CRISP-DM adaptation for text and voice-based chatbot project

19 July 2024 09:28

1. Business Understanding

- **Objectives:**
  - Define clear chatbot goals (e.g., customer support, sales assistance, information retrieval).
  - Determine the target audience and user personas.
  - Identify key performance indicators (KPIs): User satisfaction, task completion rate, average handling time, customer retention, cost savings.
- **Success Criteria:**
  - Establish quantifiable targets for each KPI (e.g., 85% user satisfaction, 95% task completion, 15% reduction in support costs).
- **Constraints:**
  - Define budget, timeline, and available resources.
  - Consider ethical and legal constraints related to data privacy and voice recordings (GDPR compliance).
- **Deliverables:**
  - Business objectives document.
  - User persona profiles.
  - KPI targets.

2. Data Understanding

- **Data Collection:**
  - Include relevant **knowledge sources** for RAG: Internal knowledge bases, FAQs, product documentation, etc. These will be used to retrieve contextual information.
  - Gather **conversation history data** for LLM fine-tuning: This can help the LLM learn the nuances of your chatbot's specific interactions.
- **Data Quality Assessment:**
  - Assess the **quality and relevance** of knowledge sources. Make sure the information is accurate and up-to-date.
  - Consider the **structure of your knowledge sources**. Will they be easily searchable by the RAG system?
- **Data Exploration:**
  - Identify potential **retrieval challenges** (e.g., ambiguous queries, complex information needs).
  - Explore the distribution of **topics and entities** in your knowledge sources to understand potential areas of focus.
- **Deliverables:**
  - Data collection and quality assessment report.
  - Initial data exploration findings.

3. Data Preparation

- **Text Data:**
  - **Knowledge Source Preparation:**
    - **Chunking:** Divide knowledge documents into smaller, manageable chunks for efficient retrieval.
    - **Indexing:** Create an index of your knowledge chunks for fast search (e.g., using vector databases like Pinecone or Faiss).
    - **Metadata Enhancement:** Add metadata to knowledge chunks (e.g., topic, keywords) to improve retrieval relevance.
  - **Conversation History Data:**
    - Preprocess the conversation data in a format suitable for fine-tuning the LLM (e.g., dialogue pairs, turn-based conversations).
- **Voice Data:**
  - Transcription: Use a high-quality transcription service (e.g., Google Cloud Speech-to-Text, Amazon Transcribe).
  - Data Augmentation: Apply techniques like adding noise, varying pitch, or time stretching to improve model robustness.
  - Feature Extraction: Generate mel-frequency cepstral coefficients (MFCCs) or other acoustic features suitable for ASR models.
- **Deliverables:**
  - Preprocessed and annotated datasets.
  - Feature extraction reports.

4. Modeling

- **Model Selection:**
  - **LLM:** Choose a powerful LLM like GPT-4, GPT-3.5, or other similar models.
  - **RAG Component:** Select or build a retrieval system that works well with your chosen LLM. There are libraries like LangChain that can help streamline this.
- **Training:**
  - **LLM Fine-tuning (Optional):** Fine-tune the LLM on your conversation history data to adapt it to your specific chatbot domain and style.
  - **Retrieval Model Training:** If you're building a custom retrieval system, you might need to train or fine-tune it based on your indexed knowledge sources.
- **Validation:**
  - Evaluate model performance using held-out validation sets.
  - Test the chatbot with simulated conversations and collect feedback from a small group of real users.
- **Deliverables:**
  - Trained and validated models.
  - Performance evaluation reports.

5. Evaluation

- **Quantitative Evaluation:**

    - Measure **retrieval performance:** Precision, recall, and F1-score of the RAG component.
    - Track the **impact of the LLM** on overall chatbot performance.
- **Qualitative Evaluation:**
    - Pay close attention to how well the chatbot handles **complex queries** that require information retrieval.
    - Assess the **coherence** and **informativeness** of responses generated with RAG augmentation.
- **Deliverables:**
    - Evaluation metrics and performance reports.
    - User feedback summaries.

6. Deployment

- **Integration:**
    - Ensure the LLM and RAG system are **seamlessly integrated** into your chatbot infrastructure.
    - Choose a deployment platform that supports the computational requirements of LLMs (e.g., cloud-based solutions).
- **Feedback Loop:**
    - Collect feedback on both **retrieval accuracy** (did the chatbot find the right information?) and **response quality** (was the answer helpful?).
    - Update your knowledge base regularly to keep the information fresh.
- **Hallucination Mitigation:**
    - Implement strategies to detect and mitigate instances where the LLM might generate incorrect or nonsensical information.
- **Ethical Considerations:**
    - Be mindful of potential biases in your training data and knowledge sources.
- **Deliverables:**
    - Deployment and integration plan.
    - Monitoring and feedback loop documentation.

Additional Considerations

Personalization:

- Collect user preferences and historical data to tailor responses.
- Use personalization algorithms to enhance user experience.

Sentiment Analysis:

- Integrate sentiment analysis models to understand user emotions and adjust responses accordingly.
- Tools like VADER or BERT-based sentiment models can be employed.

Omnichannel Experience:

- Ensure a consistent and seamless experience across all deployment channels.
- Use frameworks supporting multiple platforms to maintain consistency.

Error Handling & Fallbacks:

- Develop strategies for handling out-of-scope questions and escalating to human agents when necessary.
- Implement fallback mechanisms to maintain a smooth user experience.

Deliverables:

- Personalization strategy document.
- Sentiment analysis integration plan.
- Omnichannel deployment guidelines.
- Error handling and fallback strategies.