations from different perspectives.

PROJECT MODULES AND PHASES OF IMPLEMENTATION OF WEBpedia

Retrieval-augmented generation (RAG) frameworks represent an important step forward by integrating external sources to ground the generation process. However, these systems are typically focused on providing factual support during the writing stage, while neglecting the equally critical pre-writing phase. Effective human writing, especially in informational domains like Wikipedia, involves a structured workflow: conducting extensive background research, formulating focused and multifaceted questions, synthesizing multiple perspectives, and crafting a coherent outline before drafting. Current RAG systems and instruction-tuned LLMs fall short in replicating this pre-writing workflow. The questions they generate are often overly simplistic or fact-seeking, rarely encouraging the kind of nuanced exploration necessary for well-rounded content. Without the ability to pose deep, exploratory questions and reason across diverse information sources, LLMs remain ill-equipped to produce articles that are both informative and intellectually rigorous. To address these limitations, there is a growing need for systems that go beyond superficial prompting and instead emulate human-like research behaviors. Such systems must be capable of initiating and sustaining structured inquiry—generating insightful, open-ended questions, seeking information from multiple credible sources, and iteratively refining their understanding of a topic. Crucially, this research-driven approach should culminate in the construction of a detailed, logically organized outline that captures the breadth and depth of the subject matter. By enabling LLMs to engage in this comprehensive pre-writing process, we can significantly improve the quality of generated long-form content. This work aims to fill that gap by developing methods that empower LLMs to perform topic exploration, research question generation, and outline synthesis in ways that mirror human writing workflows. Ultimately, such capabilities would move us closer to automating the production of accurate, thorough, and well-structured Wikipedia-style articles across a broad range of domains.

The objective of this work is to develop a systematic and robust approach for generating high-quality, Wikipedia-style articles from scratch by significantly enhancing the pre-writing capabilities of large language models (LLMs). Unlike traditional generation techniques that rely heavily on parametric knowledge and simple prompting, this work focuses on equipping LLMs with tools and strategies that closely emulate the structured and iterative research process followed by human writers. To achieve this, the first goal is to automate the pre-writing process by designing a framework that allows LLMs to conduct structured, multi-perspective research, formulate meaningful and diverse questions, and synthesize comprehensive topic outlines grounded in information retrieved from external sources. This process ensures that the models are not simply generating text from memory, but rather engaging with real-world, verifiable content. Central to this approach is the introduction of the WEBPEDIA paradigm Synthesis of Topic Outlines through Retrieval and Multi-perspective Question Asking a novel multi-stage system that simulates the human research process. WEBPEDIA guides LLMs through a series of structured steps, including retrieving relevant documents, generating in-depth and perspective-rich questions, and refining their understanding of a topic through iterative interactions with the information landscape. This paradigm aims to replicate how human writers explore a subject from multiple angles, identify gaps in knowledge, and incrementally build a coherent and detailed outline before drafting content.

Another core objective is to overcome the inherent limitations of parametric memory by integrating external information sources. Through retrieval-based methods, the system grounds its outputs in reliable, up-to-date content, reducing hallucinations and enhancing factuality. To ensure fair and realistic evaluation of the proposed system, the work also introduces a new benchmark dataset, UtliedWIKI, which consists of recently published Wikipedia articles that are unlikely to have been part of any LLM's training data. This benchmark serves as a rigorous test bed for assessing the capabilities of outline and article generation systems without the confounding influence of data leakage. Moreover, the project defines comprehensive evaluation criteria that combine automatic metrics—such as factual accuracy, coverage, and structure—with human-centered assessments of clarity, coherence, and informativeness. This dual approach to evaluation enables a more nuanced and reliable understanding of system performance. Finally, the work also aims to identify and analyze the current limitations in grounded language generation, such as the difficulty of handling biased or conflicting internet content, and the risk of generating hallucinated connections between unrelated facts. By documenting these challenges, the work provides valuable insights and lays the groundwork for future research focused on developing more reliable, human-aligned writing systems powered by LLMs.

This work addresses the challenge of generating grounded, high-quality Wikipedia-like articles from scratch using large language models (LLMs). Traditional approaches that rely solely on the models' internal knowledge often produce shallow or inaccurate content, especially on niche topics. Retrieval-augmented generation (RAG) improves grounding but struggles with the crucial pre-writing phase, such as research and outline creation. To overcome these limitations, the authors introduce WEBPEDIA—Synthesis of Topic Outlines through Retrieval and Multi-perspective Question Asking—a novel framework that mimics the human writing process. WEBPEDIA operates in multiple stages: it retrieves relevant documents, generates diverse and in-depth questions from different perspectives, and iteratively gathers information to construct detailed article outlines. To support this approach, the team curated

UtliedWIKI, a dataset of recent Wikipedia articles designed to avoid overlap with pretraining data. Evaluation using both automated metrics and expert Wikipedia editors shows that WEBPEDIA outperforms baseline methods, especially in breadth, organization, and relevance of content. The study also surfaces key challenges such as dealing with internet bias and preventing fabricated connections future directions for grounded long-form text generation.

Front-End Implementation, UI Design uses Figma to design an intuitive user interface. Create high-fidelity wireframes for key user flows. Establish a clean and minimal visual style. Streamlit Implementation, Set up Streamlit project structure. Create UI components (text input, file uploader, etc.). Display real-time responses from the LLaMA backend. Implement session state handling for user interaction history. Add responsiveness and basic styling using Streamlit themes or CSS hacks. **Back-End Implementation,** LLaMA Backend Integration, Set up Python environment with required libraries (transformers, torch, etc.). Load and configure the LLaMA model (e.g., via Hugging Face or local inference). Implement a Fast API/Flask server to expose the model via REST API. Add CORS middleware to allow front-end access. API Endpoint Design, Create a POST endpoint to receive prompt and return response. Add request validation and error handling. Optimize response generation with streaming/chunked output (if supported). Performance Optimization, Enable model quantization or GPU acceleration (e.g., via CUDA). Use batching or caching for frequently asked prompts. Monitor memory and compute usage. **Authentication (Optional),** User Management, Add login/signup system with JWT authentication. Store user sessions and logs (e.g., SQLite or Firebase). Rate-limit API access per user. **Testing and Validation,** Unit Testing, use pytest for backend unit tests. Validate Streamlit UI elements with Streamlit. testing utilities. Integration Testing, Test end-to-end flow: prompt input → backend inference → response rendering. Simulate edge cases and error scenarios. **Deployment,** Backend Deployment- Deploy API using services like AWS EC2, Azure, or Railway. Set up Gunicorn/Uvicorn server for FastAPI. Use Nginx as a reverse proxy if needed. Front-End Deployment, Deploy Streamlit app via Streamlit Community Cloud or Docker container. Connect the frontend with the backend via secure API endpoints. Ensure scalability and logging for production use.

THE PROTOTYPE AND PROGRAMMING LOGIC

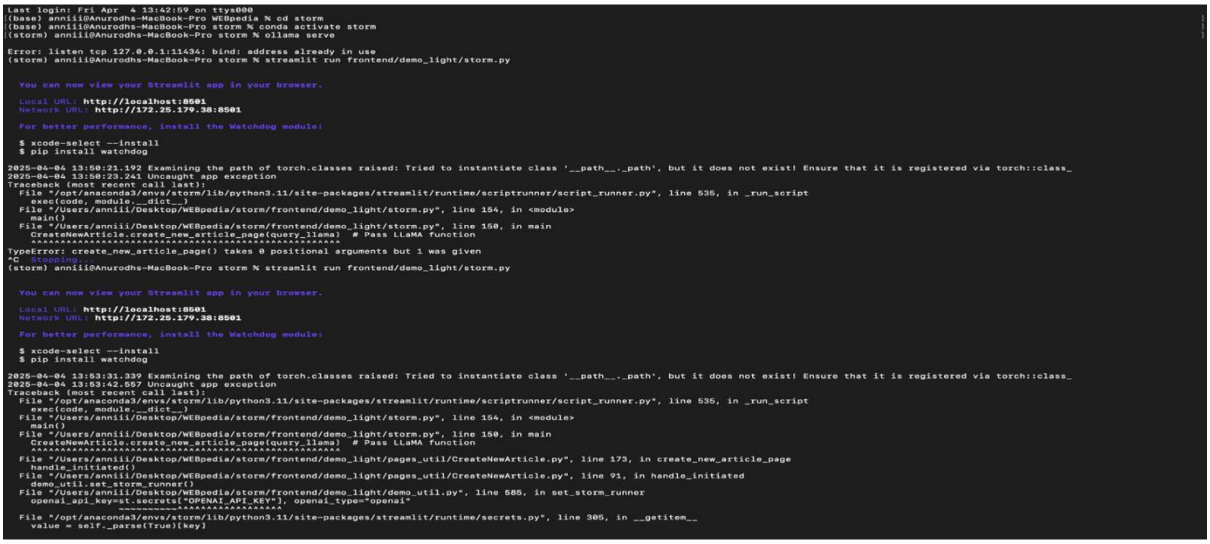
The prototype is a web-based AI chatbot interface that enables users to interact with a large language model (LLaMA) through a simple and intuitive frontend. The system architecture is divided into two main components: a Streamlit-based frontend and a backend server that hosts and serves the LLaMA model. **System Components** Frontend (Streamlit App): A responsive and interactive UI built with Streamlit, allowing users to input prompts, view real-time model responses, and optionally upload files for processing. The frontend handles user session states and presents responses in a conversational format. Backend (LLaMA Model Server), A Python-based REST API (using FastAPI or Flask) hosts the LLaMA model, which processes incoming prompts and returns generated responses. The backend ensures efficient model inference using quantization and GPU acceleration if available. Communication Layer, The frontend communicates with the backend via HTTP requests. Prompts are sent to the backends' /generate endpoint, and responses are displayed in real-time. **Key Features of the Prototype and Prototype Workflow** Real-time question answering and content generation using LLaMA. Streamlit-powered user interface for easy deployment and usage. Backend model served through a REST API. Input/output history stored in session state for continuity. Lightweight, scalable design suitable for cloud deployment. Prototype Workflow, User Interaction: The user opens the Streamlit app and enters a query in the text input box. Request Handling: The Streamlit app sends the query to the backend via an HTTP POST request to the /generate endpoint. Model Inference: The backend receives the prompt, forwards it to the LLaMA model, and generates a response. Response Delivery: The response is returned to the frontend and displayed in the chat window. Session Management: Conversation history is maintained using Streamlit's session state.

Sample Interface Screenshot (Mockup)

```
      +--------------------------------------------+
      | Chat with LLaMA                             |
      +--------------------------------------------+
  | User: How does photosynthesis work?          |
  | LLaMA: Photosynthesis is the process by...   |
            |                               |
    | [Enter your message here...]  [Send]      |
      +--------------------------------------------+
```

THE SCREENSHOTS AND DEPLOYMENT OF WebPedia

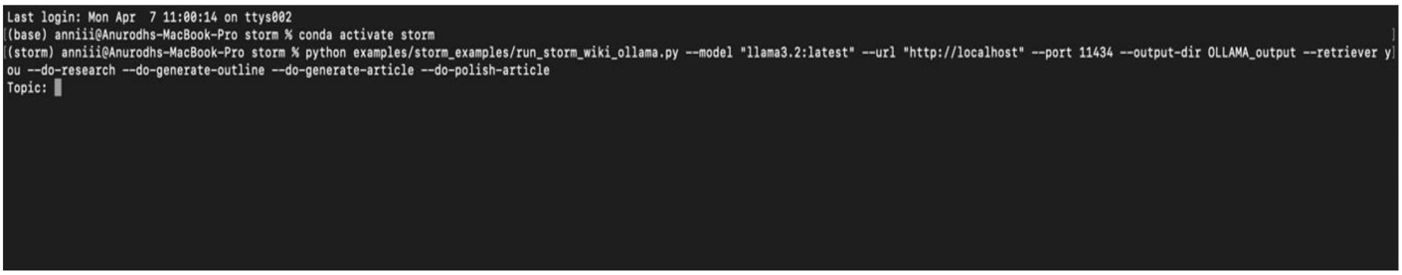
The User Interface (UI) Design for the WEBPEDIA system should aim at enabling seamless, intuitive, and transparent interaction with the automated Wikipedia article generation pipeline. Its core purpose is to guide users, from researchers to content editors, through the complex backend stages via a simple, structured interface. Design Principles. Minimalist & Clean, avoid clutter; surface only what's needed per stage. Progressive Disclosure: Reveal steps as needed, e.g., outline appears only after references. Human-in-the-loop supports manual intervention to improve factuality and structure. Explainability, Tooltip explanations, or "Why was this generated?" prompts. Accessibility, Support keyboard navigation, screen readers, and high-contrast mode.



```
Last login: Fri Apr  4 13:42:59 on ttys000
(base) anniii@Anurodhs-MacBook-Pro WEBpedia % cd storm
(base) anniii@Anurodhs-MacBook-Pro storm % conda activate storm
(storm) anniii@Anurodhs-MacBook-Pro storm % ollama serve

Error: listen tcp 127.0.0.1:11434: bind: address already in use
(storm) anniii@Anurodhs-MacBook-Pro storm % streamlit run frontend/demo_light/storm.py


  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.25.179.38:8501

  For better performance, install the Watchdog module:

  $ xcode-select --install
  $ pip install watchdog

2025-04-04 13:50:21.192 Examining the path of torch.classes raised: Tried to instantiate class '__path__._path', but it does not exist! Ensure that it is registered via torch::class_
2025-04-04 13:50:23.241 Uncaught app exception
Traceback (most recent call last):
  File "/opt/anaconda3/envs/storm/lib/python3.11/site-packages/streamlit/runtime/scriptrunner/script_runner.py", line 535, in _run_script
    exec(code, module.__dict__)
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/storm.py", line 154, in <module>
    main()
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/storm.py", line 150, in main
    CreateNewArticle.create_new_article_page(query_llama)  # Pass LLaMA function
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TypeError: create_new_article_page() takes 0 positional arguments but 1 was given
^C  Stopping...
(storm) anniii@Anurodhs-MacBook-Pro storm % streamlit run frontend/demo_light/storm.py


  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.25.179.38:8501

  For better performance, install the Watchdog module:

  $ xcode-select --install
  $ pip install watchdog

2025-04-04 13:53:31.339 Examining the path of torch.classes raised: Tried to instantiate class '__path__._path', but it does not exist! Ensure that it is registered via torch::class_
2025-04-04 13:53:42.557 Uncaught app exception
Traceback (most recent call last):
  File "/opt/anaconda3/envs/storm/lib/python3.11/site-packages/streamlit/runtime/scriptrunner/script_runner.py", line 535, in _run_script
    exec(code, module.__dict__)
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/storm.py", line 154, in <module>
    main()
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/storm.py", line 150, in main
    CreateNewArticle.create_new_article_page(query_llama)  # Pass LLaMA function
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/pages_util/CreateNewArticle.py", line 173, in create_new_article_page
    handle_initiated()
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/pages_util/CreateNewArticle.py", line 91, in handle_initiated
    demo_util.set_storm_runner()
  File "/Users/anniii/Desktop/WEBpedia/storm/frontend/demo_light/demo_util.py", line 585, in set_storm_runner
    openai_api_key=st.secrets["OPENAI_API_KEY"], openai_type="openai"
                   ~~~~~~~~~~^^^^^^^^^^^^^^^^^^
  File "/opt/anaconda3/envs/storm/lib/python3.11/site-packages/streamlit/runtime/secrets.py", line 305, in __getitem__
    value = self._parse(True)[key]
```

Figure 3: The Implementation of WebPedia to Redefine Wikipedia

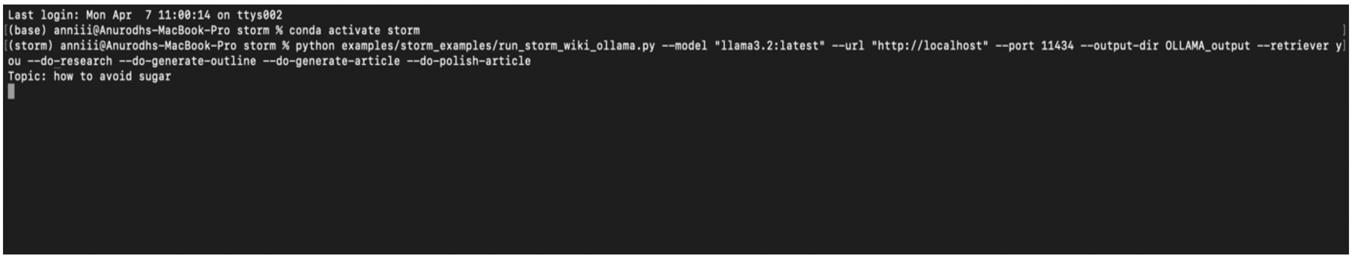
Here we activate our environment and the llama server using the code in terminal. //Conda activate// for activating the environment with all the packages in the folder containing all the codes and using //ollama server// for starting llama3 model to start running for working in the backend.



```
Last login: Mon Apr  7 11:00:14 on ttys002
(base) anniii@Anurodhs-MacBook-Pro storm % conda activate storm
(storm) anniii@Anurodhs-MacBook-Pro storm % python examples/storm_examples/run_storm_wiki_ollama.py --model "llama3.2:latest" --url "http://localhost" --port 11434 --output-dir OLLAMA_output --retriever y
ou --do-research --do-generate-outline --do-generate-article --do-polish-article
Topic: 
```

Figure 4: The activate our environment and the llama server using the code in a terminal of WebPedia

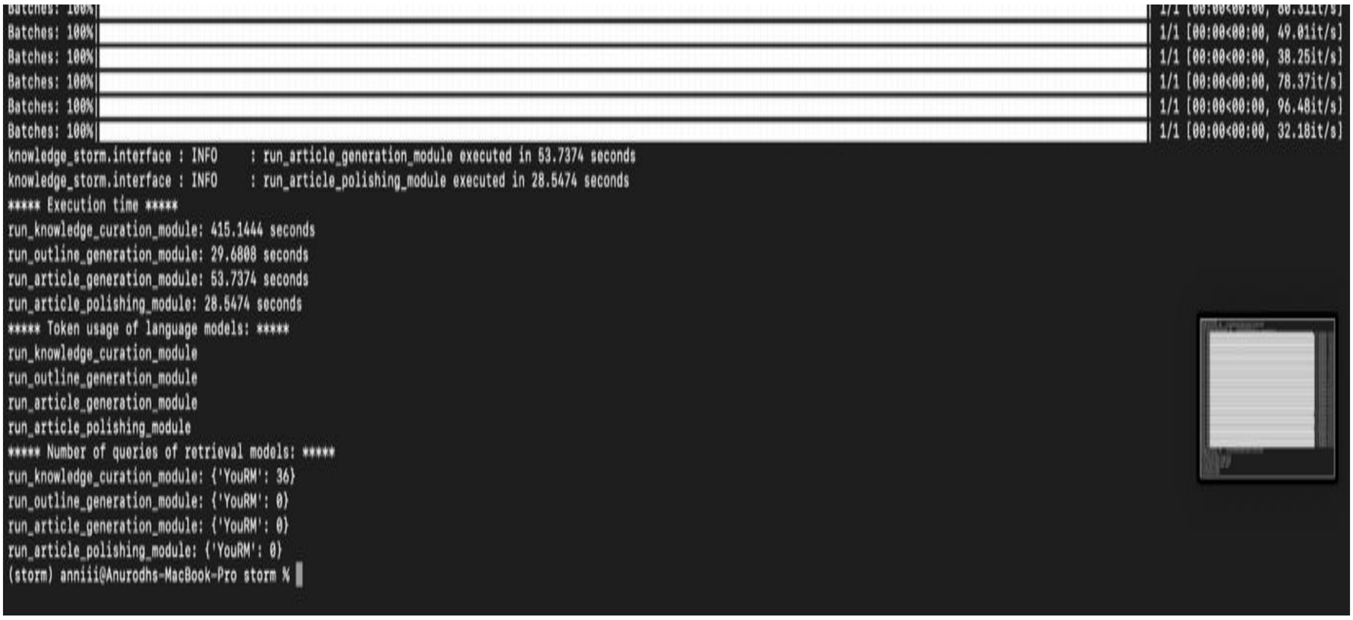
Now that the environment is activated we give command to make our model to start working and ask for an topic these is the code we type in terminal. //python examples/WEBpeida_examples/run_WEBpedia_wiki_ollama.py --model "llama3.2:latest" --url "http://localhost" --port 11434 --output-dir OLLAMA_output --retriever you --do-research --do-generate-outline --do-generate-article --do-polish-article //. We use these code in such a way that after giving the topic to the model it gets saved in a specific folder for the output where the model first do research then give outlines then generate an article after that it finally give us an polished article.



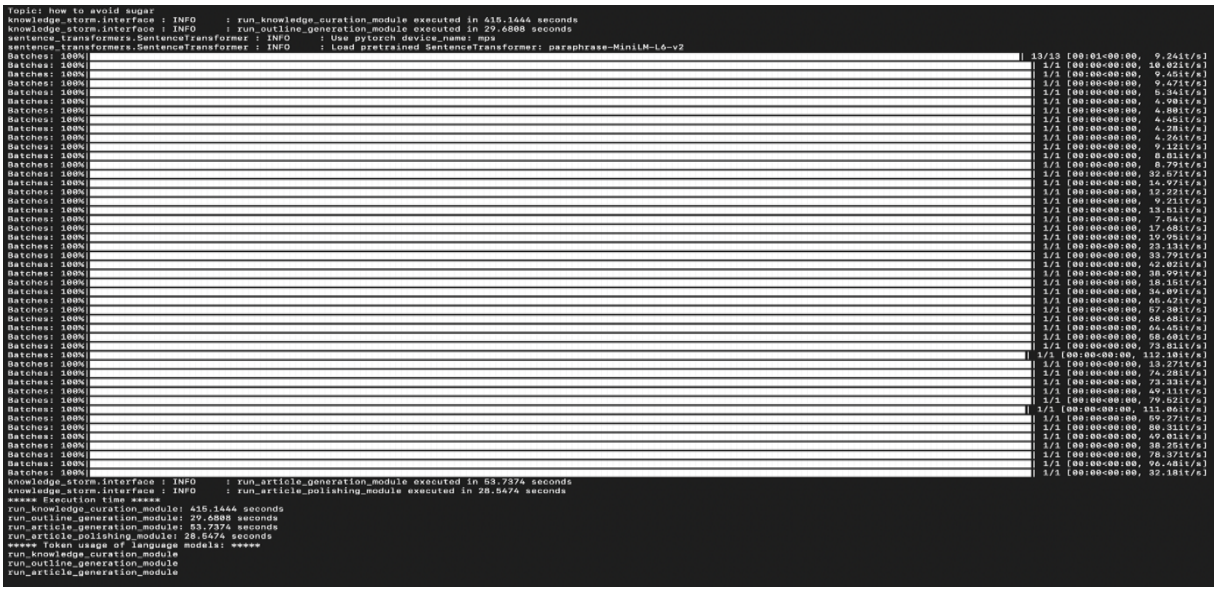
```
Last login: Mon Apr  7 11:00:14 on ttys002
(base) anniii@Anurodhs-MacBook-Pro storm % conda activate storm
(storm) anniii@Anurodhs-MacBook-Pro storm % python examples/storm_examples/run_storm_wiki_ollama.py --model "llama3.2:latest" --url "http://localhost" --port 11434 --output-dir OLLAMA_output --retriever y
ou --do-research --do-generate-outline --do-generate-article --do-polish-article
Topic: how to avoid sugar
```

Figure 5: The Implementation of outline --do-generate-article --do-polish-article WebPedia

Here we gave the topic to the model //how to avoid sugar//. Now our model will take around 3-4 minutes to generate and give an polished article. Now after 3-4 minutes that the outlines are created we would be able to see an polished articled that got saved in the output folder.



```
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 80.31it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 49.01it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 38.25it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 78.37it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 96.48it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 32.18it/s]
knowledge_storm.interface : INFO     : run_article_generation_module executed in 53.7374 seconds
knowledge_storm.interface : INFO     : run_article_polishing_module executed in 28.5474 seconds
***** Execution time *****
run_knowledge_curation_module: 415.1444 seconds
run_outline_generation_module: 29.6808 seconds
run_article_generation_module: 53.7374 seconds
run_article_polishing_module: 28.5474 seconds
***** Token usage of language models: *****
run_knowledge_curation_module
run_outline_generation_module
run_article_generation_module
run_article_polishing_module
***** Number of queries of retrieval models: *****
run_knowledge_curation_module: {'YouRM': 36}
run_outline_generation_module: {'YouRM': 0}
run_article_generation_module: {'YouRM': 0}
run_article_polishing_module: {'YouRM': 0}
(storm) anniii@Anurodhs-MacBook-Pro storm % 
```



```
Topic: how to avoid sugar
knowledge_storm.interface : INFO     : run_knowledge_curation_module executed in 415.1444 seconds
knowledge_storm.interface : INFO     : run_outline_generation_module executed in 29.6808 seconds
sentence_transformers.SentenceTransformer : INFO     : Use pytorch device_name: mps
sentence_transformers.SentenceTransformer : INFO     : Load pretrained SentenceTransformer: paraphrase-MiniLM-L6-v2
Batches: 100%|████████████████████████████████████████| 13/13 [00:01<00:00,  9.24it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 10.02it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  9.45it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  9.47it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  5.34it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  4.90it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  4.80it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  4.45it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  4.28it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  4.26it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  9.12it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  8.81it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  8.79it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 32.57it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 14.97it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 12.22it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  9.21it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 13.51it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00,  7.54it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 17.68it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 19.95it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 23.13it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 33.79it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 42.02it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 38.99it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 18.15it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 34.09it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 65.42it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 57.30it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 68.68it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 64.45it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 58.60it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 73.81it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 112.10it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 13.27it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 74.28it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 73.33it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 49.11it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 79.52it/s]
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00, 111.06it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 59.27it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 80.31it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 49.01it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 38.25it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 78.37it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 96.48it/s]
Batches: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 32.18it/s]
knowledge_storm.interface : INFO     : run_article_generation_module executed in 53.7374 seconds
knowledge_storm.interface : INFO     : run_article_polishing_module executed in 28.5474 seconds
***** Execution time *****
run_knowledge_curation_module: 415.1444 seconds
run_outline_generation_module: 29.6808 seconds
run_article_generation_module: 53.7374 seconds
run_article_polishing_module: 28.5474 seconds
***** Token usage of language models: *****
run_knowledge_curation_module
run_outline_generation_module
run_article_generation_module
```

Figure 6: The polished article that got saved in the output folder of WebPedia

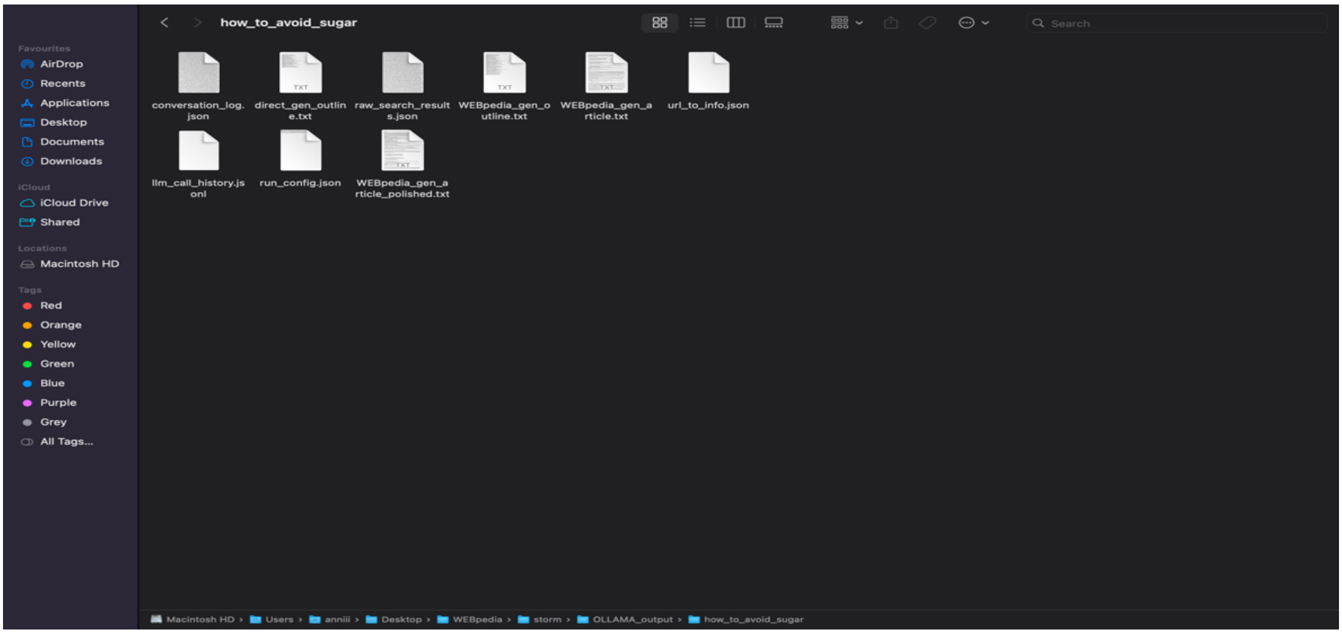
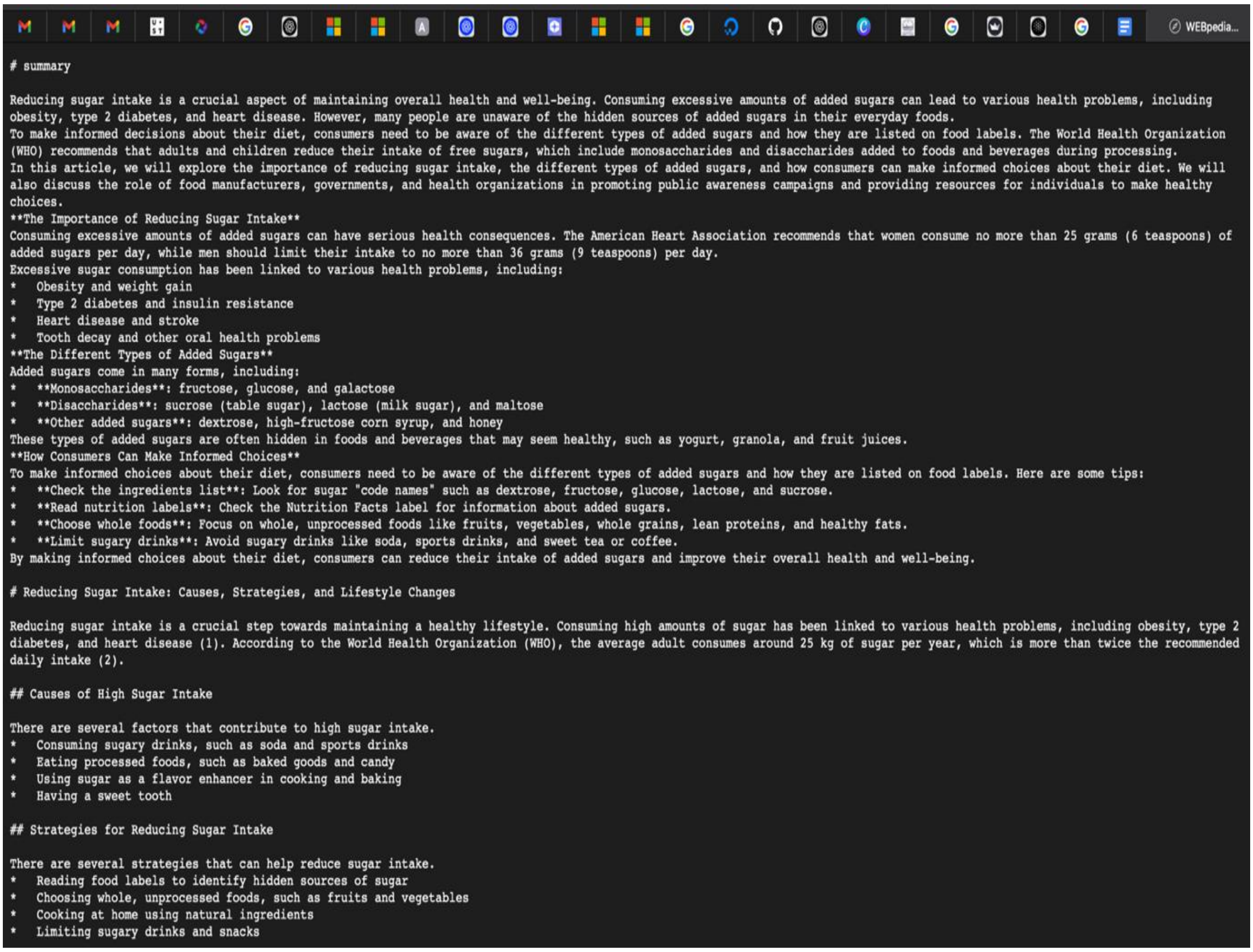


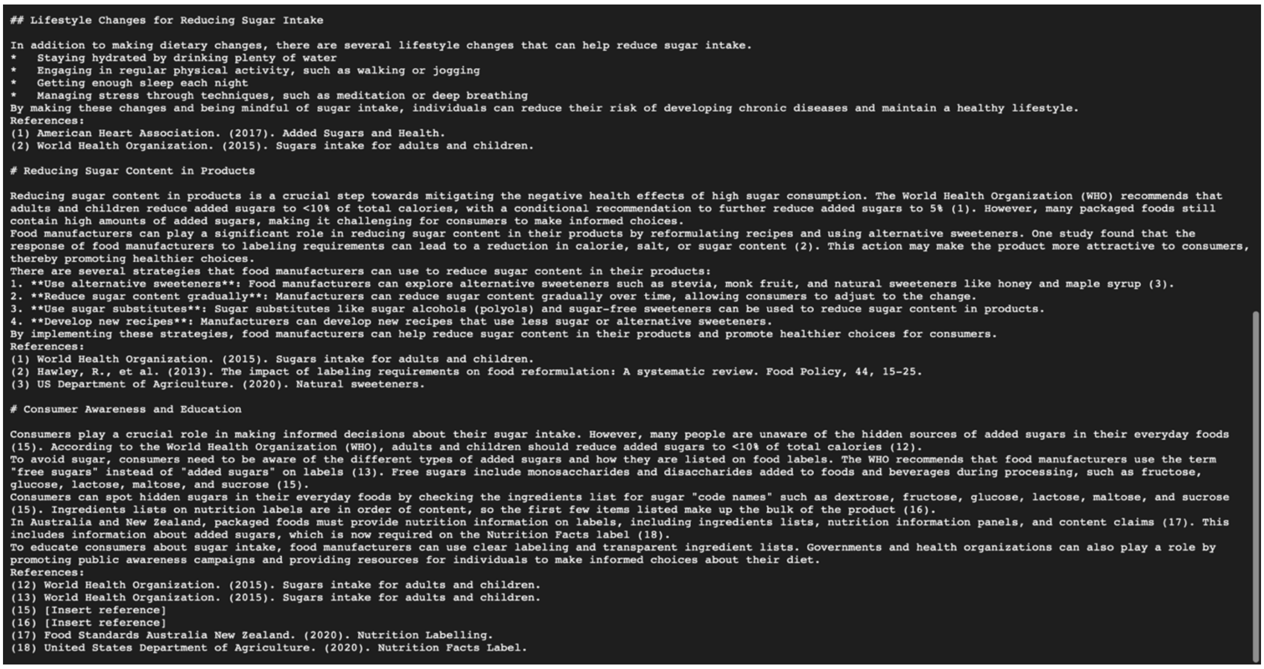
Figure 7: The Implementation of the generated article in a folder inside the code folder WebPedia

We would be able to find the generated article in a folder inside the code folder looks like these where we can other than the final generated polished article we can also see the outlined that got created. Now these is how the generated and the curated final article looks like. Or we can do the same on the front end and get the result article shown in figures.



```
# summary

Reducing sugar intake is a crucial aspect of maintaining overall health and well-being. Consuming excessive amounts of added sugars can lead to various health problems, including
obesity, type 2 diabetes, and heart disease. However, many people are unaware of the hidden sources of added sugars in their everyday foods.
To make informed decisions about their diet, consumers need to be aware of the different types of added sugars and how they are listed on food labels. The World Health Organization
(WHO) recommends that adults and children reduce their intake of free sugars, which include monosaccharides and disaccharides added to foods and beverages during processing.
In this article, we will explore the importance of reducing sugar intake, the different types of added sugars, and how consumers can make informed choices about their diet. We will
also discuss the role of food manufacturers, governments, and health organizations in promoting public awareness campaigns and providing resources for individuals to make healthy
choices.
**The Importance of Reducing Sugar Intake**
Consuming excessive amounts of added sugars can have serious health consequences. The American Heart Association recommends that women consume no more than 25 grams (6 teaspoons) of
added sugars per day, while men should limit their intake to no more than 36 grams (9 teaspoons) per day.
Excessive sugar consumption has been linked to various health problems, including:
*   Obesity and weight gain
*   Type 2 diabetes and insulin resistance
*   Heart disease and stroke
*   Tooth decay and other oral health problems
**The Different Types of Added Sugars**
Added sugars come in many forms, including:
*   **Monosaccharides**: fructose, glucose, and galactose
*   **Disaccharides**: sucrose (table sugar), lactose (milk sugar), and maltose
*   **Other added sugars**: dextrose, high-fructose corn syrup, and honey
These types of added sugars are often hidden in foods and beverages that may seem healthy, such as yogurt, granola, and fruit juices.
**How Consumers Can Make Informed Choices**
To make informed choices about their diet, consumers need to be aware of the different types of added sugars and how they are listed on food labels. Here are some tips:
*   **Check the ingredients list**: Look for sugar "code names" such as dextrose, fructose, glucose, lactose, and sucrose.
*   **Read nutrition labels**: Check the Nutrition Facts label for information about added sugars.
*   **Choose whole foods**: Focus on whole, unprocessed foods like fruits, vegetables, whole grains, lean proteins, and healthy fats.
*   **Limit sugary drinks**: Avoid sugary drinks like soda, sports drinks, and sweet tea or coffee.
By making informed choices about their diet, consumers can reduce their intake of added sugars and improve their overall health and well-being.

# Reducing Sugar Intake: Causes, Strategies, and Lifestyle Changes

Reducing sugar intake is a crucial step towards maintaining a healthy lifestyle. Consuming high amounts of sugar has been linked to various health problems, including obesity, type 2
diabetes, and heart disease (1). According to the World Health Organization (WHO), the average adult consumes around 25 kg of sugar per year, which is more than twice the recommended
daily intake (2).

## Causes of High Sugar Intake

There are several factors that contribute to high sugar intake.
*   Consuming sugary drinks, such as soda and sports drinks
*   Eating processed foods, such as baked goods and candy
*   Using sugar as a flavor enhancer in cooking and baking
*   Having a sweet tooth

## Strategies for Reducing Sugar Intake

There are several strategies that can help reduce sugar intake.
*   Reading food labels to identify hidden sources of sugar
*   Choosing whole, unprocessed foods, such as fruits and vegetables
*   Cooking at home using natural ingredients
*   Limiting sugary drinks and snacks
```

Figure 8: The final generated polished article of WebPedia to Redefine Wikipedia



```
## Lifestyle Changes for Reducing Sugar Intake

In addition to making dietary changes, there are several lifestyle changes that can help reduce sugar intake.
*   Staying hydrated by drinking plenty of water
*   Engaging in regular physical activity, such as walking or jogging
*   Getting enough sleep each night
*   Managing stress through techniques, such as meditation or deep breathing
By making these changes and being mindful of sugar intake, individuals can reduce their risk of developing chronic diseases and maintain a healthy lifestyle.
References:
(1) American Heart Association. (2017). Added Sugars and Health.
(2) World Health Organization. (2015). Sugars intake for adults and children.

# Reducing Sugar Content in Products

Reducing sugar content in products is a crucial step towards mitigating the negative health effects of high sugar consumption. The World Health Organization (WHO) recommends that
adults and children reduce added sugars to <10% of total calories, with a conditional recommendation to further reduce added sugars to 5% (1). However, many packaged foods still
contain high amounts of added sugars, making it challenging for consumers to make informed choices.
Food manufacturers can play a significant role in reducing sugar content in their products by reformulating recipes and using alternative sweeteners. One study found that the
response of food manufacturers to labeling requirements can lead to a reduction in calorie, salt, or sugar content (2). This action may make the product more attractive to consumers,
thereby promoting healthier choices.
There are several strategies that food manufacturers can use to reduce sugar content in their products:
1. **Use alternative sweeteners**: Food manufacturers can explore alternative sweeteners such as stevia, monk fruit, and natural sweeteners like honey and maple syrup (3).
2. **Reduce sugar content gradually**: Manufacturers can reduce sugar content gradually over time, allowing consumers to adjust to the change.
3. **Use sugar substitutes**: Sugar substitutes like sugar alcohols (polyols) and sugar-free sweeteners can be used to reduce sugar content in products.
4. **Develop new recipes**: Manufacturers can develop new recipes that use less sugar or alternative sweeteners.
By implementing these strategies, food manufacturers can help reduce sugar content in their products and promote healthier choices for consumers.
References:
(1) World Health Organization. (2015). Sugars intake for adults and children.
(2) Hawley, R., et al. (2013). The impact of labeling requirements on food reformulation: A systematic review. Food Policy, 44, 15-25.
(3) US Department of Agriculture. (2020). Natural sweeteners.

# Consumer Awareness and Education

Consumers play a crucial role in making informed decisions about their sugar intake. However, many people are unaware of the hidden sources of added sugars in their everyday foods
(15). According to the World Health Organization (WHO), adults and children should reduce added sugars to <10% of total calories (12).
To avoid sugar, consumers need to be aware of the different types of added sugars and how they are listed on food labels. The WHO recommends that food manufacturers use the term
"free sugars" instead of "added sugars" on labels (13). Free sugars include monosaccharides and disaccharides added to foods and beverages during processing, such as fructose,
glucose, lactose, maltose, and sucrose (15).
Consumers can spot hidden sugars in their everyday foods by checking the ingredients list for sugar "code names" such as dextrose, fructose, glucose, lactose, maltose, and sucrose
(15). Ingredients lists on nutrition labels are in order of content, so the first few items listed make up the bulk of the product (16).
In Australia and New Zealand, packaged foods must provide nutrition information on labels, including ingredients lists, nutrition information panels, and content claims (17). This
includes information about added sugars, which is now required on the Nutrition Facts label (18).
To educate consumers about sugar intake, food manufacturers can use clear labeling and transparent ingredient lists. Governments and health organizations can also play a role by
promoting public awareness campaigns and providing resources for individuals to make informed choices about their diet.
References:
(12) World Health Organization. (2015). Sugars intake for adults and children.
(13) World Health Organization. (2015). Sugars intake for adults and children.
(15) [Insert reference]
(16) [Insert reference]
(17) Food Standards Australia New Zealand. (2020). Nutrition Labelling.
(18) United States Department of Agriculture. (2020). Nutrition Facts Label.
```

Figure 9: The generated and the curated final article looks like WebPedia

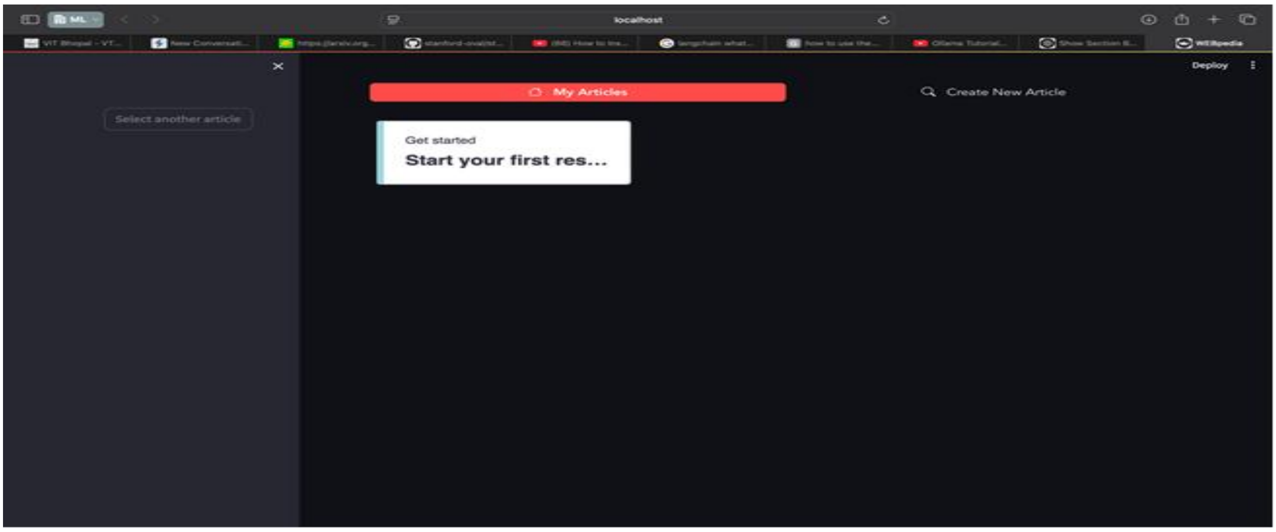
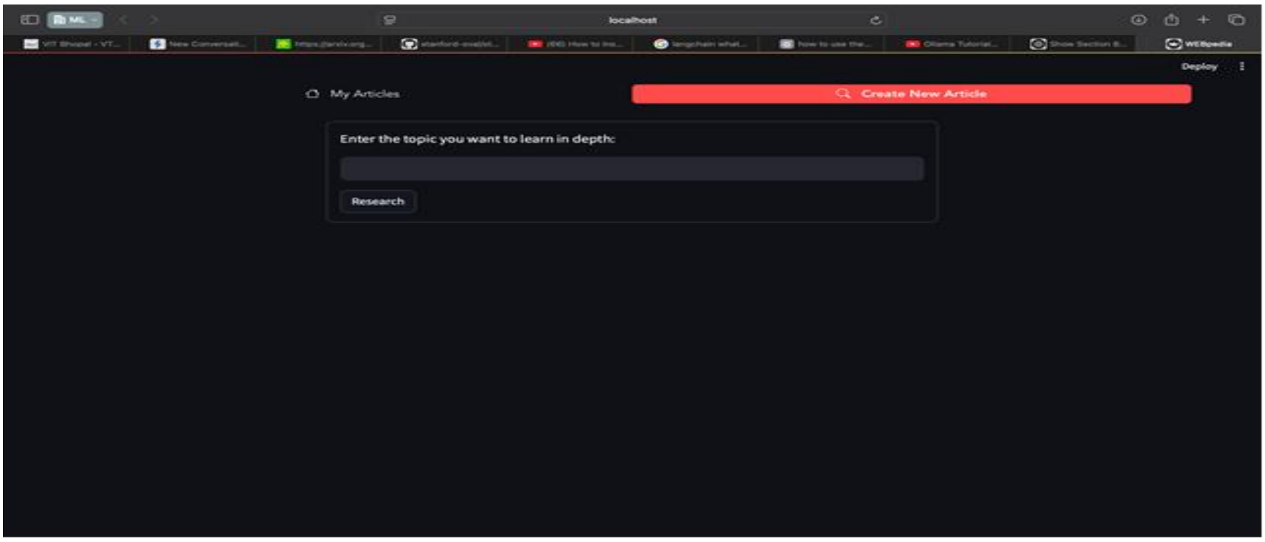


Figure 10: The Implementation of result of WebPedia to Redefine Wikipedia



Enter the topic you want to learn in depth:

Figure 11: The Implementation of Topic Content WebPedia to Redefine Wikipedia

THE CONTRIBUTION AND FINDINGS OF WebPedia 2.0

Through rigorous evaluation, the study finds that WEBPEDIA significantly improves the organization and breadth of **generated articles**. Outlines created using the system enhanced the final article’s structure and scope, with a 25% increase in organization and a **10% improvement in coverage** compared to baseline methods. Incorporating diverse perspectives proved instrumental in creating nuanced outlines that comprehensively address a topic. This approach highlights the importance of **accounting for different viewpoints** in the research phase, a step often overlooked in traditional content generation models. Despite its advancements, the study identifies critical challenges in automated **long-form article generation**. One key issue is source bias transfer, where biases in the retrieved information can influence the neutrality of the content. Another is the over-association of unrelated facts, where language models fabricate connections that do not exist. These challenges underscore the need for further refinement in **LLM-based** writing systems to ensure accuracy and reliability.

CONCLUSION

We are going to introduce WEBpedia, an **LLM-based writing system** designed to automate the pre-writing stage of creating Wikipedia-like articles from scratch and to facilitate our study on generating grounded, long-form articles, we will curate a UtliedWiki dataset and establish evaluation criteria. By increasing the breadth and depth of the article, WEBpedia helps identify new challenges for grounded writing systems, as highlighted by many research and evaluations. The WEBPEDIA framework represents a significant advancement in automating the process of writing Wikipedia-like articles from scratch. By focusing on the often-overlooked pre-writing stage, WEBPEDIA emphasizes thorough research and structured content generation, addressing key challenges such as diverse perspective gathering, iterative question asking, and grounded information retrieval. Through its multi-stage approach, WEBPEDIA demonstrates the capability to generate detailed and well-organized outlines, which form the foundation for comprehensive and coherent articles. Evaluation using the **UtliedWIKI dataset** and feedback from experienced Wikipedia editors highlight the system's strengths in creating articles with improved breadth, depth, and organization compared to baseline methods. However, challenges such as mitigating source bias and avoiding the fabrication of connections between unrelated facts underscore the need for further refinement in grounded article generation. Overall, WEBPEDIA showcases the potential of large language models in supporting and enhancing **expository writing** tasks, particularly for topics requiring meticulous research and planning. This work lays the groundwork for future innovations in long-form content creation, bridging gaps in automation and human-level expertise.

REFERENCES

1. Attributed Question Answering: Evaluation and Modelling for Attributed Large Language Models https://arxiv.org/pdf/2212.08037.
2. Suresh Kallam , M K Jayanthi Kannan , B. R. M. , . (2024). A Novel Authentication Mechanism with Efficient Math Based Approach. International Journal of Intelligent Systems and Applications in Engineering, 12(3), 2500–2510. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/5722
3. Automatically Generating Wikipedia Articles: A Structure-Aware Approach https://aclanthology.org/P09-1024.pdf
4. Balajee RM, Jayanthi Kannan MK, Murali Mohan V., "Image-Based Authentication Security Improvement by Randomized Selection Approach," in *Inventive Computation and Information Technologies*, Springer, Singapore, 2022, pp. 61-71
5. Siddhartha Banerjee and Prasenjit Mitra. 2015. WikiKreator: Improving Wikipedia Stubs Automatically. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)
6. B. R M, S. Kallam and M. K. Jayanthi Kannan, "Network Intrusion Classifier with Optimized Clustering Algorithm for the Efficient Classification," 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2024, pp. 439-446, doi: 10.1109/ICICV62344.2024.00075.
7. Tira Nur Fitria. 2023. Artificial intelligence (ai) technology in openai chatgpt application: A review of chatgpt in writing english essay. In ELT Forum: Journal of English Language Teaching, volume 12
8. M. K. Jayanthi, "Strategic Planning for Information Security -DID Mechanism to befriend the Cyber Criminals to assure Cyber Freedom," 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 142-147, doi: 10.1109/Anti-Cybercrime.2017.7905280.

9. Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing.
10. Kavitha, E., Tamilarasan, R., Baladhandapani, A., Kannan, M.K.J. (2022). A novel soft clustering approach for gene expression data. Computer Systems Science and Engineering, 43(3), 871-886. https://doi.org/10.32604/csse.2022.021215
11. G., D. K., Singh, M. K., & Jayanthi, M. (Eds.). (2016). Network Security Attacks and Countermeasures. IGI Global. https://doi.org/10.4018/978-1-4666-8761-5
12. R M, B.; M K, J.K. Intrusion Detection on AWS Cloud through Hybrid Deep Learning Algorithm. Electronics 2023, 12, 1423. https://doi.org/10.3390/electronics12061423
13. Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. Teaching language models to support answers with verified quotes.
14. Christina Sauper and Regina Barzilay. 2009. Automatically generating Wikipedia articles: A structure-aware approach. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 208–216, Suntec, Singapore. Association for Computational Linguistics.
15. Naik, Harish and Kannan, M K Jayanthi, A Survey on Protecting Confidential Data over Distributed Storage in Cloud (December 1, 2020). Available at SSRN: https://ssrn.com/abstract=3740465 or http://dx.doi.org/10.2139/ssrn.3740465
16. Kavitha, E., Tamilarasan, R., Poonguzhali, N., Kannan, M.K.J. (2022). Clustering gene expression data through modified agglomerative M-CURE hierarchical algorithm. Computer Systems Science and Engineering, 41(3), 1027-141. https://doi.org/10.32604/csse.2022.020634
17. Kumar, K.L.S., Kannan, M.K.J. (2024). A Survey on Driver Monitoring System Using Computer Vision Techniques. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds) Innovative Computing and Communications. ICICC 2024. Lecture Notes in Networks and Systems, vol 1021. Springer, Singapore. https://doi.org/10.1007/978-981-97-3591-4_21
18. Dr. M.K. Jayanthi Kannan, Anurodh Pancholi, Priyaanshu Singh, “WEBpedia Redefining Wikipedia @ Crafting Wikipedia Articles with AI-based Large Language Models”, International Journal of Innovative Research in Computer and Communication Engineering, e-ISSN: 2320-9801 DOI: 10.15680/IJIRCCE.2024.1212073, 12(12), December 2024, PP. 13517-13527. https://www.ijircce.com/get-current-issue.php?key=MTY0
19. M. K. J. Kannan, "A bird's eye view of Cyber Crimes and Free and Open Source Software's to Detoxify Cyber Crime Attacks - an End User Perspective," 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 232-237, doi: 10.1109/Anti-Cybercrime.2017.7905297.
20. P. Jain, I. Rajvaidya, K. K. Sah and J. Kannan, "Machine Learning Techniques for Malware Detection- a Research Review," 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), BHOPAL, India, 2022, pp. 1-6, doi: 10.1109/SCEECS54111.2022.9740918.
21. Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In Findings of the Association

for Computational Linguistics: EMNLP 2023, pages 5687–5711, Singapore. Association for Computational Linguistics.

22. Dr. M.K. Jayanthi Kannan, Satyajit Patel (2024). Sustainable Information Retrieval Techniques for Onion Market Instability Prediction using Machine Learning and Deep Learning Approaches. International Journal of Advance Research, Ideas and Innovations in Technology, 10(6) www.IJARIIT.com. https://www.ijariit.com/manuscripts/v10i6/V10I6-1455.pdf
23. B. R. M, M. M. V and J. K. M. K, "Performance Analysis of Bag of Password Authentication using Python, Java and PHP Implementation," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1032-1039, doi: 10.1109/ICCES51350.2021.9489233.
24. Harrison, Stephen (2023-01-12). "Should ChatGPT Be Used to Write Wikipedia Articles?". Slate Magazine. Retrieved 2023-01-13.
25. Dr.M.K. Jayanthi and Sree Dharinya, V., (2013), Effective Retrieval of Text and Media Learning Objects using Automatic Annotation, World Applied Sciences Journal, Vol. 27 No.1, 2013, © IDOSI Publications,2013, DOI: 10.5829/idosi.wasj.2013.27.01.1614, pp.123-129. https://www.idosi.org/wasj/wasj27(1)13/20.pdf
26. Nielsen, Finn Årup (2012). "Wikipedia Research and Tools: Review and Comments". SSRN Working Paper Series. doi:10.2139/ssrn.2129874. ISSN 1556-5068.
27. Dr. Naila Aaijaz, Dr. K. Grace Mani, Dr. M. K. Jayanthi Kannan and Dr. Veena Tewari (Feb 2025), The Future of Innovation and Technology in Education: Trends and Opportunities, ASIN : B0DW334PR9, S&M Publications, Mangalore, Haridwar, India-247667, ISBN-13 : 978-8198488824,https://www.amazon.in/gp/product/B0DW334PR9/ref=ox_sc_act_title_1?smid=A2DVPTOROMUBNE&psc=1#detailBullets_feature_div
28. Python for Data Analytics: Practical Techniques and Applications, Dr. Surendra Kumar Shukla, Dr. Upendra Dwivedi, Dr. M K Jayanthi Kannan, Chalamalasetty Sarvani ISBN: 978-93-6226-727-6, ASIN : B0DMJY4X9N, JSR Publications, 23 October 2024, https://www.amazon.in/gp/product/B0DMJY4X9N/ref=ox_sc_act_title_1?smid=A29XE7SVTY6MCQ&psc=1
29. Verger, Rob (7 August 2018). "Artificial intelligence can now help write Wikipedia pages for overlooked scientists". Popular Science
30. Harish Naik and M K Jayanthi Kannan, A Research on Various Security Aware Mechanisms in Multi-Cloud Environment for Improving Data Security, ISBN:979-8-3503-4745-6, DOI: 10.1109/ICDCECE57866.2023.10151135, 2nd IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics ICDCECE 2023, https://ieeexplore.ieee.org/document/10151135
31. Piscopo, Alessandro (1 October 2018). "Wikidata: A New Paradigm of Human-Bot Collaboration?". arXiv:1810.00931 [cs.HC].
32. Harish Naik Bheemanaik Manjyanaik, Rajanikanta, Jayanthi Mangayarkarasi Kannan, Preserving Confidential Data Using Improved Rivest-Shamir Adleman to Secure Multi-Cloud, International Journal of Intelligent Engineering and Systems, Vol.17, No.4, 2024 pp .162-171, DOI: 10.22266/ijies2024.0831.13, https://inass.org/wp-content/uploads/2024/02/2024083113-2.pdf
33. Teaching language models to support answers with verified quotes, https://arxiv.org/pdf/2203.11147
34. Gertner, Jon (18 July 2023). "Wikipedia's Moment of Truth - Can the online encyclopedia help teach

A.I. chatbots to get their facts right — without destroying itself in the process? + comment". The New York Times. Archived from the original on 18 July 2023. Retrieved 19 July 2023.