COSC 2670/2732 Practical Data Science with Python

Project Assignment 2, Semester 2, 2021

Marks : This assignment is worth 35% of the overall assessment for this course.

Due Date : Mon, 11 October 2021, 11:59PM (Week 11), via `canvas`. Late penalties apply. A penalty of 10% of the total project score will be deducted per day. No submissions will be accepted 5 days beyond the due date.

Objective

The key objectives of this assignment are to learn how to train and evaluate a non-trivial machine learning model. More specifically, the task is called "learning-to-rank". You will be given a set of training features that contain the relevance labels 0 (not relevant), 1 (partially relevant) and 2 (relevant), for a large set of query-document pairs. Your task is to research this problem, find a suitable solution, train a model, and produce a result file from a training set that will be scored using standard evaluation measures in Information Retrieval.

If you are unfamiliar with IR, you might want to look through the book Learning to Rank for Information Retrieval by Tie-Yan Liu, which is available online in the RMIT library and also on canvas now.

Provided files

The following files are provided:

- **A2.pdf** : This specification file.
- **train.tsv** : A large file of labelled query-document pairs suitable for training.
- **test.tsv** : The holdout set that you will use to create a runfile.
- **documents.tsv**: A 3 field file containing the document id, original html, and clean text parse of each document.
- **query.tsv**: A 2 field file containing the query id and the query text for each query.

  The A2.pdf file is on canvas, the train and test files are in a zip file you can download using the URL below called `A2data.zip` and the document.tsv and query.tsv files can be found in the optional download below called `extradata.zip`.

The Features Provided

Rules of the game

You are allowed to use any python library you like to solve the problem. The are a wealth of tools to choose from, including pandas, numpy, and scikit-learn for the basic processing, and multiple libraries designed specifically for Learning to Rank. I will let you find these on your own. It should be easy to find several that will work, and you can try several to determine which works the best. We will need you to ensure your environment is reproducible though, so the correct way to do this is to

Identifiers	**Static Document Scores**	**BM25 Scoring**
QueryID	PageRank	BM25Body
Docid	InlinkNum	BM25Anchor
Label	OutlinkNum	BM25Title
	NumSlashURL	BM25URL
	NumChildPages	BM25WholeDocument
Raw Counts/Lengths	**TFIDF related scoring**	**Language Model Scoring**
BodyTerms	IDFBody	LMIRABSBody
AnchorTerms	IDFAnchor	LMIRABSAnchor
TitleTerms	IDFTitle	LMIRABSTitle
URLTerms	IDFURL	LMIRABSURL
TermsWholeDocument	IDFWholeDocument	LMIRABSWholeDocument
LengthBody	TFIDFBody	LMIRDIRBody
LengthAnchor	TFIDFAnchor	LMIRDIRAnchor
LengthTitle	TFIDFTitle	LMIRDIRTitle
LengthURL	TFIDFURL	LMIRDIRURL
LengthWholeDocument	TFIDFWholeDocument	LMIRDIRWholeDocument
		LMIRDIRWholeDocument
		LMIRIMBody
		LMIRIMAnchor
		LMIRIMTitle
		LMIRIMURL
		LMIRIMWholeDocument

Table 1: Precomputed features by type included in the test collection.

create an anaconda environment for a specific version of python (I strongly suggest it be 3.8, install any packages you need using pip (**not** anaconda), and then generate a requirements.txt file to include with your submission. So, something like:

```
conda create -n SXXXXXX python=3.8
conda activate SXXXXXX
pip install pandas numpy scikit-learn
pip freeze > requirements.txt
```

This will create a new environment you can start in Anaconda using "`conda activate SXXXXX`" and exit from using "`conda deactivate`".

Rubric

The marking will be defined as follows:

- Making your environment reproducible (5/35 marks). So heed my suggestion above. If you do not submit a correct requirements file and we cannot reproduce your results, you will lose marks.

- A short write-up of your methodology and model decisions. This should be no more that 4 pages single column, 12pt font. We will provide an exemplar template to help you know what you cover. (10/35 marks)
- The remaining marks (20/35) are based on effectiveness. We will define 4 cutoff scores. If you achieve only the lowest, it is +5, the second +10, the third +15, and fourth +20. The scores will be based on model complexity. So a very simple model would achieve the bottom quartile score, and a state-of-the-art one should achieve the top quartile score. The score boundaries will be released next week when we have had a chance to finalise all the possibilities. Note that you can also achieve the top quartile even with a simple model if you decided to try your hand at feature engineering. I implemented a pretty simple one myself already and it had a significant impact on performance. So, we are providing the raw document and query data but you do not have to use this if you do not want to do any feature engineering on your own. You will be able to achieve the top quartile if you use the right model and tune it properly.

Once you have your model working, you should generate a single file called `A2.run`. There should be 3 **tab separated** columns of data in the file which are query id, document id, and score. These will look something like this:

Query ID	Document ID	Score
ff098645	7318933826e5	1.286884571732504
ff098645	277c9afdafd3	1.2867450764432404
ff098645	49a432153c63	0.9089192528980292
ff098645	140947f5670d	0.8923819355323458
ff098645	ade09567775c	0.6383698094322833
ff098645	5e9f0ae7ec18	0.5154112180160674
ff098645	3822873e8391	0.49284627218888555
ff098645	5eb78fecc57d	0.17202148076799256
ff098645	f43e3f85eba0	0.1210381723133895
ff098645	24f2085598ef	-0.005703002133336521
ff098645	7758d84f2f68	-0.10890231193119822

Table 2: An ordered list sorted by Score and by Query ID.

Test Collection

You can download the two data files from:
`http://wight.seg.rmit.edu.au/jsc/A2/A2data.zip`
`http://wight.seg.rmit.edu.au/jsc/A2/extradata.zip`

The extra data file contains raw data from the documents and queries and is optional. You only need to get it if you intend to try to create your own features. Be warned, this file is reasonably large, even though it is compressed. It is around 985MB of raw data.

Hints

The best solutions are very likely to be pairwise or listwise regression algorithms. Also, you will want to group by Query ID for reasons I hope are obvious. In other words if you are going to create a pair of positive and negative instances, you would want them to be from the same query so that your model gets better at discriminating between a relevant and non-relevant document. You want to treat it as a regression as you are predicting the most likely score for a document, but all evaluation in web search is based on the idea of you returning a "ranked list" (think 10 blue links in Google). These are just documents sorted from most likely to least likely to be relevant for a particular query. We will cover this more in the lectorial this week and next week.

Submission

All submissions must be made through Canvas. A link will be provided within canvas, under the Assignments tab for submissions. Assignments submitted through any other method will not be marked. To submit your files, create a top level directory which is your student number. Inside that folder, there should be exactly like the zip file provided to you. You **should not submit the data**. For example, if your student ID is S101010, when I unzip the file `S101010.zip`, I would see the following:

```
$ unzip S101010.zip
Archive:  S101010.zip
   creating: S101010/
  inflating: S101010/A2.py
  inflating: S101010/requirements.txt
  inflating: S101010/A2.run
```

NOTE 3 : There is a discussion group available in the course canvas. Please do not post code snippets in this forum. Do ask questions if you have them! We will do our best to answer them as quickly as we can.

Plagiarism Warning

University Policy on Academic Honesty and Plagiarism: It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student concerned. Please see the RMIT policy for more details: `http://rmit.info/browse;ID=sg4yfqzod48g1`. **THIS IS NOT A GROUP PROJECT.** Students are reminded that this assignment is to be attempted individually. Plagiarism of any form will result in zero marks being given for this assessment, and can result in disciplinary action. We routinely use plagiarism software on projects! Please, please don't do it. Be aware that paying someone on a coding site to do it for you is a form of plagiarism. If you are submitting work that is not your own, regardless of how you got it – you are in breach of this policy.

Extension Policy

Individual extensions will **NOT** be considered or granted by the PDS team. We are not monsters, but doing things on time matters. We consider it a core component of the assessment. If we decide to extend the deadline, the only fair way we can do this is to extend it for *everyone*, and that may not be possible given the the size of the course and university policies. We have set deadlines as late as we can in order to meet the university timelines we cannot control. So be early and not late. If you procrastinate and suddenly have other priories, you will be in a real bind, and that is not a valid reason for an extension.

If you have suffered a personal tragedy or illness, there is a University process in place to grant extensions. Our preference is that you go this route if you must, as they have very clear criteria to grant exemptions. For more information about applying for Special Consideration, see the rules and regulations at `http://www.rmit.edu.au/browse;ID=b1wqvnwk8aui`.

Getting Help

Come talk to us. Email us. Use the discussion board. Ask a question in a lectorial or a practical. There is help available if you need it.