**News Article Classifier**

Functional Specification

Date: 30-10-2021

Prepared By: Phani Kumar. T, Raghu

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

[1 Overview 3](#__RefHeading___Toc27957_3360867896)

[Purpose 3](#__RefHeading___Toc27959_3360867896)

[Scope 3](#__RefHeading___Toc27961_3360867896)

[Assumptions 3](#__RefHeading___Toc27963_3360867896)

[Risks, Restrictions, and 3rd Party Impacts 3](#__RefHeading___Toc27965_3360867896)

[Detailed Requirements List 3](#__RefHeading___Toc27969_3360867896)

[Predict the news article category for a valid URL provided as input 3](#__RefHeading___Toc27971_3360867896)

[Find invalid URLs and throw error message to user. 3](#__RefHeading___Toc27973_3360867896)

[Application requires to find out invalidate URLs provide and should throw error to the user. 3](#__RefHeading___Toc27975_3360867896)

[Process the Data and clean it using nltk per-processioning techniques. 4](#__RefHeading___Toc27977_3360867896)

[Train the model with the processed data. 4](#__RefHeading___Toc27979_3360867896)

[Predict on Test data set. Obtain the Accuracy of the model. 4](#__RefHeading___Toc27981_3360867896)

[Provide a provision to retrain the model. 4](#__RefHeading___Toc27983_3360867896)

[4](#__RefHeading___Toc27985_3360867896)

[2. Processing Model Data: 5](#__RefHeading___Toc27987_3360867896)

# 1 Overview

This section provides a high level overview of the entire software product. This document describes all data, architectural, interface, component-level design, implementation and test for the software product.

The News Article Classifier provides the interface to predict a new article category. It will provide interfaces to retrain the model also.

## Purpose

Describe the purpose of the FITS and its intended audience. Describe the functional software design, implementation and test approaches.

The purpose of this document is to describe the functional software design, implementation and test approach for developers that will maintain and interface with the News Article Classifier.

## Scope

The scope should include the following details:

(1) Identify the software product to be produced by name; for example, Cash Dispenser Service Object, MSR Firmware Update Script, Coin Recycler Detection, etc.

(2) Explain what the software product will and will not do.

(3) Describe all relevant benefits, objectives, and goals.

The News Article Classifier application is a trained ml model which will classify a news article url provided as an input to it and predict to which category the article falls into. The current version of the implementation classifies the Business, Sports, Science and World categories only. For predicting other categories the model should be trained with the data required for the same.

The following features will be supported by the News Article Classifier application:

1. Ingest the data from various sources on demand

2. Train the model using both predefined data-set and from the ingested data.

3. Predict the news article category

4. Retrain the model with new prediction data set.

## Assumptions

List all relevant assumptions or N/A

1. We are using a pre defined data set and retrieving the categories from it. We trained the model with around 50k records from the data set. We assume that the data set is fair enough because of the accuracy of the model was reported as 89%. We verified this with some test data and the results met the expectations.

2. We are relying on the docker hub provided pre-built images. Our application works as long as the docker images are not changed.

## Risks, Restrictions, and 3rd Party Impacts

List all relevant risk, restriction and 3rd party impacts or N/A

1. For parsing the news url data we are depending on a 3rd party newspaper api. If the api server is un available that would make our application to fail to proceed for prediction.

2. The current version of the implementation classifies the Business, Sports, Science and World categories only.

3. All the modules/docker images are expected to be used as it is without any changes to the dependent library versions. Any changes to them may have different behavior or it may break the application.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Detailed Requirements List

List all requirements – the section number will be the reference number to the requirement for traceability. The requirements are not explained here

### **Predict the news article category for a valid URL provided as input**

Application requires to predict the category of the news article by parsing the content of the URL provided as input. It should validate the URL and should do prediction only for valid URLs

### **Find invalid URLs and throw error message to user.**

Application requires to find out invalidate URLs provide and should throw error to the user.

### **Process the Data and clean it using nltk per-processioning techniques.**

Data should pre processed and filtered by applying the pre processing techniques

### Train the model with the processed data.

Model training should be performed with the provided processed data. A pipeline should be created to fit the training data set and get the model. Trained model should be saved and loaded on demand.

### **Predict on Test data set. Obtain the Accuracy of the model.**

The test data should be provided to the model and do predictions on it. The accuracy parameters of the model should collected.

### **Provide a provision to retrain the model.**

The application should provide an interface to retrain the model on the prediction data set. Re train model should reload the saved model and it should train and save the model.

**Provide a provision to list the prediction history.**

Provide an interface to retrieve and list the predictions in the database.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

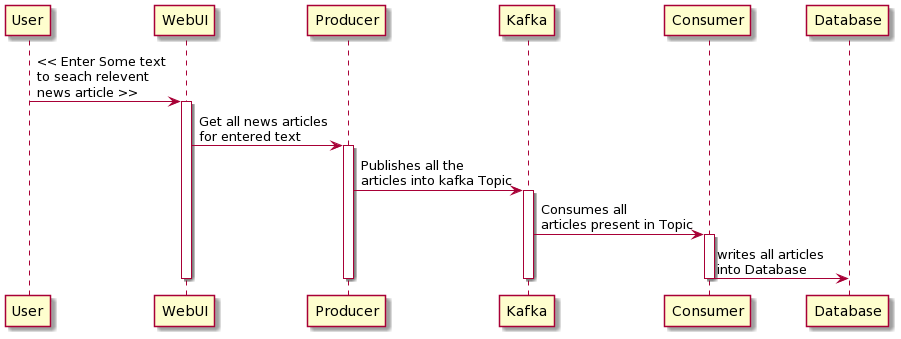
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**2 Sequence diagrams**

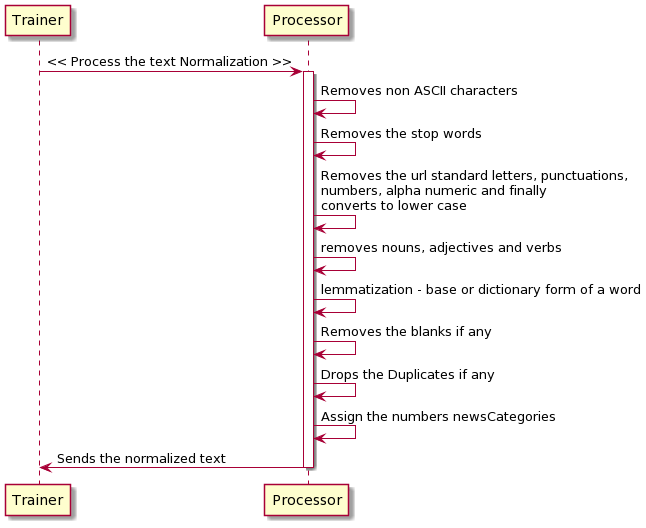
**1. Data Ingestion:**



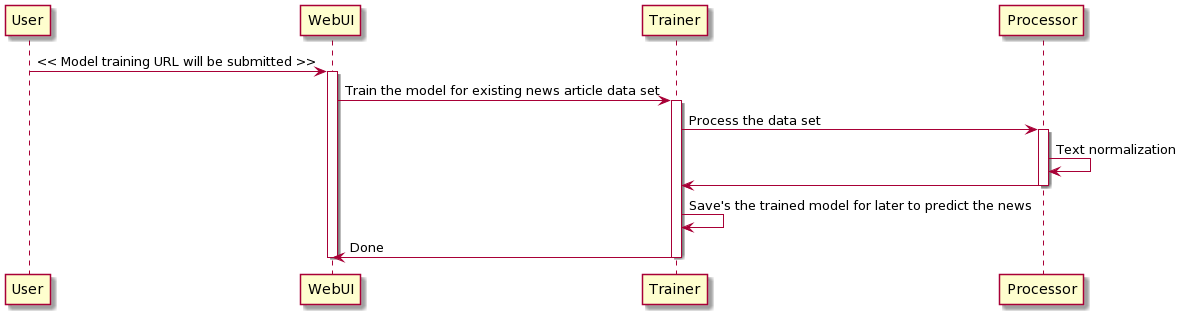
### 

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

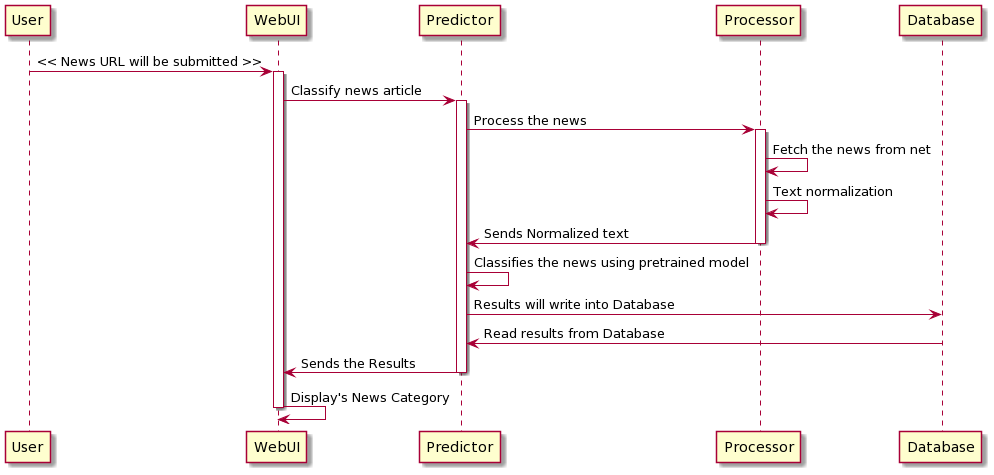
### 2. Processing Model Data:



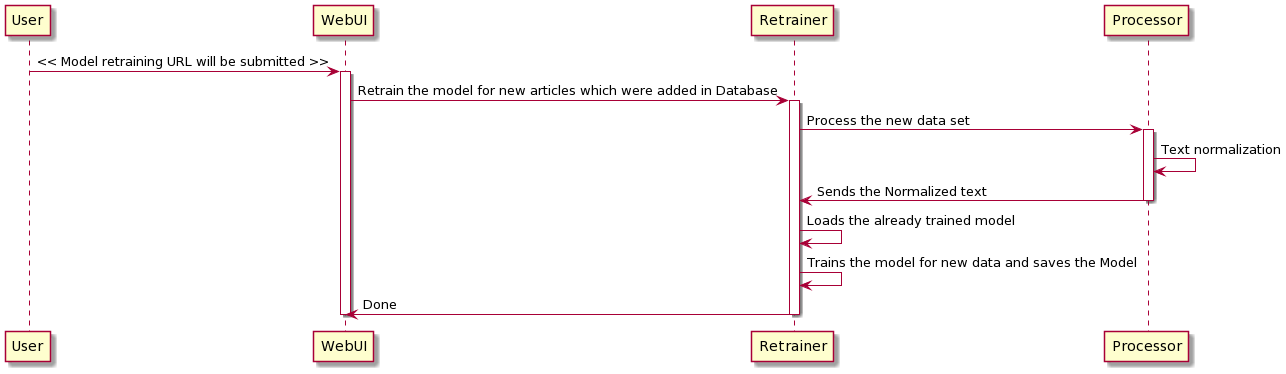
3. Training Model:



4. Predict Category of the News Article:

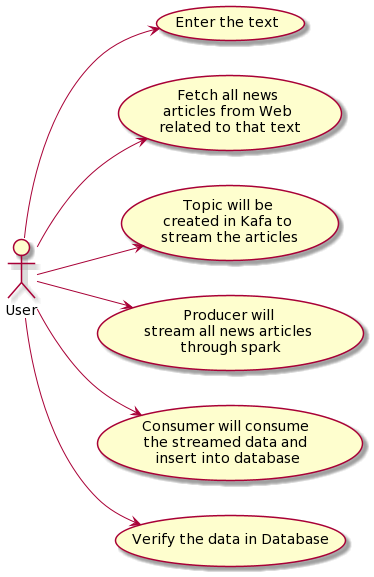


5. Retrain the model:

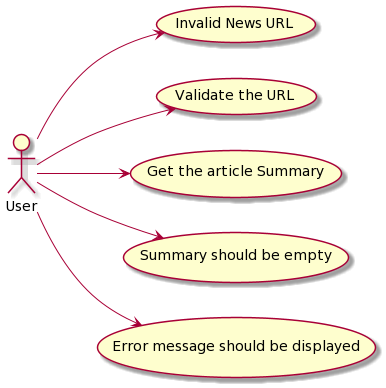


**Use Case Diagrams:**

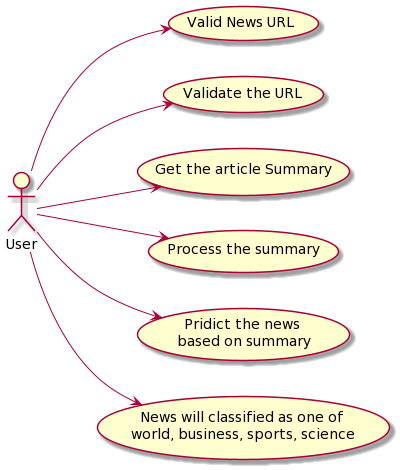
1. Data Ingestion:



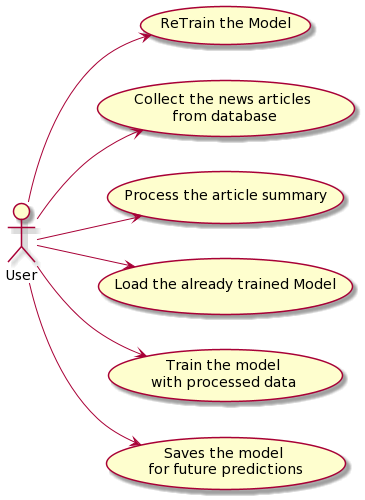
2. Invalid URL path provided to classify.



3. Valid URL provided for classification



4. Retraining News Classifier:



5. Train News Classifier Model:

