University of Amsterdam at TREC 2022: Deep Learning Track

David Rau Jaap Kamps

University of Amsterdam

Abstract

This paper documents the University of Amsterdam's participation in the TREC 2022 Deep Learning Track. We investigate novel document representation approaches capturing long documents within the input token length of neural rankers, and even in a fraction of the maximum input token length. Reducing input length of the document representation leads to dramatic gains in efficiency, as the self-attention over token length is the main culprit of the high gpu memory footprint, low query latency, and small batch sizes. Our experiments result in a number of findings. First, we observe dramatic gains in efficiency of the document representation approaches mindful of what tokens matter for the neural rankers. Second, we also observe a trade-off between effectiveness and efficiency, but also observe that document native approaches retrieve in numerous documents missed by passage based approaches. This leads to a significant underestimation of their effectiveness, but also highlights their potential to retrieve documents not considered by traditional rankers or passage based neural rankers. There is great potential to study these in future editions of the track.

1 Introduction

This paper documents the University of Amsterdam's participation in the TREC 2022 Deep Learning Track. The Deep Learning Track started at TREC 2019 and is in it's fourth year [Craswell et al., 2020, 2021, 2022]. For the 2022 edition, we decided to focus on the document retrieval task and experimented with document representations approaches that aim to encode the content of full documents within the constraints of transformers for text ranking. We are interested in document native approaches rather than view document retrieval as an afterthought of effective passage retrieval approaches, and care as much about efficiency as about effectiveness. We experiment with approaches that allow for using transformers in one pass document re-ranking with cross-encoders, or alternatively full collection ranking with bi-encoders. We are particularly interested in very succinct, and hence very efficient, document representations still representing the entire document.

This paper is structured in the following way. Our simple experiment is described in Section 2 and the results of these experiments in Section 3. Finally, we end in Section 4 with a discussion of our main findings.

2 Experimental Design

In this section we detail our document representation experiments.

Our main experimental parameter is to investigate more efficient document representations. We prioritize *efficiency*, and aim to achieve very significant efficiency gains by reducing the input length of neural document rankers. So our main aims are:

- Can we reduce document representations of full documents of any length to the sub-word input token length of neural rankers?
- Can we further reduce the input length of document representations aiming for a Pareto-optimal trade-off between efficiency and effectiveness.

This approach is attractive as in case we can control the input length of the document representation in transformers for text ranking, this directly translates into significant efficiency gains due to the very costly self-attention mechanism. Recall, self-attention compares any input token against any other input token, on all transformer layers, leading to a quadratic complexity over input length.

Document and query representation have been studied since the beginning of the field of information storage and retrieval, giving us many options to incorporate some of the deepest and most foundational results of the field into modern neural rankers. For example, one may consider the following approaches:

Truncate A poor man's approach to encoding long documents is simply to truncate the input at a particular k or at the maximum token length, e.g. at 512 sub-word tokens for BERT-based cross-encoders and bi-encoders (or 512 minus query and separator tokens).

tf.idf Given the relative effectiveness of bag-of-word neural rankers [Rau and Kamps, 2022b], an alternative is to go back to representing documents by their word distributions, and select the first k words or sub-words based on their classic vector-space term weight, such as tf.idf [Robertson and Spärck Jones, 1976, Salton and Buckley, 1988].

PLM Given that neural rankers are based on large pre-trained language models, we can also create top k term distributional document representations using language modeling framework approaches. For example, the clean and interpretable document representations of parsimonious language models [Hiemstra et al., 2004, Kaptein et al., 2010], or of Luhnian significant word language models [Dehghani et al., 2016a,b].

In pre-submission experiments, the PLM approach seemed to retrieve the highest number of novel unjudged documents high in the rankings, and we decided to base our official submissions on PLM. After pre-computing a query independent parsimonious language model for each of the MS Marco documents, we submitted runs based on the top 514, 128, and 64 terms based on this model. Our official submissions are based on a cross-encoder (CE) reranking the track's provided top 100 BM25 run for the document retrieval re-ranking subtask. We consider this BM25 ranking as reference, and also report bi-encoder or full-rank results [based on Splade, Formal et al., 2022].

In earlier years, we also analyzed recall aspects of various models and observed very high fractions of unjudged documents throughout the passage retrieval runs [Kamps et al., 2021, Rau and Kamps, 2022a]. Based on a small sample of evidence, this looks particularly to effect novel approaches that are no close variant of systems dominating the pools. Here, the fraction of unjudged is worryingly high, even for official submissions after the top 10 pooling cut-off. This is partly due to the use of an active learning approach focusing on one particular stream of documents to be judged, rather than top-n pooling favoring original systems. Novel approaches not contributing to the pools, also show large numbers of unjudged in the top 10's.

One affected category is non-passage based document retrieval approaches, and we decided to submit such approaches in order to increase document retrieval pool diversity. As it turned out, none of our submissions contributed to the pool, due to the pragmatic decision to locate the limited available resources exclusively to the passage retrieval track. The negative impact of this is a significant underestimation of the performance of these submission. As a positive side effect, this

Table 1: Effectiveness on the NIST judgments of the TREC 2021 Deep Learning Document Task

Ranker	**MRR**	**Prec**			**NCDG**			**BPref**	**MAP**
		5	**10**	**30**	**5**	**10**	**30**		
BM25	0.8367	0.7053	0.6684	0.5561	0.5231	0.5116	0.4874	0.2784	0.2126
CE PLM 512	0.8425	0.7228	0.7035	0.5573	0.5316	0.5385	0.4946	0.2773	0.2128
CE PLM 128	0.8956	0.7789	0.7281	0.6006	0.6167	0.5888	0.5487	0.2824	0.2300
CE PLM 64	0.9324	0.8386	0.7456	0.6117	0.6722	0.6245	0.5672	0.2842	0.2354
CE MaxP	0.9620	0.8281	0.7860	0.6392	0.6668	0.6446	0.5871	0.2935	0.2453

Table 2: Effectiveness on the NIST judgments of the TREC 2022 Deep Learning Document Task

Ranker	**MRR**	**Prec**			**NCDG**			**BPref**	**MAP**
		5	**10**	**30**	**5**	**10**	**30**		
BM25	0.6416	0.4605	0.3789	0.2746	0.3368	0.2983	0.2555	0.1633	0.0924
CE PLM 512⋆	0.6371	0.4132	0.3553	0.2877	0.2920	0.2689	0.2514	0.1661	0.0874
CE PLM 128⋆	0.7070	0.4632	0.4276	0.3219	0.3562	0.3391	0.2940	0.1673	0.1016
CE PLM 64⋆	0.6665	0.4474	0.4158	0.3254	0.3428	0.3252	0.2890	0.1655	0.1003
CE MaxP	0.7533	0.5132	0.4724	0.3588	0.4133	0.3839	0.3315	0.1711	0.1132
Splade PLM 64	0.5251	0.3237	0.2697	0.1899	0.2216	0.2017	0.1745	0.1148	0.0506
Splade PLM 64 (1k)	0.5255	0.3237	0.2697	0.1899	0.2216	0.2017	0.1745	0.2328	0.0678

⋆ *Official submissions.*

turns our participation into a pooling and re-usability experiment for document retrieval, extending and adding to our previous analysis of the passage retrieval pooling and re-usability at TREC 2021 [Rau and Kamps, 2022a].

3 Experimental Results

This section contains the main results of our experiments, focusing on the effectiveness and efficiency of shorter document representations, and an analysis of pool coverage of non-passage-based neural document retrieval approaches.

3.1 Effectiveness

Our main aim is not to improve effectiveness, but to improve efficiency (discussed in the next subsection). We have to accept that large gains in efficiency may come at some loss of effectiveness. So how much performance is retained?

Tables 1 and 2 show the effectiveness results for document retrieval task at TREC 2021 and TREC 2022 (based on the official, non-expanded, qrels). For 2021, where our systems didn't contribute to the document retrieval pools, we observe favorable performance in Table 1. First, we are substantially outcompeting lexical systems. Second, our far more efficient approach is still close to expensive alternatives such as MaxP. Third, the most efficient approach can even out-compete MaxP for early rank cut-offs.

Table 3: Efficiency on the NIST judgments of the TREC Deep Learning Document Task 2022

Ranker	Max. Batch Size	Total time	Query Latency	P@30
CE MaxP*	256	2h 27m 39s	44.49 ms	0.3588
CE PLM 512	256	8m 27s	4.26 ms	0.2877
CE PLM 128	2,560	2m 30s	0.50 ms	0.3219
CE PLM 64	4,608	1m 38s	0.35 ms	0.3254

* *Sliding window over the document.*

As our approach is radically different from the runs contributing to the document retrieval pools in 2021, we observed very high fractions of unjudged documents leading to an underestimation of performance. We particularly submitted these runs as official submissions in 2022, in order to get a better estimate of their performance. How did this turn out? For 2022, we see broadly the same qualitative pattern but also observe far lower performance throughout in Table 2, for all document retrieval systems. The gap with passage-based MaxP seems larger, rather than smaller, where we hoped that our non-passage based document retrieval approaches could gain in performance as being official submissions for 2022.

Closer inspection reveals far higher fractions of unjudged documents in our official runs, far higher than observed in 2021 for post-submission experiments. This effect is even worse for the full-rank Splade run, evaluated over top-100 and top-1k in Table 1. As it turns out no document retrieval submission was pooled, leading to a significant underestimation of performance for non-passage based approaches like ours. We report a detailed analysis later in this section.

In this subsection, we reported the effectiveness of efficient document retrieval approaches. A positive outcome is that these approaches gain efficiency at only a minor loss of effectiveness, and still retain the improvement of neural rankers over classic lexical approaches. A less positive outcome is that document retrieval approaches haven't contributed to the pools, leading to an underestimation of their effectiveness.

3.2 Efficiency

Our main aim was not to improve effectiveness, but to improve efficiency. Reducing the input to only a few expressive terms allows us to reduce the input length to a fraction of 512 tokens. As the self-attention in transformer-based models grows quadratically in memory with the input length, reducing the input leads to a dramatic decrease in GPU memory used. This can be exploited by fitting way larger batch sizes into GPU memory resulting in faster inference times.

To quantify efficiency we measure query latency. To this end, we measure the contextualization of 1 query-passage pair based on a batch and multiply it by the number of documents (100) to be re-ranked. We carry out all our efficiency experiments on a single NVIDIA V100 with 16GB memory with the maximum batch size in PyTorch inference mode. We determine the maximum batch size by increasing the batch size until we run out of GPU memory. We discard the first warm-up batch from the measurement. For the query latency, we measure the bare forward-pass and do not include pre-processing, or disk-access times. Additionally, we report the *total run time* indicating the actual run time including reading the input, tokenization and forward-pass.

Table 3 shows the total time needed to contextualize all query-document pairs (50,000) within the TREC Deep Learning Track 2022 re-ranking task. Additionally, we report query latency, indicating the bare GPU time that is needed to propagate the input through the model. Here we

Table 4: Distribution of document judgments (TREC Deep Learning Track 2020–2022)

Year	**Fraction**				**CCDF**			
	0	**1**	**2**	**3**	$\geq$ **0**	$\geq$ **1**	$\geq$ **2**	$\geq$ **3**
2019	59.42	28.34	7.07	5.17	100.00	40.58	12.24	5.17
2020	80.58	13.05	3.46	2.91	100.00	19.42	6.38	2.91
2021	37.18	32.00	21.21	9.62	100.00	62.82	30.82	9.62
2021 (exp)	51.56	26.05	15.29	7.10	100.00	48.44	22.40	7.10
2022 (inf)	55.25	24.56	13.64	6.55	100.00	44.75	20.19	6.55

are utilizing the maximum batch size that can be fit onto the GPU. We compare PLM 64, 128, 512 to MaxP. MaxP utilizes the same passage model applying a sliding window of 512 tokens (with an overlap of 256 tokens) over the entire document to capture the context of the entire document. We limit the total number of tokens processed per document to 8,192 tokens to make inference feasible. Therefore, our time measurements of MaxP are even an underestimate of the true cost when applied to the entire document.

We observe drastic gains in efficiency for query latency thus in total run time. Compared to MaxP reducing documents to 512 tokens (PLM 512) we are able to reduce the Query Latency by an order of magnitude from 44.49 ms to 4.26 ms. For an even stronger reduction of the documents to 64 tokens we are able to bring Query Latency down to 0.35 ms. Note that PLM with 64 tokens performs best in terms of P@30 while being the most efficient variant. As expected MaxP outperforms all PLM variants. MaxP performs around 10% better than PLM 64, while taking about 2h 27m 39s in total to run versus 1m 38s for PLM 64 demonstrating that a reduction of documents to only meaningful terms can lead to an immense efficiency gain while maintaining strong performance.

In this subsection, we reported the efficiency of efficient document retrieval approaches reducing full documents to a fraction of their tokens. We observed very dramatic gains in efficiency resulting in a very favorable trade-off between effectiveness and efficiency. This may make these models attractive under resource limited conditions or high volume production system settings. It may also help significantly increase the scope and number of use cases and applications in which neural rankers are currently not economically viable to deploy. Fortunately, for efficiency analysis we do not depend on scarce and expensive editorial judgments. In fact, our efficiency results generalize to any test collection or retrieval setting, and to entire classes of neural models.

3.3 Analysis

In this rest of this section, we provide a deeper analysis of the recall base and pool coverage of the document retrieval qrels.

To provide context, we first give some details about the document retrieval judgments in 2019–2022. The distribution of judgments is shown in Table 4. We see that in 2022, 55% of the judgments is non-relevant, and hence 45% of the judged passages has some relevance, whereas 20% of the judgments is 'Highly Relevant' or 'Perfect'. These fractions are similar to the 2021 expanded queries, which may come as no surprise as the organizers decided to following a pooling approach based on the analysis and further judgments obtained after the TREC 2021 track was completed. On the one hand, this is a positive outcome: the distribution of labels doesn't exhibit the high fraction of 'relevant' of the official judgments in 2021, indicating a more complete recall base. On

Table 5: Judged documents across ranks (TREC Deep Learning Track 2021)

		Rank				
		1	**5**	**10**	**50**	**100**
BM25 (official)	Relevant (%)	75.44	70.53	67.02	47.23	36.60
	Non-relevant (%)	24.56	29.47	32.98	19.75	17.05
	Unjudged (%)	0.00	0.00	0.00	33.02	46.35
	Rel/Judged (%)	75.44	70.53	67.02	70.51	68.21
BM25 (expanded)	Relevant (%)	75.44	70.53	67.02	50.67	41.14
	Non-relevant (%)	24.56	29.47	32.98	26.35	25.32
	Unjudged (%)	0.00	0.00	0.00	22.98	33.54
	Rel/Judged (%)	75.44	70.53	67.02	65.79	61.91

the other hand, the document retrieval judgments are 'inferred' from the passage level pooled runs and passage level assessments, making it far from guaranteed that this positive outcome translates to the document retrieval task.

In order to study the effect of pool inclusion, or lack thereof, let us first analyze the expected number of unjudged documents for pooled runs. Table 5 shows the pool coverage of the officially provided BM-25 run in 2021, used by all participants submitting re-rank submissions to the document retrieval track. In 2021, this run was indeed pooled, resulting in 0% unjudged at rank 10. This percentage increases to 33% at rank 50, and 46% of the official set to rerank in the track. With more assessment resources used in the "expanded" qrels, these fractions are lowered to 23% at rank 50, and 34% at rank 100. For the expanded qrels, there is also a decrease in the fraction of relevant over judged, going down from 75% to 62%—generally a positive signal of making progress in covering the entire recall base. Although not completely judged, this results in a fair fraction of the BM25 run being judged in 2021, and consequently a reasonably fair evaluation of all rerank submissions.

Table 6 shows a similar analysis of the 2022 inferred document qrels. For the official BM25 run used in rerank submissions, we immediately see the effect of not pooling document retrieval submissions. We observe a far higher fraction of unjudged documents, up to 71% over the top 100, but already 44% of the top 10 making official rank based measures such as NDCG@10 a significant under-estimation of performance. We observe even higher fractions of non-passage based neural document retrieval approaches, with PLM 512 retrieving about 50% unjudged in the top 10, and the bi-encoder full-rank even 60%.

Our analysis leads to three general conclusions. First, official document retrieval submissions are underestimated with the 2022 qrels, and the coverage of 2022 official submissions is significantly lower than non-pooled runs in 2021. Second, there are always unjudged documents and no test collection has a complete recall base, but the pool bias is a cause of worry (privileging passage-based approaches over native-document approaches), as are high fractions of unjudged in the top of ranking (affecting not only recall-based measures but also early precision measures). Third, the high fraction of unjudged can also be interpreted as a positive observation, as it demonstrates that non-passage retrieval approaches are able to retrieve many documents missed by passage-based approaches. The fraction of relevant over judged is quite high with 62%–75%, indicating the potential to increase pool diversity and to overcome limitations of the passage based neural rankers dominating the pools.

Table 6: Judged documents across ranks (TREC Deep Learning Track 2022)

		Rank				
		1	**5**	**10**	**50**	**100**
BM25	Relevant (%)	53.95	46.05	37.89	23.24	18.22
	Non-relevant (%)	22.37	19.21	17.89	14.11	11.01
	Unjudged (%)	23.68	34.74	44.21	62.66	70.76
	Rel/Judged (%)	70.69	70.56	67.92	62.23	62.33
CE PLM 512	Relevant (%)	51.32	41.32	35.53	24.63	18.22
	Non-relevant (%)	17.11	18.68	15.26	13.29	11.01
	Unjudged (%)	31.58	40.00	49.21	62.08	70.76
	Rel/Judged (%)	75.00	68.86	69.95	64.95	62.33
CE PLM 128	Relevant (%)	60.53	46.58	42.50	26.08	18.22
	Non-relevant (%)	25.00	21.84	19.74	14.39	11.01
	Unjudged (%)	14.47	31.58	37.76	59.53	70.76
	Rel/Judged (%)	70.77	68.08	68.29	64.43	62.33
CE PLM 64	Relevant (%)	55.26	44.74	41.58	26.66	18.22
	Non-relevant (%)	25.00	22.63	19.87	15.13	11.01
	Unjudged (%)	19.74	32.63	38.55	58.21	70.76
	Rel/Judged (%)	68.85	66.41	67.67	63.79	62.33
Splade PLM 64	Relevant (%)	34.21	32.11	27.11	16.71	12.50
	Non-relevant (%)	11.84	12.37	12.50	9.24	7.66
	Unjudged (%)	53.95	55.53	60.39	74.05	79.84
	Rel/Judged (%)	74.29	72.19	68.44	64.40	62.01

4 Conclusions

This paper documented our participation in the TREC 2022 Deep Learning Track, focusing on efficient document representation approaches for the document retrieval task. We care as much about efficiency as about effectiveness, and submitted only native, non-passage based, neural document ranking runs.

Our main conclusions are the following. First, we are able to achieve very favorable efficiency compared to passage-based document retrieval approaches. This opens up novel options to scale neural models to far longer documents, think of books or other aggregates, without a loss of efficiency. It also opens up new ways to trade-off efficiency and performance in a highly dynamic way, tailored to the specific user or specific request at hand. Second, we observe the expected trade-off between efficiency and effectiveness, where significant gains in efficiency can result in a moderate loss of effectiveness – but overall a very favorable Pareto frontier in the trade-off. This helps opening up new large-scale application areas where current neural models are not economical, and informs responsible business decisions optimizing costs and benefits. Third, the evaluation of document retrieval submissions seems completely dominated by passage retrieval approaches, and our native (non-passage based) document retrieval runs are able to retrieve significant fractions of documents missed by passage retrieval systems. This points to the potential value of alternative native document retrieval approaches, and the importance to reflect those in the assessment pools. While this is in itself a positive observation it comes with the downside that our submissions cannot be fairly evaluated given the TREC provided qrels, and their effectiveness is significantly underestimated. Our general conclusions are that native document retrieval approaches are an attractive area of research, with large potential efficiency gains, and that evaluating their effectiveness requires a larger recall base with unbiased pooling. There is great potential to study these in future editions of the track.

Acknowledgments We thank the track organizers for their amazing service and effort in making realistic benchmarks for neural ranking available. This research is funded in part by the Netherlands Organization for Scientific Research (NWO CI # CISC.CC.016), and the Innovation Exchange Amsterdam (IXA POC grant). Views expressed in this paper are not necessarily shared or endorsed by those funding the research.

References

N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. Overview of the TREC 2019 deep learning track. In *TREC 2019: Proceedings of the Twenty-Eighth Text REtrieval Conference*. NIST Special Publication 1250, 2020. URL https://trec.nist.gov/pubs/trec28/papers/Overview.DL.pdf.

N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the TREC 2020 deep learning track. In *TREC 2020: Proceedings of the Twenty-Ninth Text REtrieval Conference*. NIST Special Publication 1266, 2021. URL https://trec.nist.gov/pubs/trec29/papers/Overview.DL.pdf.

N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and J. Lin. Overview of the TREC 2021 deep learning track. In *TREC 2021: Proceedings of the Thirtieth Text REtrieval Conference*. NIST Special Publication, 2022. URL https://trec.nist.gov/pubs/trec30/papers/Overview-DL.pdf.

M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 1301–1310. ACM, 2016a. URL https://doi.org/10.1145/2983323.2983814.

M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. On horizontal and vertical separation in hierarchical text classification. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, pages 185–194. ACM, 2016b. URL https://doi.org/10.1145/2970398.2970408.

T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11–15, 2022*, pages 2353–2359. ACM, 2022. URL https://doi.org/10.1145/3477495.3531857.

D. Hiemstra, S. E. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 178–185. ACM, 2004. URL https://doi.org/10.1145/1008992.1009025.

J. Kamps, D. Rau, and N. Kondylidis. Impact of tokenization, pretraining task, and transformer depth on text ranking. In *The Twenty-Ninth Text REtrieval Conference Proceedings (TREC 2020)*. National Institute for Standards and Technology. NIST Special Publication 1266, 2021. URL https://trec.nist.gov/pubs/trec29/papers/UAmsterdam.DL.pdf.

R. Kaptein, D. Hiemstra, and J. Kamps. How different are language models and word clouds? In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings*, volume 5993 of *Lecture Notes in Computer*

Science, pages 556–568. Springer, 2010. URL https://doi.org/10.1007/978-3-642-12275-0_48.

D. Rau and J. Kamps. Recall aspects of transformers for text ranking. In E. M. Voorhees and A. Ellis, editors, *The Thirtieth Text REtrieval Conference Proceedings (TREC 2021)*. National Institute for Standards and Technology. NIST Special Publication 2022, 2022a. URL https://trec.nist.gov/pubs/trec30/papers/UAmsterdam-DL.pdf.

D. Rau and J. Kamps. The role of complex NLP in transformers for text ranking. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022*, pages 153–160. ACM, 2022b. URL https://doi.org/10.1145/3539813.3545144.

S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*, 27(3):129–146, 1976. URL https://doi.org/10.1002/asi.4630270302.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, 24(5):513–523, 1988. URL https://doi.org/10.1016/0306-4573(88)90021-0.