Composable NLP Workflows for BERT-based Ranking and QA System

Murali Mohana Krishna Dandu
MS in MLDS, ECE
UC San Diego
mdandu@ucsd.edu

Gaurav Kumar
MS in Computer Science
UC San Diego
gkumar@ucsd.edu

Abstract

There has been a lot of progress towards building NLP models that scale to multiple tasks. However, real-world systems contain multiple components and it is tedious to handle cross-task interaction with different levels of text granularity. In this project, we built an end-to-end Ranking and Question-Answering (QA) system using Forte, a toolkit that makes composable NLP pipelines. We utilized state-of-the-art deep learning models in our pipeline, evaluated the performance on MS-MARCO and Covid-19 datasets, and compare the results of ranking and QA systems with their corresponding benchmark results.

1 Introduction

Building Natural Language Processing (NLP) applications involve multiple data systems and require several NLP components to be stitched together. For example, a QA pipeline will have the following components - query understanding, full-search from the entire document corpus, re-ranking to fine-tune a subset of results, identifying the answer phrase/sentence from the top documents, and finally showing the response in a consumable way to the end user. Such a complete end-to-end system requires multiple NLP techniques and may have different input output formats. To achieve this, we need a composable pipeline with standardized data formats coupled with the power of state-of-the-art NLP models.

The objective of this project is to build and test an end-to-end Ranking and QA system utilizing Forte [1], a toolkit for building composable NLP pipelines. We also wanted to build and test a real-world applicable Covid-19 QA system using our created pipeline.

[0]Our code is open-source and is available at https://github.com/gaurav5590/Doc-Ranker

We have contributed during the project in the following ways:

- We utilized Forte's ecosystem and showcased it's various functionalities
- We built a Ranking and QA system using MS-MARCO Passage Ranking and QA datasets [2] and thoroughly evaluated each sub-component of it
- We built a Covid QA system which extends the previous pipeline and can be used in the real-world with few modifications
- We are contributing to Forte by adding the required missing processors and evaluators in the pipeline
- We provided examples for end-users to build a Retrieval and QA system with high quality code and documentation

2 Forte

Forte was build with one goal in mind - to standardize NLP interfaces. It is part of CASL (Composable Automatic and Scalable Learning) [3] open-source toolkit and provides functionalities to put together various NLP components together. Here are the major building blocks of Forte:

- **Data Structures & Ontology:** Usually NLP applications need different information at different data granularity about the text pieces like named-entity mentions, POS tags, dependency links and some user defined groups. To support this, Forte has three template data types: Span (being, end), Link (parent, child) and Group (members) which are common across most of the NLP tasks. This supports interoperations across different NLP functions in the pipeline.

- **DataPack:** DataPack contains all the information stored in the above datatypes and gets updated with new information as it passes through the pipeline. The elements of DataPack can be accessed at any stage of the pipeline using required granuarity and structure which perfectly supports cross-talk and interoperations

- **Pipeline:** A pipeline is a collection of several re-usable readers, indexers, processors and evaluators. Each component in the pipeline will process the incoming DataPack and updates it's internal ontology structure

A clear understanding of the pipeline and the information flow will be clear through Figure 3 after which will be discussed in the next section.

3 Ranking and QA System

With the abundance of web documents, retrieving relevant and specific information from huge corpuses is crucial for lot of applications. Chatbots, Voice agents, Search engines - all of them use key components of ranking and QA. In light of the latest pandemic and the lack of unified information about Covid, we also wanted to build a retrieval system for Covid QA using CORD-19 dataset [4]. However, since the dataset isn't suitable for evaluating intermediate systems, we first built and test the pipeline on MS-MARCO passage ranking and QA datasets.

3.1 Components of the System

An end-to-end QA system consists of three major components as shown in Figure 1.

- **Full-Ranker:** A system which ranks the entire corpus using a simple and faster algorithm based on keywords. The goal is to ideally have 100% retrieval recall and fetches top-N documents with very less latency. We have used Okapi-BM25 algorithm [5] for the full-ranking and have tested the system with multiple top-N documents

- **Re-Ranker:** The top-N documents fetched from the Full-ranker are ranked again using an advanced model like BERT [6]. Here, the ranking will be fine-tuned using the query-document feature representations learnt by the model. We have used a HuggingFace BERT model [7] trained on MS-MARCO itself where it will provide a score for a query-document combination

- **Question-Answering:** The top-1 document scored by the re-ranker will be used as a context text for MRC-style QA [8]. A specific phrase in the document is selected by the model as the answer span and we took the entire sentence of the document for evaluation and easier consumption. We have used a QA BERT model [9] trained on SQUAD data.

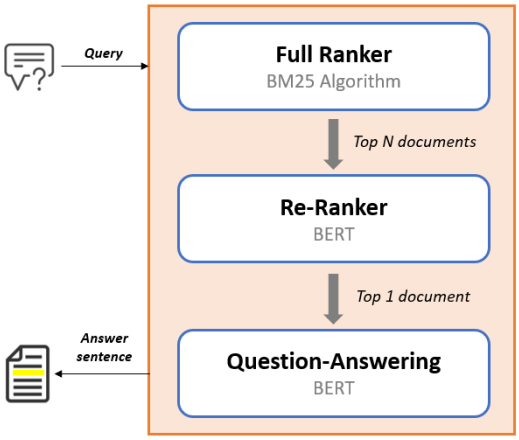

Figure 1: Ranking and QA System

3.2 Complete Pipeline

The code pipeline in Figure 2 consists of a series of Forte processors where the data flows from one to next and the internal MultiPack gets modified accordingly. Here is the code snippet for indexing, ranking, answering and evaluating the results for a given set of user queries.

```
# Elastic Search Indexer Pipeline
pipeline = Pipeline[DataPack]()
pipeline.set_reader(MSMarcoPassageReader())
pipeline.add(ElasticSearchIndexProcessor(), config.create_index)
pipeline.run(dataset_dir)

# Ranking QA Pipeline
pipeline = Pipeline[MultiPack]()
pipeline.set_reader(EvalReader(), config.reader)
pipeline.add(ElasticSearchQueryCreator(), config.query_creator)
pipeline.add(ElasticSearchProcessor(), config.full_ranker)
pipeline.add(MSMarcoEvaluator(), config.fullranker_evaluator)
pipeline.add(BertRerankingProcessor(), config.reranker)
pipeline.add(MSMarcoEvaluator(), config.reranker_evaluator)
pipeline.add(QAProcessor(), config.qa_system)
pipeline.add(QAEvaluator(), config.qa_evaluator)
pipeline.initialize()
```

Figure 2: Forte Pipeline

Figure 3 provides the complete picture on how the data flows through each component of the

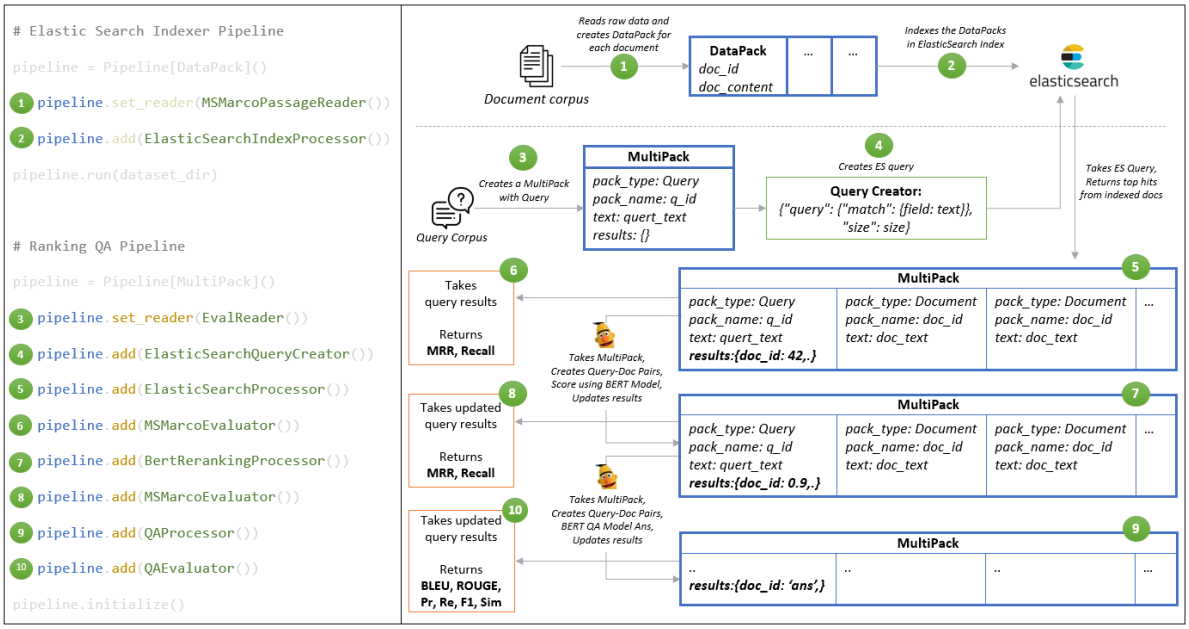

Figure 3: Information flow across the Forte pipeline

pipeline combining readers, ranking and QA processors and evaluators. You can also observe that the use of DataPack and MultiPack makes it easier for information flow and access at each stage.

As you can see, such a pipeline makes it easier to build similar QA systems. For example, by changing the reader to CordReader() with the relevant datasets provided in the config.yml, we have created a Covid QA. We will mention about the ohter engineering changes will discussing the results.

4 Results and Discussion

4.1 Overview

We have used MS-MARCO passage ranking and QA dataset for building the complete system. The dataset contains 8.1 million passages collected from real Bing questions and documents. We have used 1000 to 7000 queries from the development set to test our pipeline and showcase the results. Here are the different evaluation metrics used to test the systems:

- **MRR@N:** Mean-Reciprocal-Rank is the average of multiplicative inverse of the rank of the first correct answer. It measures where exactly the relevant document resides in the top-N ranking
- **Recall@N:** Recall ignores the position and checks whether the relevant document is present in the top-N. This is useful when you have multiple ranking systems and wants to maintain high-recall in the full-ranking search results
- **BLEU-1:** BLEU score measures the precision between reference and predicted text at uni-gram level
- **ROUGE-L:** ROUGE is a recall oriented metric that measures the longest common subsequence between two pieces of text
- **F1:** F1 is calculated using the total uni-gram tokens that are matched
- **Semantic Similarity:** Measures SpaCy's similarity which is computed through cosine of word representations (using GloVe etc.)

System	Metric	Value
Full-Ranking	MRR@10	0.16
	Recall@10	34%
Re-Ranking	MRR@10	0.34
	Recall@10	58%
QA	BLEU-1	0.32
	ROUGE-L	0.32
	F1 (Tokens Match)	0.32
	Semantic Similarity	80%

Table 1: Overall Metrics

Table 1 shows the above metrics for different systems. We have achieved a Full-Ranking MRR@10 of 0.16 and Recall@10 of 34%. Note that we have

		Full-Ranker				Re-ranker			
Re-Ranking Size	**Time per query (s)**	**MRR@10**	**MRR@100**	**Recall@10**	**Recall@100**	**MRR@10**	**MRR@100**	**Recall@10**	**Recall@100**
1	0.48	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
10	0.54	0.16	0.16	0.34	0.34	0.23	0.23	0.34	0.34
50	0.85	0.16	0.17	0.34	0.50	0.28	0.28	0.45	0.50
100	1.24	0.16	0.17	0.34	0.59	0.30	0.30	0.50	0.59
500	4.61	0.16	0.17	0.34	0.59	0.33	0.33	0.56	0.73
1000	**8.81**	**0.16**	**0.17**	**0.34**	**0.59**	**0.34**	**0.35**	**0.58**	**0.77**

Table 2: Full-Ranking and Re-Ranking results for 1000 queries

	QA								
Re-Ranking Size	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**ROUGE-L**	**PRECISION**	**RECALL**	**F1**	**Semantic Sim**
1	0.24	0.15	0.11	0.09	0.22	0.20	0.23	0.21	0.75
10	0.30	0.21	0.17	0.15	0.29	0.26	0.32	0.29	0.79
50	0.31	0.23	0.19	0.17	0.31	0.27	0.34	0.30	0.79
100	0.31	0.24	0.20	0.18	0.32	0.28	0.35	0.31	0.80
500	0.32	0.24	0.21	0.19	0.32	0.29	0.36	0.32	0.80
1000	**0.32**	**0.25**	**0.21**	**0.19**	**0.32**	**0.29**	**0.36**	**0.32**	**0.80**

Table 3: QA results on 1000 queries

used 1000 full-ranking size for the overall metrics. As we go to the re-ranker, we can observed that MRR@10 doubles and Recall@10 almost improves by 80%. These re-ranking metrics are in a very similar range of MS-MARCO re-ranking leaderboard metrics. [10]. Going to the QA, we see that BLEU, ROGUE and F1 are in the range of 0.32. The MS-MARCO QA leaderboard scores [10] are in range of 0.50 and the difference is due to the error propagation from the upstream rankers. For example, we see that the re-ranking recall is 58% and hence the QA metrics will be atleast down by 40% compared to the standalone QA benchmarking tasks. Also, note that we are using the top-1 re-ranked document for QA and increasing the search size for QA will definitely improve the results. MRR of 0.34 implies that we the correct document is on average in top-3 and search for answer in top-3 documents will definitely improve the QA results.

4.2 Sample Results

Table 4 show some sample results of the pipeline with re-ranking size of 1000. We can see that the system is able to handle different types of question structures such as 'what', 'who' as well as phrasal questions like 'tristesse definition' which are commonly used in search engines.

4.3 Re-Ranking Size

Re-ranker is a much costlier operation since it uses bigger BERT architectures. You can see in Table 2 that the retrieval per query increases with the size of documents we are re-ranking. Although we optimized the BERT re-ranking using GPU-batching, we would like to see how the results vary with different search sizes. If you observe the re-ranking results across different re-ranking sizes, there is a consistent improvement till 100-500 size and the returns get diminished after that. At size 500, we achieve very close results of 0.33 and 56% of MRR@10 and Recall@10 compared to 0.34 and 56% for size 1000 with a reduction in retrieval time by 50%.

When we look at the similar Table 3 for QA results, we observe that all the results in terms of BLEU, ROUGE and F1 get saturated after 50-100 size. Combining the results from these two tables, we can say that getting top 100 size for re-ranking would be ideal for this experimental setup.

Q: What is priority pass **A:** Priority Pass is an independent airport lounge access program.
Q: tristesse definition **A:** Tristesse is a French word meaning sadness.
Q: tricuspid atresia definition **A:** Tricuspid atresia is a type of heart disease that is present at birth (congenital heart disease), in which the tricuspid heart valve is missing or abnormally developed.
Q: who proposed the geocentric theory **A:** The geocentric model, also known as the Ptolemaic system, is a theory that was developed by philosophers in Ancient Greece and was named after the philosopher Claudius Ptolemy who lived circa 90 to 168 A.D.

Table 4: QA results for sample queries

4.4 Covid QA

Given the wide-spread pandemic and the public curiosity to understand it better, we wanted to build a COVID QA system by tweaking the above pipeline. We have used the CORD-19 dataset [4] as our doc-

		QA								
Re-Ranking Size	**Time per query (s)**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**ROUGE-L**	**PRECISION**	**RECALL**	**F1**	**Semantic Sim**
100	1.21	0.20	0.15	0.13	0.12	0.22	0.18	0.29	0.22	0.71
1000	6.64	0.20	0.15	0.13	0.12	0.22	0.18	0.29	0.22	0.71

Table 5: QA results for Cord-19 dataset on 2019 queries

ument corpus, and the COVID-QA dataset [11] to test the results. Note that the Covid QA dataset has been created using only 147 documents out of 200K+ CORD-19 documents.

Since the CORD-19 documents are very huge and the pre-trained BERT models under use are restricted by 512 token input size, we pre-processed the documents before indexing the documents. We followed the chunking and striding approach [12] to chunk our documents into smaller texts, which is a common strategy while using such BERT models. We chunked the long text by every 60 tokens keeping an overlapping stride of 15 tokens. With this, the total 212K documents have become around 18M short text passages as an input to our full-search. We have used the deepset.ai Covid model [13] for QA but used the same previous MS-MARCO model for re-ranking.

Additionally, Covid-QA is SQUAD style dataset where the reference answers were short phrases (given the context) and is usually evaluated by F1 and exact match. However, we have used the full-sentence as answers and used other evaluation metrics as well along with F1.

Table 5 shows the results of the QA system where the BLEU-1, ROUGE-L and F1 are in the range of 0.22 with Semantic Similarity being in range of 0.71. We can also see that there is no difference using higher re-ranking size (100 vs 1000), implying that the re-ranker model is not adding much value to this particular task. It is due to the fact that we are using a generic BERT ranker model which will not have technical vocabulary common in CORD dataset. Hence training and fine-tuning a model to this dataset would be an ideal next step to improve the results. [14] shows that they the benchmark F1-score is in the range of 0.25-0.30 showing that our pipeline is not far away to achieve better results.

Table 6 shows some sample queries along with both ground truth and predicted answers from our pipeline. As you can see, it is able to identify the sentence containing the answer.

Q: What is the main cause of HIV-1 infection in children? **G:** Mother-to-child transmission (MTCT) is the main cause of HIV-1 infection in children worldwide. **A:** Mother-to-child transmission (MTCT) is the main cause of HIV-1 infection in children worldwide.
Q: What is the size of bovine coronavirus? **G:** 31 kb **A:** The BCoV is a RNA virus, nonenveloped, diameter of 120 nm, single-stranded (ssRNA) positive-sense, and non-segmented with 27-32 Kb size (ICTV 2015)
Q: How long is the SAIBK gene? **G:** 27,534 nucleotides **A:** The complete genome of the SAIBK strain is 27,534 nucleotides (nt) in length, including the poly(A) tail.
Q: What does the hamster model for HCPS caused by? **G:** by capillary leak that results in pulmonary edema and the production of a pleural effusion with exudative characteristics **A:** the hamster model for HCPS appears to be caused by capillary leak that results in pulmonary edema and the production of a pleural effusion with exudative characteristics.

Table 6: QA results for sample queries of CORD dataset. Q stands for question, G stands for ground truth answer, A stands for predicted answer

5 Conclusion and Future Work

In this project, we build a composable ranking and QA pipeline with comparable results on MS MARCO and Covid-19 QA datasets. The modular nature of our pipeline as depicted in Figure 2 makes it easy to build complex NLP applications easily. In future, we would like to train and fine-tune the deep learning ranking models behind our processors for specific datasets. Also, decreasing the latency of re-ranker while maintaining the accuracy will further help in real-world search applications. Covid-QA ranking performance can be further evaluated by utilizing reference context on top of the existing metrics.

6 Acknowledgements

We would like to thank Professor Dr. Zhiting Hu and Dr. Zhengzhong (Hector) Liu for guiding us throughout the project. We also extend our gratitude towards Petuum Inc. for providing us the computing support needed to run our pipeline on GPU.

References

[1] Forte, https://github.com/asyml/forte.

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

[3] Casl, https://www.casl-project.ai/.

[4] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset, 2020.

[5] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389, 01 2009.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[7] Rodrigo Nogueira and Kyunghyun Cho. Passage reranking with bert, 2020.

[8] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[9] Huggingface qa model, https://huggingface.co/deepset/roberta-base-squad2.

[10] Ms marco document ranking and qa leaderboard, https://microsoft.github.io/msmarco/.

[11] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, July 2020. Association for Computational Linguistics.

[12] Edwin Zhang, Nikhil Gupta, Raphael Tang, Xiao Han, Ronak Pradeep, Kuang Lu, Yue Zhang, Rodrigo Nogueira, Kyunghyun Cho, Hui Fang, and Jimmy Lin. Covidex: Neural ranking models and keyword search infrastructure for the covid-19 open research dataset, 2020.

[13] Roberta covid model, https://huggingface.co/deepset/roberta-base-squad2-covid.

[14] Hillary Ngai, Yoona Park, John Chen, and Mahboobeh Parsapoor. Transformer-based models for question answering on covid19, 2021.