Intelligent Software Agents Project Report

Creator Automatic Assistant

TEAM MEMBERS

Tang Haoran A0285856J

Chen Haoquan A0286061E

Liang Zhu A0285771R

Zhang Yusen A0285839H

Chen Zhiwei A0285952N

MASTER OF TECHNOLOGY

Contents

1.0 Executive Summary

In today's fast-paced digital marketing landscape, the ability to produce and disseminate engaging content across diverse social media platforms is crucial for businesses aiming to forge meaningful connections with their audience. However, the traditional manual methods of crafting and distributing content are not only labor-intensive but also prone to inefficiency and errors. To address these challenges, we have crafted an innovative solution designed to streamline and automate the entire content generation and publication workflow.

Our cutting-edge tool harnesses the power of Larger Language Model (LLM) and Robotic Process Automation (RPA) to revolutionize how content is created and shared. By simply inputting keywords or uploading images, users can prompt our platform to generate dynamic text and visually appealing image content that is specifically tailored to resonate across different social media channels. This integration of LLM with robust RPA technology ensures that our tool not only saves valuable time but also minimizes the risk of human error, thereby boosting the overall quality and effectiveness of online content.

Additionally, our platform is equipped with the capability to generate up-to-date content by analyzing real-time online materials. This feature addresses the common issue of knowledge latency in large models, ensuring that the content remains fresh and relevant. Currently, our tool supports automated publishing across multiple platforms, including XIAOHONGSHU, ZhiHu, Twitter, and Weibo, facilitating a seamless and efficient content management process across the most influential social media networks.

Moreover, our solution is designed with a user-centric approach, featuring an intuitive interface that makes it accessible to marketers of all skill levels. It offers extensive customization options, allowing users to fine-tune their content to meet specific branding requirements and audience preferences. The reliability and performance of our tool are unmatched, providing businesses with a dependable resource that enhances their digital footprint and competitive edge in the bustling digital marketplace.

By automating the content creation and publication process, our solution empowers businesses to focus more on strategic initiatives and less on the operational complexities of digital marketing, paving the way for enhanced productivity and success in their digital engagement efforts.

2.0 Problem Description

In digital marketing, manually creating and publishing content across various social media platforms is both time-consuming and prone to errors, which can negatively impact brand perception and audience engagement. Additionally, traditional methods often fail to quickly incorporate the latest trends and real-time data, resulting in outdated content. Each platform's unique requirements further complicate content adaptation, making the process cumbersome. There is a critical need for an automated solution that streamlines the content creation and publication process, enhances accuracy, ensures relevance, and adapts content to meet the diverse specifications of different platforms, thereby improving engagement and effectiveness.

2.1 Project Objective

In response to the increasing demand for streamlined and efficient content creation and publication within the digital marketing landscape, our project introduces an innovative automated tool that harnesses the capabilities of Large Language Models (LLM) and Robotic Process Automation (RPA) technologies. Our primary objective is to significantly simplify the content creation and publishing process, thereby saving users substantial time and effort. Additionally, we aim to enhance the quality and extend the reach of the content produced, ensuring it resonates effectively with diverse audiences across multiple platforms.

To achieve these goals, we have structured our system work procedure around several key actions:

1. Development of a User-Friendly Interface: We have created an intuitive interface that allows users to easily input keywords and manage content preferences. This interface is designed to be accessible to users of all technical skill levels, facilitating a smoother content creation process.
2. Integration of Large Language Models (LLM): Our tool utilizes advanced LLM technology to generate compelling text content based on the provided keywords and images. This integration ensures that the content is not only relevant but also engaging and tailored to capture the audience's interest.
3. Implementation of Robotic Process Automation (RPA): We have incorporated RPA tools to automate the publishing process across various social media platforms. This automation reduces the manual effort required and minimizes the risk of human error during publication.
4. Provision of Customization Options: Recognizing the unique requirements of different social media platforms and audience demographics, our tool offers extensive customization options. Users can tailor their content to align with specific platform norms and audience preferences, enhancing the effectiveness of their digital marketing efforts.

Through these strategic initiatives, our project aims to empower businesses and content creators with a robust tool that not only streamlines the content creation and publication process but also boosts the overall impact and efficiency of their digital marketing campaigns.

2.2 Project Team

The project members’ name and work items are listed below:

Name	Work Items
Tang Haoran	Leader Frontend Development Documentation Video
Chen Haoquan	Model Development Documentation Video
Liang Zhu	Backend Development Documentation
Zhang Yusen	RPA Development Video
Chen Zhiwei	RPA Development Video

Table 1. Project contribution

Our team met once a week to discuss project status and updates. Tasks were assigned and areas of responsibilities were clearly outlined for each member.

2.3 Key Features

1. Content Tailoring: Generate text and image content based on user inputs and pre-set preferences tailored for specific platforms.
2. Multi-platform Integration: Seamlessly publish content to various social media platforms such as Twitter, Facebook, Instagram, and more.
3. User Experience: Simple, intuitive user interface for easy operation by users of all technical skill levels.
4. Customization and Flexibility: Options to customize content style, timing, and platform-specific adjustments.
5. Scalability and Reliability: Designed to handle high volumes of content generation and publication with minimal downtime.

2.4 Commercial viability

2.4.1 Market Research

The market for content automation tools is expanding rapidly, driven by the increasing importance of digital marketing and the growth of social media as a primary platform for consumer engagement. Research indicates that the market for marketing automation tools is expected to grow significantly in the coming years, with a compound annual growth rate (CAGR) of over 9.2% from 2020 to 2027. Businesses, especially small and medium

enterprises (SMEs), are looking for solutions that can help them compete effectively without the need for extensive marketing departments.

2.4.2 Competitive Analysis

Existing Solutions: Several platforms currently offer partial solutions, such as automated scheduling or content curation. However, few integrate both content creation and publication seamlessly.

Market Gaps: There is a significant opportunity for a tool that not only automates content generation based on contextual understanding but also customizes this content for different social media platforms.

Target Audience: Our primary market includes SMEs, digital marketers, and social media managers who seek to optimize their content strategy without increasing their workload.

Customer Feedback and Validation:

Surveys and Interviews: Initial feedback from potential users highlights a strong interest in reducing the time spent on content creation and publication, with a keen interest for integration and usability being major deciding factors.

Pilot Testing: Plans to conduct pilot testing with select businesses to gather detailed feedback on the system's effectiveness and user-friendliness before a full-scale launch.

By leveraging our findings from this extensive market research, we aim to refine our product to meet the specific needs of marketers and businesses, ensuring a strong product-market fit and a successful entry into the competitive landscape of digital marketing tools.

3.0 Solution

3.1 System Architecture

The system is designed following the Frontend and Backend Separation Architecture, which makes the system highly componentized with high scalability and maintainability. Based on function, this system architecture involves front-end, back-end, AI, automation, and integration with social platforms. The architecture is designed to provide efficient and automated service workflows through close collaboration between various components. Here is a detailed analysis of the system architecture:

User Interface (Front-end)

- React: The front-end is built using the React framework, a robust JavaScript library for building user interfaces. React enables the system to deliver dynamic single-page applications (SPAs), enhancing user interaction experiences.
- Request and Response Handling: Users initiate requests through the front-end interface. The system sends these requests to the back-end and receives processed responses asynchronously.

Server Side (Back-end)

- Server: The server component is responsible for receiving requests from the front-end, performing preliminary processing such as validation and routing. The back-end utilizes the Flask framework, a lightweight web application framework developed in Python. Flask's simplicity and ease of use facilitate rapid development and deployment.
- Agent: The agent acts as an intermediary, handling complex logic, interactions with downstream AI services, data processing, and result aggregation. LangChain is used for executing complex text analysis and generation tasks.

AI Integration

- The design supports both cloud and local AI processing, allowing the selection of the most suitable execution environment based on task requirements and resource optimization.
- OpenAI: The system integrates OpenAI's APIs to leverage powerful machine learning models (like the GPT series) for tasks requiring deep language understanding and generation.
- Computer Vision (CV): Custom computer vision models are employed for image recognition or processing tasks.

Automation and Platform Integration

- RPA: TagUI is used for robotic process automation, automating the writing, and posting of social media content, reducing manual intervention and increasing efficiency.
- Social Platform Integration: The system is integrated with multiple social platforms (such as Xiaohongshu, Weibo, etc.), automating content publication and social media management.

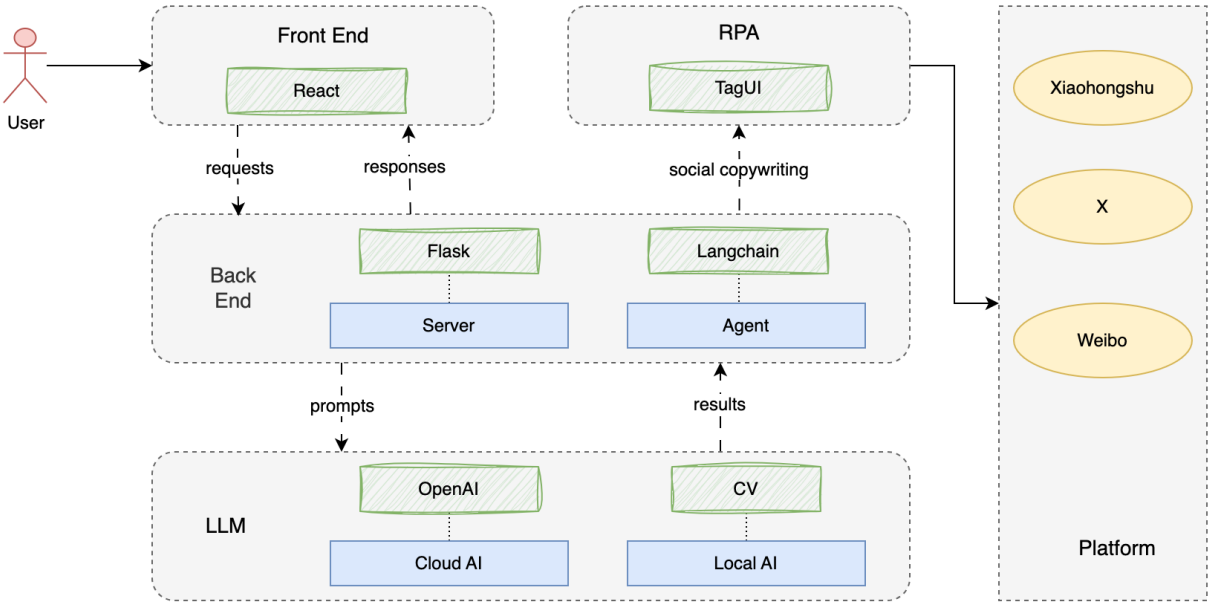

Figure 1. System Architecture

3.2 System implementation

3.2.1 Front end

Technology Stack

- React: Chosen for building the user interface due to its component-based architecture and efficient update mechanisms.
- Redux or Context API: Used for state management to handle user inputs, server responses, and overall application state.
- Axios: Used to handle HTTP requests for data interaction with the backend.

Frontend Component Design

1. User Input Interface
   - Input Form: A form component that allows users to input necessary information such as topic, style, platform, etc.
   - Submit Button: A button used to trigger the data submission process.
2. Data Display Interface
   - Results Display: Displays data returned from the backend and LLM.
   - Editable Area: Allows users to edit the results generated by the LLM.
3. Publishing Interface
   - Publish Button: Used to publish the finalized content to the specified platform.

Technical Implementation Details

- React Components: Utilize React Hooks (such as useState, useEffect) to manage component states and lifecycles.
- State Management: Use Redux for sharing state across components, such as user inputs and server responses.
- Error Handling: The frontend needs to handle errors effectively, such as failed network requests or backend service unavailability.
- User Feedback: Provide clear feedback to users at every critical step (like submission, publishing), including loading indicators, success/failure notifications, etc.

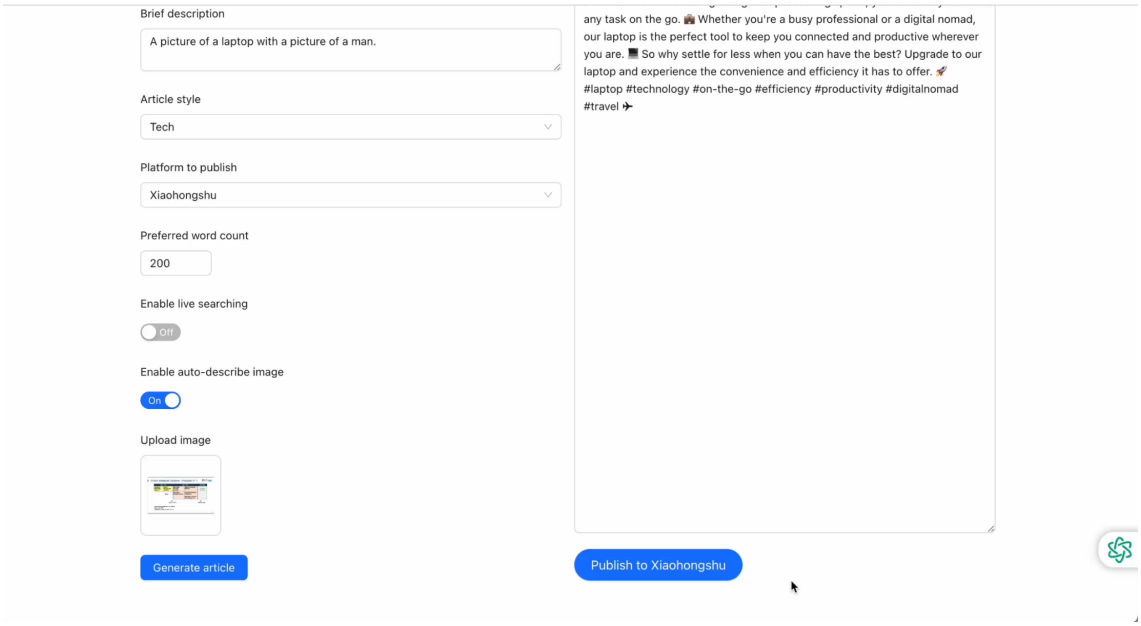

Figure 2. User Interface 1

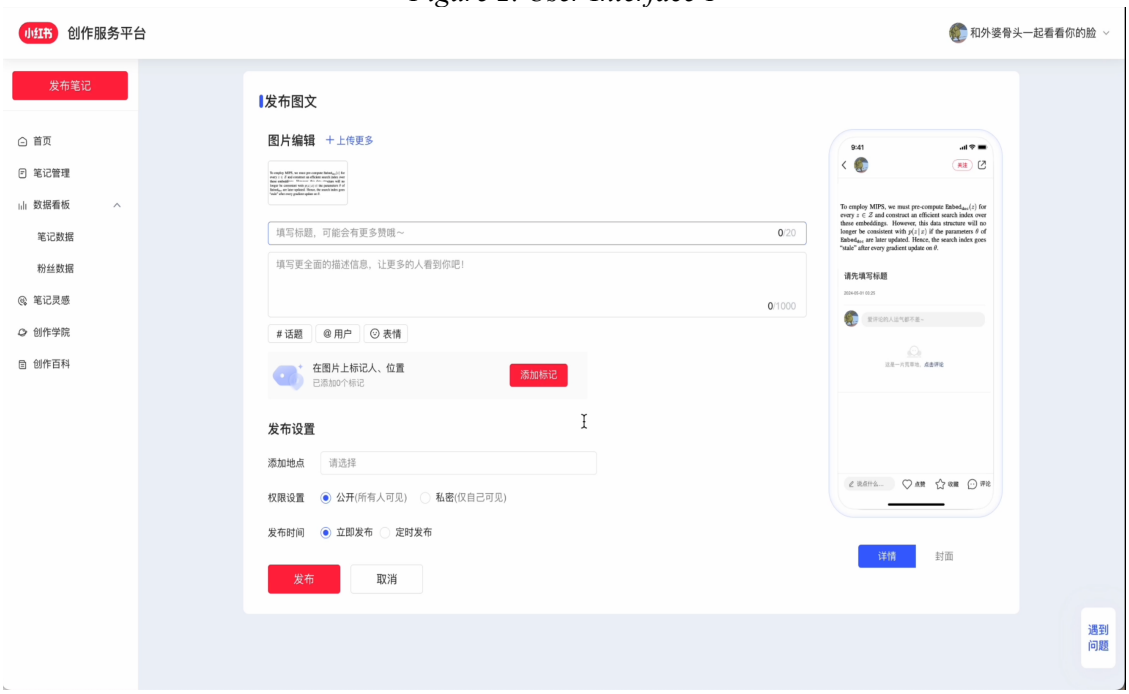

Figure 3. User Interface 2

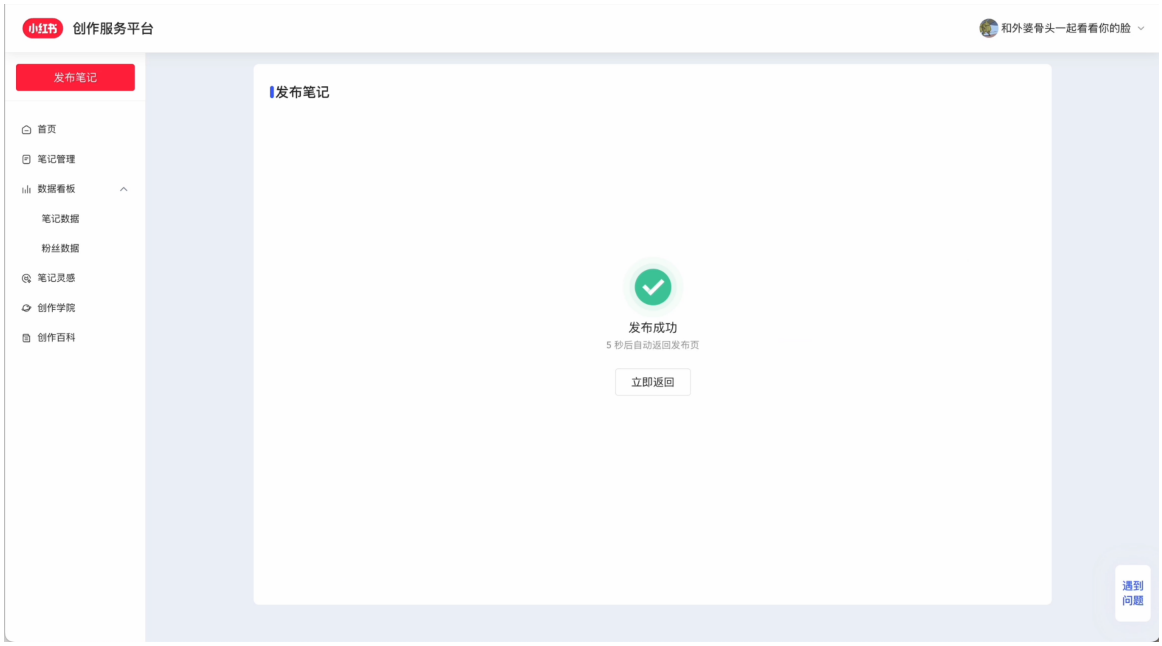

Figure 4. User Interface 3

3.2.2 Back end

Technology Stack

- Flask: A lightweight and flexible Python web framework to handle HTTP requests and serve as the backend server.
- LangChain: A library to integrate advanced NLP functionalities, particularly useful for leveraging ChatGPT in generating content based on user inputs.

Backend Architecture Design

The backend not only supports real-time content generation using advanced NLP but also seamlessly automates the content publication process across multiple platforms, significantly reducing manual effort and enhancing the efficiency of digital content management.

1. API Layer
   - RESTful APIs: Develop APIs to receive user inputs (keywords or tags), trigger content generation, and handle requests for publishing content.
2. Business Logic Layer
   - Content Generation Service: Utilizes NLP models via LangChain to generate text content based on the provided keywords or tags.
   - Publishing Service: Coordinates with RPA tools to automate the process of posting content to designated platforms such as Xiaohongshu, Weibo, Zhihu, and Twitter.
3. Data Access Layer
   - Models: Define data models for users, content, and publishing logs.
   - DAOs (Data Access Objects): Abstract and encapsulate all database interactions.
4. Integration Layer
   - NLP Integration: Integrate with NLP models through LangChain to process and generate content.

- RPA Integration: Implement connections to RPA scripts or tools that handle the login and posting mechanics on various platforms.

Key Functional Implementations

1. Processing User Inputs
   - APIs to capture and validate user inputs.
   - Pass inputs to the NLP service for content generation.
2. Content Generation
   - Use LangChain to interface with ChatGPT, generating content that is contextually aligned with the provided keywords.
   - Handle post-generation processing such as content formatting and validation before it's sent for publication.
3. Automated Publishing
   - Once content is generated and approved, trigger RPA workflows to log into various social media platforms and post the content.
   - Manage credentials and session information securely to ensure robust automation.

3.2.3 RPA

Technology Stack

- TagUI: Chosen for its ability to automate web and desktop applications, with native support for integrating AI capabilities. It's particularly adept at handling both text and image content across different platforms.

RPA Workflow Design

1. Initialization

Configuration: Set up TagUI workflows tailored to each target social media platform. This includes configuring the navigation paths, login procedures, and specific steps needed to post content on each site.

2. Authentication

Automated Login: Develop scripts to handle the login process for each platform securely. This includes managing credentials securely, filling in login forms, and handling any CAPTCHAs or two-factor authentication steps that might be required.

3. Content Posting

Content Formatting and Upload: Automate the steps involved in formatting the content as required by each platform and uploading any associated images or media.

Submission: Automate the final steps to submit or post the content, including handling any confirmation dialogs or additional input required by the platform.

4. Multi-platform Support:

Create distinct RPA workflows for each platform (Xiaohongshu, Weibo, Zhihu, Twitter), customized to handle the unique elements of each platform's posting process.

Ensure that the RPA tool can adapt to different authentication mechanisms and content posting interfaces.

5. Scheduled and Event-Triggered Execution:

Utilize cron jobs or a task scheduler to execute RPA workflows either at predetermined times or in response to specific triggers, such as the completion of content generation.

Implement queue management to handle scenarios where multiple pieces of content are ready for posting simultaneously, balancing the load to avoid platform rate limits and ensure timely publication.

3.2.4 LLM

Cloud AI – OpenAI

To utilize the LangChain framework to interact with the OpenAI API, developers can harness the power of advanced language models to build and deploy AI-driven applications. LangChain, a library designed for building language-centric applications, integrates seamlessly with various AI technologies, including those provided by OpenAI. OpenAI, known for its cutting-edge AI research and development, offers robust APIs like GPT (Generative Pre-trained Transformer) models that excel in natural language understanding and generation. By leveraging LangChain, developers can easily orchestrate these powerful models within their applications, enabling sophisticated language-based interactions and functionalities.

Local AI - Image caption

1. Introduction:

Image captioning is a challenging task in computer vision and natural language processing, where the goal is to generate natural language descriptions for images automatically. This project implements an image captioning solution using our custom model architecture.

2. Model Architecture:

- Vision Transformer Backbone: The backbone architecture processes input images and extracts high-level visual features. It employs a transformer-based architecture that replaces convolutional layers with self-attention mechanisms to capture global dependencies in the image.
- Transformer Decoder: The transformer decoder generates captions autoregressively by attending over both the embedded image features and the generated partial captions. It utilizes self-attention mechanisms to capture contextual information and generate coherent and contextually relevant captions.

3. Algorithm Principles:

- Transformer Architecture: The transformer architecture, introduced in the "Attention is All You Need" paper by Vaswani et al., forms the basis of our model. It relies on self-attention mechanisms to capture global dependencies between input tokens, enabling effective modeling of both short and long-range dependencies.
- Self-Attention Mechanisms: Self-attention mechanisms allow the model to weigh the importance of different input tokens based on their relevance to each other. This enables the

model to focus on relevant parts of the image and partial captions during caption generation, improving the quality of generated captions.

- Autoregressive Generation: The transformer decoder utilizes autoregressive generation, where each token in the output sequence is generated sequentially based on previously generated tokens. This allows the model to capture dependencies between tokens and generate coherent and contextually relevant captions.

4. Functionality:

- Training: The project supports training our model on custom datasets for image captioning tasks. It provides functionality for loading and preprocessing image and caption data, defining model architecture, loss functions, optimizers, and training procedures.
- Fine-tuning: It includes functionality for fine-tuning pre-trained models on specific datasets to adapt them to new domains or tasks. Fine-tuning allows leveraging pre-trained weights to achieve better performance on domain-specific datasets with limited training data.
- Inference: Once trained, the model can generate captions for input images using the provided predict.py script. Inference involves passing images through the trained model and generating captions based on the extracted features.
- Hub Integration: Pre-trained versions of our model are available through torch.hub, allowing easy access to different model versions. Users can load pre-trained models directly from the hub and use them for inference or fine-tuning.

3.3 Process Flow

Here is a detailed description of the process flow, which you can use to write your report:

1. User Interaction (User):

Users input keywords, style, platform choice, and images through a front-end interface. These inputs form the basis for customized content generation.

2. Front End Processing (Front End):

The front-end system collects input from users and sends this data to the back-end server. The data includes topics, styles, target platforms, and images.

3. Back End Processing (Back End):

The back end receives the data from the front end and processes it to prepare for sending to the Language Model (LLM).

The back end also handles user login details to ensure content is posted to the correct social media accounts.

4. Language Model Generation (LLM):

The back end converts user inputs (such as topics and keywords) into prompts and sends these to the language generation model.

The language model generates text content based on these prompts and sends the generated content results back to the back end.

5. Content Editing and Preparation for Publishing (Edit LLM Results + Publish):

Users receive the text generated by the language model on the front-end interface and can edit or modify it to meet their needs.

Once editing is complete, users can opt to publish the content. The edited content is then sent back through the back end.

6. Robotic Process Automation (RPA):

The back end sends the edited content to the RPA system.

The RPA is responsible for automatically logging into specified social media platforms (such as Xiaohongshu, Weibo, Zhihu, and Twitter) and publishing the content.

7. Feedback on Publication Results (Return Post Results):

Once the content is successfully published, the RPA sends the results back to the back end.

The back end then relays these results to the front end, ultimately displaying them to the user to inform them of the status of the publication (success or failure).

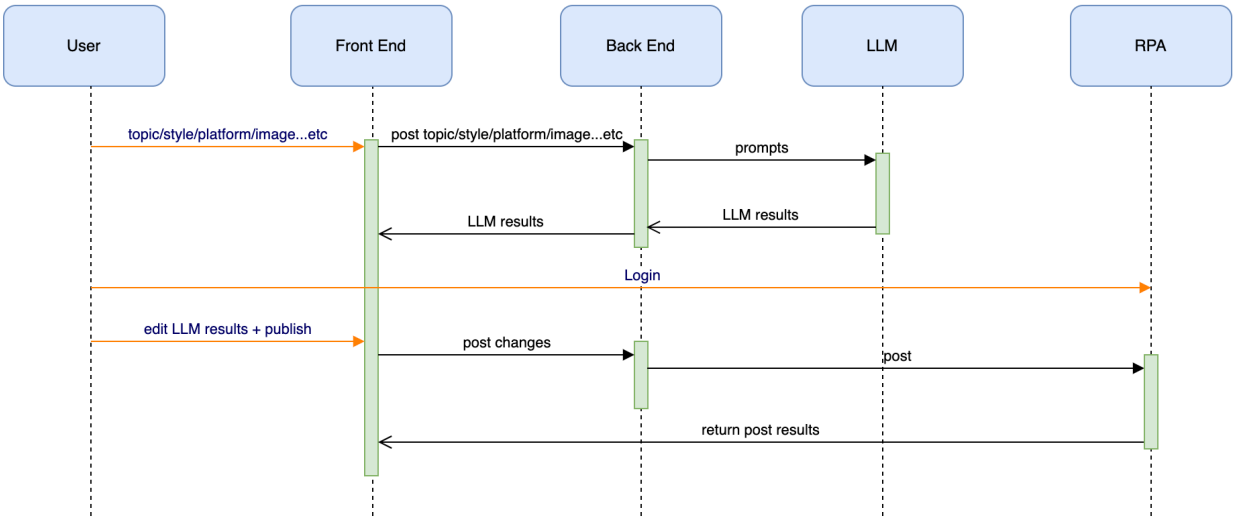

Figure 5. System Architecture

3.4 System's Features

Time Saving

- Automated Content Generation and Publishing: The system automates the entire process of content creation and publication, leveraging NLP and RPA technologies. Users input their preferences once, and the system generates and posts content across multiple platforms without further manual intervention. This significantly reduces the time spent on repetitive tasks such as writing, editing, and logging into various social media accounts to post content manually.
- Efficiency in Handling Multiple Platforms: By integrating with multiple social media platforms like Xiaohongshu, Weibo, Zhihu, and Twitter, the tool allows users to manage content distribution from a single point. This is much more efficient compared to the traditional method of accessing each platform individually, potentially saving users hours of work each day.

Easy Using

- User-Friendly Interface: The tool features a straightforward front-end interface where users can easily input their content preferences such as keywords, styles, and images. This simplicity ensures that even users with minimal technical skills can operate the system effectively.
- Seamless Integration and Login: Like the Intelligent Job Hunter system, users could potentially log in via simple methods such as QR codes or basic authentication processes. This reduces the complexity typically associated with multiple logins for different platforms.

Productive Result

- Tailored Content Creation: The use of a sophisticated language model ensures that the content generated is not only relevant to the input keywords but also styled appropriately for different platforms. This tailored approach can increase engagement on social media, as content that resonates well with a specific audience is more likely to perform better.
- Increased Opportunities for Engagement: By consistently creating and posting high-quality, platform-optimized content, the tool helps in building a stronger online presence. This can lead to increased followers, higher engagement rates, and more opportunities for monetization or influence.

4.0 Conclusion

In this project, our team successfully developed an innovative platform based on React, Flask, LangChain technology, and artificial intelligence with large language models (LLM). The main functionality of this platform is to utilize LLM to automatically generate copy that aligns with the styles of various social media platforms and supports the auto-publishing of this content. Additionally, our system is capable of real-time online content search, providing the necessary data support for LLM to generate copy.

This platform not only supports users in generating copy based on specified themes and styles but also automatically generates relevant copy from user-uploaded images. This feature significantly enhances the flexibility and adaptability of content creation. Before posting to social media, users can also personalize and adjust the copy generated by LLM, ensuring that the final content fully meets their expectations and brand image.

By adopting the Minimum Viable Product (MVP) methodology, we effectively controlled the development process, ensuring the project's efficient progression. Throughout the development process, we faced numerous challenges, particularly in optimizing the inference results of the large models.

In conclusion, through the implementation of this project, we have not only significantly enhanced the level of automation in content generation but also provided users with a powerful tool to help them more effectively conduct marketing and brand building in a diverse social media environment.

5.0 Bibliography

[1] **React**: https://reactjs.org/

[2] **LangChain**: https://langchain.com/

[3] **OpenAI**: https://www.openai.com/

[4] **TagUI**: https://tagui.readthedocs.io/en/latest/

[5] **Flask**: https://flask.palletsprojects.com/

[6] **SerpApi**: https://serpapi.com