**AI Youtube Trend Finder Based On Niche**

Overview

* **Voice Input & Transcription**: The system receives a voice message via a webhook. The audio is transcribed into text using OpenAI's Speech-to-Text functionality.
* **Context Management**: A memory manager (buffer) stores the conversation history (last 20 messages) so that the AI agent can use this context in subsequent interactions.
* **AI Agent**: The AI agent uses the transcribed text and context to generate responses using a language model, in this case, Google Gemini Chat Model.
* **Audio Response**: The generated text response is then converted into speech using ElevenLabs’ Text-to-Speech API.
* **Response Delivery**: The audio response is sent back via a webhook response, so that the voice chat client (e.g., a mobile app or web interface) can play the generated audio.

Workflow Breakdown

1. **Transcription and Memory Management**
   * **Webhook Listener & Transcription**: When a voice message is received, it is transcribed to text via the "OpenAI - Speech to Text" node.
   * **Context Storage**: The "Window Buffer Memory" and "Get Chat" nodes capture and store the conversation context in a buffer memory. This context is used to maintain continuity across the conversation.
2. **AI Processing**
   * **AI Agent**: The "AI Agent" node takes the transcribed text (and possibly other conversation context) and generates a text response. This response is determined by the system prompt provided to the agent.
   * **Language Model**: The agent leverages Google Gemini Chat Model, as configured in the "Google Gemini Chat Model" node, to generate detailed and context-aware responses.
3. **Voice Generation**
   * **Text-to-Speech with ElevenLabs**: The text response is then passed to the "ElevenLabs - Generate Audio" node, which converts it into an audio file using the ElevenLabs API. This node requires an API key and a specific Voice ID, which should be configured as per your ElevenLabs account.
   * **Audio Output**: The resulting audio is then sent back through the "Respond to Webhook" node as a binary response.
4. **User Interaction**
   * **Continuous Chat Flow**: The user interacts with the system by sending voice messages and listening to the generated audio responses, making the conversation feel natural and fluid.
   * **Context Retrieval & Update**: With each new interaction, the conversation history is updated to ensure that the AI agent always has context for its responses.

Key Advantages

* **Seamless Voice Interaction**: Converts voice inputs into context-aware, intelligent responses, and then back into voice for a full-duplex chat experience.
* **Context Preservation**: By using memory management nodes, the workflow ensures that the conversation context is maintained over multiple exchanges.
* **Flexible Integration**: Combines multiple technologies (OpenAI, Google Gemini, ElevenLabs) for a robust voice chat solution that can be customized for various use cases.
* **Real-time Response**: The integration of Webhook and voice synthesis ensures that responses are delivered in real time to end users.

How to Use

1. **Setup Credentials**:
   * Configure your OpenAI API credentials and ensure your Speech-to-Text and language model nodes are properly authenticated.
   * Set up your ElevenLabs account and obtain the necessary API key and Voice ID for the Text-to-Speech conversion.
   * Ensure the Google Gemini (PaLM) API is properly connected with valid credentials.
2. **Deploy the Workflow**:
   * Activate the workflow in n8n.
   * Set up your voice chat client (web or mobile) to send voice messages to the designated webhook URL.
3. **Testing & Iteration**:
   * Use the "Test workflow" functionality in n8n to simulate a voice message input.
   * Monitor the transcription, AI response, and audio output to ensure everything is working as expected.
   * Adjust parameters (e.g., context window length, model temperature, etc.) as needed to optimize the conversational experience.