

Topics in Data Analytics S23

Assignment 3

June 27, 2023

1 Submission

You need to submit your analysis as an executable Python Jupyter Notebook file, to onq. This Jupyter Notebook file should be named “1234-Assn3.ipynb”, where 1234 stands for the last 4 digits of your Queen’s student ID.

You should use Markdown cells in Jupyter Notebook to describe the motivation of questions and findings. Make sure the TA can find your answer to each question easily.

An “I uploaded the wrong file” excuse will result in a mark of zero.

2 Background

We will continue the practices of using pre-trained models on NLP tasks. Specifically, we would like to study a data set named ContractNLI. ContractNLI is a dataset for document-level natural language inference (NLI) on contracts whose goal is to automate/support a time-consuming procedure of contract review. In this task, a system is given a set of hypotheses and a contract, and it is asked to classify whether each hypothesis is entailed by, contradicting to or not mentioned by (neutral to) the contract as well as identifying evidence for the decision as spans in the contract. The complete description of this dataset could be found here: <https://stanfordnlp.github.io/contract-nli/>.

Natural Language Inference (NLI) is the task of determining whether the given “hypothesis” logically follows from the “premise”. This is a popular NLP task which also considered as a pre-training task in BERT. More SOTA approaches on this task can be found here: <https://paperswithcode.com/task/natural-language-inference>

Q1 (30 points) Our goal is to perform natural language inference on the given task. The first task is to check the statistics of the dataset and design how to generate new files from the raw dataset to support your NLI task training, validation and testing. You should have a textual cell in your notebook to summarize your approach (how you prepare the train/test/validation set).

Q2 (40 points) Perform NLI using two transformer based approaches. You can check the SOTA approaches from [paperwithcode](https://paperswithcode.com) and then see if hugging face has implemented the pre-trained model. If you pick a pre-trained model such as DeBerta (https://huggingface.co/docs/transformers/model_doc/deberta), make sure you fine-tuned the model on the training dataset. You are free to use public available model/code to implement your solution. Report the performance of two models on the dataset.

Q3 (20 points) Analyze the performance of your two models and perform error analysis. You should have a textual cell explaining how you design the methodology to perform error analysis with a goal of understanding what are the potential reasons behind the errors.

Q4 (10 points) You do not need to write any code in this question. Based on Q3 and external readings, propose two ideas to improve the approaches (i.e., directly fine-tune pre-trained models) used in Q2. Explain your answers in a textual cell.