# Exploring methods to improve effectiveness of ad-hoc retrieval systems for long and complex queries

**Author:** Dorian Erhan (D.Erhan@student.tudelft.nl)
**Supervisor:** Jurek Leonhardt
**Responsible Professor:** Avishek Anand
TU Delft — Delft University of Technology

## 1. Background

**Ad-hoc retrieval:** ranking a list of documents from a large collection based on their relevance to a given input query (e.g. web search).
**Sparse models:** efficient and fast, depends on exact term-matching, cannot capture context/semantics.
**Dense models:** based on neural models, can capture semantics thus better results. However, much more computationally expensive.
**Retrieve-and-re-rank:** use a sparse retriever to retrieve an initial set of candidates, then a dense model is used to re-rank them in a second stage [1].
**Dual encoders:** use neural models to encode the query and document separately. The queries and documents are mapped to a common vector space and the similarity between them is computed [2].
**Fast Forward indexes:** a framework that uses dual encoders as re-rankers. Final ranking score is the interpolation of the 1st stage sparse score and the 2nd stage dense score [3].

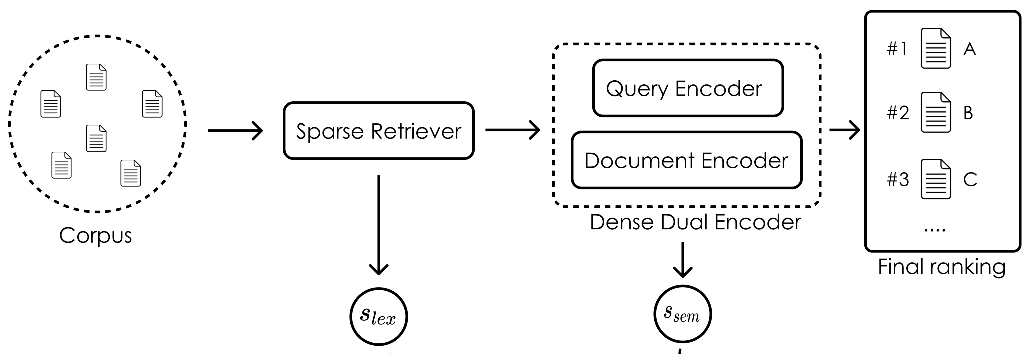
## 2. Research Question

**Can the effectiveness for long and complex queries be improved on Fast-Forward indexes?**

**RQ1:** How does query length and complexity affect the re- ranking performance of different encoders on Fast-Forward indexes?

**RQ2:** What strategies can be employed to improve the effectiveness for long and complex queries on Fast Forward indexes?

## 3. Fast-Forward Indexes




**Final Ranking Score =** $\alpha * s_{lex} + (1-\alpha) * s_{sem}$

## 4. Methodology

To evaluate the impact of long & complex queries on performance, retrieval is performed on datasets that feature a wide range of average query lengths.

| Dataset | Avg. query word length | Task |
|---|---|---|
| TREC-COVID | 10.60 | Biomedical IR |
| SciFact | 12.37 | Fact Checking |
| HotpotQA | 17.61 | Question Answering |
| Arguana | 192.9 | Argument retrieval |

**Table 1:** Overview of the utilised datasets

Two approaches are explored to improve effectiveness of long queries:

- **Query reduction using LLM's:** "I read that ions can't have net dipole moments why not" ⟶ "Why can't ions have net dipole moments?"
- Using **multiple dense models** for the re-ranking stage.

## 5. Results

### Effect of query length on performance

| Dataset | Avg. query word length | BM25 RR@10 | BM25 nDCG@10 | BM25 MAP@1000 | Fast Forward: BM25 >> artic-m RR@10 | Fast Forward: BM25 >> artic-m nDCG@10 | Fast Forward: BM25 >> artic-m MAP@1000 |
|---|---|---|---|---|---|---|---|
| TREC-COVID | 10.60 | 0.8172 | 0.5761 | 0.1835 | 0.9600 | 0.8093 | 0.2569 |
| SciFact | 12.37 | 0.6440 | 0.6839 | 0.6378 | 0.7070 | 0.7427 | 0.7030 |
| HotpotQA | 17.61 | 0.6624 | 0.5128 | 0.4344 | 0.8693 | 0.7181 | 0.6402 |
| Arguana | 192.68 | 0.2408 | 0.3662 | 0.2520 | 0.2511 | 0.3792 | 0.2626 |

**Table 2:** Comparison of retrieval performances for datasets of different average query lengths.

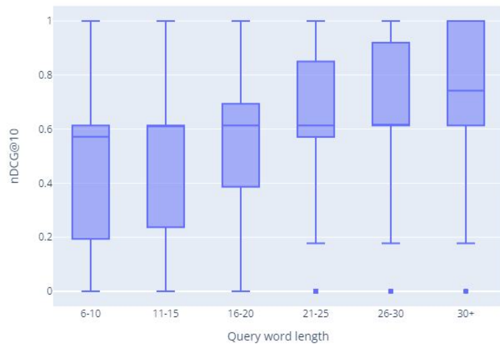



**Figure 1:** Query word length vs. nDCG@10 on HotpotQA.

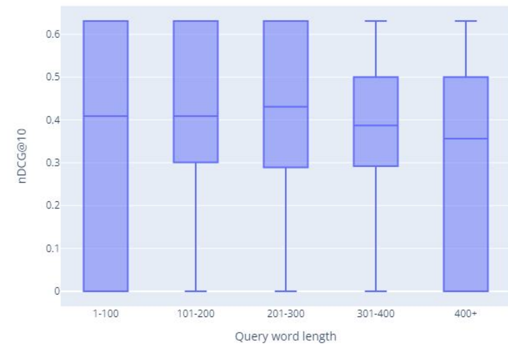



**Figure 2:** Query word length vs. nDCG@10 on Arguana.

### Multiple dense re-rankers

| | SciFact $\alpha_S$ | SciFact $\alpha_{D1}$ | SciFact $\alpha_{D2}$ | SciFact $\alpha_{D3}$ | SciFact nDCG@10 | HotpotQA $\alpha_S$ | HotpotQA $\alpha_{D1}$ | HotpotQA $\alpha_{D2}$ | HotpotQA $\alpha_{D3}$ | HotpotQA nDCG@10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **One re-ranker** | | | | | | | | | | |
| Artic | 0.5 | 0.5 | - | - | 0.7522 | 0.3 | 0.7 | - | - | 0.7255 |
| BGE | 0.3 | 0.7 | - | - | 0.7695 | 0.5 | 0.5 | - | - | 0.6957 |
| GTE | 0.5 | 0.5 | - | - | 0.7694 | 0.5 | 0.5 | - | - | 0.6864 |
| **Two re-rankers** | | | | | | | | | | |
| Artic + BGE | 0.0025 | 0.2975 | 0.7 | - | 0.7688 | 0.25 | 0.5 | 0.25 | - | 0.7340 |
| Artic + GTE | 0.0025 | 0.3975 | 0.6 | - | 0.7756 | 0.075 | 0.725 | 0.2 | - | 0.7310 |
| BGE + GTE | 0.0025 | 0.7 | 0.2975 | - | 0.7790 | 0.25 | 0.5 | 0.25 | - | 0.7122 |
| **Three re-rankers** | | | | | | | | | | |
| Artic + BGE + GTE | 0.0025 | 0.1975 | 0.5 | 0.3 | 0.7765 | 0.05 | 0.55 | 0.3 | 0.1 | 0.7305 |

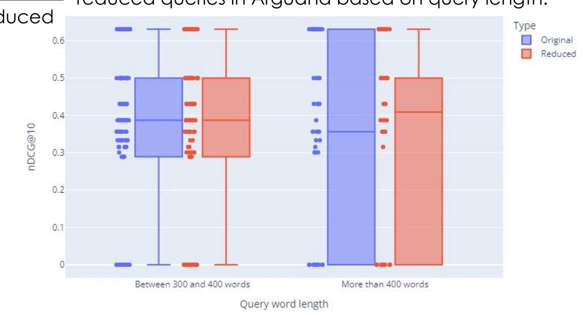
**Table 3:** Performance comparison (nDCG@10) with varying numbers of re-rankers.

### Query Reduction

| Dataset | Avg. word length | BM25 >> Artic-m $\alpha = 0$ | BM25 >> Artic-m $\alpha = 0.1$ | BM25 >> Artic-m $\alpha = 0.3$ |
|---|---|---|---|---|
| **TREC-COVID** | | | | |
| Original | 10.60 | 0.7984 | 0.8092 | 0.8074 |
| Original Reduced | 7.48 | 0.7901 | 0.8006 | 0.7985 |
| Keyword | 3.48 | 0.6577 | 0.6656 | 0.6688 |
| Narrative | 24.96 | 0.6576 | 0.6672 | 0.6686 |
| Narrative Reduced | 10.26 | 0.6910 | 0.6988 | 0.6877 |
| **Arguana** | | | | |
| Original | 192.9 | 0.3764 | 0.3792 | 0.3869 |
| Reduced | 65.38 | 0.3652 | 0.3691 | 0.3751 |

**Table 4:** Performance comparison (nDCG@10) between the reduced and unreduced queries in TREC-COVID and Arguana.

**Figure 3:** Comparison between the original and reduced queries in Arguana based on query length.




## 6. Limitations

- Limited sample of datasets - testing the methods on additional datasets would be beneficial leading to more robust results.
- The effectiveness of LLM generated reductions is influenced by the input and system prompts used. While multiple configurations were tested to optimize the results, better configurations may still exist.

## 7. Conclusions

### Effect of query length on performance

- Retrieval effectiveness decreases as average query length of the dataset increases.
- However, in some datasets shorter queries are more challenging than longer ones due to their **ambiguous** nature.

### Multiple dense re-rankers

- **Effective in increasing ranking quality** for long & complex queries.
- Using two models for re-ranking provides the best balance between performance and ranking quality.
- Optimal configuration consists of assigning greater weights to the best performing models and including the sparse score.

### Query reduction

- Performance comparable to original, but **not effective in increasing ranking quality** for long & complex queries.
- Limited success in improving performance of queries that surpass the context length of the dense model.

## 8. Future Work

Recommendations for future research include:

- Improving effectiveness by using multi-vector representation for queries & documents
- Applying query reduction to only a subset of corpus, based on specific criteria
- Query extension for difficult and ambiguous short queries

## References

[1] R. F. Nogueira and K. Cho, "Passage re-ranking with BERT," CoRR, vol. abs/1901.04085, 2019. [Online]. Available: http://arxiv.org/abs/1901.04085
[2] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: Proceedings of the 2020 Conference on Empircal Methods in Natural Language Processing (EMNLP) (2020), pp. 6769–6781
[3] Jurek Leonhardt et al. "Efficient Neural Ranking using Forward Indexes". In: Proceedings of the ACM Web Conference 2022. WWW '22. , Virtual Event, Lyon, France, Association for Computing Machinery, 2022, pp. 266–276.