High Level Design (HLD)

Document Tagging Project

Author Name: Dibyendu Biswas.

Revision Number: 1.0
Last Date of revision: 25-September-2023

Document Version Control

Date issued	Version	Description	Author
25-09-2023	1.0	Initial HLD – v1.0	Dibyendu Biswas

Contents

Abstract

The Document Tagging Project is an innovative and scalable solution designed to address the challenge of organizing and categorizing vast amounts of unstructured textual data. In today's data-driven world, organizations accumulate massive repositories of documents, articles, reports, and other textual content. Efficiently tagging and classifying these documents based on their content is essential for improving searchability, knowledge management, and information retrieval.

The impact of the Document Tagging Project extends across various industries and applications. It empowers content creators, knowledge managers, and data analysts to streamline document organization and retrieval processes, thereby saving time and improving decision-making. Additionally, the project's scalability ensures adaptability to diverse data sources and domains, making it a valuable asset for organizations seeking to unlock the full potential of their textual data.

1 Introduction

1.1 Why this High-Level Design Document

The purpose of this High-Level Design (HLD) document is to add the necessary details to the current project descriptions to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a references manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the software and hardware interfaces.
- Describe the performance requirements.
- Include design features and the architecture of the project.
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Applications Compatibility
    - Resource Utilization
    - Serviceability

1.2 Scope

The HLD document presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrations of the system.

1.3 Definitions

Term	Description
DB or db	Database, collection of all the information monitored by system.
IDE	Integrated Development Environment.
AWS	Amazon Web Service.
Tags	Define the tags of your document.

2 General Description

2.1 Product Perspective

The Document-Tagging solution system is an NLP based model which help us to find the relevant tags of our document/ documents.

2.2 Problem Statement

In today's data-driven world, organizations and individuals alike face the formidable challenge of managing vast repositories of unstructured textual data, including documents, articles, reports, and more. These collections of textual information often lack a consistent and organized tagging or categorization system, making it challenging to efficiently search, retrieve, and utilize the valuable knowledge contained within.

The problem at hand is the need for an automated and scalable solution to classify and tag documents effectively. The Document-Tagging project aims to address this problem by developing a robust system that can analyze textual content and assign relevant tags or categories to each document.

2.3 Proposed Solution

Document-Tagging system that leverages advanced natural language processing (NLP) techniques and transformer based pre-trained models for efficient document categorization. This system aims to automate the process of assigning relevant tags or categories to large volumes of unstructured textual data. Users can upload documents, which undergo preprocessing and are then classified into multiple relevant categories using NLP models.

2.4 Further Improvements

In this Document-Tagging system, users have the option to review and modify suggested tags. The tagged documents are stored for easy retrieval and management. The system offers significant benefits, including increased efficiency, improved knowledge retrieval, and scalability for growing document collections. Performance is measured using precision, recall, F1-score, accuracy, and scalability metrics.

2.5 Technical Requirements

Technical requirements of Document Tagging system will be:

- **Data Ingestion:** Develop mechanisms to ingest documents in various formats, such as text, PDF, and HTML. Ensure compatibility with a wide range of data sources, including local storage, cloud storage, and web scraping.

- **Data Preprocessing:** Implement text preprocessing techniques, including tokenization, stop-word removal, stemming, and entity recognition, to clean and prepare the text data for analysis.

- **Multi-label Classification Models:** Utilize state-of-the-art pre-trained transformer models like BERT, DistilBERT, or RoBERTa for multi-label document classification. Fine-tune these models on a labeled dataset for tag prediction.

- **Scalability:** Design the system to handle large volumes of documents efficiently. Optimize model inference and data processing pipelines for scalability using distributed computing and parallel processing.

- **User Interface:** Develop a user-friendly web-based or desktop application that allows users to upload, tag, and manage documents easily. Provide features for bulk tagging, document search, and tag management.

- **Tag Taxonomy:** Implement a hierarchical or flat taxonomy system for document categorization. Allow users to create, modify, and delete tags/categories as needed.

- **Tag Recommendation:** Incorporate tag recommendation algorithms to assist users in the tagging process. Suggest relevant tags based on document content and existing tags.

- **Version Control:** Implement version control for tagged documents to track changes and revisions over time. Ensure that previous versions of tagged documents can be retrieved.

- **Security:** Implement robust security measures to protect sensitive documents and user data. Ensure data encryption, user authentication, and access control mechanisms.

- **Performance Metrics:** Develop mechanisms to evaluate the performance of the tagging system. Metrics should include precision, recall, F1-score, accuracy, and scalability benchmarks.

- **API Integration:** Offer RESTful APIs to enable seamless integration with other applications and systems, facilitating data exchange and automation.

- **Monitoring and Logging:** Implement monitoring and logging tools to track system performance, user activities, and errors. Enable proactive issue resolution and system optimization.

- **Documentation:** Provide comprehensive technical documentation, including installation guides, API documentation, and user manuals.

- **Deployment Options:** Offer both on-premises and cloud-based deployment options to accommodate various user preferences and infrastructures.

- **Compliance:** Ensure compliance with relevant data protection regulations (e.g., GDPR) and industry-specific standards.

- **Support and Maintenance:** Establish a support and maintenance plan to address user inquiries, bug fixes, updates, and improvements.

- **Training:** Provide user training and on boarding materials to assist users in effectively utilizing the Document-Tagging system.

- **Backup and Recovery:** Implement regular data backups and disaster recovery procedures to prevent data loss and ensure business continuity.

- **Performance Optimization:** Continuously monitor and optimize the system's performance, including model retraining, query optimization, and infrastructure scaling.

- **Cost Management:** Implement cost-monitoring and resource-optimization strategies, especially in cloud-based deployments, to control operational expenses.

These technical requirements form the foundation of the Document-Tagging project, ensuring that the system efficiently handles document categorization needs while meeting security and scalability standards.

2.6 Data Requirements

In this Document-Tagging project, we need two features that is text data and tags or label data.

- **Document Texts:** The primary data consists of the documents you want to tag. These documents can be in various formats, such as plain text, PDFs, HTML, or other structured formats. The documents should cover the topics or subjects you intend to categorize or tag.

- **Tags or Labels:** You need a set of predefined tags or labels that represent the categories or topics you want to assign to the documents. These tags can be hierarchical or flat, depending on your taxonomy structure.

- **Labeled Data:** To train and fine-tune machine learning models for document tagging, you need a labeled dataset. This dataset should consist of documents paired with the corresponding tags or labels. Each document can have one or multiple associated tags, depending on the multi-label or multi-class nature of your task.

2.7 Tool Used

2.7.1 Hardware Requirements

- High-Performance CPU and GPU,
- Sufficient RAM (at least 16 GB),
- Storage Device, etc.

2.7.2 Software Requirements

- MS-Office,
- VS-Code,
- Anaconda,
- SQLite Database,
- Git and GitHub,
- Docker,
- DVC,
- Kubernetes,
- Heroku,
- AWS, etc.

2.7.3 Operating System

- Window 10 OS
- Ubuntu OS

2.8 Constraints

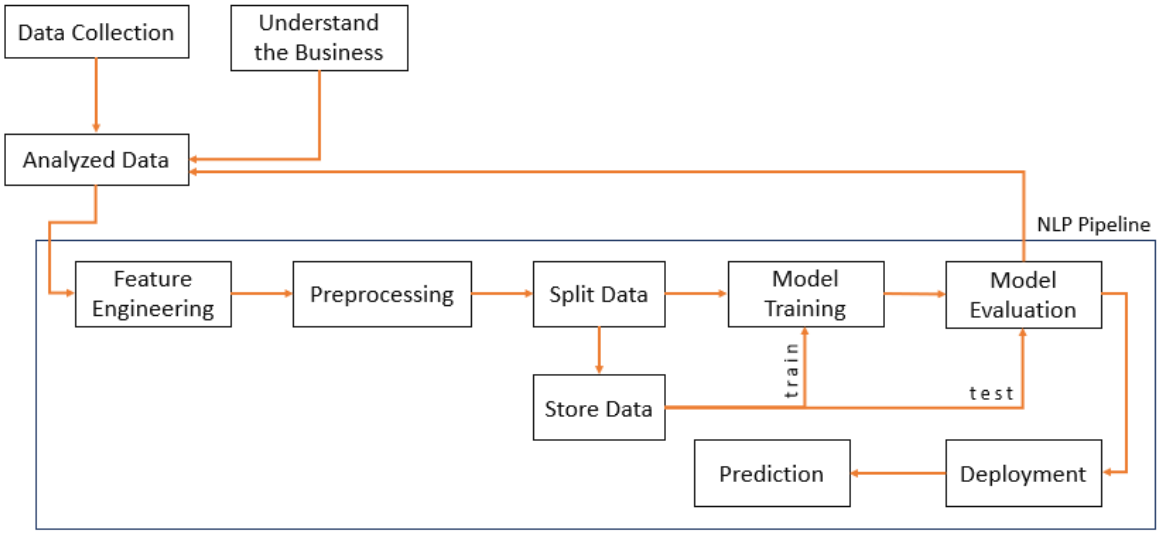
The Document-Tagging based system solution must be user friendly, as automated as possible and users should be required to know any of the working.

2.9 Assumptions

The main objective of the Document-Tagging project is to develop a robust and efficient system for automatically assigning relevant tags or labels to documents based on their content (2.2 Problem Statement). This project aims to streamline the process of categorizing and organizing a large corpus of documents, making it easier for users to search, retrieve, and manage information.

3 Design Details

3.1 Process Flow



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-by-Step Description:

- The system identifies at what step logging required.
- The system should be able to log each and every system flow.
- Developer can choose logging method (training_logs and prediction_logs).
    - training_logs is logging the training pipeline.
    - prediction_logs is logging the prediction pipeline.

- System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usages.

3.4 Performance

Document-Tagging project is to develop a robust and efficient system for automatically assigning relevant tags or labels to documents based on their content. It categorizing and organizing a large corpus of documents, making it easier for users to search, retrieve, and manage information. Also, model retraining is very important to improve the performance.

3.5 Reusability

The code written and the components used should have the ability to reused with no problems.

3.6 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

3.7 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.8 Deployment

Either I can deploy on AWS or Heroku or both.

4 Conclusion

Document Tagging project represents a significant step forward in automating and optimizing the categorization and organization of documents within organizations. By leveraging machine learning and natural language processing techniques, this project offers a powerful solution for improving document management workflows.

Through the development of a robust and scalable system, users can efficiently assign relevant tags to documents, enabling easier search, retrieval, and organization. The multi-label classification capability ensures that documents can be associated with multiple categories or topics, reflecting the complex nature of real-world document collections.

The project's user-friendly interface and customization options empower users to tailor the tagging system to their specific needs, whether it's in a corporate setting, academic institution, or any knowledge-intensive organization. Integrating the system with existing document management tools further enhances its utility and impact.

With a focus on accuracy, performance optimization, and ongoing maintenance, the Document Tagging project aims to provide a reliable and efficient solution that streamlines document management processes, saves time, and boosts productivity.

---------- **Thank You**