CSI4107 - Information Retrieval
Winter 2019

Search Engine Programming Project

VERSION 1.4
April 6th 2019

New in this Version - Information about WHAT TO SUBMIT, for the final 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%) ATTENTION: Some students did not meet the requirements for the Vanilla System. To encourage you to perform better on the Final System, I will allow a 5% transfer from Vanilla to Final if your Final system is much better, so you can choose to make the weight (10% Vanilla, 20% Final). IF you wish for this transfer, you MUST show Yash (during your demo) that all the elements that were not working well in the Vanilla system have now been fixed.
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 FINAL system:**

You must submit a single zipped file containing 3 files:

(a) A small report (pdf) with:

- For each of the mandatory module, provide a short description of your implementation (how did you do it, any challenges?)

- Describe how you dealt with the Reuters collection.
  - Were you able to process all the files? If not, why? What caused problems? How many documents did you end up with?
  - Did you have any execution time issues? (searches being too long for example). If yes, what did you do?
  - How long does it take to generate the dictionary? Did you set up some constraints to make the dictionary generation faster?
  - In general, describe how more challenging it is to work with the Reuters collection than with the CSI collection that was used for the vanilla system.

- Provide the results of 5 queries (screenshot of your search engine):
  - shareholder AND security (boolean)
  - oil AND profit (boolean)
  - Canada canola oil (VSM)
  - European banks stockholders (VSM)
  - U.S. corn market (VSM)

- Take 5 test words to illustrate the behavior of your modules 8 and 9:
  - coffee, stock, oil + 2 words of your choice
  - Bigram Language Model
    - For each of the 5 test words, show the top 10 words for completion that your system generates
  - Query Expansion
    - For each of the 5 test words, show the top most similar words which would then become part of the expanded query. How many are you adding?

- For the topic classification:
  - Choose 5 different topics and give examples of documents which were automatically classified with these topics.
  - What k did you use for kNN?
  - We are not doing a formal evaluation, but does the classification seem to make sense on the 5 examples you chose?
  - Do you think the kNN is a good approach here?

- If you are working in teams, mention which additional modules you implemented. Mention who did what (how did you split the work). Then, for each additional module, provide a short description of your implementation (how did you do it, any challenges?). Also provide some results to illustrate what has been achieved.

(b) A README file that will explain how to run your code. The TA MUST be able to run your code. Yash is organizing live demos if possible. You have received an email from him.

	(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.
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. *ATTENTION: Students who did not provide references to sources only lost 1 mark in the Vanilla System as a warning. But for the Final System, you MUST provide the source of your code to not lose all marks.*

2. *Collections*

Your Final Search Engine will have to index and retrieve documents from 2 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/ I've also included the collection in Brightspace.

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

The final system must include a second collection, the Reuters news stories.

The final system must build on top of your Vanilla System. ALL mandatory modules required for the Vanilla System MUST be part of the Final System.

Additionally, the Final system must include the following mandatory modules. Students working alone are required to program the mandatory modules. Students working in team must program the mandatory modules as well as 2 optional modules.

All modules are individually described in file CSI4107-SearchEngine-Modules-FinalSystem.pdf.

Mandatory -- Adapting Vanilla System for Reuters Collection

The first requirement for you final system is to make sure your Vanilla System is fully functional on the Reuters Collection. You will need a pre-processing module for Reuters which you are free to program the way you want to generate a collection in the format used by your Search Engine.

Each Reuters document (from the files reu2-000.sgm to reu2-021.sgm) looks like the document below. We will use the text, the title, and the topics. You can ignore the other fields.

```
<REUTERS TOPICS="YES" LEWISSPLIT="TEST" CGISPLIT="TRAINING-SET" OLDID="20430"
NEWID="21007">
<DATE>19-OCT-1987 15:30:22.56</DATE>
<TOPICS><D>acq</D></TOPICS>
<PLACES><D>usa</D><D>france</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&#5;&#5;&#5;F
&#22;&#22;&#1;f2842&#31;reute
r f BC-BROWN-DISC-TO-BUY-RHO   10-19 0076</UNKNOWN>
<TEXT>&#2;
<TITLE>BROWN DISC TO BUY RHONE-POULENC &lt;RHON.PA> UNIT</TITLE>
<DATELINE>    COLORADO SPRINGS, Colo., Oct 19 - </DATELINE><BODY>Brown Disc Products
Co
Inc, a unit fo Genevar Enterprises Inc, said it has purchased
the ongoing business, trademarks and certain assets of
Rhone-Poulenc's Brown Disc Manufacturing unit, for undisclosed
terms.
    Rhone-Poulenc is a French-based chemical company.
    Under the agreement, Rhone-Poulenc will supply magnetic
tape and media products to Brown Disc Products.
 Reuter
&#3;</BODY></TEXT>
</REUTERS>
```

Figure 1 - Example of Reuters document with topic assigned

Final System Mandatory modules (for Reuters)

8. Bigram Language Model + Query Completion Module
9. Automatic thesaurus construction + Global Query Expansion with thesaurus (in VSM)
10. Text categorization using kNN + Topic Restriction

Final System Optional modules

- Global Query Expansion (with Wordnet)
- Visualization of automatic thesaurus (showing the related words to the user in a graphical way)
- Local query expansion in the VSM model (Rocchio algorithm)
- Dynamic clustering of documents (k-Means or HAC) with adapted UI to allow visualization and/or selection (counts for 2 optional modules)
- Text categorization (on Reuters collection) with Naive Bayes