NLP Assignment 2
Question Answering

Description

A general question-answering system would be able to answer a question about any domain, based on the world knowledge. This system would consist of three stages. A given question is read and reformulated in the first stage, followed by information retrieval via a search engine. An answer is then synthesized based on the query and a set of retrieved documents.

The SearchQA is a retrieval-based question-answer task challenge. This task provides questions and answers from Jeopardy! (An American TV show), as well as snippets of web pages from Google search. You need to use the data to train a model that provides answers to the given questions and search result snippets. The following is an example of instance.

———

Question: born poland , first prime minister israel

Answer: ben gurion

Snippets:

- <s> shimon peres' family 5 fast facts need know heavy com sep 27 , 2016 six decades public life click learn former israeli prime minister 's family born august 21 , 1923 vishnyeva , poland time part belarus according academy </s>

- <s> 16 day history read happened today short display day gr daily infobits website , favorite first polish pope reigned pope 26 years prime minister israel , david **ben gurion** , born plonsk , poland </s>

- <s> shimon peres wikipedia shimon peres israeli statesman ninth president israel , serving 2007 2014 peres served twice prime minister israel twice interim prime minister , peres told rabbi menachem mendel schneerson born result blessing parents </s>

- <s> israel palestine google books result </s>

———

It is worth noting that not all the search result snippets in this task contain answers. Therefore, you need to consider how to employ a model to pick out the snippets that can answer the question and retrieve the most appropriate answer spans. Each question has approximately 40 corresponding snippets. There is no question that cannot be answered. All text in the data is in lowercase.

The final system can combine multiple models. We recommend that you use the pre-trained language models. In addition to deep learning models, you can consider traditional text retrieval methods, such as TF-IDF, BM25.

For this assignment, you are offered ***NLP-2-Dataset.zip*** containing training data, validation data and testing data(no answers). You need to use the training data for training model, the validation data for tuning hyper-parameters to improve performance of the model, the testing data for final evaluation. When the training process is done, you use final system containing the well-trained model to predict answer text as submission. We will evaluate your submissions with ground truth.

Requirements

- **Python programming language only.**
- **You can use any machine learning library (** ***TensorFlow***, ***Keras***, ***Pytorch***, etc.**).**
- You can refer to the snippets of code from open-source projects, tutorials, blogs, etc. Do not clone the entire projects directly. **Try to implement a system by yourself.**
- Libraries that provide pre-trained language models such as ***HuggingFace(Transformers)*** are allowed to be used. However, you should process the data, write the output layer of the model and design training process yourself. Therefore, libraries that package the entire process completely such as ***simpletransformers*** and ***sentence-transformers*** are not allowed to be used. Also, you should not use models that have been fine-tuned on the SearchQA dataset.

Grading

We follow the definition of metrics introduced at SQuAD 1.1 to measure the performance of the systems in terms of Exact Match and F1-score, where:

"The F1 metric, which we have already referenced many times before, is the measure of average overlap between a model's prediction and ground-truth answer span. Formally, F1 = (2·Precision·Recall)/(Precision+Recall) where precision is defined as the ratio of correctly predicted words in the answer span to the total number of predicted answer span words. Recall, meanwhile, is the ratio of correctly predicted answer span words to total number of words in answer span. The exact match metric is a binary value that takes on a value of 1 if the predicted answer and true answer are exactly equal (not counting punctuation and articles) and zero otherwise."

Models are evaluated based on Exact Match and F1-score on the testing data.

- **Completed Source Code (20%)**
- **EM & F1-Score (40%)**
- **Report (40%)**

Submission Rule

Please pack up your **source code**, **test-submit.txt** and **report** into a .zip file named as *<studentid>_hw2.zip*, and upload it to the ee-class system. The following files must be included in your submission:

- one project source code *<studentid>.py* (Please consolidate all your code into one file when you submit.)
- *test-submit.txt*: test-submit.txt with questions and predicted answers only **(Please do not include search results snippets)**

- *requirements.txt*: installed libraries in your Python environment. To create a requirements file, use: *pip freeze > requirements.txt* or *conda list -e > requirements.txt*
- *<studentid>_report.pdf*: assignment report. You can write your data preprocessing, model architecture, training process, hyperparameters, evaluation scores or anything else you want.

Deadline: 2022/11/22 23:59

Key words: Extractive Question Answering, Text Retrieval, Pre-training Language Model