Improving ExpertSearch Progress Report

[**Plan** 1](#_Toc56973106)

[**Automate scraping process** 2](#_Toc56973107)

[**Perform topic mining** 2](#_Toc56973108)

[**Additional improvement** 2](#_Toc56973109)

[**Improve UI** 2](#_Toc56973110)

[**Progress** 2](#_Toc56973111)

[**Automated scrapping process** 2](#_Toc56973112)

[**1.** **Dataset preparation** 2](#_Toc56973113)

[**2.** **Scraper** 3](#_Toc56973114)

[**3.** **Text classification** 3](#_Toc56973115)

# **Plan**

## **Automate scraping process**

* To identify faculty directory pages
* To identify faculty home pages

## **Perform topic mining**

* To identify top-k topics associated with each faculty

## **Additional improvement**

* To improve email extraction for each faculty

## **Improve UI**

* To display top-5 topics associated with each retrieved faculty
* To allow search based on any of the topics from the displayed topic cloud
* To prepopulate email content when clicked on a faculty’s email address

# **Progress**

## **Automated scrapping process**

The automated scrapping process involves text classification for each of the subtasks listed above.

### **Dataset preparation**

First we need to prepare the dataset for training and testing the model. The following approach was used to prepare the dataset.

* + Downloaded the known faculty directory pages from the sign-up sheet for MP 2.1. These will serve as the “positive” examples.
  + Collected top URLs from Alexa. These will serve as the “negative” examples.
    - Collected the global top-50 pages of Alexa.
    - Collected the top-50 pages for different countries. Manually verified that the pages are in English.
  + About 900 URLs were obtained from the sign-up sheet data, which was partitioned into 500 for training and 400 for test data.
  + URLs for total 14 countries + top-50 global URLs from Alexa were collected. This gave 750 “negative” URLs.
  + Wrote a python module for data handling that does the following:
    - Convert the MP 2.1 sign-up data from csv to a file containing only the directory URLs. Perform any cleanup as necessary and labels them as “faculty”.
    - Combine the top-50 Alexa URLs for 10 countries and labels them as “alexa”. Use these 500 pages for training.
    - Combine the top-50 Alexa URLs for 5 countries and labels them as “alexa”. Use these 250 pages for testing.
    - Mix the 500 faculty directory training URLs with the 500 Alexa training URLs. Remove duplicates if any. This gives 734 URLs as the final training URLs.
    - Mix the 400 faculty directory test URLs with the 250 Alexa training URLs. Remove duplicates if any. This gives 548 URLs as the final test URLs.

### **Scraper**

Wrote a python module for scraping the URLs collected from the above step.

The scraper does the following:

* + Gets the contents of each URL as text.
  + Performs clean-up of non-ascii characters from the content.
  + Performs other clean-ups such as substituting newlines, tabs, multiple whitespaces into single whitespace.
  + Substitutes contents such as “403 Forbidden”, “404 Not found”, etc. with single whitespace.
  + Writes contents of each webpage as a single line of space separated words in a file meant to be the final corpus.
    - This is done to prepare both the training corpus (“train\_dataset.cor”) and the test corpus (“test\_dataset.cor”).

### **Text classification**

Wrote a python module for performing the text classification task of identifying valid faculty directory pages from the test corpus.

The classification module does the following:

* + Use gensim to build a Doc2Vec model for feature vector representation of each document.
  + Use the train\_dataset.cor to build the vocabulary and train the model.
  + Save the model so that it can be reloaded while running next time on the same dataset.
  + Use LogisticRegression as the classifier from scikit-learn module.
  + Now use LogisticRegression to predict the categories of the test URLs given the test dataset.