Project 3 Language Model

COMP4901K and MATH 4824B

Fall 2018

Notes

- Report, code and result submission due: December 16 at 23:59. **No late submission is accepted.**

1 Content

1.1 Task

Project 3 is related to lab 8 (language model). The task is to predict the last token given a sequence.

```
Training corpus:
    D1: clear sky blue sky
    D2: the blue car
    D3: sky is nice

Pre-defined vocabulary:
    [blue, car, clear, nice, sky, <unk>, <eos>]

Test sequence:
    The blue sky is

Sample output:
    [0.82, 0.04, 0.05, 0.08, 0.01, 0.00, 0.00]
```

1.2 Sample Code

In this project, you can use any programming languages/third-party libraries to implement your algorithm. If you have trouble on implementing it on your own, you can use the sample code [1] implemented by our TAs. Here we will give a brief tutorial on the sample code. To run the sample code successfully, you should install `keras, tensorflow, sklearn`.

[1] You can download sample code together with the data

After downloading and unzipping the whole file, you can find these following files:

```
./data
    train.csv
    valid.csv
    test.csv
    vocab.json
./baselines
main.py
data_helper.py
utils.py
scorer.py
```

The function of files is as follows:

- **data/*.csv**: These files have three columns `id / sentencce / label`, where `sentence` contains all previous tokens of a sentence except the last one and `label` is the last token. Similar to project 2, we replace all `label` field of `test.csv` by -1.
- **vocab.json**: This file is a dictionary which map tokens into indexes.
- **baselines/*.csv**: Result files for `valid.csv` produced by different algorithms.
- **main.py**: Language model skeleton code.
- **data_helper.py**: Data loader.
- **utils.py**: Tools involved in this projects.
- **scorer.py**: You can score your model on `valid.csv` offline. The input of this file is your result file for `valid.csv`. After the deadline, TAs will score your result file for `test.csv` using the same evaluation metrics. Unlike project 2, you have only one chance to submit your result file this time.

You can use the following command to train a model:

```
python main.py -mode train -saved_model models/model.h5  -student_id 12345678
-epochs 1 -batch_size 32 -embedding_dim 100 -hidden_size 500 -drop 0.5
```

If you encounter a "TypeError" in the end of the program, you can just ignore it. If you have GPU, you can use option `-gpu` to assign a GPU device. Once you have trained a model, you can use test mode to predict the last words of the sentences in `valid.csv` usiing the following command:

```
python main.py -mode test -saved_model models/model.h5 -input
data/valid.csv -student_id 12345678
```

The result file named `12345678_valid_result.csv` for `valid.csv` will be generated. You can score it on the validation set. You can then compare your score with baselines to know the performance of your model.

```
python scorer.py -submission 12345678_valid_result.csv
```

Once you have finished tuning your model, you can make a submission for `test.csv`.

```
python main.py -mode test -saved_model models/model.h5 -input
data/test.csv -student_id 12345678
```

The result file named `12345678_result.csv` for `test.csv` will be generated. This time you cannot score it since there is no ground truth in `test.csv`. You should submit it along with your report and code, and our TAs will score your submission after the deadline.

1.3 Regulations

The following items provide you some regulations of the project:

- The naming convention of result file should be `student_id_result.csv`, e.g. `12345678_result.csv`. Other naming methods **get 0 point**.
- No late submission is accepted, otherwise **gets 0 point**.
- You are supposed to finish this project on your own. Plagiarism and teamwork is not allowed.
- You should submit result file for `test.csv`, not for `valid.csv`.
- You can use any programming language you like and any third-party libraries.

2 Submission

You need to submit three files, `12345678_result.csv` for `test.csv`, your code `12345678_code.zip` and report `12345678_report.pdf` to briefly describe your algorithm. Note that "`12345678`" should be replaced by your student id. Please do NOT put three files into one folder, zip it and submit it. You should submit all three files separately.

The result submission file should satisfy this format for each row:

$$id\ ,\ P(w_0)\ ,\ P(w_1)\ ,\ ...\ ,\ P(w_{N-1})$$

where $P(w_i)$ is the probability of i^{th} word in the vocabulary, and N is the vocabulary size.

In the report, you need to include the following points:

- Your name, student_id, your scores on the validation set. (10%)
- What algorithms/architecture are you using in this project? (30%)
- How do you conduct parameter tuning? List all the parameters and results you have tried (30%)
- How to run your code? Which third-party libraries are you using? (30%)

3 Grading Rubrics

We will follow the following grading rubrics for the final grade of this project.

Grade	Model (80%)	Report (15%)	Code (5%)
60%	baseline 1	Submission of the report	
80%	baseline 2	Showing algorithms you used	
90%	baseline 3	Detailed explanation of the algorithms	
100%	baseline 4	Very detailed and insightful analysis	Submission of the code

Table 1: Grading Rubrics.

The 15% of report weight will be applied as a weight for each of the four items shown in Section 2.