Option 1 - Search Engine for the UCL Website

1. Do a literature review about general algorithms for Search Engines and enterprise search algorithms.

2. Build a live search engine that indexes and searches content within domain ucl.ac.uk. Try at least three different ranking algorithms

3. You can use any open source IR package whenever it is needed.

4. Implement and use PageRank algorithm for ranking.

5. Generate your own test set with queries and judgements (To be submitted together with the code)

6. Using the right metrics (use at least two metrics), and the queries in your test set, compare the results with:
    a. https://www.google.co.uk/#q=computer+science+site:ucl.ac.uk
    b. http://search2.ucl.ac.uk/s/search.html?query=computer%20science&collection=website-meta&profile=_degrees&tab=degrees

Option 2 - Learning to Rank

1. Do a literature review on learning to rank, including different types of methods and the advantages/disadvantages of each method over others.

2. Run LambdaRank, RankNet and one more learning to rank algorithm of your choice, tune the parameters of the selected models properly and explain the tuning procedure.
   a. Useful implementations of some learning to rank algorithms can be found in RankLib as a component of The Lemur Project. You are free to use other implementations/toolboxes.
   b. Dataset: MSRDataSet. More details about the properties and format can be found at: https://www.microsoft.com/en-us/research/project/mslr/

3. Treat ranking as a classification problem by building a logistic regression classifier that predicts relevance and use that for ranking. Implement and submit the code of your own classifier.

4. Using appropriate metrics (use at least two metrics, implemented by yourself), compare the quality of different algorithms and comment on their performance. What was the effect of parameter tuning on the performance? Which learning to rank algorithm outperforms the others? Why?

Option 3 - Home Depot Kaggle Challenge

The Home Depot Kaggle challenge requires participants to produce a model ranking products sold by Home Depot by relevance to user queries. Your goal is to improve their customers' shopping experience by developing a model that can accurately rank search results. This requires development of a Learning to Rank model. You will be using the Home Depot Kaggle Data.

1. Do a literature review on learning to rank, including different types of methods and the advantages/disadvantages of each method over others.

2. Run a learning to rank algorithm of your choice and tune the parameters of the model. Using appropriate metrics, show how parameter tuning affects the performance.
    a. Useful implementations of some learning to rank algorithms can be found in RankLib as a component of The Lemur Project. You are free to use other implementations/toolboxes.

    b. It is important to come up with good features to solve this problem well. Come up with your own features and analyse which features play an important role in ranking.

Submission Instructions

Submission:

1. A detailed report (Up to 8 pages group report + Up to 3 pages individual report. Latex is preferred).
    - The group report should include **introduction of the problem**, **methods used, analysis and comparative evaluation** of the methods (submitted only once per group), naming the file as Group xx.zip (e.g: Group 10.zip).
    - In the individual report, each team member should provide a **detailed literature review by yourself** and **your own contributions/methods to the project** with additional experiments/analysis which are otherwise not reported in the group project. You also have a paragraph commenting your role in the group and how the team functioned overall (submitted separately by each team member). Naming the file as Groupxx_YourName.zip (e.g. Group10_EmineYilmaz.zip)
    - A clear distinction between the two reports: the individual report focusing on methods developed by each group member, while the group report focusing on comparing them and possible combing them together.
2. Write a clear manual and upload the code to github with a clear indication it is for IRDM 2017 group project (with project number) at UCL.
3. You can use either Java, Python or C++ as the programming language.

Deadlines and Marking

Deadlines:

- Forming a group of up to 4 people and sending us their information: March 3rd, midnight
  Enter information about team members here
- Final submission: April 10th, midnight through Moodle (separate links for group and individual submissions provided in Moodle). Group projects can be submitted by one member; each member has to submit separate individual reports

Marking

- The total marks for both group and individual reports are 40% (the rest 60% is written exam)
- Out of 40%, the group report has 15% mark, while the individual report count for 25%
- Assessment Criteria (both reports):
  - Overall presentation of the report (problem definition, literature review etc)
  - Implementation (whether you have your own methods etc)
  - Evaluation (sensible metrics. convincing results. significance test etc)
  - Discussion and Analysis (critical thinking derived from your results etc)

  The above four criteria are equally weighted.

Contacts:

TAs:

Rui Luo r.luo@cs.ucl.ac.uk

Rafał Muszyński rafalmuszynski@gmail.com

Yixin Wu <hzwyx17@gmail.com>

Lecturers:

Dr. Jun Wang, jun.wang@cs.ucl.ac.uk

Dr. Emine Yilmaz, emine.yilmaz@ucl.ac.uk