

Minor Project Semester V

Group: B3

Project Guide:

Prof. Nisha Vanjari

Group Members:

Hiral Shah (49)

Jai Shah (50)

Sarvesh Soni (65)

Week 1 Progress & Detail Flowchart – 10/08/2021:

Problem Statement:

Department Newsletter Generator is a project that simplifies and eases the task of a editor of a newsletter. As the current system, editor has to manually gather all information of the department throughout they year, edit it, format it and make a document with the same.

Our project aims to remove all the hassles related to this process and automate almost every aspect of creating a newsletter by taking inputs from various faculties about the achievements and automatically creating a newsletter with pre-selected format.

Literature Survey:

A. Tools & Techniques:

- 1) Automated Local Newsletter Generator: For SVM classifier construction, we use LibSVM and for the geographic reference application, we employ Gazetteer, WordNet and Named Entity Tagger. The text preprocessing process includes stemming, stop words removal, too short/long or frequent/infrequent words deletion. The TF-IDF weight calculation is applied to convert text into vectors and scripts provided by LibSVM are applied to search optimal parameters for classifier construction.
- 2) Automatic Article Generator from Extracted Databases: ContentBot has been built as a web application tool to guide the user through the steps of creating new articles from its raw database. The tool has been built to ease the creation of this template that will then be used to generate the articles.
- 3) REAL TIME OPERATION OF NEWSLETTER GENERATION: Spring Boot application (Java 8). Docker container. Maven. Jenkins. JUnit for automated functional tests.

B. Methodology:

- 1) The automatic newsletter generator is composed of two main parts: topic classification and geographic reference application. The Support Vector Machine is employed for topic classification, which is well known and widely used in text classification area, and also proved effective in classification. For the geographic reference part, we apply Worldwide Gazetteer to check whether the user entered word is a place name and use WordNet service to collect all possible alias names for the place.
- 2) The generation tool then uses a random number to select the good synonym for each tag and adds for each of the articles the correct database entry. In our example, we have a total of six sentences, having a total of eleven synonym tags. to find out the amount of different sentence possibilities we multiply the amount of synonym per synonym tags together to find.
- 3) It is in the interest of Paf that I use these programs for development, so the new program can easily be ported into their system. Also, I am in contact with my supervisors at Paf during the course of this thesis. Thus, they gave me useful suggestions and ideas. And of course, I shall be using the knowledge benefited from both programming and database courses from the university.

Research Gap:

The 20 newsgroups data sets are collected several years ago and most of the information are retrieved from United State related news resources. Hence, when we want to evaluate locations in United Kingdom, the available documents are limited. The articles are still far from perfect to beat a traditional author. The purpose of this component is to help reduce time consumption and money by eliminating the possibility whereby a user mistakenly introduces a bug into program code when manually copying templates from email-template-gui, and making it possible to test templates.

Scope:

A. Required tools & technology:

Since our project does not go very deep into creating newsletter using machine learning or AI hence, we do not require many high-level technologies. We are using a basic website for input and output purposes and using python docx library to generate the document.

B. Required Methodology:

We are going to implement a website with forms for faculties to fill with numbers and description for any events throughout the year, the same information will be visible to the admin on the admin login where they can select all relevant information and on click of a button can generate pre-formatted newsletter in any format (.doc or .pdf).

Implementation:**A. Frontend Requirements:**

For frontend of the project, we are going to use HTML5, CSS and JavaScript for all the pages. We are also going to use Bootstrap framework for better visuals on the page.

B. Backend Requirements:

For backend of the project, we are going to use Django for handling overall website and for database purposes MySQL or Firebase will be used. We also plan to attach the Python docx library in the backend to generate the newsletter with already set format.

Expected Result:

We expect our final result to be a perfect website where any information provided by faculties regarding events conducted during the year will be seen by the admin, selected and on click of a button a newsletter for the department will be generated. We also plan to implement sending the same newsletter to everyone by mail from the website directly.