WA3540-CAP Lab Guide

Deep Learning Gen AI Capstone

ascendient learning

Part of Accenture

Revision 1.0.0 published on 2025-08-05.

To obtain authorization for any such activities (e.g., reprint rights, translation rights), to customize this book, or for other sales inquiries, please contact:

Ascendient, LLC
1 California Street Suite 2900
San Francisco, CA 94111
https://www.ascendientlearning.com/

USA: 1-877-517-6540, email: getinfousa@ascendientlearning.com
Canada: 1-877-812-8887 toll free, email: getinfo@ascendientlearning.com

Table of Contents

BenefitsSync AI for TechLance HR

MODULE 1

1.1. Introduction

This capstone challenges you to create **BenefitsSync AI**, an integrated AI system for TechLance's HR department. The system processes employee benefits feedback, indexes complex policy documents, and delivers clear, personalized policy explanations via a chatbot interface. It leverages a unified Generative AI framework to categorize feedback, simplify policy information, answer employee queries, and ensure responsible AI practices using Giskard.

1.2. Scenario

TechLance is a mid-sized technology company with 5,000 employees, specializing in cloud-based software solutions. The HR department manages a comprehensive benefits package, including health insurance, retirement plans, tuition reimbursement, wellness programs, and paid time off. However, the department faces challenges:

- **Feedback Overload**: Employees submit extensive feedback (e.g., via `feedback_data.csv`), but manual analysis is slow, delaying responses to issues like complex reimbursement processes.
- **Policy Complexity**: Jargon-heavy benefits policies confuse employees, especially new hires or part-time staff, leading to misunderstandings about eligibility or processes.
- **Low Benefits Utilization**: Data shows underutilization of benefits (e.g., BenefitID 30 with "Not worth the hassle" comments), reducing ROI.
- **Fairness and Compliance**: With a diverse workforce, TechLance must ensure equitable access to benefits information and comply with GDPR and HIPAA.

The goal is to build **BenefitsSync AI** to automate feedback analysis, simplify policy communication, optimize benefits utilization, and maintain fairness and accuracy.

1.2.1. BenefitsSync AI System

A Python-based web application (e.g., using Streamlit or Gradio) will integrate all components into a single tool for HR professionals. The system will process feedback, index policies, and provide a chatbot for employee queries. It will run from a single command line and include real-time feedback processing and policy retrieval.

1.2.2. Feedback Data

Feedback data is provided in `feedback_data.csv`, under `assets/data/` containing employee comments on benefits. The file includes columns like `EmployeeID`, `BenefitID`, `Comment`, and `Timestamp`. This data will be used to test feedback categorization and action routing.

1.2.3. Policy Documents

Benefits policy documents are provided as PDFs in the `assets/benefits/` directory. These contain detailed information on benefits like health insurance, tuition reimbursement, and maternity leave. The documents will be indexed for Retrieval-Augmented Generation (RAG).

1.3. Capstone Instructions

This capstone is open-ended, allowing you to design and implement **BenefitsSync AI** to meet TechLance’s needs. The following sections outline the key phases, requirements, and integration steps.

1.3.1. Phase 1: Feedback Categorization and Action Routing

Develop a prompt chain pipeline to process employee feedback, categorize issues, assess sentiment, prioritize urgency, and route tasks to HR or benefits providers.

1. **Feedback Classification**: Use a representational LLM to classify feedback into categories (e.g., Process Issues, Coverage Issues, Benefit Value).

2. **Sentiment Analysis**: Analyze comment sentiment (Positive, Negative, Neutral) and assign a severity score (1–5, where 5 is highly negative).

3. **Action Identification**: Identify actionable tasks (e.g., simplify reimbursement process) based on category and sentiment.

4. **Task Routing**: Generate routing instructions to assign tasks to departments (e.g., Benefits Administration) with issue summaries and priority levels.

5. **Synthesize Results**: Summarize findings, emphasizing subcategory analysis (e.g., recommend cutting “Gym Membership: Tier 3 Partners”) and prioritizing by impact and feasibility.

1.3.2. Phase 2: Policy Document Indexing for RAG

Index benefits policy PDFs for Retrieval-Augmented Generation (RAG) to enable efficient policy retrieval.

1. **Text Extraction**: Use PyPDF2 to extract text from policy PDFs, removing noise (e.g., page numbers).

2. **Text Preprocessing**: Segment text into chunks (e.g., 200 words per chunk).

3. **Vector Store Creation**: Build a Chroma vector store using embeddings from `sentence-transformers` or OpenAI.

4. **Retrieval Testing**: Test retrieval with 5 HR policy queries (e.g., “What is the eligibility for Tuition Reimbursement?”) and evaluate chunk relevance.

5. **Advanced RAG Methods**: Research and test 3 advanced RAG techniques (e.g., metadata filtering, query expansion, HyDE) and document findings.

1.3.3. Phase 3: Policy Explainer System

Develop a chatbot-based Policy Explainer System to transform complex policy documents into clear, employee-friendly explanations.

1. **Policy Retrieval**: Retrieve relevant policy sections using RAG based on employee queries (e.g., “What’s the maternity leave policy?”).

2. **Jargon Simplification**: Rewrite complex policy text into plain language using a Generative AI prompt.

3. **Personalized Explanation**: Tailor explanations to employee context (e.g., family status, tenure) using HR data.

4. **Follow-Up Handling**: Handle follow-up questions, ensuring consistency with original policy.

1.3.4. Phase 4: Testing and Evaluation

Evaluate the system using DeepEval/Giskard to ensure robustness, fairness, and ethical compliance.

1. **Robustness Testing**: Test resilience to query variations (e.g., “maternity leave” vs. “pregnancy leave”) using Giskard.

2. **Fairness Evaluation**: Analyze explanations for bias across demographics (e.g., gender, age).

3. **Ethical Risk Assessment**: Use Giskard’s hallucination and toxicity detection to identify inaccuracies or inappropriate language.

4. **Performance Metrics**: Measure accuracy (policy alignment), coherence, and user satisfaction via simulated interactions.

5. **Prompt Refactoring**: Refine prompts based on Giskard outputs to improve performance.

6. **Optimization**: Implement performance enhancements (e.g., caching FAQs) to speed up processing.

1.3.5. Phase 5: Integration - BenefitsSync AI Application

Integrate all components into a Python-based web application (e.g., Streamlit or Gradio) that runs from a single command line.

1. **Application Setup**: Build a web app with interfaces for feedback upload, chatbot queries, and policy retrieval.
2. **Feedback Processing**: Integrate the feedback categorization pipeline to process `feedback_data.csv` and display results.
3. **Policy Chatbot**: Embed the Policy Explainer System as a chatbot interface for employee queries.
4. **Performance Optimization**: Use caching (e.g., for FAQs) to improve response times.
5. **Command-Line Execution**: Ensure the app runs with a single command (e.g., `streamlit run app.py`).

> 💡 Tip
>
> Ensure all components (feedback pipeline, RAG, chatbot) are integrated seamlessly for HR usability.

1.4. Conclusion

In this capstone, you will design and implement **BenefitsSync AI** to streamline TechLance’s HR operations. By automating feedback analysis, indexing policies, and providing clear policy explanations, the system will improve employee satisfaction, optimize benefits utilization, and ensure fairness and compliance.