CSI4107 - Information Retrieval
Winter 2019

Search Engine Programming Project

VERSION 1.2
February 13th 2019

DIFFERENCE between Version 1.2 (Feb. 13th) and Version 1.1 (Feb. 9th)

I had wrongly included a wildcard (*) in the example queries (What to submit?) for the VSM model, and you are not asked to do that. I just removed it.

DIFFERENCE between Version 1.1 and Version 1.0

- **EMPHASIS in RED** in the document for the VANILLA system to submit soon, February 20th, 2019 !!!!
- SPECIFICATIONS (more details) about **WHAT TO SUBMIT** for your Vanilla System.

1. *Overview of the Search Engine Programming Project*

The table below provides the important information about the logistics of the project. Please read carefully to make sure you do not miss any information and deadlines.

Purpose	Develop your own Search Engine (SE). Have hands-on experience with the various modules which could be part of a search engine: retrieval model, user interface, relevance feedback, document clustering, etc.
Development	The project will be done in 2 phases. In phase 1, you must develop a Vanilla System which will be a first complete search engine with limited capabilities. In phase 2, you must include additional features to your system and perform a formal evaluation.
Teams	The programming project can be done in teams (max 2) or individually. The number of modules to be included in your Search Engine will be in relation to the number of students working together.
Programming language	MUST be either Java or Python.
Percentage	**Vanilla system (15%)**, Final system (15%)
Important dates	**Vanilla system → Feb. 20, 2019**, Final system → April 12, 2019 ***No extensions will be given. -10% per day late.***

What to submit ?	**For the VANILLA system (INFO ADDED HERE - Please READ) :** You must submit a single zipped file containing 3 files: (a) A small report (pdf) with: ● a title page indicating the names and student numbers of team members, and the name of your system ● a page showing a diagram of your system architecture ● for each module in your system architecture diagram, provide: ○ functionality / limitations (cases not handled) ○ problems encountered (if any, as you developed the module) ● pages containing some screenshots of the results of your search engine for each model for the following queries: ○ boolean model ■ (operating AND system) ■ (comput* AND graph*) ■ (crypto* OR security) ○ VSM ■ operating system ■ computers graphical ■ cryptographic security ● pages containing screenshots for additional queries (you decide which ones) that would highlight the particularities of your system. For example if you implemented spelling correction, you should show that. (b) A README file that will explain how to run your code. The TA MUST be able to run your code. (c) the Java or Python classes of your system The UI must be self-explanatory so that your search engine could be easily tested. If you work in teams, a single student should submit.
How to submit ?	Two links will be provided in Brightspace, in the Search Engine Project Module, one in February for the Vanilla system and one in April for the final system. **SUBMISSION LINK in Brightspace is NOW active for February 20th deadline.**
Mandatory source acknowledgement	In today's world of available snippets of code everywhere on the Internet, you will certainly copy/paste from various sites in addition to writing your own code. You MUST provide the source (URL links) of each site where you copied code from and say "Inspired from X" or "modified from X" in your class comments. Important: Failure to include links to sources will result in a grade of 0 for the project.

2. Collections

Your Search Engine will have to index and retrieve documents from 3 collections.

Name	Description	Link
UofO-CSIcourses	**A list of course descriptions. This collection will be used for the vanilla system.**	https://catalogue.uottawa.ca/en/courses/csi/
Reuters 21578	A news collection, often used for text categorization and information retrieval. This will be required for the final system.	http://www.daviddlewis.com/resources/testcollections/reuters21578/
Web collection	Your choice of a site (or a few sites) to crawl.	

3. Development of the Vanilla System

The vanilla (or baseline) system must perform retrieval operations on the UofO-CSIcourses collection. **The vanilla system must include the following mandatory modules, and either 1 (for people working alone) or 3 (for people working in teams) optional modules.** All modules are individually described in file CSI4107-SearchEngine-Modules.pdf.

Mandatory modules

1. Corpus pre-processing
2. User Interface
3. Dictionary building
4. Inverted Index Construction
5. Corpus Access
6. Boolean model of information retrieval
7. Vector Space Model of information retrieval

Optional modules

- Phrase query indexing
- Spelling correction with Edit Distance
- Spelling correction with Soundex
- Wildcard management with additional bigram indexing
- Relevance feedback (choose this one ONLY if you also do the probabilistic model)
- Probabilistic Relevance Retrieval Model

4. Development of the Final System

Requirements for the final system will be provided at a later time (during study break). But basically, it will be the inclusion of more modules (some mandatory, some optional). Already, we have seen in class query completion and query likelihood retrieval models that will be in the final system. The final system will also minimally include a second collection (Reuters).