Vigicovid Question Answering system at EPIC-QA

Arantxa Otegi
HiTZ Center - Ixa
UPV/EHU, Spain
arantza.otegi@ehu.eus

Iñaki San Vicente
Elhuyar fundazioa
Spain
i.sanvicenteg@elhuyar.eus

Borja Lozano
NLP & IR Group
UNED, Spain
blozano@lsi.uned.es

Xabier Saralegi
Elhuyar fundazioa
Spain
x.saralegi@elhuyar.eus

Jon Ander Campos
HiTZ Center - Ixa
UPV/EHU, Spain
jonander.campos@ehu.eus

Anselmo Peñas
NLP & IR Group
UNED, Spain
anselmo@lsi.uned.es

Eneko Agirre
HiTZ Center - Ixa
UPV/EHU, Spain
e.agirre@ehu.eus

Abstract

We report here the participation of the VIGICOVID project Question Answering system to the Epidemic Question Answering track (EPIC-QA) Task A (Expert QA). The system receives a set of questions asked by experts about the disease COVID-19 and its causal virus SARS-CoV-2, and provides a ranked list of expert-level answers to each question.

1 Introduction

Many biosanitary researchers around the world are directing their efforts towards the study of COVID-19. This effort generates a large volume of scientific publications and at a speed that makes the effective acquisition of new knowledge difficult. Information Systems are needed to assist biosanitary experts in accessing, consulting and analyzing these publications. This is precisely the general objective of the VIGICOVID project[1], that motivated our participation in the Epidemic Question Answering track (EPIC-QA) Task A (Expert QA).

In this task, systems receive a set of questions asked by experts about the disease COVID-19 and its causal virus SARS-CoV-2, and systems in return must provide a ranked list of expert-level answers to each question, extracted from CORD-19 document collection (Wang et al., 2020).

CORD-19 is a resource of over 400,000 scholarly articles, including over 150,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. The CORD-19 dataset represents the most extensive machine-readable coronavirus literature collection.

In the following sections we describe the VIGICOVID Question Answering system presented to EPIC-QA Task A (Expert QA).

2 System overview

VIGICOVID QA system has an architecture in three steps: Document retrieval; document scanning and answer extraction; and ranking of answers.

2.1 Information Retrieval

Our Information Retrieval module follows three steps: Filtering and indexing, preliminary retrieval, and re-reanking.

1. Filtering and Indexing. A keyword-based filter is applied to select covid related documents. Keywords are different variants of the "COVID-19" term. The indexing has been performed only at the whole document level.

2. Preliminary retrieval. From the collection of full texts of the scientific articles, we obtain a preliminary ranking for the query. We use a language modeling based information retrieval approach (Ponte and Croft, 1998) including pseudo relevance feedback. For that purpose, we used the Indri search engine (Strohman et al., 2005), which combines Bayesian networks with language models. We construct a complex query, assigning different weights to the query, question and narrative (background) fields. Our final configuration uses the following query setting:

[1] http://nlp.uned.es/vigicovid-project

$$Q = 0.8 * (query + question) + 0.2 * background$$

All fields have been tokenized and stemmed using Krovetz stemmer (Krovetz, 1993). We recover up to 5,000 documents per query.

3. Re-ranking. The preliminary ranking obtained in the previous step is re-ranked using a BERT-based relevance classifier, following a strategy similar to the one proposed by (Nogueira and Cho, 2019). For each candidate document given by the preliminary ranking, its abstract and the corresponding query are processed through a BERT-based relevance classifier, which returns a probability of an abstract to be relevant with respect to the given query.

   For this purpose, we fine-tuned the Clinical BERT pretrained model (Alsentzer et al., 2019) on the task of identifying relevant abstracts with respect to queries. The training dataset was compiled using the qrels provided in the TREC-COVID task, and also pseudo qrels generated by using titles and abstracts from the CORD-19 dataset.

   As mentioned, the classifier returns a relevance probability of an abstract with respect to a given query. This probability is linearly combined with the score of the first ranking according to a coefficient k, and the ranking is rearranged based on that new value. For our submissions using neural reranking k was optimized with the EPIC-QA primary round collection ($k = 0.1$). Our Final reranking combination uses the following setting:

$$score_d = k * (RerankerScore_d) + (1 - k) * (IndriScore_d) \quad \text{where} \quad k = 0.1$$

After the re-ranking process we select the top 400 documents to be scanned by the answer extraction module.

2.2 Document Scanning and Answer Extraction

The answer extraction module is based on neural network techniques. More specifically, we have used the SciBERT language representation model, which is a pretrained language model based on BERT, but trained on a large corpus of scientific text, including text from biomedical domain (Beltagy et al., 2019). This model has shown strong performance on several downstream NLP tasks in the scientific domain.

We fined-tuned SciBERT for QA using the following two datasets: SQuAD2.0 (Rajpurkar et al., 2018), which is a reading comprehension dataset widely used in the QA research community, and QuAC (Choi et al., 2018), which is a conversational QA dataset containing a higher rate of non-factoid questions than SQuAD.

Following the usual answer extraction method we used the fined-tuned SciBERT model as a pointer network, which selects an answer start and end index given a question and a context. According to the EPIC-QA guidelines, the answers returned by the QA system must be a sentence or several contiguous sentences. In our case, we select those sentences which contain the answer span delimited by the start and end indexes given by the neural network.

Since the documents are too long to feed the network with them, (they exceed the maximum input sequence length) we follow the sliding window approach where the documents are splitted into overlapping passages. As query we just use the text in the "question" field.

For the maximum sequence length, stride parameters and other parameters we used the default values of the implementation (Wolf et al., 2020).

After scanning the whole document, we keep the 10 most probable answers to the question for each given document. Each of these answers have a score given by the neural network. However, since each candidate document has also a relevance score given by the search engine, we produce the final ranking of answers after the combination of both scores. After trying several combinations over the preliminary datasets, we submitted two options: the joint probability (implemented as the simple product of both scores), and a equiprobable linear combination of both scores (implemented as the sum of both scores).

Run ID	Description	NDNS-Partial	NDNS-Relaxed	NDNS-Exact
1	IR without reranking, ir*qa	0.3148	0.3152	0.3587
2	IR with reranking, ir*qa	0.3290	0.3295	0.3741
3	IR with reranking, ir+qa	0.3437	0.3442	0.3907
	Median	0.3377	0.3387	0.3802
	Max	0.3699	0.3709	0.4207

Table 1: Results of VIGICOVID runs compared with the median of participants and the best system.

At this point, we have a ranking of around 4,000 answers per question. After filtering out the repeated answers and sentences we finally submit the top 1,000 answers following the organization guidelines.

3 Results

We submitted three runs:

1. IR results without neural re-ranking, product combination of IR and answer extraction scores.

2. IR results with neural re-ranking, product combination of IR and answer extraction scores.

3. IR results with neural re-ranking, sum combination of IR and answer extraction scores.

Table 1 shows the results of each run submitted by VIGICOVID. Comparing *run 1* and *run 2* we can observe the positive effect of the neural reranking of retrieved results. Comparing *run 2* and *run 3* we can observe also how sensitive the combination of retrieval weights and answer extraction weights can be. In our case, the linear combination of both scores perform better than the joint probability (computed as the product).

Acknowledgements

This work has been funded by VIGICOVID project FSuperaCovid-5 (Fondo Supera COVID-19 / CRUE-CSIC-Santander)

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SCIBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Robert Krovetz. 1993. Viewing morphology as an inference process. In *SIGIR*, pages 191–202.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Trevor Strohman, Donald Metzler, Howard Turtle, and W Bruce Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proceedings of the international conference on intelligent analysis*, volume 2, pages 2–6. Citeseer.

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. 2020. Cord-19: The covid-19 open research dataset.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.