Assignment 2

Credits: Andrea Galassi, Federico Ruggeri, Paolo Torroni

Keywords: Transformers, Question Answering, CoQA

Deadlines

- **December 11**, 2022: deadline for having assignments graded by January 11, 2023
- **January 11**, 2023: deadline for half-point speed bonus per assignment
- **After January 11**, 2023: assignments are still accepted, but there will be no speed bonus

Overview

Problem

Question Answering (QA) on CoQA dataset: a conversational QA dataset.

Task

Given a question Q, a text passage P, the task is to generate the answer A.
$\rightarrow A$ can be: (i) a free-form text or (ii) unanswerable;

Note: an question Q can refer to previous dialogue turns.
$\rightarrow$ dialogue history H may be a valuable input to provide the correct answer A.

Models

We are going to experiment with transformer-based models to define the following models:

1. $A = f_\theta(Q, P)$
2. $A = f_\theta(Q, P, H)$

where f_θ is the transformer-based model we have to define with θ parameters.

The CoQA dataset

Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80. Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well. Jessica had . . .

Q_1: Who had a birthday?
A_1: Jessica
R_1: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.

Q_2: How old would she be?
A_2: 80
R_2: she was turning 80

Q_3: Did she plan to have any visitors?
A_3: Yes
R_3: Her granddaughter Annie was coming over

Q_4: How many?
A_4: Three
R_4: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

Q_5: Who?
A_5: Annie, Melanie and Josh
R_5: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

For detailed information about the dataset, feel free to check the original paper.

Rationales

Each QA pair is paired with a rationale R: it is a text span extracted from the given text passage P.
$\rightarrow R$ is not a requested output, but it can be used as an additional information at training time!

Dataset Statistics

- **127k** QA pairs.
- **8k** conversations.
- **7** diverse domains: Children's Stories, Literature, Mid/High School Exams, News, Wikipedia, Reddit, Science.
- Average conversation length: **15 turns** (i.e., QA pairs).
- Almost **half** of CoQA questions refer back to **conversational history**.
- Only **train** and **validation** sets are available.

Dataset snippet

The dataset is stored in JSON format. Each dialogue is represented as follows:

```
{
    "source": "mctest",
    "id": "3dr23u6we5exclen4th8uq9rb42tel",
    "filename": "mc160.test.41",
    "story": "Once upon a time, in a barn near a farm house, there lived a little white
kitten named Cotton.
    Cotton lived high up in a nice warm place above the barn where all of the farmer's
horses slept. [...]" % <-- $P$
    "questions": [
        {
            "input_text": "What color was Cotton?",   % <-- $Q_1$
            "turn_id": 1
        },
        {
            "input_text": "Where did she live?",
            "turn_id": 2
        },
        [...]
    ],
    "answers": [
        {
            "span_start": 59,   % <-- $R_1$ start index
            "spand_end": 93,    % <-- $R_1$ end index
            "span_text": "a little white kitten named Cotton",   % <-- $R_1$
            "input_text" "white",   % <-- $A_1$
            "turn_id": 1
```

```
        },
        [...]
    ]
}
```

Simplifications

Each dialogue also contains an additional field `additional_answers`. For simplicity, we **ignore** this field and only consider one groundtruth answer A and text rationale R.

CoQA only contains 1.3% of unanswerable questions. For simplicity, we **ignore** those QA pairs.

[Task 1] Remove unaswerable QA pairs

Write your own script to remove unaswerable QA pairs from both train and validation sets.

Dataset Download

```
import os
import urllib.request
from tqdm import tqdm

class DownloadProgressBar(tqdm):
    def update_to(self, b=1, bsize=1, tsize=None):
        if tsize is not None:
            self.total = tsize
        self.update(b * bsize - self.n)

def download_url(url, output_path):
    with DownloadProgressBar(unit='B', unit_scale=True,
                             miniters=1, desc=url.split('/')[-1]) as t:
        urllib.request.urlretrieve(url, filename=output_path, reporthook=t.update_to)

def download_data(data_path, url_path, suffix):
    if not os.path.exists(data_path):
        os.makedirs(data_path)
```



```
    if not os.path.exists(data_path):
        print(f"Downloading CoQA {suffix} data split... (it may take a while)")
        download_url(url=url_path, output_path=data_path)
        print("Download completed!")
```

```
# Train data
train_url = "https://nlp.stanford.edu/data/coqa/coqa-train-v1.0.json"
download_data(data_path='coqa', url_path=train_url, suffix='train')

# Test data
```

```
test_url = "https://nlp.stanford.edu/data/coqa/coqa-dev-v1.0.json"
download_data(data_path='coqa', url_path=test_url, suffix='test')  # <-- Why test? See next
```

Data Inspection

Spend some time in checking accurately the dataset format and how to retrieve the tasks' inputs and outputs!

[Task 2] Train, Validation and Test splits

CoQA only provides a train and validation set since the test set is hidden for evaluation purposes.

We'll consider the provided validation set as a test set.
$\rightarrow$ Write your own script to:

- Split the train data in train and validation splits (80% train and 20% val)
- Perform splits such that a dialogue appears in one split only! (i.e., split at dialogue level)
- Perform splitting using the following seed for reproducibility: 42

Reproducibility Memo

Check back tutorial 2 on how to fix a specific random seed for reproducibility!

[Task 3] Model definition

Write your own script to define the following transformer-based models from huggingface.

- [M1] DistilRoBERTa (distilberta-base)
- [M2] BERTTiny (bert-tiny)

Note: Remember to install the `transformers` python package!

Note: We consider small transformer models for computational reasons!

[Task 4] Question generation with text passage P and question Q

We want to define $f_\theta(P, Q)$.

Write your own script to implement f_θ for each model: M1 and M2.

Formulation

Consider a dialogue on text passage P.

For each question Q_i at dialogue turn i, your model should take P and Q_i and generate A_i.

[Task 5] Question generation with text passage P, question Q and dialogue history H

We want to define $f_\theta(P, Q, H)$. Write your own script to implement f_θ for each model: M1 and M2.

Formulation

Consider a dialogue on text passage P.

For each question Q_i at dialogue turn i, your model should take P, Q_i, and $H = \{Q_0, A_0, \ldots, Q_{i-1}, A_{i-1}\}$ to generate A_i.

[Task 6] Train and evaluate $f_\theta(P, Q)$ and $f_\theta(P, Q, H)$

Write your own script to train and evaluate your $f_\theta(P, Q)$ and $f_\theta(P, Q, H)$ models.

Instructions

- Perform multiple train/evaluation seed runs: [42, 2022, 1337].[1]
- Evaluate your models with the following metrics: SQUAD F1-score.[2]
- Fine-tune each transformer-based models for **3 epochs**.
- Report evaluation SQUAD F1-score computed on the validation and test sets.

[1] Remember what we said about code reproducibility in Tutorial 2!

[2] You can use `allennlp` python package for a quick implementation of SQUAD F1-score: `from allennlp_models.rc.tools import squad`.

[Task 7] Error Analysis

Perform a simple and short error analysis as follows:

- Group dialogues by `source` and report the worst 5 model errors for each source (w.r.t. SQUAD F1-score).
- Inspect observed results and try to provide some comments (e.g., do the models make errors when faced with a particular question type?)[1]

[1] Check the paper for some valuable information about question/answer types (e.g., Table 6, Table 8)

Assignment Evaluation

The following assignment points will be awarded for each task as follows:

- Task 1, Pre-processing → 0.5 points.
- Task 2, Dataset Splitting → 0.5 points.
- Task 3 and 4, Models Definition → 1.0 points.
- Task 5 and 6, Models Training and Evaluation → 2.0 points.
- Task 7, Analysis → 1.0 points.
- Report → 1.0 points.

Total = 6 points

We may award an additional 0.5 points for outstanding submissions.

Speed Bonus = 0.5 extra points

Report

We apply the rules described in Assignment 1 regarding the report.

- Write a clear and concise report following the given overleaf template (**max 2 pages**).
- Report validation and test results in a table.[1]
- **Avoid reporting** code snippets or copy-paste terminal outputs → **Provide a clean schema** of what you want to show

Comments and Organization

Remember to properly comment your code (it is not necessary to comment each single line) and don't forget to describe your work!

Structure your code for readability and maintenance. If you work with Colab, use sections.

This allows you to build clean and modular code, as well as easy to read and to debug (notebooks can be quite tricky time to time).

FAQ (READ THIS!)

Question: Does Task 3 also include data tokenization and conversion step?

Answer: Yes! These steps are usually straightforward since `transformers` also offers a specific tokenizer for each model.

Example:

```
tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
encoded_text = tokenizer(text)
%% Alternatively
inputs = tokenizer.tokenize(text, add_special_tokens=True, max_length=min(max_length,
```

```
512))
input_ids, attention_mask = inputs['input_ids'], inputs['attention_mask']
```

Suggestion: Hugginface's documentation is full of tutorials and user-friendly APIs.

Question: I'm hitting **out of memory error** when training my models, do you have any suggestions?

Answer: Here are some common workarounds:

1. Try decreasing the mini-batch size
2. Try applying a different padding strategy (if you are applying padding): e.g. use quantiles instead of maximum sequence length

---

Contact

For any doubt, question, issue or help, you can always contact us at the following email addresses:

Teaching Assistants:

- Andrea Galassi → a.galassi@unibo.it
- Federico Ruggeri → federico.ruggeri6@unibo.it

Professor:

- Paolo Torroni → p.torroni@unibo.it

The End!

Questions?