Assignment 2
NLP 203: Natural Language Processing III

University of California Santa Cruz

All assignments are to be completed individually. You may discuss with the TAs and the instructor, but you may not receive help from anyone else.

Instructions. Submit a zip or tgz file containing your writeup, and output and code to Canvas. Although not required, we encourage you to use LaTeXto typeset your writeup.

Problem 1: Robustness in QA [15 points]

In this question you will test the robustness of the RoBERTa model for question answering. There is a demo of the model here for you to use:

`https://demo.allennlp.org/reading-comprehension/transformer-qa`

This model has been fine-tuned on the SquAD 2.0 dataset, which contains questions about passages from English Wikipedia.

1. Attempt to make the model give a incorrect answers on in-domain passages. For this question, the passages should be in-domain passages "in the wild," i.e. ones that you have found but haven't edited. You can use suggested passages from the demo. Try to give three examples of questions that give incorrect answers, possibly on different passages. How difficult is it to fool RoBERTa? Remember that the model should also correctly know when the answer isn't present in the passage.
2. Repeat part 1, but this time you may create or edit the passages as well as the questions. Using this approach, how hard is it to fool RoBERTa?
3. Repeat part 1, but this time use passages in the wild that are out of domain. Examples include: scientific articles, medical literature, social media, chat or Reddit posts, movie dialogs, Shakespeare, poetry, etc. How difficult is it to fool RoBERTa on out of domain data?

Problem 2: Domain Adaptation for QA [30 points]

The purpose of this homework is to train a Question Answering model. Transformer-based pre-trained models have reasonably good performance when fine-tuned on domain-specific data. However this performance greatly decreases on data from other domains. For example, top models trained on SQuAD don't do well on out-of-domain event-based questions derived from the ACE dataset [3]. This assignment is designed to study and improve upon this performance gap.

You will need to fine-tune a QA model on in-domain and out-of-domain data and report performance on the Covid-QA [1] dataset.

Data format and Evaluation

The Covid-QA dataset contains 2,019 question answer pairs annotated by biomedical experts from scientific articles related to COVID-19, in a format similar to SQuAD. The dataset contains 147 articles and all questions are answerable. We have already created train, dev and test splits from this dataset as shown in table 1, which are downloadable here[1].
Also provided is the evaluation script from SQuAD v2.0, named evaluate.py, which can automatically computes the performance of your QA models in terms of EM and F1. To use this script, save the predictions from your QA model in the form of a dictionary, with the dictionary key being the question ID, and the value being the answer text, and dump it into a json file. This is your $pred_file, and the $gold_file is covid-qa-dev.json or covid-qa-test.json, based on the split you are evaluating. The outputs (EM and F1) are stored in $eval_file, which is also a json file.

To run evaluate.py, use the command

```
python3 evaluate.py $gold_file $pred_file --out-file $eval_file
```

split	# articles
train	104
dev	21
test	22

Table 1: Statistics for the Covid-QA dataset

Part 1 [15 points]

Use RoBERTa finetuned on SQuAD as the baseline model to find out out-of-domain performance of the model on Covid-QA. Run the roberta-base-squad2 model from Hugging Face[2] and report the performance on the dev and test splits in terms of EM and F1.

Part 2

Next, write code to further fine-tune your model from Part 1 on the train split of Covid-QA and report results on the dev and test splits in terms of EM and F1. You may make use of the built-in Trainer class in the Huggingface Transformers library for training. Refer to this tutorial here for more details on how to finetune a pretrained model[3]. This may take a very long time to train, so you do not need to actually do the fine-tuning. Proceed to part 3 for a faster training method that you will use.

[1] https://drive.google.com/drive/folders/1QbZYiTOm4pRh41UG1UvImoGMWuXFIWNH?usp=sharing

[2] https://huggingface.co/deepset/roberta-base-squad2

[3] https://huggingface.co/docs/transformers/tasks/question_answering

Part 3 [15 points]

An alternative to finetuning the full pretrained model is to use an **Adapter**, which introduces and updates only a tiny set of newly introduced parameters at every transformer layer. Learn more about Adapter here[4] Train an adapter-transformer ([2]) model starting from roberta-base-squad2 in Part 1 on the train split of Covid-QA and report the performance on the dev and test splits in terms of EM and F1. You will find information about how to train an adapter for QA here [5]

Deliverables In the writeup, be sure to answer the questions from problem 1, and for problem 2, fully describe your models, experimental procedure and hyperparameters used, along with EM and F1 scores on both the dev and test splits for each model. Provide graphs, tables, charts or other summary evidence to support any claims you make. Submit all your code, trained models, and the predictions on the dev and test splits for each model in the zipfile containing your writeup.

References

[1] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. Covid-qa: a question answering dataset for covid-19. 2020.

[2] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

[3] Elior Sulem, Jamaal Hay, and Dan Roth. Do we know what we don't know? studying unanswerable questions beyond squad 2.0. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4543–4548, 2021.

[4] https://adapterhub.ml/blog/2020/11/adapting-transformers-with-adapterhub/

[5] https://github.com/adapter-hub/adapter-transformers/tree/master/examples/pytorch/question-answering