THE UNIVERSITY of EDINBURGH
School of informatics

- University Homepage
- School Homepage
- School Contacts
- School Search

COURSEWORK 2

Based on lectures 9, 10, 13, and 14

This coursework is split into two main parts:

- **IR Evaluation**: based on lectures 9 and 10
- **Text Classification**: based on lectures 13/14 and labs 6

IR EVALUATION

In the first part of the coursework, you are required to build a module to evaluate IR systems using different retrieval scores. The input to your module is a retrieval results of a given IR system and a file that has the list of relevant documents for each of the queries.

Please follow the following steps:

- Download the following compressed file `systems.zip`. It has 7 files as follow:
  - 6 results files for 6 different IR systems, named `S[1-6].results`, each contains the retrieved set of documents for 10 queries numbered from 1 to 10. The format of the files is as follow:
  ```
  1 0 710 1 5.34 0
  1 0 213 2 4.23 0
  2 0 103 1 6.21 0
  ```
  The numbers above represent the following in order: `query_number 0 doc_number rank_of_doc score 0`.
  - `qrels.txt` file, which contains the list of relevant documents for each of the 10 queries. The format of the file is as follows:
  ```
  1: (9090,3) (6850,2) (9574,2)
  ```
  where the first number is the query number (`1:`), the remaining is the list of tuples of the document numbers and the value of relevance. e.g. `(9090,3)` means that document `9090` has a relevance value of 3. This value is only important for measures such as DCG and nDCG; while for measures such as P, R, and MAP, all listed documents as relevant are treated the same regardless to the value.
- Develop a module `EVAL` that calculates the following measures:
  - **P@10**: precision at cutoff 10 (only top 10 retrieved documents in the list are considered for each query).
  - **R@50**: recall at cutoff 50.
  - **r-precision**
  - **MAP**: mean average precision over all the retrieved results
  hint: for all previous scores, the value of relevance should be considered as 1. Being 1, 2, or 3 should not make a difference on the score.
  - **nDCG@10**: normalized discount cumulative gain at cutoff 10.
  - **nDCG@20**: normalized discount cumulative gain at cutoff 20.
  Note: Please use the equation in Lecture 9. Any other implementation for nDCG will not be accepted.

  Function name?
- The following files need to be created in the exact described format:
  - **S[1-6].eval**: these are 6 files for each of the 6 systems, named from `S1.eval` to `S6.eval` that corresponds to `S1.results` to `S6.results` respectively. Each file should contain a table of the above scores for each of the 10 queries. An example output file for a given system S9 could be found here. As shown in file, scores and heading are all tab separated. Before submission, please check that your out files for these 6 files is correct using the Perl script.

每个系统有10×6个值
共#sys个文件

将每系统有1×6个值.合到一个里

- **All.eval**: this file contains the average scores of each of the 6 systems. An example formatted file is here. As shown in file, scores and heading are all tab separated. Before submission, please check that your format is correct using the Perl script.
- Based on the average scores achieved for each system, you need to add a section in your report to describe the best system according to each score (i.e. what is the best system when evaluated using with P@10, and what is the best system with R@50, and so on). For each best system with a given score, please indicate if this system is statistically significantly better than the second system with that score or not. Please explain why.
  hint: using 2-tailed t-test, with p-value of 0.05. You are free to use existing tool for calculate the p-value. No need to implement this one.
- **NOTE**:
  - All files of results will be marked automatically. Therefore, please be careful with using the correct format.
  - Please round the scores to 3 decimal points (e.g.: 0.046317 --> 0.046).

TEXT CLASSIFICATION

In this part, you are required to apply text classification task on the same collection used in Lab 6

Please apply the following:

- Apply all the steps in this lab Lab6 & 7
- For the baseline system that you created with SVM classifier and BOW as features, print the measures in a file called `Eval.txt` with the following format:

```
Accuracy = 0.673
Macro-F1 = 0.631
Results per class:
1: P=0.8 R=0.6 F=0.685
2: P=0.45 R=0.712 F=0.551
3: . . .
```

- Try to improve the results of your classifier (you should have already did in the lab), the report a better performing system in a file: `Eval2.txt`.
- In your report on this assignment, please explain how you managed to improve the performance compared to the baseline system, and mention how much gain in the Macro-F1 and accuracy you could achieve with your improved method.

SUBMISSIONS AND FORMATS

You need to submit the following:

1. **`S1.eval`, `S2.eval`, `S3.eval`, `S4.eval`, `S5.eval`, `S6.eval`**, and **`All.eval`**: 7 files containing the IR scores in the format described above.
2. **`Eval.txt`** and **`Eval2.txt`**: 2 files containing the classification results of the baseline system and the improved system as described above.
3. **`code.zip`**: a compressed directory contains all the files of your code that produces the IR evaluation scores of the IR part, with a **readme file** of the steps to run it. Please try to make your code as readable as possible (commented code would be highly appreciated).
4. **`Report.pdf`**: Your report on the work. It should contain:
   - 1 page on the work you did in the assignment in general, which can include information on your implementation code, summary on what was learnt, challenges faced, comment on any missing part in the assignment.
   - 1 page on the best performing IR system for each score (you can put in a table), and an explanation of if the best system is significantly better than the second system or not, and why.

- 1-2 pages on the work you did on classification, and how much improvement you could achieve over the baseline, and how it was achieved (new features? learning method? more training data? ... etc.)

On a DICE machine, create a directory called tts2, and place the following files into it. When you're ready to submit, run the following command:

```
submit ttds cw2 tts2
```

Submission deadline: **11:59pm, 17 November 2019**

MARKING

The assignment is worth **10%** of your total course mark and will be scored out of **10 points** as follows:

- **4 points** for the outputs of the IR Evaluation, namely: `S[1-6].eval` and `All.eval`. These marks would be assigned automatically. Any problem in following the format will lead to dramatic decrease in your mark.
- **2 points** for the explanation in the report to the best IR system for each score and if it is significant or not.
- **3 points** for the improved system in classification.
- **1 point** on the code of the IR evaluation.
- **-1 point** *as penalty* in case the format of submission is not as required

ALLOWED / NOT ALLOWED

- For the IR measures, scores should be 100% calculated with your code. It is **NOT** allowed to use ready implementations of these scores. Only the ttest, you can use libraries (or any tool) to do it.
- For the classification, you can directly use your work in the lab. No need to do any new work as long as you managed to achieve improvement of the baseline system. However, your mark depends on the amount of work and improvement you achieve.

Home : Teaching : Courses : Tts