CSI5386: Natural Language Processing

Assignment 2

Text classification using deep learning

Due: Nov 27, 2022, 10pm

**Note:** This assignment should be done in groups of students. We use the formed BrightSpace groups. When one person in the group submits, the submission can be seen by all the members of the group.

# Part 1: Legal text classification [50 points]

Use the Atticus dataset of legal contacts: <https://zenodo.org/record/4595826#.YyXT6HbMI2w>

from Assignment 1.

Download the file CUAD\_v1.zip and unzip it. The folder full\_contact\_txt/ contains 510 files with full text contracts (a collection of TXT files of the underlying contracts). Each file is named as “[document name].txt”. These contracts are in a plaintext format and are not labeled in that directory, but there are 41 labels provided (in the xls files), for 41 types of relevant text (paragraphs) to be extracted from the contracts.

There is GitHub code provided, feel free to use it, extend it, or to develop your own.

ReadMe and Datasheet are [here](https://drive.google.com/drive/u/1/folders/1Yu-JnZj1LbVBfTdPiHfMDnaKZj4eqks8), the code for replicating the results, together with the model trained on the dataset is on Github [here](https://github.com/TheAtticusProject/cuad).

The data was split into 80% for training and 20 % for test. Please use the provided split.

The task is to highlight salient portions of a contract that are important for a human to review, that is, to extract and classify the relevant pieces of text into the 41 classes. This means to detect where one text of interest starts and where it ends. It can be framed as a question answering task, where the name of the class is the question. There are also irrelevant sentences that should not be extracted. The data is imbalanced because there are many irrelevant sentences. This is why the evaluation measures should be some that are appropriate for imbalanced data (not accuracy). Precision, Recall, F-score are ok. Area under the PR curve (AUPR) is used in the provided code, so please report it and we will use it to determine which team obtained the best results. There is an evaluation script provided. Please include in your submission (zip file) a file with your output for the test data.

You can experiment with several available trained models (at least 5) and implement/train your own (at least one). More information about the task and ten prepared models is published [here](https://arxiv.org/abs/2103.06268). If you prefer to not use the provided code, please experiment with 3 models of your own (or three versions of your model). Please explain in the report the results of different models and what was the highest AUPR score and for what model it was.

# Part 2: Training sentence similarity models  [50 points]

Use the dataset from the Semeval 2016-Task1 Semantic Textual Similarity (STS), as in Assignment 1. Use any of the training data provided there to try to improve your results from your Assignment 1. You can use the best pre-trained model from you Assignment 1, or any model of your choice. Then fine-tune it (train it) for the task of calculating sentence similarity.

Use the test data [STS Core (English Monolingual subtask) - test data with gold labels](http://alt.qcri.org/semeval2016/task1/data/uploads/sts2016-english-with-gs-v1.0.zip) for the final evaluation of your trained system.

Read more about the task at [https://alt.qcri.org/semeval2016/task1/#](https://alt.qcri.org/semeval2016/task1/)

You can write your code or reuse existing code or tools (as long as you acknowledge them and extend them if needed).

Include in your report details about what sentence embedding you used (with what parameters and what mechanism is behind them) and explain how you did the fine turning using the training data.

Include in your submission your system output (the scores for each pair of sentences) for your best fine-tuned model for all the files in the test data.

As in Assignment 1, the evaluation score to report is the Pearson correlation between the score obtained by your model and the expected solution, for the test data. The expected solution scores are numeric values between and 5 (from low similarity to high similarity). There is script that you can use to compute the correlations. Include in your submission (zip file) a file with your predictions for the test data).

**Sentence Similarity**

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Pearson correlation before fine-tuning** | **Pearson correlation after fine-tuning** |
| STS test data |  |  |

# Submission instructions

1. Prepare a report with written answers for the two parts. Summarize the methods that you implemented, any additional resources that you used, present the results that you obtained, and discuss them. Write the names and student numbers of the team members at the beginning of the report and explain how the tasks were divided.

2. Submit your report, results file (results files for part 1 and for part 2), and code electronically through the Virtual Campus or by email. Archive everything in a .zip file. Do not include the initial data files or any external tools or word/sentence embedding pre-trained models. Include a readme file explaining how to run each program that you implemented and how to use any external resources.