Low Level Design (LLD)

Document Tagging

Revision Number: 1.0

Last date of revision: 23/12/2023

Triveni Medici

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 23th Dec 2023 | 1.0 | First Draft | Triveni Medici |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_gjdgxs)

[**Abstract** 4](#_30j0zll)

[1](#_1fob9te) Introduction 5

[1.1](#_3znysh7) Why this Low-Level Design Document? 5

[1.2](#_2et92p0) Scope 6

[1.4](#_3dy6vkm) 1.3 Risks 6

[1.5](#_1t3h5sf) 1.4 Out of Scope 7

[2](#_4d34og8) Technical specifications 7

[2.1 Predicting Results 7](#_2s8eyo1)

[2.2 Logging 7](#_17dp8vu)

[2.3 Database 7](#_3rdcrjn)

[3](#_26in1rg) Technology stack 8

[4](#_lnxbz9) Proposed Solution 9

[5](#_35nkun2) Model training/validation workflow 10

[6](#_1ksv4uv) User I/O workflow 12

[7](#_44sinio) Exceptional scenarios 13

**Abstract**

Several thousand blog entries are actively shared on social media every hour; for example, blogging services like Tumblr1 had over 70 billion postings across various communities by January 2014. Keywords or "#tags" (hashtags) have been shown to serve as group identities and brand labels in addition to being topic markers. Authors on Tumblr can create their own tags or use pre-existing tags to label their blogs. It can be difficult to create or choose tags for optimum dissemination, and authors may not be able to attach all of the essential tags. In recent years, algorithm-driven document tagging has emerged as a possible answer to this challenge. Blog search, clustering related blogs, displaying subjects connected with trending tags, and personalization of blog postings are just a few of the downstream applications of automatically labelling these blogs. The customization algorithm could match user interests with tags linked with a blog post to improve user engagement.

The main objective here is -

1. Tagging news articles or blog posts with relevant tags from a collection of predefined ones is coined as document tagging.

2. To provide an accurate tagging of articles which can beneficial for several downstream applications such as recommendation and search.

3. To demonstrate the effectiveness of the approach, do experiments on several datasets and show promising results against state-of-the-art methods

# Introduction

## Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Document Tagging Model . It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which documents are clasified. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

Several thousand blog entries are actively shared on social media every hour; for example, blogging services like Tumblr1 had over 70 billion postings across various communities by January 2014. Keywords or "#tags" (hashtags) have been shown to serve as group identities and brand labels in addition to being topic markers. Authors on Tumblr can create their own tags or use pre-existing tags to label their blogs. It can be difficult to create or choose tags for optimum dissemination, and authors may not be able to attach all of the essential tags. In recent years, algorithm-driven document tagging has emerged as a possible answer to this challenge.

The main objective of the project is to classify the document or given input text for authors and also help authors extract the keywords from the text for tagging.

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

## Scope

This software system will be a Web application This system will be designed to classify the document or given input text for authors and also help authors extract the keywords from the text for tagging.

## Constraints

The model is built to support only few types of documents when user uploads and also model has the dataset taken for training as famous dataset from Kaggle 20 new groups dataset.

## Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

# Technical specifications

## 2.1 Dataset

Data were acquired through the publicly available lists of articles from 20 news groups

Data source: <http://qwone.com/~jason/20Newsgroups/>

Data format Raw: group of files with multiple folders as categories

**Value of the Data**

* This dataset is a collection newsgroup documents. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.
* There is file (list.csv) that contains a reference to the document\_id number and the newsgroup it is associated with.  
  There are also 20 files that contain all of the documents, one document per newsgroup.
* In this dataset, duplicate messages have been removed and the original messages only contain "From" and "Subject" headers (18828 messages total).
* This dataset can help researchers and practitioners easily build classification models in systems which helps the authors for tagging their content and also classify them
* Finally, the provided datasets could also be used To demonstrate the effectiveness of the approach, do experiments on several datasets and show promising results against state-of-the-art methods.

## 2.1.1 Data Description

The presented dataset was collected and prepared for the purpose of building and evaluating various classification methods for Tagging news articles or blog posts with relevant tags from a collection of predefined ones is coined as document tagging and to provide an accurate tagging of articles which can beneficial for several downstream applications such as recommendation and search.

There are also 20 files that contain all of the documents, one document per newsgroup.

In this dataset, duplicate messages have been removed and the original messages only contain "From" and "Subject" headers (18828 messages total).

Each new message in the bundled file begins with these four headers:

Newsgroup: alt.newsgroup

Document\_id: xxxxxx

From: Cat

Subject: Meow Meow Meow

The Newsgroup and Document\_id can be referenced against list.csv

Organization

* Each newsgroup file in the bundle represents a single newsgroup
* Each message in a file is the text of some newsgroup document that was posted to that newsgroup.

This is a list of the 20 newsgroups:

* comp.graphics
* comp.os.ms-windows.misc
* comp.sys.ibm.pc.hardware
* comp.sys.mac.hardware
* comp.windows.x rec.autos
* rec.motorcycles
* rec.sport.baseball
* rec.sport.hockey sci.crypt
* sci.electronics
* sci.med
* sci.space
* misc.forsale talk.politics.misc
* talk.politics.guns
* talk.politics.mideast talk.religion.misc
* alt.atheism
* soc.religion.christian

## 2.2 Document Tagging

The customization algorithm could match user interests with tags linked with a blog post to improve user engagement.

Tagging news articles or blog posts with relevant tags from a collection of predefined ones is coined as document tagging and to provide an accurate tagging of articles which can beneficial for several downstream applications such as recommendation and search.

## 2.3 Logging

We should be able to log every activity done by the user.

* The System identifies at what step logging required
* The System should be able to log each and every system flow.
* Developers can choose logging methods. You can choose database logging/ File logging as well.
* System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## 2.4 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the disease.

2. The User gives required information.

3. The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice (Cassandra).

**2.5 Deployment**

1. AWS



# Technology stack

|  |  |
| --- | --- |
| **Front End** | HTML/CSS/JS/React |
| **Backend** | Python Django |
| **Database** | Cassandra |
| **Deployment** | AWS |

# Proposed Solution

The classical machine learning tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that’s best fit for the above case.

For Feature Engineering show:-

1. Document Text cleaning

Baseline Model for Classification: LogisticRegression, DecissionTree, SVM, NaiveBayes

Actual Model for Classification: Logistic Regression

Baseline Model for Keywords Extraction: Bert, TF-ID , Yake

Actual Model for Keywords Extraction: Yake

# Model training/validation workflow



# User I/O workflow

Diagram

Description automatically generated

# Exceptional scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Exception | Mitigation | Module |
|  |  |  |  |
|  |  |  |  |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Steps to perform test case | Module | Pass/Fail |
|  |  |  |  |

# Key performance indicators (KPI)

* Tagging news articles or blog posts with relevant tags from a collection of predefined ones is coined as document tagging.
* To provide an accurate tagging of articles which can beneficial for several downstream applications such as recommendation and search.