ProposaGen - Gen AI For Proposals : Stage 2

Easwari Engineering College

Saranath P
310621243048

Janani D
310621243022

Hitesh S
310621243020

Meet Our Team - ProposaGen

- **Saranath P**
  - Role: *Project Lead*
  - Expertise: AI Integration, Proposal Management
- **Janani D**
  - Role: *Developer*
  - Expertise: Backend Development, Data Processing
- **Hitesh S**
  - Role: *Data Engineer*
  - Expertise: Data Preparation, Data Visualization

Important Links - ProposaGen

- Slides link - Slides
- Code Base - GitHub : Fun-with-LLMs : ProofOfConcept
- Video Link : Click for Video

Problem Statement

Significant Market Opportunity

- **Expanding AI Market**
  - Global AI market projected to reach **$390.9 billion by 2025**.
- **Growth in Proposal Software**
  - Proposal management software market expected to grow at a **CAGR of 14%** from 2021 to 2026.
- **Rising Demand for Automation in India**
  - The size of the Indian industrial automation market is valued at 15 billion dollars in 2024. The market is expected to grow to around 29 billion dollars by 2029.
- **Investment in AI Technologies**
  - Organizations investing heavily in AI to gain competitive advantage.

Problem Statement - (contd)

Challenges in Proposal Development

- **Time-Consuming Process**
  - Manual creation of proposals is labor-intensive.
- **Lack of Personalization**
  - Difficulty tailoring proposals to specific client needs.
- **Inconsistent Quality**
  - Variability in content and presentation across proposals.
- **Limited Data Utilization**
  - Challenges in analyzing vast technical documents and designs.
- **Inefficient Resource Allocation**
  - High manual effort leads to increased costs and missed deadlines.

Problem Statement - (contd)

Addressing Real-World Challenges

- **Inefficiency in Traditional Methods**
  - Manual proposal creation is time-consuming and prone to errors.
- **Competitive Business Environment**
  - Companies need quick turnaround times to stay ahead.
- **Customization Demand**
  - Clients expect proposals tailored to their specific needs.
- **Cost Reduction Pressure**
  - Automating proposals reduces operational costs.

Demography

Objective and Approach

Overview

- **Goal**: Develop an AI-driven system to automate and personalize client proposals.
- **Approach**: Utilize advanced AI technologies and frameworks to generate customized proposals efficiently.

Objective and Approach - (contd)

Key Objectives

- **Data Collection**
  - Gather extensive data from technical documents and engineering designs relevant to client needs.
- **Data Processing**
  - Analyze and understand complex requirements using advanced algorithms and machine learning techniques.
- **Content Generation**
  - Employ Large Language Models (LLMs) to generate precise, customized proposal content aligned with client specifications.

Objective and Approach - (contd)

Integration with LangChain

- **Automation**
  - Implement LangChain to automate the end-to-end proposal generation workflow.
- **Scalability**
  - Leverage LangChain’s infrastructure to handle large data volumes and generate multiple proposals efficiently.
- **Flexibility**
  - Adapt to various data sources and client requirements using LangChain’s modular design.

Objective and Approach - (contd)

Personalization and Accuracy

- **Tailored Proposals**
  - Customize proposals to include specific technical details and design elements required by clients.
- **Precision**
  - Ensure high accuracy by continuously refining LLM models with client feedback and additional data.

Objective and Approach - (contd)

Efficiency and Competitive Edge

- **Reduced Manual Effort**
  - Automate repetitive tasks involved in proposal creation.
- **Time Savings**
  - Decrease the time needed to develop proposals, enabling quick responses to client needs.
- **Competitive Advantage**
  - Deliver high-quality, personalized proposals faster than competitors.

Objective and Approach - (contd)

Future Enhancements

- **Continuous Improvement**
  - Implement feedback loops to refine LLM models and the proposal generation process.
- **Expansion**
  - Explore expanding the solution to other domains beyond technical and engineering proposals.

Solution Overview

Three Approaches

1. **Using LangChain**
   - Utilize LangChain to automate the proposal generation task.
2. **Using DeepSpeed (Dspy)**
   - Fetch individual components for proposals using DeepSpeed for optimized processing.
3. **Fine-Tuning LLaMA Model on GPUs**
   - Fine-tune a LLaMA model using GPUs for enhanced performance and customization.

Solution Overview - (contd)

Three Approaches

1. **Using LangChain**
   - Utilize LangChain to automate the proposal generation task.
2. **Using DeepSpeed (Dspy)**
   - Fetch individual components for proposals using DeepSpeed for optimized processing.
3. **Fine-Tuning LLaMA Model on GPUs**
   - Fine-tune a LLaMA model using GPUs for enhanced performance and customization.

Solution Overview - (contd)

Approach 1: Using LangChain

- **Automation**
  - Streamline the workflow from data input to proposal output.
- **Modularity**
  - Easily integrate with various data sources and APIs.
- **Scalability**
  - Efficiently handle multiple proposals and large datasets.

Solution Overview - (contd)

Three Approaches

1. **Using LangChain**
   - Utilize LangChain to automate the proposal generation task.
2. **Using DeepSpeed (Dspy)**
   - Fetch individual components for proposals using DeepSpeed for optimized processing.
3. **Fine-Tuning LLaMA Model on GPUs**
   - Fine-tune a LLaMA model using GPUs for enhanced performance and customization.

Solution Overview - (contd)

Approach 2: Using DeepSpeed (Dspy)

- **Optimized Performance**
  - Utilize DeepSpeed for faster data processing and model training.
- **Component Retrieval**
  - Fetch and assemble individual proposal components efficiently.
- **Resource Efficiency**
  - Leverage DeepSpeed’s optimizations to reduce computational costs.

Solution Overview - (contd)

Three Approaches

1. **Using LangChain**
   - Utilize LangChain to automate the proposal generation task.
2. **Using DeepSpeed (Dspy)**
   - Fetch individual components for proposals using DeepSpeed for optimized processing.
3. **Fine-Tuning LLaMA Model on GPUs**
   - Fine-tune a LLaMA model using GPUs for enhanced performance and customization.

Solution Overview - (contd)

Approach 3: Fine-Tuning LLaMA Model on GPUs

- **Customization**
  - Fine-tune the LLaMA model to align closely with client-specific terminology and requirements.
- **Enhanced Performance**
  - Use GPUs to accelerate model training and inference.
- **Improved Accuracy**
  - Achieve higher precision in generated proposals through tailored fine-tuning.

Technical Implemetation

List of Technologies, Tools, and Frameworks Used

- **LangChain**: Used for chaining various LLM models and facilitating the integration with LLaMA 8B model.
- **LLaMA 8B/70B Model**: Pretrained model which has understanding of the language.
- **ChromaDB**: A vector store for document embeddings, used to store and retrieve contextual information during proposal generation.
- **DSPy**: Used for extracting detailed sections of the proposal such as executive summaries, client needs, and proposed solutions.
- **Python**: Core programming language for implementing the logic and handling backend processes.
- **PyTorch**: For training and fine-tuning the LLaMA models on GPUs.

Technical Implemetation - (contd)

Explanation of How These Technologies Were Applied

- **LangChain Integration**: Facilitated the connection between the LLaMA 8B model and the proposal generation pipeline, allowing for dynamic response generation based on previous client examples.
- **ChromaDB**: Utilized to store historical client documents and examples, which were later retrieved and used to enhance proposal content.
- **DSPy**: Extracted and structured various parts of the proposal, including client needs analysis, risk assessment, and executive summaries, ensuring consistency across genres.

Technical Implemetation - (contd)

Technical Difficulties Faced and Resolved

Challenge: Handling large datasets and documents with high variability in structure.

- **Solution**: Implemented `RecursiveCharacterTextSplitter` for efficient text processing, ensuring even large documents were split and processed correctly.

Challenge: Managing memory usage with large-scale embeddings.

- **Solution**: Used Chroma's efficient database and retrieval system to offload large embeddings, allowing for faster and scalable proposal generation.

Challenge: Training the LLaMA model for fine-tuning.

- Due to lack of resources, we did not yet fine-tune the LLaMA model, but we plan to do so before the stage 3 presentation.

Technical Implemetation - (contd)

Scalability and Flexibility of the Technical Architecture

- **Scalable Architecture**: By leveraging ChromaDB, the system is designed to scale as more client documents and proposals are added. The system can efficiently handle large datasets and provide quick retrieval of relevant information.
- **Flexibility**: The integration of DSPy allows for modular updates and customizations to the proposal sections, making the system adaptable to various client requirements across different domains.

Novelty

Prior Art Search on Published Literature

- Conducted a comprehensive prior art search across published literature in domains such as AI-powered proposal generation, LLMs for personalized content, and AI-driven contextual analysis.
- Key sources included academic databases, patents, and industry white papers focused on AI in document processing and proposal customization.
- No existing solution combines LangChain, DSPy, and LLaMA 8B with ChromaDB for real-time, personalized proposal generation with industry-specific contextualization.

Novelty - (contd)

Unique Differentiators of Our Solution

- **Dynamic Proposal Generation**: Leveraging LangChain and DSPy, our system offers real-time proposal creation that adapts to specific client needs, incorporating personalized industry examples that cannot be easily replicated by competitors.
- **Contextual Retrieval with ChromaDB**: Unlike traditional systems, our solution uses ChromaDB to store and retrieve relevant client documents as context, making it nearly impossible for competitors to replicate the same level of customization and contextual understanding.
- **Integration of DSPy Modules**: DSPy modules allow for structured, detailed, and domain-specific proposal generation that incorporates industry standards, providing a unique edge over competitors.

Novelty - (contd)

Competitor Benchmarking

- **Feature 1**: Real-time, personalized proposal generation based on specific client needs.
  - **Competitor**: Limited personalization, often requiring manual inputs.
  - **Our Solution**: Fully automated and scalable proposal generation tailored to industry and client.
- **Feature 2**: Contextual document retrieval and dynamic updates.
  - **Competitor**: Basic static template-based proposals.
  - **Our Solution**: Uses ChromaDB for retrieving relevant documents, ensuring proposals are always up to date.
- **Feature 3**: Modular architecture using DSPy.
  - **Competitor**: Non-modular, hard-coded proposal systems.

- **Our Solution**: Flexible DSPy modules allow for easy customization and addition of new features, maintaining competitive advantage.

Novelty - (contd)

Novelty Claim Support

- **Prior Art Search Findings**: No existing solution offers the same combination of LangChain, DSPy, LLaMA 8B, and ChromaDB for personalized proposal generation.
- **Competitor Benchmarking**: Our solution significantly outperforms existing competitors in flexibility, scalability, and context-based generation.
- **Expert Validation**: Novelty claim backed by published literature, competitor analysis, and unique technical architecture that cannot be easily replicated.

Challenges Faced

Identification of Specific Challenges

- **AI Hallucination**: At times, the LLaMA 8B model generated outputs that were irrelevant or inaccurate, affecting the quality of the proposals.
- **GPU Training Constraints**: Finetuning the model using Low-Rank Adaptation (LoRA) was challenging due to GPU limitations and availability.
- **Dataset Collection**: Gathering relevant and high-quality datasets for fine-tuning and training the model was time-consuming and required extensive validation.

Challenges Faced - (contd)

Strategies and Solutions Used

- **Mitigating AI Hallucination**: Implemented context management strategies using ChromaDB to ensure the model has access to relevant and accurate information when generating proposals.
- **Overcoming GPU Constraints**: We have requested our college GPUs using which we can train the model and can use DSPy in that model instead of a pretrained, original llama model.
- **Improving Dataset Collection**: We expanded our data collection efforts to include publicly available datasets and collaborated with domain experts to ensure the relevance and quality of the data used for model finetuning.

Challenges Faced - (contd)

Lessons Learned

- **AI Hallucination**: Ensuring proper context retrieval is critical for maintaining the accuracy and relevance of AI-generated content.
- **GPU Training**: LoRA proved to be an effective strategy for overcoming GPU limitations and achieving model finetuning with fewer resources.
- **Dataset Collection**: Collecting and curating high-quality datasets is essential for successful model training and significantly impacts the performance and reliability of the AI system.

Results and Achievements

- **LangChain Integration**
  - Generated complete proposals instantly upon inputting requirements.
  - **Time Efficiency**: Reduced proposal generation time to **10 seconds** per proposal.
- **Dspy (DeepSpeed) Implementation**
  - Extracted specific sections for detailed proposals.
  - **Detail Enhancement**: Provided in-depth content, improving proposal quality.
  - **Processing Time**: Took **2-3 minutes** per proposal.

Demonstration Video

Video Link

Project Plan for Completion of Prototype and Future Enhancements

- **Prototype Completion Timeline**
  - Build a working prototype by **mid-December**.
  - **Milestones**:
    - **August**: Finalize project requirements and design specifications.
    - **September**: Develop and integrate core functionalities - LangChain and DSpy for POC.
    - **October**: Integrating GPUs and training a full fldged model for client proposals.
    - **November**: Testing and Debugging.
    - **Mid-December**: Prototype deployment and demonstration.

Project Plan for Completion of Prototype and Future Enhancements

- **Resources and Tools Required**
  - **Hardware**:
    - GPUs for model training and fine-tuning.
  - **Software**:
    - LangChain, DeepSpeed (Dspy), LLaMA models.
  - **Procurement Plan**:
    - Secure necessary hardware resources by **end of October**.
    - Obtain software licenses and set up development environment.