Meeting Minutes

# March – April

Read literature to understand transformers, ODQA, and synthetic data.

These are a sample of the key readings to read and understand the topic:

1. [Recent advances in Open-Domain QA](https://lilianweng.github.io/lil-log/2020/10/29/open-domain-question-answering.html#fast-maximum-inner-product-search-mips).
2. [RAG from FacebookAI - We will build our AI model on top of this](https://ai.facebook.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/).
3. [End-to-End QA on COVID-19: Domain Adaptation with Synthetic Training - This is our benchmark paper](https://arxiv.org/pdf/2012.01414.pdf).
4. [Training Question Answering Models From Synthetic Data](https://www.aclweb.org/anthology/2020.emnlp-main.468.pdf)
5. …

Created a document to take notes of our understanding of the topics as well as writing down any questions we had for Shamane.

Note taking document: <https://docs.google.com/document/d/1Ucz2T_uHKCq0CkPM3d0Hybzw3IpjjNLLM-emqltJ7uY/edit>

Questions for Shamane: <https://docs.google.com/document/d/1HonQcDevIugc_4J5TAIODYxjKRWEpablCdrFhDIVcxI/edit>

We also discussed Shamane’s Huggingface PR for full end-2-end fine-tuning of RAG <https://github.com/huggingface/transformers/pull/10410>

# April – May

April 10th Meeting: To do before the next week:

1. Going through final theory section of RAG and understanding BERT, BART and DPR.
2. Getting familiar with cord-19 dateset.
3. Find a way to filter the synthetic QA dataset.
4. Round trip consistency method - Find a machine comprehension model from Huggiface (Model that answer a question given a passage) and create a pipeline.

Started worked on the round-trip consistency method to filter data.

After we created the first version of the filtration method, Shamane asked us to make the code cleaner, and make a function that will use different models for filtering and compare the performance between models.

Look at adding multiple criteria for the filtration method to assess whether to keep an answer or not.

# May – June

Had several meetings on round trip consistency to discuss our code.

We cleaned several functions, wrote scripts and readmes.

While we were completing the round-trip consistency method to filter data, we begain work on generating simple synthetic data using a simple approach. Where the sentence of a passage is used as the answer to generate the question. We used the CORD-19 data from <https://github.com/allenai/cord19> as our knowledge base.

# June – July

Final Part of Project Meeting Minutes (RAG pipeline)

* We discussed the final part of the project.
* This part can be more of an exploration.
* We want to test how our end-to-end model's retriever work against the DPR model trained on domain specific data.
* In order to train a DPR we need a passage and a question.

Using synthetic data, generate hard negatives with the BM25 Kaggle notebook. In that notebook you can see we need to have something called **hard negatives**(please read the [DPR](https://arxiv.org/pdf/2004.04906.pdf) paper | section 3.2 ). Simply hard negatives are similar passages to a question that **doesn't consist of the answer**. In order to select hard negative passages we can use a lexical matching engine like BM25.I found this amazing kaggle BM25  [notebook](https://www.kaggle.com/ideanlabib/bm25-search-query-similarity-ranking)  you just need to go through it and set it up on your computers.

Put the hard negatives into a format the DPR can understand. Start fine-tuning the DPR by separating processed synthetic data into training and test sets.

Clean up the pipeline so that when we come up with new filtration techniques we can use the pipeline to evaluate the performance of the new filtration technique.

We want to test the end2end model's retriever against DPR trained on domain specific data

DPR fine-tuning

1. We need passage and question i.e. the synthetic question and its passage
2. We need something called hard negatives
3. We need to work on a data-set building process for training the DPR
4. Figure out a way to build a data set that can be used to train the DPR on domain specific data

We consider three different types of negatives:

* 1. Random: any random passage from the corpus;
  2. BM25: top passages returned by BM25 which don’t contain the answer but match most question tokens;
  3. Gold: positive passages paired with other questions which appear in the training set. We will discuss the impact of different types of negative passages and training schemes in Section 5.2.

Our best model uses gold passages from the same mini-batch and one BM25 negative passage. In particular, re-using gold passages from the same batch as negatives can make the computation efficient while achieving great performance. We discuss this approach below.

Approach

1. Use BM25 to get a list of passages from synthetic question. The corpus of passages should not include the context for that synthetic question. We should get top-k passages that doesn't contain the answer? This is a bit naive so we can do more ...
2. For each of the top-k passages selected using BM25 we want to generate an answer
3. Do similarity to synthetic answer
4. Further subset the top-k passages based on the lowest similarity score
5. Now we have, for each synthetic QA pair, we have: $\{q\_i,a\_i|p\_i^+,p\_{i,m}^-\}$ where m is the number of hard negative passages selected via the BM25

Shamane has asked us to train BART independently using a summarisation script from Hugginface. This BART model will be added to our pipeline.

# July – August

Shamane is finding that it’s difficult to train RAG with only synthetic data since it learns extremely slow. However, adding synthetic data with reconstruction signals often works better. The main problem seems to be the fact that the model has to find the context before it can generate an answer.

Shamane has come back saying that using only synthetic data even after filtering is still hard to train when the knowledge base is large. The retriever part is difficult, not the answer generation part. We are finding the synthetic data is not factual enough e.g. some questions are saying what does figure 1 say, but different documents could have a figure 1 but for different answers.

Looks like the next steps will be to create better synthetic question-answer data, since the current synthetic QA is good for a reading comprehension model, but not so good for the retriever.

Have a look at the Pubmed QA paper. Add the title to the question so that the model can try using the question + title to find the answer. Nothing that this won’t be useful in teh real world since if we had the title then we don’t need anything else. We are only adding the title to help train the retriever component of RAG.

Presentation feedback from Suranga:

* Why did we use RAG?
* Why did we use synthetic data?
* Make sure to simplify things so that someone without technical knowledge can understand.
* Make the presentation more of a story talk more about what we did rather than literature review.
* Talk about what has been done before, and then talk about what we did, why it’s important, and the final findings.
* Make sure these things are emphasised in the report. The introduction and motivation are important to be clear about.
* Have a hypothesis, brainstorm and use a systematic process to complete this.

Next steps of our project is to introduce RAG friendly data to help the QA framework.

Have a look at the following link:

<https://parl.ai/projects/blenderbot2/?fbclid=IwAR2g0I2spRNfGckNZzFYHHDKoIOqfGTExJTxP7jUoPmcswazJGFBvNia0qY>

Think of potential ideals for new signals. We came up with a summarisation signal.

# August – October

Another potential signal is to combine DPR and BM25. The thinking here is that when the dataset is small there is not much of a lexical match, so in these scenarios DPR is not very good. This is where BM25 can come in and boost performance.

Method will take each question, and pass it into BM25. The result will be appended to the question, we can take the top 5 contexts from BM25 if we want to. This can then be passed into the DPR, at which the DPR will retrieve some more documents. This means the BART model will have the BM25 context and the retrieved documents to use.

Overall, the main final contributions

* Synthetic data generation using summarisation
* BM25 question enhancement

Dense Vector Round Trip Consistency

* The round\_trip\_consistency folder contains two files.
* The Round\_Trip\_Consistency\_RAG.ipynb file contains the code we used to apply our filtration method to our abstractive summarisation dataset.
* The filter\_synthetic.py file contains the code we used to run round trip consistency and cosine similarity to filter synthetic question-answer pairs.

RAG Pipeline Implementation

* The RAG\_pipeline file contains two files.
* The BART\_Fine\_Tuning.ipynb file contains the code used to fine-tune BART independently.
* The BM25\_Hard\_Negative\_Selection\_+\_DPR\_Fine\_Tuning.ipynb file contains the code which creates the dataset required to fine-tune the DPR. This file also contains the code for the RAG pipeline.

Summarising the knowledge base

* This file contains a single file.
* The Summarising Knowledge Base.ipynb file contains the code used to summarise the knowledge base. This is the method where we explore summarising a knowledge base before giving it the retriever to use. It includes both approaches e.g., summarising every passage, and summarising just the abstract.

Abstractive Summarisation for Unambiguous Synthetic QA Generation

* This compendium (synthetic\_data\_gen) contains our synthetic QA data generated using our pipeline.
* The data folder contains all our final synthetic data that we generated. NOTE: we have a deprecated data folder for our old data that we generated on Google Colab.
* SyntheticQAGenerator.py is the main Python script for generating our synthetic data. To use this we run the run\_synthetic\_qa\_generator.sh file in bash
* Requirements.txt contains all the Python packages required to run our algorithm.
* Our ConcatenateData.py and PrepareRAGData.py finalises our output data into a format suitable for training RAG.

Integrating Lexical Matching results into synthetic data/Question Enhancement

* This compendium contains a data folder which houses our input data and output data for question enhancement.
* The input data contains about 175K synthetic QA pairs (generated by Shamane). This is located within the Q-covid folder. Within this folder, the main file is train.source and val.source as these two files contains the synthetic questions that we use to process.
* The output data is also 175K synthetic QA pairs, but the questions are enhanced using our algorithm. The output data is in the Q-covid-BM25 also in the same format as the input data.
* Inside the Q-covid-val folder we process a validation data set in a similar way.
* Requirements.txt contains all the Python packages required to run our algorithm.
* BM25\_question.py is the main python script file that processes input data and then outputs the data to the output\_data folder. This script is the source code for enhancing the input questions’ context to help improve the DPR retrieve better passages. We used a BM25 lexical model for lexical matching. This compendium is explained in detail in our second contribution of our report, relating to question enhancement/Improving Retriever component by integrating lexical matching results.