Intelligent Email Assistant System

This is an intelligent email assistant system based on the DeepSeek AI large language model. It helps users accomplish various email-related tasks, including grammar correction, tone/style adjustment, content rewriting suggestions, multilingual support, auto-completion, template provision, email summarization, and email generation for specific scenarios.

Feature List

- **Multi-turn Dialogue**: Understands context and supports continuous multi-turn conversations for coherent interactions.
- **Streaming Output**: Displays AI-generated content in real time, providing a better user experience and reducing wait time.
- **Grammar Correction**: Checks and corrects grammar errors in emails to improve professionalism.
- **Tone and Style Adjustment**: Adjusts the tone of emails (formal, friendly, professional, etc.) as needed.
- **Content Rewriting Suggestions**: Provides suggestions for rewriting email content to make it clearer and more effective.
- **Multilingual Support**: Supports translating emails into different languages to meet international needs.
- **Email Templates**: Offers a variety of commonly used email templates (meeting invitations, leave requests, customer complaint replies, quotations, payment reminders, etc.).
- **Email Summarization**: Automatically summarizes lengthy emails and extracts key information.
- **Scenario-based Email Generation**: Generates appropriate email content based on simple instructions (e.g., "send a follow-up reminder").

Technical Architecture

The system adopts a front-end and back-end separation architecture, mainly divided into three core parts:

1. Frontend (HTML/JavaScript)

- **UI Framework**: Uses Tailwind CSS for modern, responsive UI design.
- **Interaction**: Provides an intuitive user interface supporting various email processing functions.
- **Real-time Streaming Display**: Implements real-time streaming output using native JavaScript.
- **Markdown Rendering**: Uses marked.js to parse and render AI-generated Markdown content.
- **Code Highlighting**: Uses highlight.js for syntax highlighting of code snippets.

2. Backend (FastAPI)

- **API Framework**: Built with Python FastAPI for high-performance, asynchronous REST API services.
- **Routing Management**: Provides dedicated API endpoints for different functions.
- **Streaming Response**: Supports asynchronous streaming output.
- **Error Handling**: Comprehensive error capture and exception handling.
- **Content Validation**: Automatic request data validation and type checking.

3. AI Engine (DeepSeek)

- **Large Language Model**: Powered by DeepSeek's advanced language model capabilities.
- **Context Understanding**: Supports context management for multi-turn conversations.
- **Multilingual Processing**: Supports translation and content generation in multiple languages.
- **Domain-specific Optimization**: Specially optimized for email scenarios.

System Modules

1. Core Files

- **app.py**: Main FastAPI application file, handles HTTP requests and routing.
- **deepseek_logic.py**: Encapsulates DeepSeek API logic and manages AI model interactions.
- **static/index.html**: Frontend interface for user interaction and result display.
- **requirements.txt**: Project dependency management.
- **start.py**: System startup script.

2. Functional Modules

- **Dialogue Module**: Implements basic multi-turn conversation functionality.
- **Analysis Module**: Provides email content analysis and suggestions.
- **Rewrite Module**: Offers email content rewriting and optimization.
- **Translation Module**: Supports multilingual email translation.
- **Template Module**: Manages and fills preset email templates.
- **Follow-up Module**: Generates replies or follow-up emails based on existing emails.

Technical Implementation Details

Frontend Implementation

- **Function Switching**: Switches between different modules via the sidebar.
- **Streaming Output**: Uses fetch API and ReadableStream to handle streaming responses.
- **Data Handling**: Exchanges data between frontend and backend in JSON format.
- **Dynamic Forms**: Dynamically generates form fields based on the selected template type.
- **Responsive Design**: Adapts to different device screen sizes.

- **User Experience Optimization**: Adds message action buttons, loading indicators, error handling, etc.

Backend Implementation

- **Asynchronous API**: Utilizes FastAPI's async features to handle concurrent requests.
- **Streaming Response Handling**: Uses StreamingResponse for real-time responses.
- **Request Validation**: Uses Pydantic models for automatic request data validation.
- **Context Management**: Maintains conversation history for context-aware replies.
- **Error Handling**: Provides appropriate responses and status codes for different error types.

DeepSeek AI Integration

- **Prompt Engineering**: Designs specialized prompts for different scenarios.
- **Parameter Tuning**: Optimizes model parameters for best results.
- **Streaming Output**: Supports DeepSeek API's streaming response mode.
- **Error Retry**: Handles network exceptions and API call failures.

Usage

1. Install dependencies:

```
pip install -r requirements.txt
```

2. Start the backend service:

```
uvicorn app:app --reload
```

Or use the provided startup script:

```
python start.py
```

3. Open the frontend page in your browser:

```
http://localhost:8000/static/index.html
```

4. Enter your email content or instructions in the chat box, and the system will provide assistance based on your needs.

API Endpoints

POST /chat

- Function: Interact with the email assistant.
- Request parameters:
  - `message`: User input message.
  - `history`: Conversation history (optional).
  - `stream`: Whether to use streaming response (optional, default is false).

- Return values:
  - `code`: 200 for success, 400 for failure.
  - `data`: Assistant's reply content.
  - `msg`: Response message.

POST /chat/stream

- Function: Interact with the email assistant in streaming mode.
- Request parameters: Same as /chat.
- Return values: Streaming text response, each line is a JSON object.

Other Functional APIs

The system also provides APIs for specific functions:

- `/analyze`: Email analysis
- `/rewrite`: Email rewriting
- `/translate`: Email translation
- `/template`: Email template generation
- `/follow_up`: Follow-up email generation

Each API has a corresponding streaming version, e.g., `/analyze/stream`.

Project Dependencies

- Python 3.7+
- FastAPI
- Uvicorn
- DeepSeek API
- Frontend:
  - Tailwind CSS
  - marked.js (Markdown parsing)
  - highlight.js (code highlighting)
  - Font Awesome (icons)

If you need further customization or have any questions, feel free to contact the maintainer.