COMMENTS ANALYSIS IN SOCIAL MEDIA BASED ON LLM AGENTS

David Ramamonjisoa and Shuma Suzuki

Iwate Prefectural University, Faculty of Software and Information Science, postbox 020-0693

ABSTRACT

This paper presents an online tool that automatically filters comments on YouTube based on agents built with Large Language Models (LLMs). We designed and built agent-based comment filtering system driven by the Langchain framework and LLMs to allow users a better interface for comments reading and discussions. Specialized agents can detect spam, toxic or constructive comments. Agents are constructed as modular Langchain object with memory components and reasoning tools to make a chain of thought by interacting with LLMs. In this paper, we provide a comprehensive detail of the agent architecture and workflow to fulfil its task. Additionally, we compare the generated result by the agents to our previous tool available online. We discuss our expectations regarding the future of the comment system.

KEYWORDS

YouTube comments, Large Language Model, Agents, Filtering and Visualization

1. INTRODUCTION

In today's digital landscape, YouTube has emerged as a prominent platform for sharing and consuming video content. With its vast user base and diverse content offerings, YouTube has become a hub for engagement and interaction among individuals worldwide. This engagement is often facilitated through the comments section, where viewers can express their thoughts, opinions, and reactions to the videos they watch. While comments can provide valuable insights and foster a sense of community, they can also harbor negative content, including toxic, insulting, or offensive remarks. These detrimental comments can hinder constructive dialogue, discourage participation, and even create a hostile environment for users. Recognizing the potential impact of negative comments, YouTube has implemented measures to moderate and filter comments based on their algorithm and user feedback. However, these measures may not always capture the nuances of language and the intent behind comments. To address this challenge, we have developed an online tool that employs advanced machine learning techniques to analyze YouTube comments and categorize them based on several key criteria.

In this paper, we present an online tool that automatically filters comments on YouTube based on agents built with Large Language Models (LLMs). We designed and built an agent-based comment filtering system driven by the Langchain framework and LLMs to provide users with a better interface for reading and discussing comments. Specialized agents can detect spam, toxic or constructive comments. Agents are constructed as modular Langchain objects with storage components and inference tools to form an inference chain by interacting with LLMs.

The paper is organized as follows. First, we describe the agent framework and related research. Then, we describe our agent-based comment analysis framework and the development of the different models. Finally, we show our experimental results and deployment as a comment analysis website. We conclude this paper with some reflections on the benefits of our system and future work.

2. AGENT FRAMEWORK

LLM applications are the combination of LLM and agents. Some literature refers to these applications as bots. The most common application is chat bots. The agents are the autonomous entities that control these LLM bots. The agent can observe, act, generate tasks, and make specific decisions within a controlled environment to achieve predetermined goals or objectives. Examples of tasks such as personal assistance, information retrieval, question answering, or summarizing can be performed by agents in LLMs. More difficult tasks that require a combination of these examples and planning can be accomplished with multi-agent workflows.

Table 1. Four agent framework comparison.

Feature	**Copilot Studio**	**LangChain**	**AI Agent**	**SuperAgent**
Purpose	Customizable AI assistant for Microsoft 365	AI-driven agent for task execution	General AI agent framework	AI agent with advanced capabilities
Integration	Seamless with Microsoft ecosystem	Can integrate with software ecosystems	Broad integration capabilities	Designed for complex integrations
Customization	High, with low-code environment	Moderate, with focus on planning tasks	High, flexible agent roles	High, with emphasis on inter-agent communication
Pricing	Subscription-based after trial	Not specified	Varies by implementation	Not specified
Use Case	Business process automation	Task-oriented applications	Diverse applications	Complex task execution

Table 1 shows a comparison of four agent framework available and ready to use for LLM applications [1],[2],[3],[4]. Software agents, with their potential to automate and streamline tasks, are poised to become increasingly prevalent in the near future. In this paper, we choose LangChain because it is an open source and free to use.

2.1. LangChain Framework

LangChain [2] is a cutting-edge agent framework that enables the creation of sophisticated AI agents capable of understanding and executing complex tasks. It serves as a foundational tool for developers to construct and deploy agents that can interact with users, systems, and other agents in a coherent and context-aware manner. By utilizing LangChain, developers can craft agents that not only respond to queries but also proactively engage in problem-solving, data analysis, and decision-making processes. The framework's modular design allows for seamless integration with existing systems, providing a robust platform for developing the next generation of intelligent agents.

The LangChain documentation provides several ready-to-use agents applied to specific use cases. For example, csv_agent, pandas_dataframe_agent, and python_agent are agents that interact with files in tabular format, agents that can perform dataframe operations, and agents that can execute Python code using the Python REPL (Read-Eval-Print Loop) tool to execute commands provided by LLM. When an agent is executed in LangChain, an "agent executor" chain is initialized. This agent executor is the runtime of the agent.

2.2. Multiagent Framework

The multi-agent paradigm offers a compelling approach to tackling project complexity and supporting collaborative workflows. ChatDev by Chen Qian et al. [5] is an example aiming to achieve a virtual software company managed by AI multiagent integrating LLMs.
A library built on top of LangChain called LangGraph[1] for building stateful, multi-actor applications with language models is another example of a multiagent framework. It extends the LangChain Expression Language (LCEL) with the ability to coordinate multiple chains or actors across multiple computation steps, which is particularly useful for agent-like behaviors and workflows that involve cycles.

3. AGENT-BASED COMMENT ANALYSIS FRAMEWORK

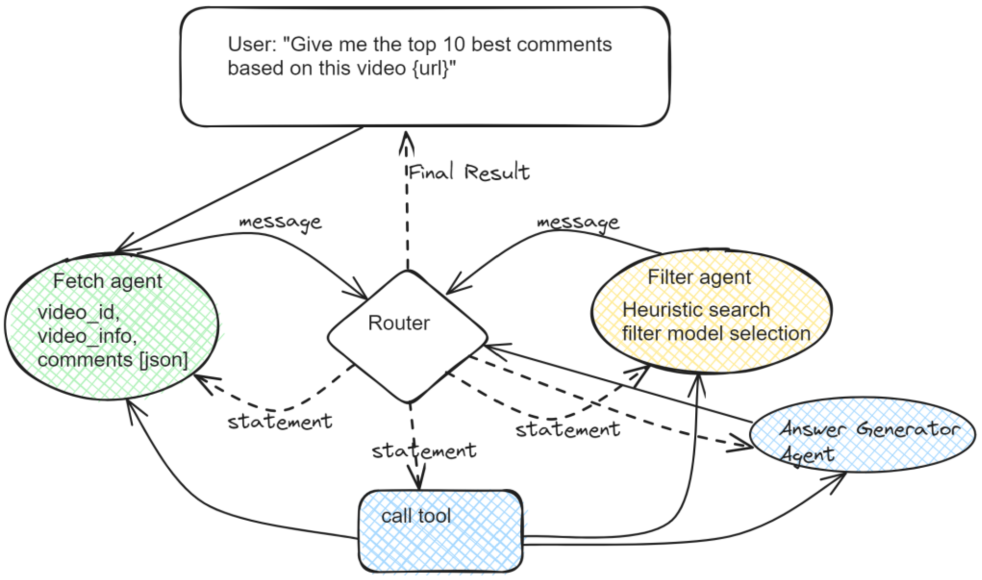


Figure 1: Our YouTube comment analyzer

We propose a multiagent system to implement our new comment filtering system as shown in Figure 1. The prompt from user is processed that it calls first the Fetch agent to collect all data concerning the video by using Youtube API. The Router is an IF statement agent that receives messages from all agents and select the appropriate agent by using the call_tool to dispatch the task according to its plan. The final result is then presented to the user after all execution.

[1] https://python.langchain.com/docs/langgraph/

4. EXPERIMENTAL RESULT

4.1. LangChain Framework Experience

Using our multiagent system, we tested the strength of LLM OpenAI GPT4 to interact with a Kaggle dataset called Tate & Morgan Viral Interview: 50K YT Comments[2]. The user input is interpreted by the OpenAI agent and passed to the dataframe agent by generating the Python code. The dataframe agent executes the code and returns the result. The answer generator agent transforms the result into natural language for the final answer. We performed the exploratory data analysis without entering the Python code, which is generated and executed by the system. We can do sentiment analysis without programming by asking the system in plain English or Japanese. We can also do topic modeling, hate speech detection, or constructive comment extraction without knowing how to program. We still configure the output to get the same graphical result as with matplotlib plotting or wordcloud result. Code generation is not straightforward because the OpenAI agent can't capture the user's intent from the first prompt, and it can't test the code directly, so it needs iterative feedback from the other agents and the user to get it right. Human user patience is still required.

4.2. Real-World Application and User Interface

We have developed an online tool that employs advanced machine learning techniques to analyze YouTube comments and categorize them based on several key criteria:

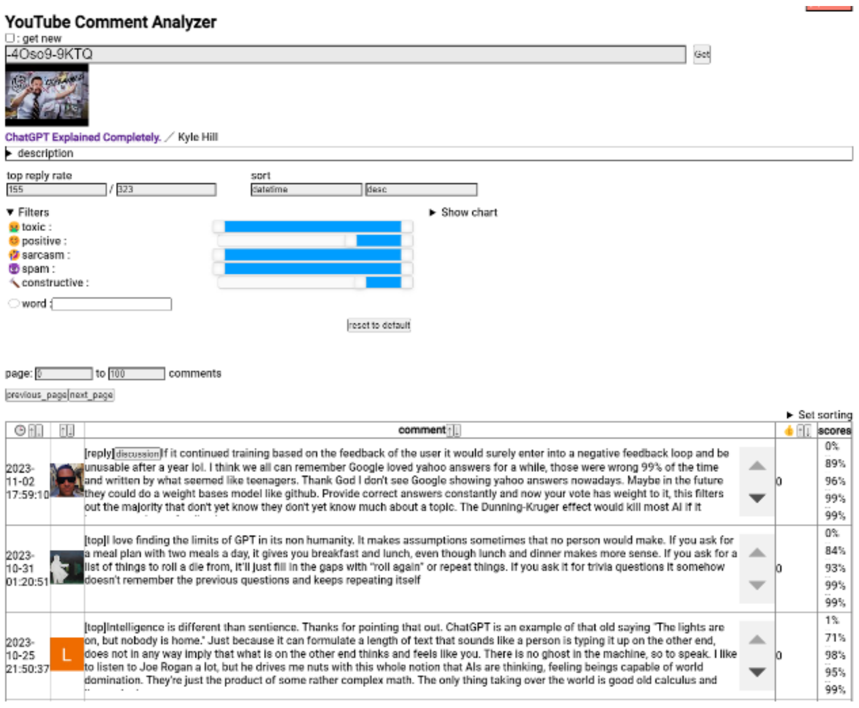
- Harmful: Comments that are discriminatory, hateful, or promote violence.
- Spam: Comments that are repetitive, irrelevant, or promotional in nature.
- Sentiment: Comments that express emotions such as happiness, sadness, anger, or fear.
- Ironic: Comments that use sarcasm or humor to convey a message.
- Constructive: Comments that provide valuable feedback, suggestions, or insights.

Our online tool utilizes six deep learning models trained on comprehensive datasets of YouTube comments to accurately classify comments based on these criteria [6].

By leveraging our six predictors, we can analyze any comments on YouTube, as illustrated in Figure 1. From left to right, the results display the post date, contributor icon, comment, like/dislike buttons, toxic (yes/no), sentiment (pos/neg), sarcastic, spam (yes/no), and constructive (yes/no) classifications. Users can click each arrow to sort the results in ascending or descending order. The percentage value in each column represents the probability result from the corresponding predictor.

Our proposed tool provides a valuable resource for YouTube users to manage their comment experience and foster a more positive and constructive online environment.

[2] https://www.kaggle.com/datasets/kanchana1990/tate-and-morgan-viral-interview-50k-yt-comments



YouTube Comment Analyzer

☐ : get new

-4Oso9-9KTQ

ChatGPT Explained Completely. ⁄ Kyle Hill

top reply rate: 155 / 323

sort: datetime desc

		comment		scores
2023-11-02 17:59:10		[reply] discussion If it continued training based on the feedback of the user it would surely enter into a negative feedback loop and be unusable after a year lol. I think we all can remember Google loved yahoo answers for a while, those were wrong 99% of the time and written by what seemed like teenagers. Thank God I don't see Google showing yahoo answers nowadays. Maybe in the future they could do a weight bases model like github. Provide correct answers constantly and now your vote has weight to it, this filters out the majority that don't yet know they don't yet know much about a topic. The Dunning-Kruger effect would kill most AI if it	0	0% 89% 96% 99% 99%
2023-10-31 01:20:51		[top]I love finding the limits of GPT in its non humanity. It makes assumptions sometimes that no person would make. If you ask for a meal plan with two meals a day, it gives you breakfast and lunch, even though lunch and dinner makes more sense. If you ask for a list of things to roll a die from, it'll just fill in the gaps with "roll again" or repeat things. If you ask it for trivia questions it somehow doesn't remember the previous questions and keeps repeating itself	0	0% 84% 93% 99% 99%
2023-10-25 21:50:37	L	[top]Intelligence is different than sentience. Thanks for pointing that out. ChatGPT is an example of that old saying 'The lights are on, but nobody is home.' Just because it can formulate a length of text that sounds like a person is typing it up on the other end, does not in any way imply that what is on the other end thinks and feels like you. There is no ghost in the machine, so to speak. I like to listen to Joe Rogan a lot, but he drives me nuts with this whole notion that AIs are thinking, feeling beings capable of world domination. They're just the product of some rather complex math. The only thing taking over the world is good old calculus and	0	1% 71% 98% 95% 99%

Figure 2: Our YouTube comment analyzer

As shown in Figure 2, the tool can sort and filter on predicted values. The “top reply rate” means the number of top-level comments divided by the number of total comments. It is the ratio. The filter sets the lower and upper bounds of the predicted value so that system displays only comments within that range.

The main advantage of our tool in comparing to the conventional comment in YouTube is the filtering commands that allow user to control the comments according the predictors. In some contents where comments are easily attaining the order of 10,000 or more, users are only viewing the most viewed according to the YouTube algorithm.

5. CONCLUSIONS

In this paper, we propose an agent-based comment filtering system to improve the readability of YouTube comments using a tool powered by LLMs. This innovative approach deviates from YouTube's default comment display, allowing users to engage with new knowledge and communities in a more meaningful and effective way.

In our experiment, we demonstrate the potential of using LLMs and multi-agent architectures to make data analysis more accessible to those without programming expertise.

As mentioned earlier, further improvements are needed, including the development of a user-friendly interface, the construction of robust multilingual models, and the expansion of our dataset collection efforts.

Future Directions and Ongoing Research

To further enhance the effectiveness of our tool, we are actively pursuing several research directions:

- User Interface Optimization: We are continuously refining the tool's user interface, ensuring a seamless and intuitive user experience.
- Multilingual Model Development: We are expanding our multilingual capabilities to cater to a wider global audience.
- Dataset Expansion: We are actively seeking out and incorporating additional high-quality datasets to enrich our modeling efforts.

REFERENCES

[1] Qingyun Wu et al.(2023) AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. https://doi.org/10.48550/arXiv.2308.08155, https://github.com/microsoft/autogen

[2] Chase, Harrison (2022) https://python.langchain.com/docs/get_started/introduction

[3] https://aiagent.app/

[4] https://superagent.sh/

[5] Chen Qian et al. (2023) Communicative Agents for Software Development. https://doi.org/10.48550/arXiv.2307.07924, https://github.com/OpenBMB/ChatDev

[6] David Ramamonjisoa, Hidemaru Iguma, Riki Murakami. "Filtering Relevant Comments in Social Media Using Deep Learning," In Proceedings of the 2022 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), pp. 335-340, Niagara Fall, Ontario, Canada, November 2022.

AUTHORS

David Ramamonjisoa is an associate professor at the Faculty of Software and Information Science of Iwate Prefectural University in Japan.

Shuma Suzuki is a 4th year undergraduate student at the Faculty of Software and Information Science of Iwate Prefectural University in Japan.