High Level Design (HLD)

Document Tagging

Revision Number: 1.0

Last date of revision: 03/11/2022

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 20/11/2023 | 1 | Initial HLD – V1.0 | Triveni M |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_heading=h.gjdgxs)

[Abstract 4](#_heading=h.30j0zll)

1. [Introduction 5](#_heading=h.1fob9te)
   1. [Why this High-Level Design Document? 5](#_heading=h.3znysh7)
   2. [Scope 5](#_heading=h.2et92p0)
   3. [Definitions 5](#_heading=h.tyjcwt)
2. [General Description 6](#_heading=h.3dy6vkm)
   1. [Problem statement 6](#_heading=h.4d34og8)
   2. [Proposed Solution 6](#_heading=h.2s8eyo1)
   3. [Data Requirements 7](#_heading=h.26in1rg)
   4. [Tools used 8](#_heading=h.lnxbz9)
   5. [Constraints 9](#_heading=h.44sinio)
   6. [Assumptions 9](#_heading=h.2jxsxqh)
3. [Design Details 10](#_heading=h.z337ya)
   1. [Process Flow 10](#_heading=h.3j2qqm3)
      1. [Model Training and Evaluation 10](#_heading=h.1y810tw)
      2. [Deployment Process 11](#_heading=h.4i7ojhp)
   2. [Event log 11](#_heading=h.2xcytpi)
   3. [Error Handling 11](#_heading=h.1ci93xb)
   4. [Performance 12](#_heading=h.3whwml4)
   5. [Reusability 12](#_heading=h.2bn6wsx)
   6. [Application Compatibility 12](#_heading=h.qsh70q)
   7. [Resource Utilization 12](#_heading=h.3as4poj)
   8. [Deployment 12](#_heading=h.1pxezwc)
4. [Dashboards 13](#_heading=h.49x2ik5)
   1. [KPIs (Key Performance Indicators) 13](#_heading=h.2p2csry)
5. [Conclusion 14](#_heading=h.147n2zr)

# 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 High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description 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 reference 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 hardware and software 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
      * Application compatibility
      * Resource utilization
      * Serviceability

## Scope

The HLD documentation 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 administrators of the system.

## Definitions

|  |  |
| --- | --- |
| *Term* | *Description* |
| *Database* | Collection of all the information monitored by this system |
| *IDE* | Integrated Development Environment |
| *AWS* | Amazon Web Services |

# General Description

#### Problem statement

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.

PROPOSED SOLUTION

The main objective here is develop a model which -

1. Tags news articles or blog posts with relevant tags from a collection of predefined ones

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.

This needs a Deep learning NLP model with the tasks like Data Exploration, Data Cleaning, Feature Engineering, Model Building and Model Testing. Try out different Deep learning algorithms that’s best fit for the above case.

For Feature Engineering show:-

1. Document -Based Features

Baseline Model: Logistic Regression since this is a classification problem.

Actual model: XG Boast

## Data Requirements

Data requirements completely depend on our problem statement.

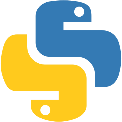
* + - We need the data to be in articles with document format.
    - The documents can be in text, pdf or any other format.
    - The document contains multiple articles and the data is classified based on different topics

## Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn , NLP Libraries are used to build the whole model.



A blue triangle with a cross

Description automatically generated

* + - PyCharm is used as IDE.
    - For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
    - AWS is used for deployment of the model.
    - Cassandra is used to retrieve, insert, delete, and update the database.
    - Front end development is done using HTML/CSS
    - Python Flask is used for backend development.
    - GitHub is used as version control system.

## Assumptions

## All the data collected is from the free source ‘http://qwone.com/~jason/20Newsgroups/’

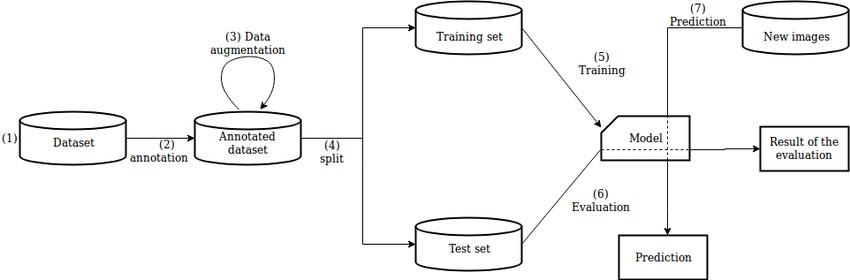
## It has all the data in the news articles as text format.

# Design Details

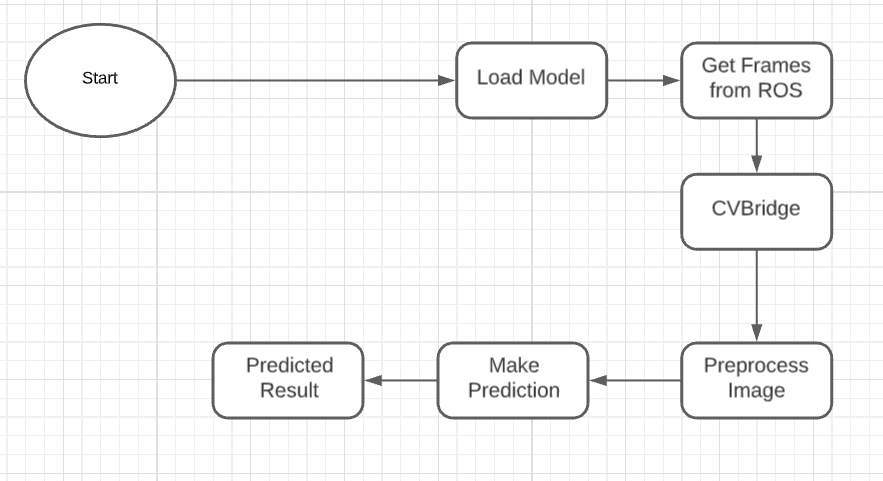
## Process Flow

For tagging the documents, we will use deep learning model. Below is the process flow diagram is as shown below.

### Model Training and Evaluation



### Deployment Process



## Event log

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

**Initial Step-By-Step Description:**

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

## 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 usage.

# Performance

Document tagging is used for tagging the documents and classifying the documents. Also, model retraining is very important to improve the performance.

## Reusability

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

## 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.

## Resource Utilization

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

* 1. **Deployment**



# Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.



As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

## KPIs (Key Performance Indicators)

Key indicators displaying the keywords used in the document and classifies the document based on the keywords extracted

# Conclusion

The Document tagging model will use the keywords of the articles to classify the documents based of the keywords used using various libraries.

# References