Extractive Question Answering with SQuAD v1.1

Report

Dimitris Mavroeidis
13/06/2022

Task

The task is described as follows:

Given a text (context) and a question, find its answer within the provided text. It is regarded a reading comprehension task.

Dataset

The dataset comprises SQuAD v1.1. This dataset contains questions, answers and contexts. Contexts are the documents from which the answers should be extracted. For each context, there are several question/answer pairs which were compiled by human annotators. There are two versions of SQuAD, v1.1 and v2.0. The two versions are mostly the same, with v2.0 also containing questions that cannot be answered using the provided context. We use the full SQuAD v1.1 dataset.

Tools

Deep learning library

A tech company usually needs to create and evaluate prototypes and then scale them to production requirements. PyTorch, while very flexible during prototyping, does not perform very well at scale. That is the reason I implemented the solution using PyTorch Lightning[1], which retains the flexibility of PyTorch while optimized for production use.

Infrastructure

All experiments were ran in Google colab and Kaggle notebooks. Google colab has low-end GPUs with a maximum of 12GB of RAM on the free plan, while Kaggle offers v100 GPUs with 16GB RAM on the same plan. Thus, we ran the prototypes -on the development set only- on Google colab and experimented with the full dataset on the Kaggle notebooks.

Model

We chose to use BERT as our underlying pre-trained model and fine-tune it. We created an additional layer adding some of the characteristics that are particular to questions answering, such

1 https://www.pytorchlightning.ai/

as “start positions” and “end positions”. We also created a custom data class and processes to preprocess and format the SQuAD dataset.

Experiments

Due to the limited time, we experimented only with two setups:

1. Reduced-size dataset
2. Full dataset

Below, you can find a table figures of how the model performed using these two setups. We report both exact match (EM) and F1 score.

Dataset size	Exact Match (EM)	F1
Small	68.5	72.4
Complete	73.3	78.5

A copy of the model binary is available upon request for verifying the results.

Improvements

The model can be improved in several ways. Improvements can happen at the level of Data and Model.

Data

Human annotators and validators make mistakes. Since the “Questions” and “Answers” portion of the dataset was created by humans, they can contain mistakes. A quick way to eliminate some of them is to run an auto-corrector on the “Questions” to fix spelling errors. By quickly going through the dataset, we found a couple of spelling errors. In other datasets, especially those that were created semi-automatically or fully-automatically, major improvements can be hugely improved. It is important to remember that models are as good as the data we feed them.

Model

One major way is tune the hyper-parameters of the model. Trying different values for each hyper-parameter and accessing the results can improve the performance of the model on the particular dataset.

Resources

We used the following resources and code repositories as both a source of code (copied/adapted major parts) and inspiration to create our own project. Below, they are presented in order of importance:

- https://github.com/kswamy15/NLP_Tasks_PyLightning/blob/main/Bert_NLP_Q%26/A_Pytorch_Squad_v3.ipynb
- https://huggingface.co/transformers/v3.2.0/custom_datasets.html
- https://arxiv.org/abs/1909.04925
- https://towardsdatascience.com/how-to-fine-tune-a-q-a-transformer-86f91ec92997
- https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/LanguageModeling/BERT
- https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15843151.pdf
- https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15792151.pdf
- https://github.com/mirbostani/QA-KD-AL
- https://huggingface.co/docs/transformers/model_doc/distilbert#transformers.DistilBertForQuestionAnswering

Code

The code is available in the form of a jupyter notebook at the following location.

https://github.com/dmavroeidis/qa_test