# Exploring First-Stage IR Approaches on the CORD-19 dataset

Leonidas Kaldanis
Radboud University
Nijmegen, Netherlands
leo.kaldanis@ru.nl

Foteini Papadopoulou
Radboud University
Nijmegen, Netherlands
foteini.papadopoulou@ru.nl

Büsra Yilmaz
Radboud University
Nijmegen, Netherlands
buesra.yilmaz@ru.nl

## ABSTRACT

As the COVID-19 pandemic in 2020 spread rapidly worldwide, healthcare professionals urgently needed reliable information to tackle the pandemic. A vast number of research papers were published every day making it difficult and crucial to quickly find valid answers and avoid misinformation. To address this need, the TREC-COVID-19 shared task was initiated, where participants built search systems using various methods. Our project is motivated by the Information Retrieval community's interest in aiding healthcare professionals and focuses on building a first-stage retrieval system using the CORD-19 dataset. We apply traditional retrieval models (TF-IDF, BM25, Language Model with Dirichlet smoothing) and a neural IR model (Deep Impact) to experiment with different variants of topics. Additionally, we explore the Doc2Query— approach during indexing, measuring its impact on the evaluation metrics and query search runtime. The results revealed that the traditional retrieval models using the standard indexing remain competitive in the TREC-COVID challenge, showing almost the same performance as the advanced neural approaches in evaluation metrics and execution time, with the TF-IDF outperforming. Moreover, the findings suggest that the choice of query variants plays a crucial role, with the description being the best choice in this context. We release the code and the data for reproduction and further exploration.[https://github.com/foteinipapadopoulou/IR-project]

## KEYWORDS

Information Retrieval, Doc2Query, TREC-COVID challenge, Deep Impact, traditional retrieval, neural models,PyTerrier

## 1 INTRODUCTION

The COVID-19 pandemic in 2020 emphasized the critical need for rapid and reliable access to information within the healthcare sector. Healthcare experts faced the challenge of accessing papers and publications related to coronavirus, through a huge amount of studies, to quickly discover valid answers and accurate insights, and avoid misleading and fake information through a huge amount. Every day during the pandemic, multiple articles were published, which made the seek for reliable sources more difficult. Within only 5 months, PubMed documented more than 60,000 articles that corresponded to search terms related to coronavirus, including COVID-19 and SARS-CoV-2[7].

In response to this critical need, the Text Retrieval Conference (TREC) set up the TREC-COVID shared task. This was a collaborative effort aiming to use Information Retrieval (IR) and text processing to support researchers and clinicians during the pandemic. The TREC-COVID challenge made use of the COVID-19 Open Research Dataset (CORD-19), which was a constant weekly updated and dynamic collection of articles from biomedical literature. In a total of five rounds, participants collaborated in teams to submit for each topic up to 1000 documents and built their own search systems using diverse approaches[9].

The primary evaluation structure of TREC, known as ad-hoc evaluation, provides participants with a database and a set of topics. Participants then formulate these topics into queries for their IR systems, submitting up to N (usually 1000) results per topic. The outcomes from all participants are combined, and an evaluation is manually conducted on the top-ranked.

In this study, we navigate through traditional retrieval models, including BM25, TF-IDF and Language Model with Dirichlet smoothing using the CORD-19 dataset. Besides, we apply the neural IR model Deep Impact[6] and the Doc2Query– indexing method[4] to investigate their capabilities in first-stage retrieval through COVID-19 biomedical literature. We run our experiments on the last round of the TREC-COVID challenge focusing on the performance of different retrieval metrics and query runtime. Therefore, we aim to answer the following research questions:

- RQ1:*What is the performance of traditional retrieval models using a standard indexing method, in terms of retrieval time and evaluation metrics in the context of the TREC-COVID challenge?*
- RQ2: *Does the utilization of the Doc2query— approach or Deep Impact neural model improve the performance of those models compared to the standard indexing method in the TREC-COVID shared task, both in terms of retrieval time and evaluation metrics?*

## 2 RELATED WORK

Roberts et al.(2020)[9] were the organizers of the TREC-COVID challenge. They provided a structured framework for evaluating information retrieval systems in the context of the pandemic. In their studies[9][10] they provide a detailed overview of the challenge. The shared task consisted of multiple rounds, each using an updated version of the dataset and COVID-related topics. The results were generated through the submission of ranked lists of documents, with human annotators providing relevance judgments. This paper is the starting point of our research as it clearly explains the whole set-up of the challenge.

Mallia et al.(2021)[6] conducted their research to explore the use of neural models implementing "a new document term-weighting scheme", named Deep Impact, which leverages contextualized language models like BERT to estimate the semantic importance of tokens in a document. More specifically, the Deep Impact model utilizes a mixture of standard inverted indexes and BERT-based

*IR'23-'24, Radboud University, Nijmegen, Netherlands*

contextualized retrieval models using DocT5Query document expansion to produce an impact score for each token in the dataset corpus. The outperformance of the Deep Impact model compared with other first-stage retrieval models forms a basis for our inclusion of Deep Impact in our comparative evaluation.

In addition, the Doc2Query-– approach, proposed by Gospodinov et al.(2023)[4], uses a filtering technique to reduce the expanded queries that are not relevant, offering a potential improvement to traditional indexing approaches. In particular, the Doc2Query approach proposed by Nogueira et al. [8] uses a sequence-to-sequence model to make predictive queries that are related to each document; however, these models tend to "hallucinate" content building queries that reduce the overall performance of the retrieval metrics. The Doc2Query– approach proposes a solution for this problem using filtering techniques, with the best one being the ELECTRA scorer(30%), to remove the unnecessary produced queries. We are interested in using Doc2Query– method since we want to experiment with techniques to improve the efficiency of the IR system on the TREC-COVID challenge. This study supports us in understanding this method and provides us with instructions on how to implement it.

Table 1 shows the results from the top five teams in the TREC-COVID challenge of 5th round[1]. Seeing the performance of those teams, motivated us to develop a system and try different approaches in order to push its limits.

Making use of all these insights and methods of previous research work, we aim to contribute to the understanding of retrieval models' performance during important events, such as the pandemic.

| team | NDCG@20 | Precision@20 | Brepf | MAP |
|---|---|---|---|---|
| Unique ptr | 0.8496 | 0.876 | 0.6372 | 0.4718 |
| covidex | 0.8311 | 0.846 | 0.533 | 0.3922 |
| Elhuyar NLP team | 0.8116 | 0.834 | 0.6091 | 0.4029 |
| UCD CS | 0.7859 | 0.844 | 0.4488 | 0.3348 |
| Udel fang | 0.7929 | 0.827 | 0.5451 | 0.3682 |

**Table 1: Evaluation metrics from 5 highest-ranking teams in the last round of the TREC-COVID challenge**

# 3 EXPERIMENTAL SETUP

## 3.1 Data

For this study, we used the CORD-19 dataset [9], which is a collection of research biomedical articles that are focused on coronaviruses[7]. CORD-19 has been built by the Allen Institute for AI with the cooperation of famous other organizations and consists of papers and pre-prints from several sources.

The total number of documents reached 193K in the last round and in the first round 30 topics were created to be used as queries with 3 different levels of expression: the query(title), which contains the most meaningful terms, the description, which is a more precise question of the topic, and the narrative, which contains a thorough description of the topic[1]. An example of a topic in the TREC-COVID task is presented in Table 2. After each round, 5 topics were added, with the fifth round having 50 topics in total. In our experiments, we utilized the titles and abstracts of the articles of the CORD-19 dataset, omitting the full text of each article. This decision aligns with the research suggested by Almeida T. and Matos S.(2020) [1]. We share their perspective that articles from different medical resources may have different representations and that abstracts, being well-written, represent an amount of valuable information regarding the conducted research.

| Query(Title) | Description | Narrative |
|---|---|---|
| Coronavirus social distancing impact | Has social distancing had an impact on slowing the spread of COVID-19? | Seeking specific information on studies that have measured COVID-19's transmission in 1 or more social distancing (or non-social distancing) approaches |

**Table 2: Example topic for TREC-COVID task - Extracted from Roberts K. et al. (2020)[9]**

## 3.2 Hardware

We performed our experiments for the indexing on the Kaggle platform and the Radboud GPU cluster server of the Science Faculty. For the retrieval phase and the comparison of the query execution time, we utilized only Kaggle's hardware. Specifications about the used hardware on each platform can be found in Table 6 in the Appendix.

## 3.3 Implementation-Methodology

For our experimental setup, we have implemented three different approaches. The first method is the standard indexing offered by PyTerrier. The default properties are used by PyTerrier[2], among these are English tokenizing, stemming, and a stopwords list removal. The second approach will involve the implementation of the Doc2Query— approach[4] which implements a refined filtering technique to filter irrelevant queries effectively. The default Doc2Query implementation by PyTerrier was applied to our experiment building the indexing pipeline with the number of generated queries to 20, the Electra scorer[3], and the Query Filtering to 30%. We have also utilized the Doc2Query– approach with the number of generated queries to 40 to understand the effectiveness between n=20 and n=40 generated queries, as Gospodinov et al. [4] showed that there was a slight improvement using these numbers.

The last approach for our experiments is the Deep Impact neural model[6]. Its implementation has been extracted and modified by the PyTerrier library of Deep Impact. We have experimented with the default properties of the Deep Impact indexer which uses the fine-tuned DeepImpact checkpoint from the original paper code and as a base model the 'bert-base-uncased'[4] pre-trained English model. However, in our experiments, we utilize as a base model the 'gsarti/covidbert-nli' pre-trained model on the CORD-19 dataset[5] since we saw a slight improvement on the evaluation.

[1]Results for each round: https://ir.nist.gov/trec-covid/archive.html

[2]A detailed list of default properties can be found at the PyTerrier documentation: https://pyterrier.readthedocs.io/en/latest/terrier-indexing.html#indexing-configuration

[3]The generated queries are scored with the crystina-z/monoELECTRA-LCE-nneg3 pre-trained model as Gospodinov et al.[4], show best performance using it

[4]https://huggingface.co/bert-base-uncased

[5]https://huggingface.co/gsarti/covidbert-nli

For all three methods, the Indexer class IterDictIndexer by PyTerrier is utilized to build the index, since it provides the possibility to iterate through the docs and process the duplicates or empty documents in the dataset as we discovered some of them during our experiments.

After the indexing stage, in the domain of Vector Space Models, we implement TF-IDF for first-stage retrieval. As for Probabilistic Models, our project involves applying both BM25 and a Language Model with Dirichlet smoothing. To answer our research questions and compare to the traditional retrieval models, we will include a Neural IR method by integrating the Deep Impact retrieval model. As mentioned earlier, this neural model uses the capabilities of a contextualized language model, BERT, to estimate the semantic importance of tokens within a document[6].

For the whole implementation, the PyTerrier library[5] is being used as it contains practical and helpful tools to conduct our research.

### 3.4 Metrics

For each approach, we investigate the outcomes employing various evaluation metrics, including NDCG@20, Precision@20, Recall@20, and MAP. NDCG measures the quality of the ranking, by comparing it with the ideal ranking. The closer this value is to 1, the better the ranking of the system. The precision indicates the accuracy of the retrieval of relevant documents compared to the total number of retrieved documents. High value in Precision@20 means that most of the top-20 retrieved documents were relevant . On the other hand, recall measures the proportion of relevant documents that were successfully retrieved out of the total number of relevant documents. Mean Average Precision (MAP) is a metric calculating the mean of Average Precision across multiple queries, that way measuring the variance between them.

The evaluation process is enabled through the application of judgments from trec-eval for the last round of the challenge. This evaluation method aims to provide a thorough understanding of the performance of our chosen models in the context of the TREC-COVID challenge.

## 4 RESULTS AND ANALYSIS

The following results illustrate the variations in evaluation metrics and average query execution times resulting from using different variants of the topics specifically, title, description, and narrative. In the context of TREC evaluations, a 'variant' refers to the diverse ways in which topics or queries can be expressed, and in this study, we examine the impact of using distinct variants on the performance of retrieval models and different indexing methods.

### 4.1 Standard Indexing

The standard indexing approach demonstrates very strong competence in comparison with the other advanced methods. As shown in Table 3, TF-IDF exhibited superior performance compared to both BM25 and the Dirichlet Language Model across all evaluation metrics and variants. While BM25 demonstrated a comparable performance, the Dirichlet Language Model performed less favorably, showing the lowest scores among the three models. Notably, the use of description as a variant significantly increased the performance for all models.

| Model | P@20 | R@20 | MAP | NDCG@20 |
|---|---|---|---|---|
| | | Title | | |
| TF IDF | 0.636 | 0.029 | 0.201 | 0.5753 |
| BM25 | 0.645 | 0.0292 | 0.1994 | 0.5801 |
| DirichletLM | 0.494 | 0.0224 | 0.1678 | 0.4336 |
| | | Description | | |
| TF IDF | $0.703^{\nabla}$ | $0.0328^{\nabla}$ | $0.217^{\nabla}$ | $0.6424^{\nabla}$ |
| BM25 | 0.666 | 0.0308 | 0.2165 | 0.6156 |
| DirichletLM | 0.563 | 0.0261 | 0.1824 | 0.5015 |
| | | Narrative | | |
| TF IDF | 0.555 | 0.0243 | 0.1516 | 0.5105 |
| BM25 | 0.542 | 0.0243 | 0.1506 | 0.497 |
| DirichletLM | 0.408 | 0.0176 | 0.1048 | 0.3554 |

**Table 3: Evaluation retrieval metrics using different variants of topics(title, description, narrative) with standard indexing approach for the TREC-COVID dataset. The ∇ symbol shows the highest value of each metric.**

### 4.2 Doc2Query– Indexing

The Doc2Query– approach revealed some interesting patterns across the selected evaluation retrieval metrics, notably aligning with the indexing approach. We have first computed its performance using 20 generated queries. As can be observed in Table 4, the TF-IDF model consistently outperforms BM25 and the language model with Dirichlet smoothing in all three topic variants. Remarkably, the TF-IDF model exhibits the largest values in the description category, indicating a high performance, with the BM25 following closely, whereas the Dirichlet smoothing model drops behind every metric.

| Model | P@20 | R@20 | MAP | NDCG@20 |
|---|---|---|---|---|
| | | Title | | |
| TF IDF | 0.643 | 0.0298 | 0.2009 | 0.5846 |
| BM25 | 0.632 | 0.0294 | 0.1990 | 0.5706 |
| DirichletLM | 0.462 | 0.0217 | 0.1601 | 0.4121 |
| | | Description | | |
| TF IDF | $0.711^{\nabla}$ | $0.0328^{\nabla}$ | $0.2191^{\nabla}$ | $0.6518^{\nabla}$ |
| BM25 | 0.679 | 0.0312 | 0.2189 | 0.6269 |
| DirichletLM | 0.542 | 0.025 | 0.1777 | 0.4889 |
| | | Narrative | | |
| TF IDF | 0.561 | 0.0246 | 0.1533 | 0.5132 |
| BM25 | 0.547 | 0.0244 | 0.1523 | 0.4971 |
| DirichletLM | 0.382 | 0.0166 | 0.1008 | 0.3305 |

**Table 4: Evaluation retrieval metrics using different variants of topics(title, description, narrative) with Doc2Query– indexing approach using 20 generated queries for the TREC-COVID dataset. The ∇ symbol shows the highest value of each metric.**

As shown in Table 7 in the Appendix, the same pattern is followed by using 40 generated queries, with a minor improvement in the metric, when using the description of the topic with the TF-IDF model outperforming.

### 4.3 Deep Impact Neural Model

| Variant of Topic | P@20 | R@20 | MAP | NDCG@20 |
|---|---|---|---|---|
| Title | 0.572$^{\nabla}$ | 0.0254$^{\nabla}$ | 0.1218$^{\nabla}$ | 0.5291 |
| Description | 0.569 | 0.024 | 0.1135 | 0.5467$^{\nabla}$ |
| Narrative | 0.419 | 0.0187 | 0.0823 | 0.4008 |

**Table 5: Evaluation retrieval metrics using different variants of topics(title, description, narrative) with Deep Impact neural model approach for the TREC-COVID dataset. The ∇ symbol shows the highest value of each metric.**

According to Table 5, using the title of the topic as query, the Deep Impact model achieved the highest values across the evaluation metrics, with the usage of the description as query coming after with a slight difference. The narrative field noted the lower performance compared to the other variants. Deep Impact consistently recorded the lowest performance when compared to the other two indexing approaches.

### 4.4 Average Query Execution Time

To address RQ1 and RQ2, we computed the average query execution time for the 50 topics with respect to their variants, for those different indexing and retrieval methods used before and we represent these findings in the bar charts depicted in Figures 1 and 2, 3 in the Appendix. The average execution query times for the Doc2Query– approach using 40 generated queries can be found in the Appendix section in Table 7, since we have not noticed any important difference with the approach using 20 generated queries, except for the slight increase in the execution time.

According to these Figures, it can be depicted that the Deep Impact neural model performed the slowest for every variant of topics, which can probably be explained by the fact that it uses the stored impact scores in the query processing. Another notable slow performance in the average execution query time is the use of the Standard Indexing with Dirichlet smoothing, which shows an almost 80-90% higher increase than the other retrieval model approaches, except for the Deep Impact model. Furthermore, we can see a slight increase of around 25% in the average query execution time of the Doc2Query– approach with TF-IDF, using the description and narrative variant of topics compared to the rest of the standard indexing models, and Doc2Query– with BM25 and Dirichlet smoothing approach.

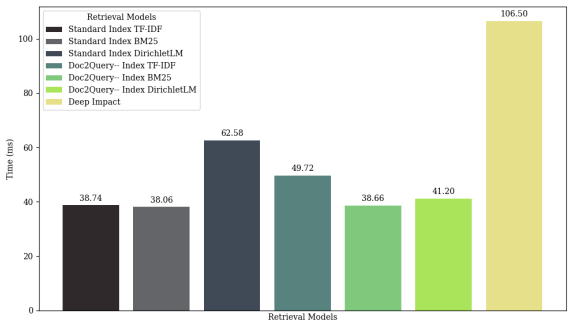


**Figure 1: Average Query Execution time using 'description' variant of queries for different indexing and retrieval models.**

## 5 DISCUSSION AND OUTLOOK

In this paper, we tried to comprehend the difference in the performance of different indexing and retrieval models in terms of the TREC-COVID challenge, in the specific domain of biomedical articles of the dataset in the first-stage retrieval.

Based on our results, overall the TF-IDF retrieval model can capture the most relevant information the best, outperforming the BM25 with slight differences, and the Dirichlet smoothing language model in every usage of the model in the CORD-19 dataset. This agrees with the evidence of Trotman et al.(2014)[11] that BM25-based models seem to outperform language modeling. Furthermore, in a dataset like CORD-19, where specific terms may be crucial for identifying relevant documents (e.g., medical or scientific terms), TF-IDF's ability to highlight such terms could be the reason why we see such a performance.

In the context of the TREC-COVID challenge, the Doc2Query– approach did not show much improvement over the standard indexing approach. This can be due to the nature of the dataset and the fact that in our experiments the Doc2Query model has been trained on the MS-MACRO dataset[2], which is not a dataset focusing on biomedical research but containing more general queries from Bing's search system. For the same reason, this generative query expansion and filtering approach did not show a large improvement on the CORD-19 dataset compared to the other followed approaches, even with 40 generated queries.

Another important finding is that the Deep Impact neural model, although displaying a competitive performance in the first-stage retrieval, did not show much improvement. This could be explained partly because the used checkpoint was the default fine-tuned Deep Impact model which is not trained to handle queries from the biomedical domain.

Finally, the results of using different types of queries of the CORD-19 dataset with different models on the first-stage retrieval, showed that the usage of the topic description can be highly effective in terms of the performance metrics, confirming the importance of using phrases in the queries instead of term as suggested by Becks et al.(2010)[3].

## 6 FUTURE WORK

Future work would include running the experiments on the re-ranking phase and investigating whether the Deep Impact neural model would speed up the query execution time, as proposed by Mallia et al. (2020)[6]. Furthermore, it would be interesting to see the performance on each round of the TREC-COVID challenge and explore the changes in the performance of each model. Finally, an important aspect that should be examined is altering the pre-trained embedding model in the Doc2Query– and Deep Impact approaches to generate the queries and see its effectiveness in capturing relevant information in the first-stage retrieval.

## ACKNOWLEDGMENTS

We would like to thank the Teaching Assistants and the Professor for the feedback and for providing us the Radboud GPU cluster to run our experiments.

## REFERENCES


[1] Tiago Almeida and Sérgio Matos. 2020. Frugal neural reranking: evaluation on the Covid-19 literature. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Karin Verspoor, Kevin Bretonnel Cohen, Michael Conway, Berry de Bruijn, Mark Dredze, Rada Mihalcea, and Byron Wallace (Eds.). Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.nlpcovid19-2.3

[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. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs.CL]

[3] Daniela Becks, Thomas Mandl, and Christa Womser-Hacker. 2010. Phrases or Terms? The Impact of Different Query Types.

[4] Mitko Gospodinov, Sean MacAvaney, and Craig Macdonald. 2023. Doc2Query–: When Less is More. *Kamps, J., et al. Advances in Information Retrieval. ECIR 2023. Lecture Notes in Computer Science, vol 13981. Springer, Cham.* 13981 (03 2023). https://doi.org/10.1007/978-3-031-28238-6_31

[5] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation inInformation Retrieval using PyTerrier. In *Proceedings of ICTIR 2020*.

[6] Antonio Mallia, Omar Khattab, Nicola Tonellotto, and Torsten Suel. 2021. Learning Passage Impacts for Inverted Indexes. *CoRR* abs/2104.12016 (2021). https://arxiv.org/abs/2104.12016

[7] Vincent Nguyen, Maciek Rybinski, Sarvnaz Karimi, and Zhenchang Xing. 2020. Pandemic Literature Search: Finding Information on COVID-19. In *Proceedings of the The 18th Annual Workshop of the Australasian Language Technology Association*, Maria Kim, Daniel Beck, and Meladel Mistica (Eds.). Australasian Language Technology Association, Virtual Workshop, 92–97. https://aclanthology.org/2020.alta-1.11

[8] Rodrigo Frassetto Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *CoRR* abs/1904.08375 (2019). arXiv:1904.08375 http://arxiv.org/abs/1904.08375

[9] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William 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 (07 2020), 1431–1436. https://doi.org/10.1093/jamia/ocaa091

[10] Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh. 2021. Searching for scientific evidence in a pandemic: An overview of TREC-COVID. *Journal of Biomedical Informatics* 121 (09 2021). https://doi.org/10.1016/j.jbi.2021.103865

[11] Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to BM25 and Language Models Examined. In *Proceedings of the 19th Australasian Document Computing Symposium* (Melbourne, VIC, Australia) *(ADCS '14)*. Association for Computing Machinery, New York, NY, USA, 58–65. https://doi.org/10.1145/2682862.2682863

# 7 APPENDIX

| Kaggle platform | |
|---|---|
| **CPU** | 3xIntel(R) Xeon(R) CPU @ 2.00GHz |
| **GPU** | Tesla P100 15GB |
| **RAM** | 32GB |
| **Radboud GPU cluster server** | |
| **CPU** | 2xIntel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz |
| **GPU** | NVIDIA GeForce 2080Ti 11 GB |
| **RAM** | 128GB |

**Table 6: Specifications of the hardware we used during our experiments.**

| Model | P@20 | R@20 | MAP | NDCG@20 |
|---|---|---|---|---|
| | | Title | | |
| TF IDF | 0.634 | 0.0295 | 0.2005 | 0.5768 |
| BM25 | 0.620 | 0.0288 | 0.1980 | 0.5640 |
| DirichletLM | 0.418 | 0.0201 | 0.1502 | 0.3706 |
| | | Description | | |
| TF IDF | 0.719$^{\nabla}$ | 0.0333$^{\nabla}$ | 0.2205$^{\nabla}$ | 0.6667$^{\nabla}$ |
| BM25 | 0.684 | 0.0315 | 0.2199 | 0.6430 |
| DirichletLM | 0.520 | 0.0238 | 0.1658 | 0.4707 |
| | | Narrative | | |
| TF IDF | 0.559 | 0.0247 | 0.1545 | 0.5095 |
| BM25 | 0.542 | 0.0244 | 0.1536 | 0.4966 |
| DirichletLM | 0.341 | 0.0146 | 0.0948 | 0.298 |

**Table 7: Evaluation retrieval metrics using different variants of topics(title, description, narrative) with Doc2Query– indexing approach using 40 generated queries for the TREC-COVID dataset. The ∇ symbol shows the highest value of each metric.**

| Retrieval Model | Time(ms) |
|---|---|
| **Title** | |
| TF IDF | 32.26 |
| BM25 | 31.74 |
| DirichletLM | 31.48 |
| **Description** | |
| TF IDF | 56.44 |
| BM25 | 43.98 |
| DirichletLM | 43.86 |
| **Narrative** | |
| TF IDF | 77.74 |
| BM25 | 63.58 |
| DirichletLM | 66.3 |

**Table 8: Average Query Execution time using different variant of queries for different retrieval models using the Doc2Query– approach with 40 number of genarated queries**

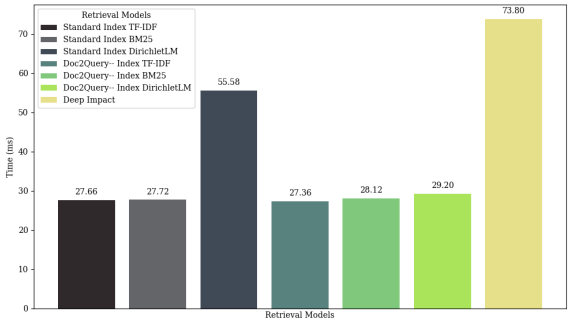


**Figure 2: Average Query Execution time using 'title' variant of queries for different indexing and retrieval models.**

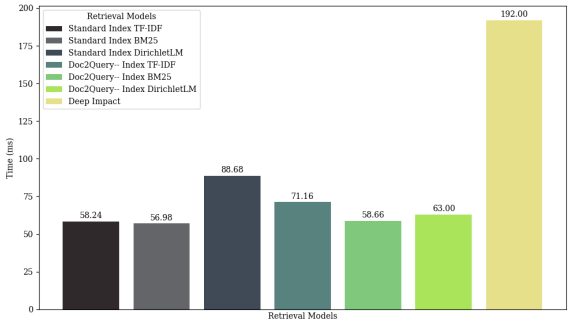


**Figure 3: Average Query Execution time using 'narrative' variant of queries for different indexing and retrieval models.**