CIS 530 Spring 2017 Project

Instructor: Ani Nenkova
Head TA: Gil Landau

Released: April 8, 2017
Due: 9:00am, April 25, 2017.
No late days allowed because presentations should happen before reading days.

Overview

For the final project you will revisit the assigned task from the second homework: predict the relative difficulty of a text. The task is to predict the difficulty of a text as a real value score. This time around, you will receive training data so that you can explore supervised approaches if you wish. You can reuse code and ideas from any of the completed class assignments.

The focus of the project is on feature engineering, so the challenge is to come up with features that distinguish the relative difficulties of different texts. Your previous experience with the task may serve you well here. You will also need to think about the applicability to this task of certain techniques that we have covered in class. The motivation and reasoning for trying out certain solutions will be rewarded as much as the actual performance of your final system.

The evaluation metric will again be the correlation between the gold standard scores and the difficulty scores produced by your system. The gold standard is the "perceived difficulty score" you assigned to text as part of the first assignment. The three teams that achieve the highest correlation will be awarded extra credit. Three teams will also get extra credit for insights they get about the strengths and weaknesses of their system. The labels for the test dataset will not be released but you can submit **up to five** preliminary predictions and receive a score to check your progress. Your sixth submission is your final submission, and it should be based on the model you describe and evaluate in your final report. We will set up a leaderboard so that you can compare your results to the other teams and the baseline.

The baseline system is the Spearman correlation of a prediction based on the fraction of words not in New York Times corpus from Homework 2. You may build this system first and improve upon it, or start your own from scratch. **Unlike previous homeworks, you will not be graded on whether you perform better than the baseline.** You will be graded on your reasoning and analysis alone.

1 Data

You will have available two sets of data: a training set and a test set. The training set will be labeled data similar to those in the perception judgments you did as part of assignment 1. The scores are out of 10, where 1 is the easiest and 10 is the most difficult.

There will be one excerpt per line in `project_train.txt`, for a total of 461 excerpts. Its score will be in `project_train_scores.txt`, on the same line as the text is on.

In addition to the texts labeled with the perceived difficulty, there will be an optional training set which will include about 5,000 medical abstracts in the folder `optional_training`. There will be one abstract per file and their scores will be stored in the file `optional_project_train_scores.txt`. Each line

in `optional_project_train_scores.txt` is tab-separated, where the first value is the name of the file and the second value is the score.

The files in the optional training set are also medical abstracts, but their scores have a different meaning. The score in this set indicates how easy it was for annotators to find specific information in the text. As a result, an abstract with a higher score indicates more agreement amongst the annotators. This could indicate an easier abstract. *Therefore, a text with a higher score may be easier.*

There are 50 excerpts in the test set. We will not provide the labels for the test set, but you will submit your predictions in a specified format (see below) and receive your correlation score. You will only be able to evaluate your performance on the test set five times before your final submission, so you will need to plan carefully which settings of your model you want to compare.

All project data are stored on `biglab` under `/home1/c/cis530/project/data`. The files include:

- `project_train`, which contains the training excerpts, one per line.
- `project_train_scores`, which contains the training scores, one per line.
- `project_test`, which contains the test excerpts without labels, one per line.
- A directory `optional_training`, which contains roughly 5000 excerpt files and a mapping score file `optional_project_train_scores`

2 Evaluation

To evaluate your system, you will submit your predictions for each test excerpt. The predictions file should contain one line for each excerpt, consisting of the predicted label, *in the same order that the excerpts appear in* `project_test`. As with the previous assignment, the predictions are expected to be real numbers, but there are no limitations on what those real numbers are.

The test set was randomly drawn, so the distribution of text difficulties is roughly, but not exactly, the same in the test set as in the training set.
As with the previous assignment, we will use Spearman correlation to evaluate your predictions. The correlation of your submission will be automatically updated on the class leaderboard. Updates will happen automatically every 30 minutes.

3 Project guidelines

- Projects can be done individually or in teams of two.
- In your project report, you should report the correlation of the system submitted last to the leaderboard. You should submit the code that generated the final leaderboard results.
- The three best systems among all groups get extra credit for the final project. First place gets 15%, second gets 10%, third gets 5%.
- **Your code does not need to be executable on our machines.**

4 Your System

Develop a supervised system to distinguish articles by their relative difficulties. You can reuse code from earlier class assignments as you implement features for your classifier. *We request that you use at least two tools for language processing* that you find useful (i.e. tokenizers, POS taggers, dependency parsers, etc).

Classifiers and other machine learning methods do not count as tools for this purpose because they are not related to language specifically. You should describe all tools you use for preprocessing in your report.

Note that we did not release a dedicated development set, so to avoid over-fitting you should perform model validation on your own using data within the training set. You can do this either by explicitly separating a validation set from the training set data, or using cross-validation.

Your final submission should include the code for the final result you submitted, which is also recorded on the leaderboard.

5 Report requirement

For the project, please feel free to use the tools provided at the project resource page. The project report should be submitted as a PDF and should be no more than 2,500 words long. Keep the discussion focused and to the point, shorter is better. **Clear and complete write-ups with strong discussion and analysis sections will receive up to 15 points extra credit for the final project.** It should include the following sections:

- **Introduction**. The general idea/method for your system: which features you chose to implement and why you expect them to be predictive.
- **Method**. What resources or tools you have used and how they are included in your model.
- **Final system**. Give a brief description of your final system.
- **Experiments**. The correlation of your system and its variants on the test set. The final correlation you report should be that for your last submission on the leaderboard. You should record any preliminary results before the final submission so that you can include them in your writeup.
- **Discussion and Analysis**. Discuss your results and any interesting comparisons/conclusions that you can draw from them.

Each of the Introduction, Methods and Analysis results will be scored separately for grading. The introduction section will be receive full scores if the proposed approach logically matches the task. The methods section will receive full scores if you use at least two types of text analysis covered in class, either as tools or implemented by you. The discussion will receive full grade if your provide an insight about why the system does or does not work well and what would be additional ways for improvement you could have explored if you had unlimited time for the task.

6 Submission

6.1 Register your team

Before getting results on the leaderboard, you need to submit your team name. **Every team member has to submit an individual registration**. The deadline for registration will be **11:59PM, April 13th**. You simply need to submit a plain text file called `team.txt` that contains a single line with your team name. We need to know the number of teams before releasing the sign-up schedule for presentations.

```
% turnin -c cis530 -p proj_groups team.txt
```

6.2 Submit to the leaderboard

You are allowed to submit five preliminary test predictions to the leaderboard. The sixth submission will be your final one – the one you need to submit code for and describe in your final report.

To get your prediction score on the test set, submit a single file called `test.txt` to the leaderboard.

```
% turnin -c cis530 -p leaderboard test.txt
```

The format of the file `test.txt` is described in Section 2. Your score on the test set will appear at the leaderboard after the next scheduled update.

The leaderboard will be automatically updated every 30 minutes, so you may not see your score immediately. The leaderboard will be hosted here: `http://www.cis.upenn.edu/~cis530/leaderboard.htm`.

6.3 Submit your code and write-up

Please submit the code for your system and a short `README` describing your submitted files and how to run your code. Your final submission is due by **09:00AM, April 25th**.

```
% turnin -c cis530 -p project project_yourpennkey.zip
```

or

```
% turnin -c cis530 -p project project_yourpennkey_teammatespennkey.zip
```

Only one group member needs to make the final submission. Your submitted `.zip` should include the following:

- Code for your system
- A short `README` describing how to run your code
- Your final project report (.pdf format)

6.4 Milestones

- April 13th: Register your team
- April 14th: Leaderboard opens and sign-up schedule released
- April 25th: Final submission due and leaderboard closes
- April 25th-26th: Project presentations