Classifying conspiracy theories based on threads on "Allmystery.de"

Sebastian Müller

1 Introduction

Classifying conspiracy theories presents a unique challenge. Definitions are often elusive and contested, and attempts to formalize them tend to fall short of capturing what people intuitively recognize. As U.S. Supreme Court Justice Potter Stewart famously said when grappling with how to define pornography, "I know it when I see it." [1] Similarly, most people can identify conspiracy theories in the wild, even if they struggle to define them precisely.

On the internet, conspiracy theories flourish in discussion forums, comment sections, and social media. Interestingly, the term "conspiracy theory" is not always used by the participants themselves, likely due to its pejorative connotations. However, in spaces where it is explicitly used, it can serve as a practical marker for categorizing content without needing to define it ourselves. This project adopts a bottom-up approach: rather than imposing my own definitions, I leverage the categorizations provided by forum users to explore and analyze conspiracy content.

To that end, I focused on the German-language forum *Allmystery.de*, one of the oldest still-active internet forums in Germany. According to Wikipedia (one of the few available sources on the site's history) Allmystery dates back to 1987 and was originally founded on the ARPANET. Its conspiracy theory subforum remains one of the most active sections, reflecting the forum's original purpose. As of May 10, 2025, the conspiracy subforum contained 3,333 threads and 416,339 posts, with discussions dating back to 2002.

The primary goal of this project was to develop a web-based AI agent capable of classifying new user-submitted conspiracy theories by matching them with similar discussions from Allmystery's archive. To achieve this, I scraped forum data, embedded the content into a vector space for semantic search, and built an AI-powered interface using Streamlit. Additionally, a statistical analysis page was developed to provide further insight into the data.

Beyond this immediate functionality, the project also serves as a first step toward training a general-purpose classifier capable of identifying conspiracy theories in German-language texts. By utilizing a large, thematically labeled

[1] Gewirtz, Paul (1996): On "I Know It When I See It". In: Yale Law Journal, vol. 105, no. 4, 1996, pp. 1023–1047.

dataset generated by forum users, I aim to lay the groundwork for a more practical and contextual classification of conspiracy content.

This project was developed during a one-week workshop organized by the Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) in Leipzig. Participants were encouraged to pursue AI-focused project ideas and present their results at the end of the workshop. I would like to thank the organizers for their support and for facilitating this valuable experience.

2 Implementation

2.1 Data Acquisition from Allmystery.de

Fortunately, the structure of the Allmystery.de website follows a conventional and consistent HTML layout, which made it suitable for data scraping. To collect the relevant content from the forum, I used Python's `requests` library to download pages and `BeautifulSoup` to parse the HTML. While I had limited prior experience with these libraries, I relied on a large language model (LLM) to guide me through their documentation and usage, and identified the necessary HTML tags manually.

While I had limited prior experience with these libraries, I relied on a large language model (LLM) to guide me through their documentation and usage, as well as to understand the provided functions and refactor code. I also identified the necessary HTML tags manually. For other libraries that will be discussed later, I continued to use the LLM for assistance.

To make the scraping process more manageable on my hardware, I divided the task into two stages. First, I created an index file containing metadata for all threads in the conspiracy subforum, specifically: the thread ID, thread title, and thread URL. This file served both as a lookup table and as a control mechanism to track progress and resume scraping if interrupted.

In the second stage, each thread listed in the index was processed individually. For every thread, a separate `.jsonl` file was created, named after its thread ID. Each file contains structured data for all posts within that thread. The following attributes were extracted from each post:

- Thread ID, thread title, and thread URL
- Page number (each thread page contains approximately 20 posts)
- Post ID
- Username of the author
- Date and time of the post
- Content of the post
- Sequential number of the post within the thread

Not all of this information was used in the current version of the project, but it may prove useful for future analysis, for example, in tracking user behavior over time or analyzing the temporal development of specific theories.

Initially, due to time constraints and technical limitations, I was only able to scrape about 15% of all threads before presenting the project during the workshop. However, since the threads on Allmystery are sorted by most recent activity, this initial dataset still captured a representative and active subset of the forum. After the workshop, I completed the scraping process and, as of May 10, 2025, the dataset includes all conspiracy threads available on the site up to that date.

2.2 Vectorization and Indexing

To enable semantic search over the scraped forum data, I implemented a vector database using the Langchain framework in Python, in combination with the Chroma vector store via the `langchain-chroma` library. For generating vector embeddings, I used the model `jina/jina-embeddings-v2-base-de` provided through the Ollama framework. This model was specifically chosen because it is optimized for the German language, unlike general-purpose multilingual models, which had yielded suboptimal results in previous topic modeling tasks for another project.

The embedding pipeline worked as follows: forum post content was wrapped into Langchain `Document` objects, enriched with thread metadata (title and URL), and then passed to the embedding function. Each document's resulting vector was stored in the Chroma database under a persistent directory.

Initially, I encountered a limitation with Chroma's batch size handling during the insertion phase. Since embedding the entire dataset was a resource-intensive process, requiring overnight execution, I was disappointed to find the next morning that the insertion had failed. To address this, I reran the embedding step on a subset of the data: specifically, only the first page of each thread, which yielded a total of around 8,000 posts, or approximately 4% of all scraped content at that time.

While this decision limited the scope of the dataset, it turned out to be a useful heuristic. The first page of a thread typically contains enough context to convey its main topic, as thread openers usually frame the discussion clearly. However, one downside of this approach is that secondary or derivative conspiracy theories, those mentioned deeper in a thread, may not be captured in the vector store if they do not appear on the first page.

Despite this limitation, the resulting retrieval system performed well in practice. The reduced dataset still allowed for effective semantic search, making the trade-off acceptable for this experimental stage of the project. The final retriever was configured to return the top semantically similar posts for any given query, enabling meaningful comparisons between new user input and existing forum discussions.

2.3 AI-Agent Development

The core functionality of the application relied on a local AI agent to interpret user-submitted conspiracy theories and match them with relevant threads from the Allmystery forum. For this task, I used the `gemma3:4b-it-qat` model provided by Google, which was compatible with my hardware setup via the Ollama framework. While this model was lightweight enough to run locally, its performance was limited; each query typically required several minutes to complete. No alternative models were tested within the scope of this project.

To maintain structure in the model's responses, I designed a prompt that explicitly instructed the AI to return output in JSON format. This format included the most relevant thread's title, URL, the content of the most similar post, and an explanation for the match. However, the model did not always adhere strictly to the expected format, which necessitated additional processing. A regular expression was used to extract JSON from the raw output, but this workaround proved imperfect, occasionally failing to capture malformed or incomplete JSON segments.

The classification process consisted of two main steps. First, the user input was embedded and compared against the posts in the vector database to retrieve the top semantic matches. These candidate posts were then passed, along with the user's input, to the AI model, which selected the most relevant thread and formulated a justification for the selection. Importantly, the system was designed to handle cases in which no good match could be found. The fallback behavior was a null response with an explanation such as *"I can't find a good fitting thread."* However, in practice, this case never occurred; even when the user input did not resemble a conspiracy theory, the model always returned a thread it considered a suitable match.

This behavior suggests that while the retrieval and generation pipeline was functional, the threshold for similarity may have been too low, or the prompt instructions insufficiently strict. Nevertheless, for exploratory and prototyping purposes, the AI agent fulfilled its intended role and successfully mapped new input onto existing forum content with reasonable semantic alignment.

2.4 Frontend with Streamlit

The user interface for this project was implemented using the Python library `Streamlit`, which allowed for rapid development of an interactive web application. The frontend was divided into two main pages: one for AI-powered classification and another for presenting statistical insights into the dataset.

AI-Agent Interface

The classification page served as the central user interaction point. Users could input a conspiracy theory in natural language and receive the most semantically similar thread from the Allmystery forum. Upon submission, the input was vectorized and compared to the embedded dataset using the retriever described

earlier. The top matching posts were passed to a local language model for interpretation and contextual reasoning.

The AI model responded in JSON format containing the title and URL of the most relevant thread, a brief explanation of the match, and the matched post's content. If no suitable thread was found, the interface was designed to gracefully handle the response by displaying a fallback message. The interface also allowed users to inspect the raw matched post, which helped maintain transparency in how matches were selected.

Statistics Dashboard

A secondary page in the frontend was created to provide a statistical overview of the dataset and scraping progress. This dashboard contextualized the scale and limitations of the data used for training and inference. Key elements included:

- **Thread and Post Coverage:** A pie chart compared the total number of threads in the forum with the number of scraped threads. Another pie chart compared the total number of scraped posts to the subset that was actually inserted into the vector database. Both charts are now outdated, as the scraping process has continued.
- **Most Active Threads:** The dashboard also included a horizontal bar chart displaying the ten threads with the highest post counts within the available data. Interestingly, three of the top five threads (and four of the top ten) were focused on the events of September 11, 2001. This observation aligns with a statement from the forum's founder that the conspiracy subforum experienced significant growth following the 9/11 attacks.[2] Other top threads included discussions about flat Earth theory, the moon landing, climate change, and general-purpose conspiracy debates. Post counts ranged from over 16,000 in the largest thread to approximately 5,000 in the tenth.

[2]Zantke, Kai; Jahn, Sönke (2013): Interview mit Allmystery.de-Gründer Dennis Kort. Available online at: https://www.computerbild.de/artikel/cb-Aktuell-Internet-Interview-Allmystery.de-Gruender-Dennis-Kort-8988245.html

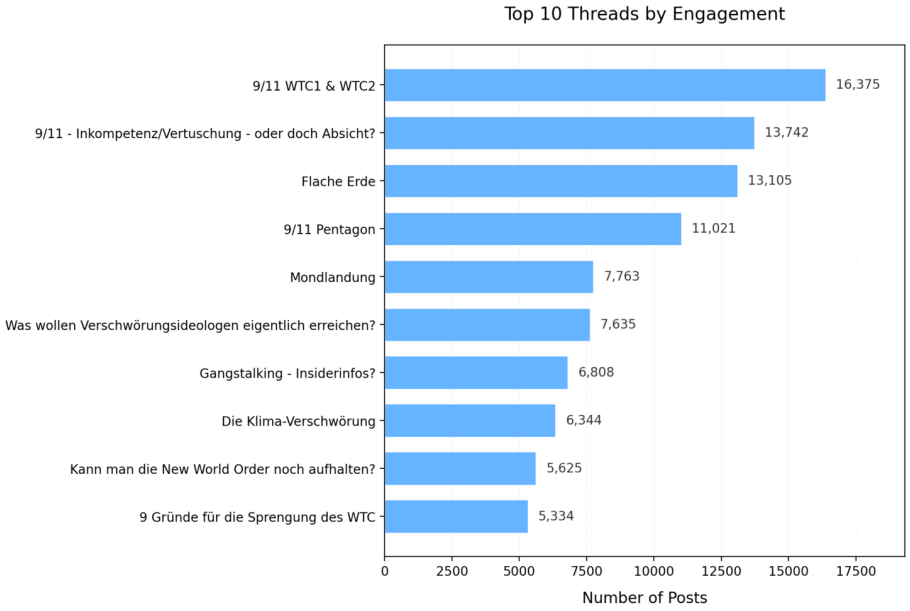

Figure 1: Top 10 threads by number of posts in the first version of the scraped dataset.

Overall, the Streamlit frontend offered a simple but effective interface for interacting with both the AI model and the dataset. It combined intuitive input/output fields with informative visualizations that helped users understand the scope and context of the project.

3 Conclusion

This project successfully demonstrates a working prototype for classifying and linking user-submitted conspiracy theory descriptions to relevant threads from the German conspiracy forum Allmystery. Despite the limited scope of the current dataset, the tool performs remarkably well. The combination of a domain-specific German embedding model (`jina/jina-embeddings-v2-base-de`), a Chroma-based vector database, and a lightweight LLM (Gemma 3B via Ollama) allowed for meaningful semantic matching between user input and forum discussions.

The Streamlit-based frontend provided an intuitive way to interact with the AI agent and visualize dataset statistics. The data exploration revealed interesting patterns, such as the dominance of 9/11-related threads among the most active discussions, which aligns with historical statements about the forum's activity growth.

Given the promising qualitative performance of the system, future work should focus on quantifying its accuracy. One possible approach would be to

build a test set from existing forum posts and measure how often the AI agent correctly predicts the originating thread based on post content. This would help assess classification quality and reveal whether removing noisy or off-topic posts could improve accuracy.

From an ethical standpoint, making this tool publicly accessible would require additional safeguards. To reduce the risk of reinforcing misinformation, especially for users who may lack critical distance to conspiracy narratives, it would be essential to provide contextual information alongside the forum threads. One approach could involve scraping and indexing Wikipedia articles related to conspiracy theories. These articles could be semantically matched to user queries using a second vector database, allowing the tool to present both relevant forum discussions and informative, fact-based resources.

While the tool is currently intended as an experimental project, primarily for users who are already critically engaged with the topic of conspiracy theories, it also carries a certain entertainment value. However, this "fun" aspect must not outweigh the potential risks when the tool is accessed by less discerning users. Responsible design must therefore prioritize the inclusion of explanatory and educational content to ensure the system is not misused or misunderstood.

In summary, the prototype demonstrates the technical feasibility and potential of AI-assisted classification in online conspiracy discussions. With further evaluation, data refinement, and ethical safeguards, this tool could serve both as a research instrument and as a resource for understanding and contextualizing conspiracy narratives online.