CSE 4392 - Assignments - Assignment 7

List of assignment due dates.

The assignment should be submitted via Canvas. Submit a file called assignment7.zip, containing all source code files needed to run your solutions for the programming tasks. Your Python code should run on Google Colab, unless permission is obtained via e-mail from the instructor or the teaching assistant.

All specified naming conventions for files and function names are mandatory, non-adherence to these specifications can incur a penalty of up to 20 points.

Your name and UTA ID number should appear on the top line of answers.pdf and all source code files.

Task 1 (100 points, programming)

File authors_main.py contains an incomplete program that trains and evaluates a neural network model that predicts the author of a short piece of text (about 40 to 300 words). The training and test data for this program are stored at file dataset.zip, which you should download and unzip.

The dataset contains text from three authors: Charles Dickens, C. S. Lewis, and Mark Twain. The dataset folder is structured as follows:

- The dataset/train_data contains data that your model should use for training. Using the data in this folder to create an appropriate training set is part of your task. The data here consists of the full text of the following books:
  - David Copperfield, by Charles Dickens.
  - Prince Caspian, by C. S. Lewis.
  - The Lion, the Witch and the Wardrobe, by C. S. Lewis.
  - Tom Sawyer, by Mark Twain.

- The dataset/test_data is provided just for reference. **YOU SHOULD NOT USE THIS DATA IN ANY WAY IN YOUR CODE.** This folder provides the full text of the books that are used in our test set:
  - Great Expectations, by Charles Dickens.
  - Out of the Silent Planet, by C. S. Lewis.
  - The Prince and the Pauper, by Mark Twain.

- The dataset/test_tf contains the test set for the program. This test set is stored in the appropriate format so that we can load it using the keras.utils.text_dataset_from_directory function. This is done in the code already provided in file authors_main.py. There are 3,000 text files here, 1,000 from each author. Each text file contains a small piece (from 40 to 348 words) from one of the three books in the dataset/test_data folder. To generate each text piece stored here, I followed this process:
  - Pick a random "sentence" S from one of the books. Here, "sentence" is simply text between two "." characters.
  - If the sentence was 40 "words" or longer, store it as one of the text files. Here, "word" is simply a string between two white space characters.

- If the sentence was shorter than 40 "words", append to it the next sentence, and keep doing so till we get a text piece with at least 40 "words".

To complete that code, you must create a file called authors_solution.py, where you implement the following Python function:

```
model = learn_model(train_files)
```

The `learn_model` function takes as argument `train_files`, which is a list of lists of filenames. Element `train_files[i]` is a list of filenames specifying the text files storing books to be used as training data for one specific author. The `train_files` variable is already defined in authors_main.py.

Your function should somehow (how exactly is up to you) the text files stored in `train_files` to create an appropriate training set, and then it should use that training set to train a neural network model. The function returns the trained model.

Here are some recommendations, that you may choose to follow or not:

- Use a fully-connected model.
- Use a bag-of-words representation, with ngrams=1 or ngrams=2.
- Use a vocabulary size of 10,000 or 20,000. I found that with ngrams=1, sometimes the extracted vocabulary was under 20,000 anyway.
- Use 3,000 training examples (1,000 per author).

You can see how the `learn_model` function is used in authors_main.py, to verify that you understand what it is supposed to do. When grading, we reserve the right to test your solution with different code (instead of authors_main.py), so your solution should comply with the specifications given above and should not assume the existence of any global variables defined in authors_main.py.

Some Additional Information

Here is some additional information about my solution.

- It takes about 30 seconds for the whole program to run on my computer.
- My solution is about 100 lines of code, including import statements and blank lines.
- I tried various models, and ran each of them five times. The simplest model I tried gave the worst average test accuracy, ranging from 62.83% to 67.73%. Your solution should be able to at least match that accuracy. The best model gave a test accuracy ranging from 67.83% to 71.13%. You may be able to get better results, I definitely did not do a very thorough search of all possibile variations.

Task 1b (Extra Credit, maximum 10 points)

10 points will be given to each of the three solutions that give the best test accuracy.

To qualify for consideration, you need to run your solution 10 times (or more, if you want), and report a summary of results in answers.pdf. Your result summary report smallest, largest, mean, and median of the test accuracies that your solution achieved. You should also document in answers.pdf the design choices that you

made.

Task 1c (Extra Credit, maximum 10 points)

Here, you are allowed to add more books to the training data, and/or do transfer learning from other models that you may find or build yourself. 10 points will be given to each of the three solutions that give the best test accuracy. One restriction: you cannot add books that are prequels or sequels to any of the books in our test data. The only books that I can think of that fall under this restriction are the other books from C. S. Lewis's Space Trilogy, but there may be other books I am not aware of that the restriction would apply to.

To qualify for consideration, you need to run your solution 10 times (or more, if you want), and report a summary of results in answers.pdf. Your result summary report smallest, largest, mean, and median of the test accuracies that your solution achieved. You should also document in answers.pdf what books you added (if you added books), and what base models you used for transfer learning (if you used transfer learning).

CSE 4392 - Assignments - Assignment 7