Procesamiento del Lenguaje Natural, Revista nº 68, marzo de 2022, pp. 133-142 recibido 29-11-2021 revisado 08-02-2022 aceptado 10-02-2022

Readers versus Re-rankers in Question Answering over COVID-19 scientific literature

Readers versus Re-rankers para la Búsqueda de Respuestas sobre COVID-19 en literatura científica

Borja Lozano, Javier Berná, Anselmo Peñas
UNED NLP and IR Group
Universidad Nacional de Educación a Distancia (UNED)
{blozano, jberna, anselmo}@lsi.uned.es

Abstract: In this work we present a comparison between the two most used neural Question Answering (QA) architectures to solve the problem of information overload on COVID-19 related articles. The span extraction (reader) and the re-ranker. We have found that there are no studies that compare these two methods even though they are so widely used. We also performed a search of the best hyperparameters for this task, and tried to conclude whether a model pre-trained with biomedical documents such as bioBERT outperforms a general domain model such as BERT. We found that the domain model is not clearly superior to the generalist one. We have studied also the number of answers to be extracted per context to obtain consistently good results. Finally, we conclude that although both approaches (readers and re-rankers) are very competitive, readers obtain systematically better results.
Keywords: Question Answering, Information Retrieval, Transformers based pre-trained models, BERT, COVID-19.

Resumen: En este trabajo presentamos una comparación entre las dos arquitecturas neuronales de Respuesta a Preguntas (QA) más utilizadas para resolver el problema de la sobrecarga de información en los artículos relacionados con COVID-19: extracción de respuestas (reader) y el reordenamiento (re-ranker). Hemos encontrado que no hay estudios que comparen estos dos métodos a pesar de que son tan ampliamente utilizados. También realizamos una búsqueda de los mejores hiperparámetros para esta tarea y tratamos de concluir si un modelo pre-entrenado con documentos del dominio biomédico como bioBERT supera a un modelo de dominio general como BERT. Encontramos que el modelo de dominio biomédico no es claramente superior al generalista. También hemos estudiado el número de respuestas a extraer por contexto para obtener resultados consistentemente buenos. Finalmente, concluimos que aunque ambos enfoques (readers y re-rankers) son muy competitivos, los readers obtienen sistemáticamente mejores resultados.
Palabras clave: Búsqueda de Respuestas, Recuperación de Información, Modelos pre-entrenados basados en transformers, BERT, COVID-19.

1 Introduction

Since the COVID-19 outbreak, a huge number of scientific articles have been published making the effective acquisition of new knowledge difficult. There are emerging requests from the medical research community for efficient management of the information about COVID-19 from this huge number of research articles[1]. Therefore, Information Systems are needed to assist biosanitary experts in analyzing these publications.

In this work, we explore full Question Answering (QA) systems, systems that given a question and a document collection, rank all the relevant answers that come from different sources. The collections used in the COVID-19 domain are large enough to require a two-stage pipeline (Chen et al., 2017) that combines an Information Retrieval (IR) step with a neural QA module.

There are two main neural strategies for combining both IR and QA in the state-of-

[1] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

the-art: readers and re-rankers. Both receive a preliminary ranking of contexts given by the IR module. Readers approach scan these contexts looking for the text spans that answer the question. Readers assign a score to each answer, so the final ranking of answers (across different sources) comes from this score or its combination with the IR scores. In the re-rankers approach, the neural model is used to directly re-rank the initial list of paragraphs or sentences given by the initial retrieval.

The advantage of re-rankers is that they provide a final ranking in a sound way. In the other side, they can't go beyond the information retrieved by the IR module. However, readers can scan larger contexts looking for answers with bigger lexical gaps to the question, missed by the classical IR engines.

We have found that, although neuronal reranking is a common method to apply after an ad-hoc information retrieval and prior to a reader, there are no studies that compare these two methods independently and fairly.

Therefore, our goal in this work is two fold:

1. Compare readers and re-rankers to determine their differences in performance, and
2. Determine which is the best configuration for answering questions about COVID-19.

The coupling of IR and QA modules hinders the independent evaluation of the QA approaches (readers vs. re-rankers). To isolate their performance and be able to compare the most popular QA architectures we will use the relevance judgements (qrels) to fix the IR variable to the subset of documents, paragraphs and sentences that contain the actual answers to the test questions.

2 Previous work

Open-domain Question Answering (QA) aims to answer questions by finding answers in a large collection of documents (Voorhees and others, 1999). Early approaches to solve this problem consisted in elaborated systems with pipelined components dealing with question analysis, document retrieval and answer extraction (Brill, Dumais, and Banko, 2002; Ferrucci, 2012). Recent advances of Machine Reading Comprehension (MRC) leaded a two-step pipeline, the *retriever-reader* (Chen et al., 2017).

State-of-the-art models for both architectures (readers vs. re-rankers) are based on pretrained models like BERT (Devlin et al., 2018) which are then finetuned for a specific task.

2.1 Readers

The two-stage pipeline for open-QA was first proposed by (Chen et al., 2017). In this architecture the retriever first extracts a small subset of contexts from a large collection. Then the second component of the pipeline, the reader, scans each context thoughtfully in search for an the answer to the question. (Chen et al., 2017) encode the retrieved contexts and the questions using different Recurrent Neural Networks (RNN). For each question-context pair, two distributions over the contexts tokens are computed using bilinear terms, one for the start of the span and the other for the end. The final answer maximizes the probability of the start and end tokens. With the advent of transformers and pre-trained language models (Devlin et al., 2019) many systems adapted them as their reader (Hao et al., 2022). These systems, although effective at extracting correct answers from a context, process each question-context pair as independent of each other. To improve on this issue (Wang et al., 2019) normalizes the probabilities of the span start and end for all tokens in all contexts whereas (Karpukhin et al., 2020) adds another distribution over the [CLS] token representation of all contexts. Recently some authors proposed generative models with enough parameters to create the answer instead of extracting it (Roberts, Raffel, and Shazeer, 2020). Although competitive in some benchmarks large generative models are expensive to train and make inferences on. To tackle this problem (Izacard and Grave, 2021) combines evidence from the retrieved passages to generate the answer.

2.2 Re-rankers

Other approaches substitutes the reader by a answer re-ranking module where the retrieved passages are divided into plausible sentences and re-ranked by a BERT based cross-encoder (Nogueira and Cho, 2019; Yang, Zhang, and Lin, 2019). In those approaches the neural model is used to rerank

an initial ranking generated by a classical information retrieval model based on term-matching techniques. Specifically, they fine-tune the BERT Large model for the task of binary classification, adding a single layer neural network fed by the [CLS] vector in order to obtain a relevance probability. It has been demonstrated that fine-tuning BERT and treating ranking as a classification problem outperforms existing neural information retrieval models by large margins (Pradeep, Nogueira, and Lin, 2021). A known issue of such neural architectures is that require a large number of query relevances (qrels) for training, but their manual generation is very expensive. Some authors (Nogueira and Cho, 2019; Yang, Zhang, and Lin, 2019) use qrel data oriented to passage retrieval such as MS-Marco (Nguyen et al., 2016) and TREC-CAR (Dietz et al., 2017). Another alternative is to generate relevance judgements automatically. (Dehghani et al., 2017), for example, propose to train neural models for ranking using pseudo-qrels generated by unsupervised models like BM25. The TREC-CAR dataset (Dietz et al., 2017) itself is automatically generated from the structure (article, section and paragraph) of the Wikipedia articles. (MacAvaney, Hui, and Yates, 2017) generate pseudo-qrels from a news collection, using the titles as pseudo-queries and their content as relevant text.

2.3 QA on COVID-19

The model vocabulary and its transfer knowledge capabilities depend on the corpus where it has been pretrained. In the same way general domain models are pretrained using general domain corpus like Wikipedia, we hypothesize that models pretrained with in-domain knowledge such as bioBERT (Lee et al., 2019) should improve the performance of downstream tasks related to biomedical information such as the COVID-19 domain is.

With the rise of the COVID-19 Pandemic the value of open-domain QA systems increased as the academic literature about the virus became unmanageable. Many systems, like Vespa[2], AWS search[3] (Bhatia et al., 2020), Google[4] (Bendersky et al., 2020) or Waterloo[5] (Zhang et al., 2020) arose during the first months of the pandemic. Albeit useful in aiding scientific search of COVID-19 literature they all lacked proper domain evaluation, which is usually perform by comparing the correct span of text with the predicted one using a set metric like *F1* or an *Exact Match* (Rajpurkar, Jia, and Liang, 2018). This evaluation is well suited for short and factoid answers but fails to capture complex responses to diverse information needs within the same question.

The Epidemic Question Answering (EPIC-QA) (Goodwin et al., 2020) was organized to aid in the creation of COVID-19 QA systems. The track evaluates capable of automatically answering ad-hoc questions about the disease COVID-19 by extracting answers from the CORD-19 dataset (Wang et al., 2020), 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 is regularly updated and represents the most extensive machine-readable coronavirus literature collection.

One way complex question answering scenarios have been evaluated has been through the use of *nuggets*, a set of atomic "facts" that answer the question. Old evaluation scenarios differentiated between "vital" nuggets and "non-vital" nuggets (Dang, Lin, and Kelly, 2008) whereas new evaluation methods consider all nuggets equally relevant and score answers based on how diverse (in terms of number of *nuggets*) their answers are (Goodwin et al., 2020).

3 Models under evaluation

Since we want to compare the reader approach versus the re-ranker approach, we will fix the retrieval variable by using directly the relevant documents with the different correct answers per question that provide the EPIC-QA dataset.

The models we will compare for the reader and the re-ranker are the following ones in the state-of-the-art:

3.1 Reader

The span extraction module is based on pretrained BERT models (Devlin et al., 2018) with two additional parameters vectors for the span start (S) and span end (E), both $S, E \in R^h$ with h being the hidden size of the last layer. The probability for a span to

[2]https://cord19.vespa.ai/
[3]https://cord19.aws/
[4]https://covid19-research-explorer.appspot.com/
[5]https://covidex.ai/

be an answer is computed in two steps:

1. The $soft$ highest logits in the start and end logits vectors are combined to form $soft^2$ plausible answers, scored by the sum of the start token and end token logits.
2. Iterate each of the answers ranked by score to discard non-valid answers (e.g. end token before start token) until the $soft$ answers are valid. Compute the probability of each answer by a softmax over their scores.

After scoring the best $soft$ answers for each document only the qa_cut best are ranked in the final ranking, up to 1000 answers per question.

The number $soft$ of scored answers for each document and the number qa_cut of selected answers for each document are hyperparameters. We experimented with $soft \in \{10, 20, 50, 100, 200, 500\}$ and $qa\_cut \in \{1 : 20\}$.

In our experimentation we consider 2 pretrained BERT models: The original BERT (Devlin et al., 2018) trained with Wikipedia and Book Corpus, a dataset containing +10,000 books of different genres and BioBERT (Lee et al., 2019) trained on large-scale biomedical corpora.

Four different datasets where consider to finetune the models for span extraction:

1. SQuAD2.0 (Rajpurkar, Jia, and Liang, 2018), which is a reading comprehension dataset widely used in the QA research community.
2. QuAC (Choi et al., 2018) a conversational QA dataset containing a higher rate of non-factoid questions than SQuAD.
3. Merge, a combination of SQuAD2.0 and QuAC with the examples shuffled.
4. Seq, a combination of SQuAD2.0 and QuAC where the model is first finetuned with SQuAD2.0 and then with QuAC.

3.2 Re-ranker

The re-ranking module is based on finetuned BERT models on the MSMARCO dataset (Nguyen et al., 2016), a passage ranking dataset which contains one million queries from real users and their respective relevant passages annotated by humans.

The documents are divided into small sentences that are re-ranked using this BERT-based relevance classifier, following a strategy similar to the one proposed by (Nogueira and Cho, 2019).

Then, as with the reader, only the qa_cut best are ranked in the final ranking, up to 1000 answers per question. The number qa_cut of selected answers for each document is $qa\_cut \in \{1 : 20\}$.

We consider two pretrained BERT models: The original BERT (Devlin et al., 2018), and BioBERT (Lee et al., 2019) trained on large-scale biomedical corpora. Both finetuned with MSMARCO for the re-ranking task. The input to the cross-encoder is formed by concatenating the question and sentence into a sequence separated by the [SEP] token. BERT then computed the probability of the sentence being relevant to the query.

4 Evaluation setting

4.1 Dataset

CORD-19 (Wang et al., 2020) 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. It is used extensively for research, including international shared tasks in the IR and QA fields, such as the CORD-19 Challenge at Kaggle[6], TREC-COVID(Roberts et al., 2020) or EPIC-QA[7]

Epidemic Question Answering (EPIC-QA) track aims to develop systems capable of automatically answering ad-hoc questions in English about COVID-19. EPIC-QA involves two tasks, Expert QA and Consumer QA. Experiment in this work are conducted with the data related to the Expert QA task, aimed to answer questions posed by experts.

The questions have three fields: a keyword-based query, a natural language question, and narrative or background. They are evaluated through the use of *nuggets*, a set of atomic "facts" that answer the question. Two datasets were compiled for the task:

[6]https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

[7]https://bionlp.nlm.nih.gov/epic_qa/

The Preliminary Round dataset uses a snapshot of CORD-19 from June 19, 2020, and it includes 45 expert level questions used in the 4th round of the TREC-COVID IR shared task. EPIC-QA Organizers annotated human-generated answers and sentence-level answer annotations (judgements for short) for 21 of those questions as evaluation set in the preliminary round.

The Primary Round dataset is compiled using a snapshot of CORD-19 from October 22, 2020, and it includes 30 expert level questions and their respective relevance judgements.

In this work we have merged the two annotated sets (21 from preliminary round plus 30 from primary round) into one single dataset (epicQA) to gather more evidence in the evaluation results (Table 1).

In order to evaluate the *reader* and *re-ranker* modules in isolation we constructed an *ideal IR*. The organization of the event released some judgements with the correct nuggets for each question and the sentence where they were. By this it is possible to generate an ideal documents level *ideal IR*. We use this *ideal IR* to evaluate our final systems in 5.

Questions	51
Docs	1446
Tokens/Doc	3351
Sentences/Doc	124
Tokens/Sentence	27
Relevant Sentences/Docs	4

Table 1: EPIC-QA dataset statistics.

4.2 Metrics

The evaluation metric, *Normalized Discount Novelty Score* (NDNS), was provided in the EPIC-QA track as a modified version of Normalized Discounted Cumulative Gain. For each answer in a ranking for a question the *Novelty Score* measures the relevant information not yet seen in previous answers of the ranked list.

$$NS(a) = \frac{n_a * (n_a + 1)}{n_a + f_a} \tag{1}$$

where n_a is the number of novel nuggets of answer a and f_a is the *sentence factor*. Three different variants of NDNS are consider based on how this factor is computed:

- **Exact**: Answers should express novel nuggets in as few sentences as possible. This scenario is more suited to evaluate system where brevity is a priority, like a chat bot which can only give one answer.

$$f_a = n_{sentences} \tag{2}$$

- **Relaxed**: Length doesn't penalise answers as long as every sentence contains novel nuggets. This variant of the NDNS metric rewards systems where brevity is not a requirement but non-redundancy is.

$$f_a = n_{non-relevant} + n_{redundant} + 1 \tag{3}$$

- **Partial**: Redundant information is not penalized which makes this metric well suited for systems solving tasks like a state-of-the-art research about a topic where some overlap in the relevant answers is expected.

$$f_a = n_{non-relevant} + 1 \tag{4}$$

The final metric is computed as the cumulative NS of answers up to rank $k = 1000$

$$NDNS(\mathbf{a}) = \frac{1}{NDNS_{ideal}} * \sum_{r=1}^{k} \frac{NS(a_r)}{\log_2(r+1)} \tag{5}$$

where $NDNS_{ideal}$ is the optimal ranking of answers that could have been found in the document collection for the given question, computed using a beam-search with a width of 10 over the annotated sentences.

4.3 Random baseline

For the creation of the baseline we randomly sorted all sentences in the ideal IR documents into groups of 1000 and evaluated them until a convergence score was reached (Table 2).

epicQA	**Baseline**
NDNS-Partial	0.1726
NDNS-Relaxed	0.1736
NDNS-Exact	0.1948

Table 2: Baseline for the three metrics.

5 Experimentation

In order to make a fair comparison between architectures, we first explore the best set

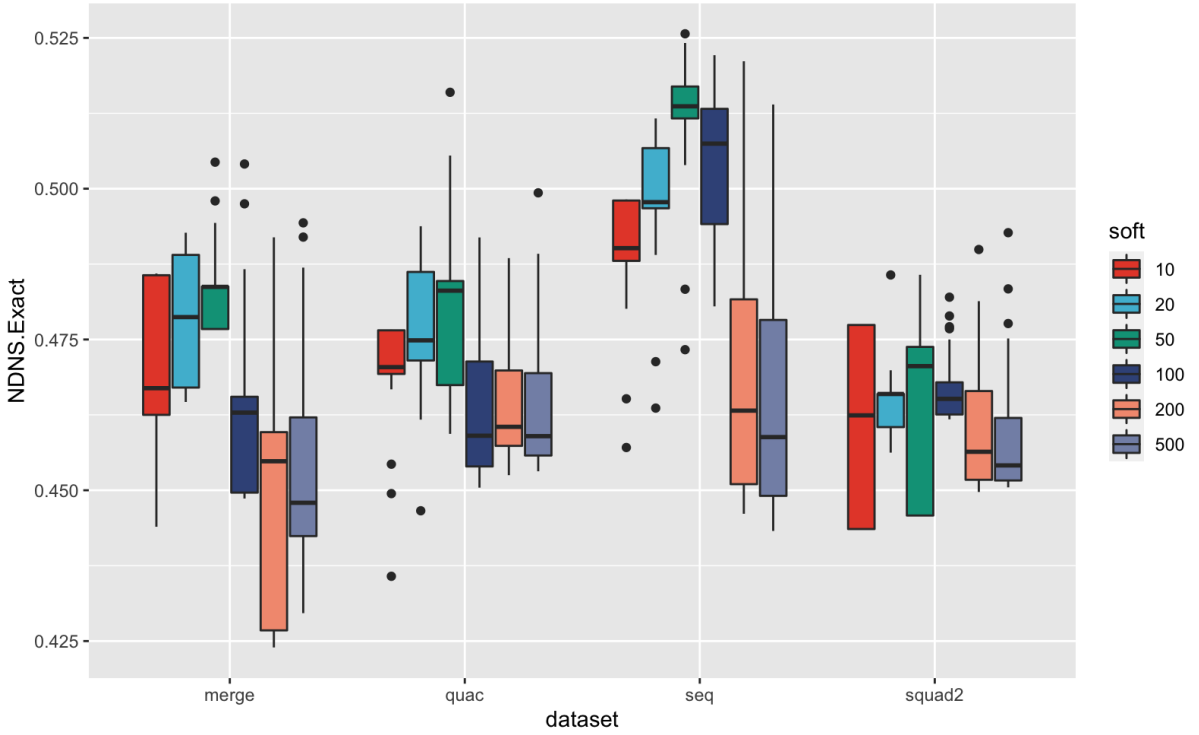

Figure 1: Average NDNS-Exact box-plot across models (BERT and BioBERT) for the reader architecture. Number of considered answers per document (qa_cut 1..20) for each fine-tuning model, breakdown by softmax size. See 3.1 for an explanation of the datasets.

of hyperparameters for each system independently. Those hyperparameters shared by both architectures (the base model and the number of answers consider per document) will be explored jointly. The three different metric scenarios (NDNS-Exact, NDNS-Relaxed and NDNS-artial) have a very strong correlation. We will use mainly NDNS-Exact for comparison since it is the most restrictive and similar to the common metrics used in MRC evaluation.

5.1 Reader

The configuration of the reader depends on two elements. First, the dataset used for tuning the pre-trained model. Second, how to calculate the score of each answer that will determine the final ranking of answers. This score depends on the size of the softmax over the scores of pre-candidate answers for each context.

5.1.1 Dataset used for tuning

The first parameter decision in the reader pipeline is choosing which dataset will be used to finetune the model. We consider four different datasets, detailed in 3.1: SQuAD 2.0, QuAC, random merge of both, and training in sequence (first SQuAD and then QuAC). Figure 1 presents the results breakdown by softmax. Training BERT models with two consecutive datasets is shown to yield the overall best results, even better than just randomly merge both datasets. This result was somehow expected: First, using both dataset gives us more training data. Second, QuAC answers are longer than the kind of factoid-like answers of SQuAD. In this sense, they are closer to the kind of complex answer we need in the COVID-19 domain. So, ending the training with QuAC benefits the model we need.

5.1.2 Softmax

The second step in the reader is to compute the probability of the answers in a document and obtain the scores we need for the final ranking. Given a context and the question, each candidate answer span is first scored by the sum of its start and end token logits. Once all the possible answers in the context are scored, then the probability is computed by a softmax. The size of this softmax, i.e. number of answers per document over which probability is distributed determine the later rank of all answers for a question. The bigger

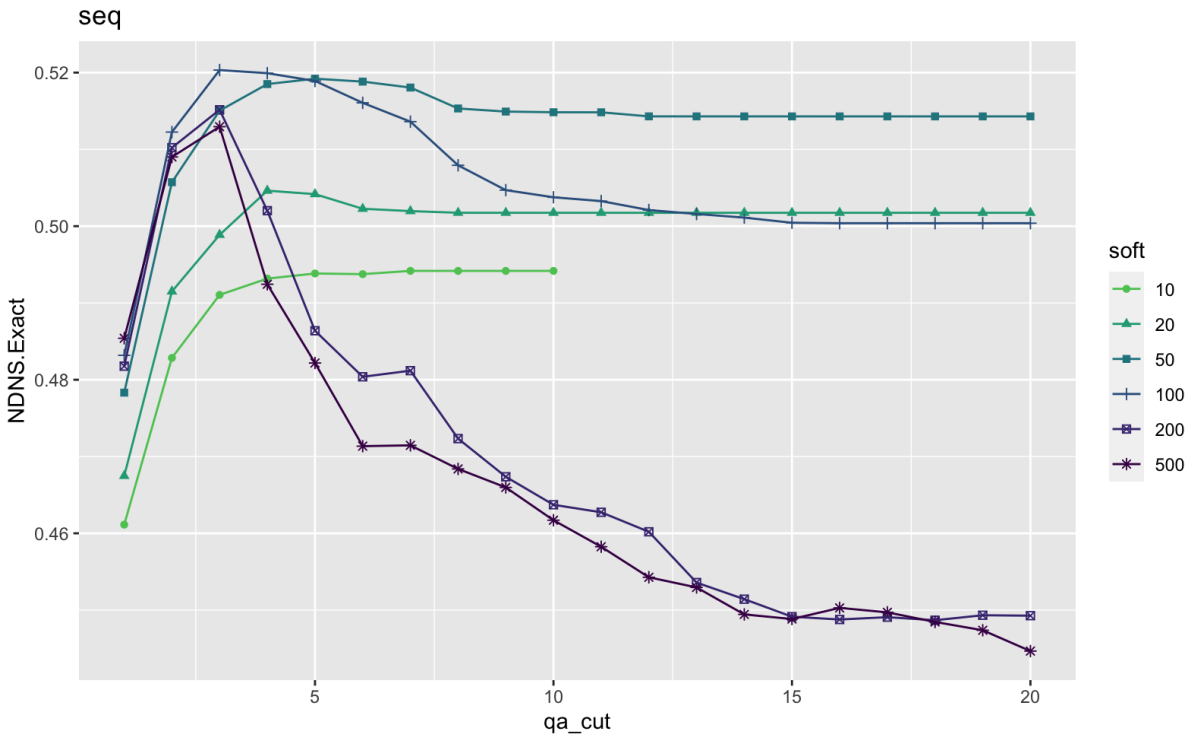

Figure 2: Average NDNS-Exact across models (BERT and BioBERT) for the reader, depending on the number of answers per document (qa_cut), breakdown by softmax size.

the size of the softmax the smaller the probability of each individual answer in absolute terms. However answers in which the model has a high confidence may stand out more from the rest. Results are ploted in Figure 2 with each softmax having a separate curve for all levels of qa_cut, the number of answers selected from each document to be ranked. All softmax sizes follow a similar trend at the beginning of the curve with bigger sizes (> 20) peaking all above 0.51 NDNS-Exact. Interestingly the highest sizes experience a step drop down in their scores. The rest of the sizes experience a small decrease followed by a flat convergence. Overall most softmax sizes perform better at qa_cuts smaller than 10. In this range both softmax sizes of 50 and 100 had the highest scores among all sizes. For this reason we have used them in the rest of the experimentation.

5.2 Re-ranker

The only hyperparameter to explore in the re-ranking is the number of responses per document (qa_cut). In Figure 3 we can see as scores improve drastically when taking between 2 and 10 responses per document, reaching the top in 8, and from 10 responses per document the quality drop decreases in a linear way. Between a qa_cut of 2 and 10, scores above 0.44 are consistently obtained.

5.3 Reader vs Ranker

Finally we compare both architectures filtering qa_values above 10 as both methods results worsen after. Results are plotted in 4. Both methods beat the random baseline 2 by a large margin of more than 25 points, proving its effectiveness.

Results show that the reader approach constantly outperforms the re-ranker one, even for its lowest score with one answer for document. We observe also that reader scores have a smaller variance over the range of qa_cut whereas the re-ranker is surprisingly bad with only one response per document. Another interesting observation is that these results are robust to the use of different pretrained models (BERT and BioBERT), and to the softmax size of the reader.

Contrary to what it might be expected, the domain model bioBERT does not outperform the generalist model BERT, specially in the case of the reader approach. This result rises questions on whether the QA task on COVID-19 benefits from domain-trained networks or if generalists are sufficient.

In the case of the re-ranking method the

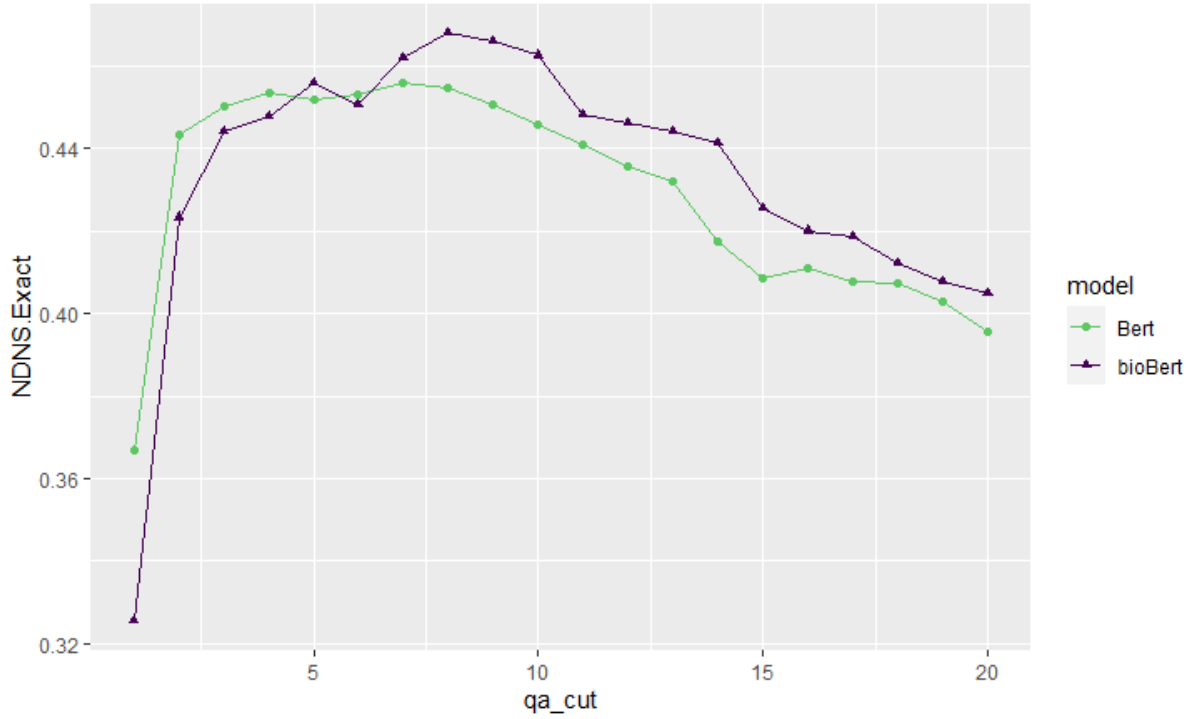

Figure 3: NDNS-Exact results for the re-ranker by number of answers per document (qa_cut) up to 20.

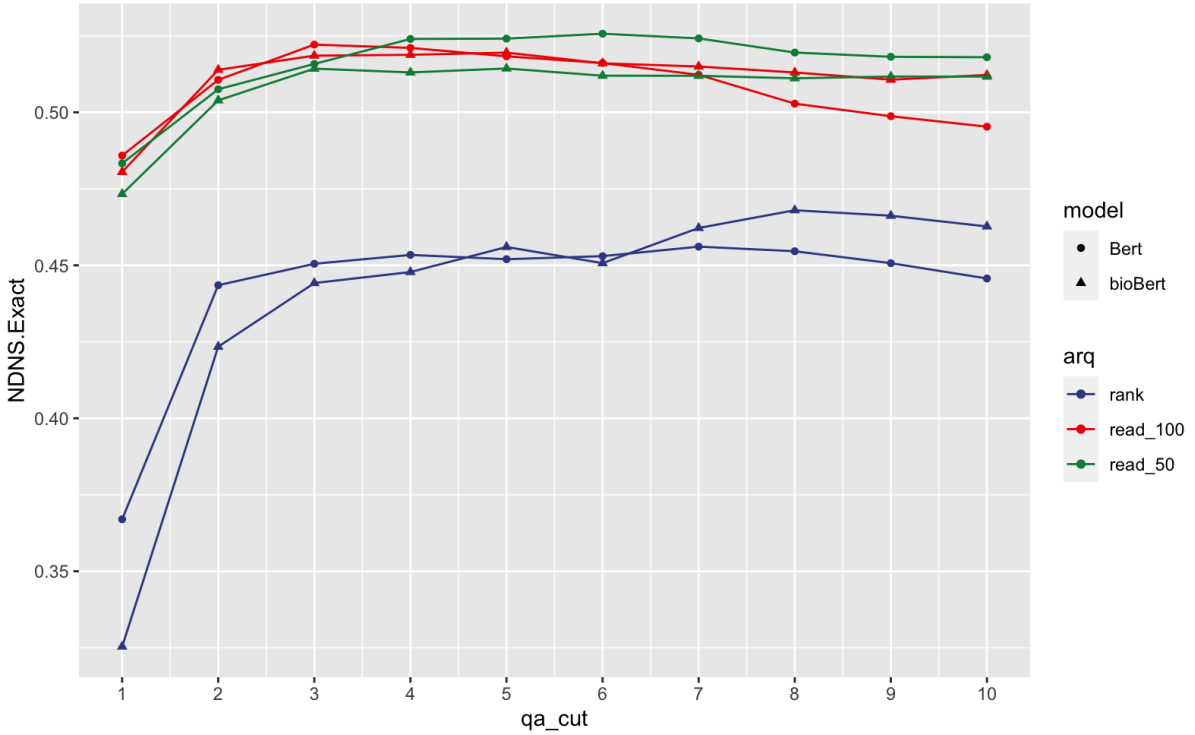

Figure 4: NDNS-Exact results for the Reader vs the Re-ranker by number of answers per document (qa_cut) up to 10.

domain model does outperform the generalist. So we can not come to a global conclusion, but it would be worthwhile to investigate further in this direction.

6 Conclusions and future work

In this work we compare two of the most popular neural QA architectures, retriever-reader and retriever-reranker. We have tested them in the domain-specific scenario of COVID-19.

Although both approaches have shown competitive results, the reader has proven to yield better results than the re-ranker.

Both architectures rely on a previous retrieval step and both return a ranking of sentences answering a given question. However, the retriever-reader approach allows the retrieval of broader contexts than a single sentence and then scan that context looking for the best match. In this case, the final selected sentence may not contain all the exact terms used for the retrieval step, but other related terms according to the language models behind. Therefore, it seems more robust to the initial keyword base retrieval step.

We also concluded that regardless of the method to be used it is always better to take several responses per document, specially in open-domain QA. We conclude that a good range is between 3 and 10 responses per document.

Both domain and generalist models have obtained similar results. We believe that there is an overestimation of the capabilities of domain models and it would be interesting to continue the research in this direction.

As future work, we plan to extend this work to other BERT models and new datasets.

Acknowledgments

This work has been partially funded by VIGICOVID project[8] FSuperaCovid-5 (Fondo Supera COVID-19/CRUE-CSIC-Santander) and by the Spanish Ministry of Science, Innovation and Universities (DeepReading RTI2018-096846-B-C21, MCIU/ AEI/FEDER, UE).

References

Bendersky, M., H. Zhuang, J. Ma, S. Han, K. Hall, and R. McDonald. 2020. RRF102: Meeting the TREC-COVID challenge with a 100+ runs ensemble. *arXiv preprint arXiv:2010.00200*.

Bhatia, P., L. Liu, K. Arumae, N. Pourdamghani, S. Deshpande, B. Snively, M. Mona, C. Wise, G. Price, S. Ramaswamy, X. Ma, R. Nallapati, Z. Huang, B. Xiang, and T. Kass-Hout. 2020. AWS CORD-19 Search: A Neural Search Engine for COVID-19 Literature.

Brill, E., S. Dumais, and M. Banko. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264.

Chen, D., A. Fisch, J. Weston, and A. Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Choi, E., H. He, M. Iyyer, M. Yatskar, W.-t. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Dang, H., J. Lin, and D. Kelly. 2008. Overview of the TREC 2006 Question Answering Track, 2008-11-05.

Dehghani, M., H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dietz, L., M. Verma, F. Radlinski, and N. Craswell. 2017. TREC Complex Answer Retrieval Overview. In *TREC*.

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

Ferrucci, D. A. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1–1.

Goodwin, T. R., D. Demner-Fushman, K. Lo, L. L. Wang, W. R. Hersh, H. T. Dang, and I. M. Soboroff. 2020. Overview of the 2020 Epidemic Question Answering Track. Technical report, Text Analysis Conference (TAC) 2020.

Hao, T., X. Li, Y. He, F. L. Wang, and Y. Qu. 2022. Recent progress in leveraging deep learning methods for question answering. *Neural Computing and Applications*, 34(4):2765–2783.

Izacard, G. and E. Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April. Association for Computational Linguistics.

Karpukhin, V., B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 09.

MacAvaney, S., K. Hui, and A. Yates. 2017. An approach for weakly-supervised deep information retrieval. *arXiv preprint arXiv:1707.00189*.

Nguyen, T., M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. 2016. MS MARCO: A human-generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Nogueira, R. and K. Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Pradeep, R., R. Nogueira, and J. Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *arXiv preprint arXiv:2101.05667*.

Rajpurkar, P., R. Jia, and P. 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.

Roberts, A., C. Raffel, and N. Shazeer. 2020. How Much Knowledge Can You Pack into the Parameters of a Language Model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Roberts, K., T. Alam, S. Bedrick, D. Demner-Fushman, K. Lo, I. Soboroff, E. Voorhees, L. L. Wang, and W. R. Hersh. 2020. TREC-COVID: rationale and structure of an information retrieval shared task for COVID-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436, 07.

Voorhees, E. M. et al. 1999. The TREC-8 question answering track report. In *Trec*, volume 99, pages 77–82. Citeseer.

Wang, L. L., K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. M. Kinney, et al. 2020. CORD-19: The COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Wang, Z., P. Ng, X. Ma, R. Nallapati, and B. Xiang. 2019. Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5881–5885.

Yang, W., H. Zhang, and J. Lin. 2019. Simple applications of BERT for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*.

Zhang, E., N. Gupta, R. Tang, X. Han, R. Pradeep, K. Lu, Y. Zhang, R. Nogueira, K. Cho, H. Fang, et al. 2020. Covidex: Neural ranking models and keyword search infrastructure for the covid-19 open research dataset. *arXiv preprint arXiv:2007.07846*.