# Project Proposal Document

## Table of Contents

## Project Title:

Automating Comment Handling in Collaborative Documents using Hierarchical Capsule Networks and Advanced AI Techniques

## Team Members:

- Sravani Pati
- Vamsi Krishna Pasam

## Abstract:

This project aims to develop a system that automates the handling of comments in collaborative documents. The system will prioritize, categorize, and triage comments based on relevance, complexity, urgency, and importance using advanced machine learning models, including Hierarchical Capsule Networks and Large Language Models (LLMs). The project will also develop a user interface to facilitate interactions, including assigning priorities, creating incident tickets, and tracking comment resolutions. The final output will be a comprehensive tool ready for real-world application and publication at the ACM CHI conference.

## Introduction:

Comments in collaborative documents play a critical role in facilitating communication and collaboration. Efficiently managing these comments can significantly enhance productivity and document quality. Our project uses advanced AI techniques to automate the prioritization and categorization of comments, making the document editing process more efficient and streamlined.

## Objectives:

1. Develop a hierarchical capsule network for multi-label classification of comments.
2. Explore zero-shot and n-shot learning capabilities of LLMs and Vision-Language Models (VLMs) for hierarchical classification.
3. Using a multi-label classification system, develop an approach to prioritize and categorize comments based on relevance, complexity, urgency, or importance.
4. Create a user-friendly interface for document upload, comment analysis, priority assignment, incident management, and dashboard insights.
5. Prepare the project for submission to the ACM CHI conference and ensure readiness for real-world applications.

## Methodology:

The project will be implemented in several phases, each focusing on different aspects of development and integration.

### Phase 1: Data Preparation and Initial Model Development

1. Data Cleaning and Preprocessing:
    - Standardize data formats and handle missing values.
    - Split the comment date and time, and ensure no empty comments.
    - Visualize and analyze the data to identify critical predictors and trends.
2. Initial Model Development:
    - Implement and evaluate DistilBERT for initial intent classification.
    - Analyze performance metrics and identify areas for improvement.

### Phase 2: Hierarchical Capsule Networks and Advanced Models

1. Hierarchical Capsule Network Development:
    - Implement a hierarchical capsule network for multi-label classification.
    - Train and evaluate the model on the cleaned dataset.
2. Zero-shot and N-shot Learning with LLMs and VLMs:
    - Investigate the capabilities of models like GPT-4 and CLIP for hierarchical classification.
    - Implement both sequential and in-build classification approaches.
    - Fine-tune models and explore in-context learning and retrieval-augmented generation (RAG).

### Phase 3: Triage Comments

1. Develop Prioritization and Categorization Approach:
    - Develop an approach to prioritize and categorize comments based on relevance, complexity, urgency, or importance.
    - Use a multi-label classification system to assign multiple tags to each comment, indicating its priority and category.
2. Integration with ML Models:
    - Integrate the triage system with the hierarchical capsule network and other models.
    - Ensure the system accurately reflects the prioritization and categorization in the user interface.

### Phase 4: User Interface Development

1. Front-end Development:

- Create a document upload interface using React.js.
- Develop components for comment analysis, priority assignment, and incident management.
- Implement a dashboard for visualizing insights and analytics.

2. Back-end Development:
   - Set up a Node.js server with Express.js.
   - Develop APIs for document processing, comment analysis, priority management, and incident tracking.
   - Integrate the back-end with the Python-based ML models using Flask or FastAPI.

**Phase 5: Integration and Testing**

1. Integration:
   - Integrate the front-end with the back-end APIs.
   - Ensure seamless communication between the user interface and ML models.
2. Testing:
   - Conduct unit and integration testing to ensure functionality.
   - Perform user testing to gather feedback and make necessary adjustments.

**Phase 6: Deployment and Finalization**

1. Deployment:
   - Containerize the application using Docker.
   - Deploy the application on a cloud platform (AWS et al.).
2. Finalization:
   - Prepare documentation and finalize the project for submission to the ACM CHI conference.
   - Ensure the system is ready for real-world application with user-friendly documentation.

## Project Delivery Timeline

| Phase | Task | Start Date | End Date | Deliverable |
|---|---|---|---|---|
| Phase 1: Data Preparation | Data Cleaning and Preprocessing | June 1, 2024 | June 15, 2024 | Cleaned Dataset |

|  |  |  |  |  |
|---|---|---|---|---|
|  | Initial Model Development (DistilBERT) | June 16, 2024 | June 30, 2024 | Initial Model and Evaluation Report |
| Phase 2: Advanced Model Development | Hierarchical Capsule Network Implementation | July 1, 2024 | July 15, 2024 | Trained Capsule Network Model |
|  | Zero-shot and N-shot Learning Implementation | July 16, 2024 | July 31, 2024 | Fine-tuned LLMs and VLMs |
| Phase 3: Triage Comments | Develop Prioritization and Categorization Approach | August 1, 2024 | August 10, 2024 | Triage System and Classification Report |
|  | Integration with ML Models | August 11, 2024 | August 15, 2024 | Integrated Triage System |
| Phase 4: UI Development | Front-end Development (React.js Components) | August 16, 2024 | August 25, 2024 | Document Upload and Analysis Interface |
|  | Back-end Development (Node.js APIs) | August 26, 2024 | September 5, 2024 | Fully Functional Back-end APIs |

| | | | | |
|---|---|---|---|---|
| | Integration of Front-end and Back-end | September 6, 2024 | September 10, 2024 | Integrated Application |
| Phase 5: Testing and Deployment | Integration Testing and User Feedback | September 11, 2024 | September 20, 2024 | Tested and Refined Application |
| | Deployment to Cloud | September 21, 2024 | September 25, 2024 | Deployed Application on Cloud |
| | Final Documentation and Conference Paper Preparation | September 26, 2024 | September 30, 2024 | Submission-Ready Project Documentation |
| Phase 6: Finalization | Final Adjustments and Submission | October 1, 2024 | October 15, 2024 | Submission to ACM CHI and Ready-for-Use Application |

## Conclusion:

Following this detailed project plan, we aim to develop a robust and effective system for automating comment handling in collaborative documents. The project will be prepared for publication at the ACM CHI conference and will be suitable for real-world applications. This proposal outlines the steps, methodologies, and timeline to ensure the project's success and impact.

## Appendices:

1. Data Schema: Detailed description of the dataset columns and their meanings.
2. Model Architecture: Diagrams and explanations of the hierarchical capsule network and other models used.
3. API Documentation: Endpoints and usage guidelines for the developed APIs.
4. User Manual: Instructions for using the user interface and system features.

This comprehensive project proposal document serves as a blueprint for the successful execution and delivery of the project by the end of September 2024, aligning with the goals for publication and real-world application.