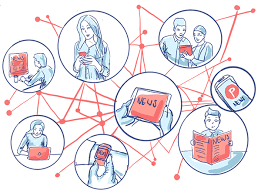
**High Level Design (HLD)**

News Articles Sorting



**Revision Number - 1.0**

**Last Date of Revision – 06/07/2022**

**SRIPHANI**

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 06/07/2022 | 1.0 | HLD Document | Sriphani |
|  |  |  |  |
|  |  |  |  |

**Contents**

Document Version Control…………………………………………………………………………2

Abstract……………………………………………………………………………………………….3

1. Introduction……………………………………………………………………………………….4
   1. Why this High-Level Design Document?..............................................................4
   2. Scope……………………………………………………………………………………4
2. General Description……………………………………………………………………………...5
   1. Product Perspective & Problem Statement…………………………………………5
   2. Tools Used……………………………………………………………………………...5
3. Design Detail……………………………………………………………………………………...6
   1. Functional Architecture………………………………………………………………...7
   2. Optimization………………………………………………………………………….….8
4. KPI……………………………………………………………………………………………….…9
5. Deployment……………………………………………………………………………………......9

# Abstract

* Over the last few years, the world has experienced a surge in the number of online news portals.
* This has caused the volume of news articles to reach an all-time high; which will only get higher with time.
* Thus, an efficient system of categorization and organization of the articles has become a necessity for various information systems like-news aggregation and association in search engines.
* It is impractical to employ humans to label this expansive volume of text data, prompting the growth of automated text categorization systems.
* And so, we devised a deep learning model that effectively categorizes news articles from the headlines and short text descriptions.
* The prime foci of our work were to design, develop, and measure the performance metrics of our proposed model.

# 1 Introduction

## 1.1 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 before 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

## 1.2 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.

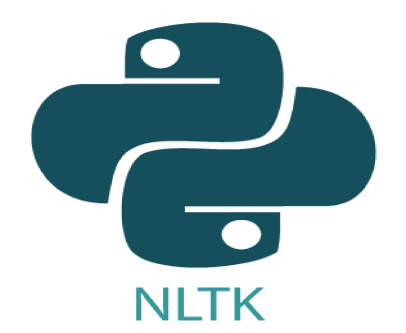
# 2 General Description

## 2.1 Product Perspective & Problem Statement

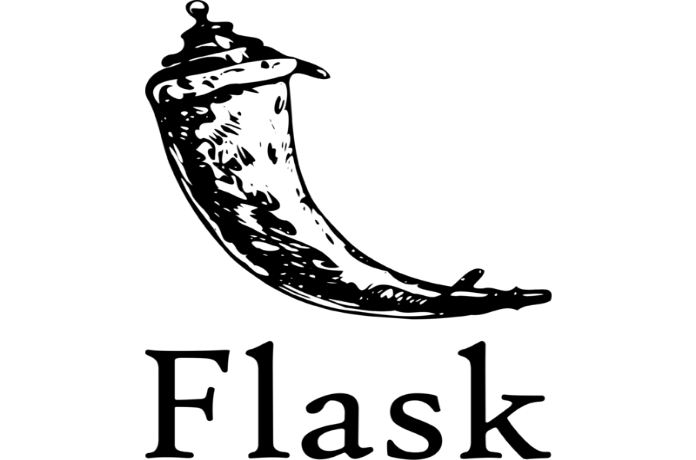
* In today’s world, data is power.
* With News companies having terabytes of data stored in servers, everyone is in the quest to discover insights that add value to the organization.
* With various examples to quote in which analytics is being used to drive actions, one that stands out is news article classification.
* Nowadays on the Internet there are a lot of sources that generate immense amounts of daily news.
* In addition, the demand for information by users has been growing continuously, so it is crucial that the news is classified to allow users to access the information of interest quickly and effectively.
* This way, the machine learning model for automated news classification could be used to identify topics of untracked news and/or make individual suggestions based on the user’s prior interests

## 2.2 Tools used

Deep Learning Technology (NLP) tools and libraries works such as NumPy, Pandas, Seaborn, Matplotlib, MS-Excel,pandas-profiling,scikit-learn,flask,nltk, Jupyter Notebook and Python Programming Language are used to build the whole framework.

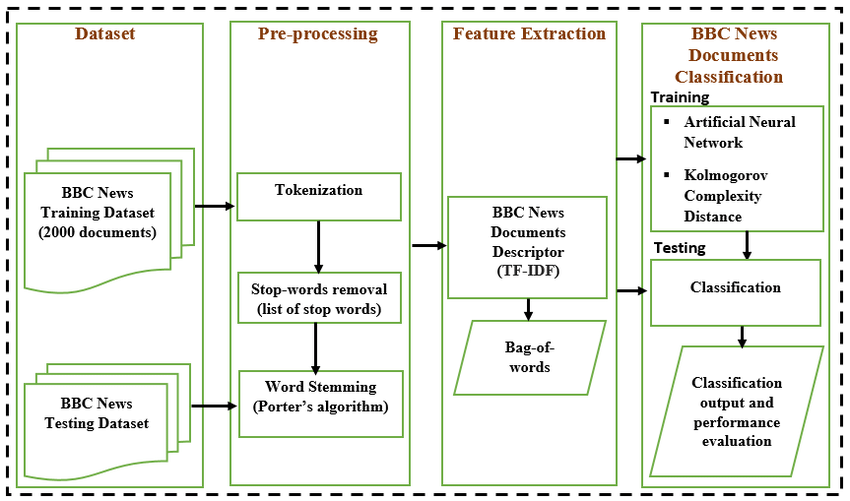






**3 Design Details**

## 3.1 Functional Architecture



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

**How NLTK WORKS**

## 3.2 Optimization

* This thesis focuses on the optimization of nlp components for robustness and scalability.
* Three kinds of nlp components are used for our experiments, a part-of-speech tagger, a dependency parser, and a semantic role labeler. For part-of-speech tagging, dynamic model selection is introduced.
* Our dynamic model selection approach builds two models, domain-specific and generalized models, and selects one of them during decoding by comparing similarities between lexical items used for building these models and input sentences.
* As a result, it gives robust tagging accuracy across corpora and shows fast tagging speed.
* For dependency parsing, a new transition-based parsing algorithm and a bootstrapping technique are introduced.
* Our parsing algorithm learns both projective and non-projective transitions so it can generate both projective and non-projective dependency trees yet shows linear time parsing speed on average.
* Our bootstrapping technique bootstraps parse information used as features for transition-based parsing, and shows significant improvement for parsing accuracy.
* For semantic role labeling, a conditional higher-order argument pruning algorithm is introduced. A higher-order pruning algorithm improves the coverage of argument candidates and shows improvement on the overall F1-score.
* The conditional higher-order pruning algorithm also noticeably reduces average labeling complexity with minimal reduction in F1-score.
* For all experiments, two sets of training data are used; one is from the Wall Street Journal corpus, and the other is from the OntoNotes corpora.
* All components are evaluated on 9 different genres, which are grouped separately for in-genre and out-of-genre experiments.
* Our experiments show that our approach gives higher accuracies compared to other state-of-the-art nlp components, and runs fast, taking about 3-4 milliseconds per sentence for processing all three components.
* All components are publicly available as an open source project, called ClearNLP.
* We believe that this project is beneficial for many nlp tasks that need to process large-scale heterogeneous data

# 4 KPI

* The procedure in this paper utilizes natural language processing (NLP) techniques to extract these terms or concepts from maintenance work orders and measure their influence on Key Performance Indicators (KPIs) as defined by managers and decision makers.
* We present a case study to demonstrate the developed workflow (algorithmic procedure) to identify terms associated with concepts or systems which have strong relationships with a selected KPI, such as time or cost.
* This proof of concept uses the length of time a Maintenance Work Order (MWO) remains open from creation to completion as the relevant performance indicator.
* By identifying tasks, assets, and environments that have significant relevance to KPIs, planners and decision makers can more easily direct investigations to identify problem areas within a facility, better allocate resources, and guide more effective analysis for both monitoring and improving a facility.

.

# 5 Deployment:

* we are going to deploy the NLP model with the flask on the Heroku platform. Here, we are going to use the GitHub method to deploy the flask application on the server.
* In the previous tutorials, we learnt how to train the movie review sentiment analysis model and how to deploy the movie reviews sentiment analysis model on the localhost using the flask application.

# 

# Machine Learning Model Deployment on Heroku Using Flask | by Charu Makhijani | Towards Data Science