

How important is Recall for Measuring Retrieval Quality?

Shelly Schwartz, Oleg Vasilyev, Randy Sawaya

Primer Technologies Inc.

San Francisco, California

shelly.schwartz,oleg,randy.sawaya@primer.ai

Abstract

In realistic retrieval settings with large and evolving knowledge bases, the total number of documents relevant to a query is typically unknown, and recall cannot be computed. In this paper, we evaluate several established strategies for handling this limitation by measuring the correlation between retrieval quality metrics and LLM-based judgments of response quality, where responses are generated from the retrieved documents. We conduct experiments across multiple datasets with a relatively low number of relevant documents (2–15). We also introduce a simple retrieval quality measure that performs well without requiring knowledge of the total number of relevant documents.

1 Introduction

As an important and well-known example of information retrieval, Retrieval-Augmented Generation(RAG) involves gathering a set of query-relevant documents (or chunks of documents) from a knowledge base (KB), then sending the highest ranked K documents to a Large Language Model (LLM) in order to generate a response to a particular query. In an ideal scenario, the majority of the top K documents would be relevant to the query in addition to making up a significant amount of the total relevant documents in the KB.

A common measure of the quality of the top K selection is the F measure, a harmonic weighted average of precision and recall (Rijsbergen, 1977; Van Rijsbergen, 1979). In a real, changing KB the total number of relevant documents is not known in advance for an arbitrary query, meaning that the recall is also not known (Webber, 2012; Yilmaz and Aslam, 2006; Kutlu et al., 2018; Shokouhi et al., 2006; Fan et al., 2022; Ferrante et al., 2021; Soboroff, 2007; Nguyen et al., 2018; Zobel, 1998).

There are several ways to circumvent the lack of knowledge required to calculate recall:

1. Creating a fixed benchmark: Given a set of queries, a corresponding set of relevant documents is known and makes up a subset of the KB. The subset is fixed and must be recreated when substantial changes are made to the KB.
2. Estimating the total number of relevant documents for each query: This can be done by (1) attempting to extract as many relevant documents as possible (known as pooling) (Upadhyay et al., 2024; Abbasiantaeb et al., 2024; Takehi et al., 2025; Ganguly and Yilmaz, 2023), (2) extrapolating the count of relevant documents to higher K (Rathee et al., 2025; Schmidt et al., 2025; Krishna et al., 2025) or (3) comparing two different retrieval methods (known as capture–recapture) (Fraysse et al., 2023; Mordido and Meinel, 2020).
3. Using an evaluation measure that does not include the total number of relevant documents.

We investigate the relative quality of a few simple measures which utilize these approaches. For example, $nDCG$ ((Järvelin and Kekäläinen, 2000, 2002)) does not use the total number of relevant (“positive”) documents N_p . In (Valcarce et al., 2020) $nDCG$ was noted for high discriminative power for top- K recommendations and, in (Radlinski and Craswell, 2010), for having strong correlations with interleaving.

We use several datasets with a low $N_p \in [2, 15]$ and K covering a range around N_p as well as a few common, small embeddings for retrieval. We judge the measures by their correlations with the quality of LLM response to the query as both the measures and the response use the top K retrieved documents. This is illustrated in Figure 1 and explained in the next section.

In related work, the recall was noted as more important than precision (Li and Ouyang, 2025). It is also known that the irrelevant documents in the top- K do cause problems (Amiraz et al., 2025;

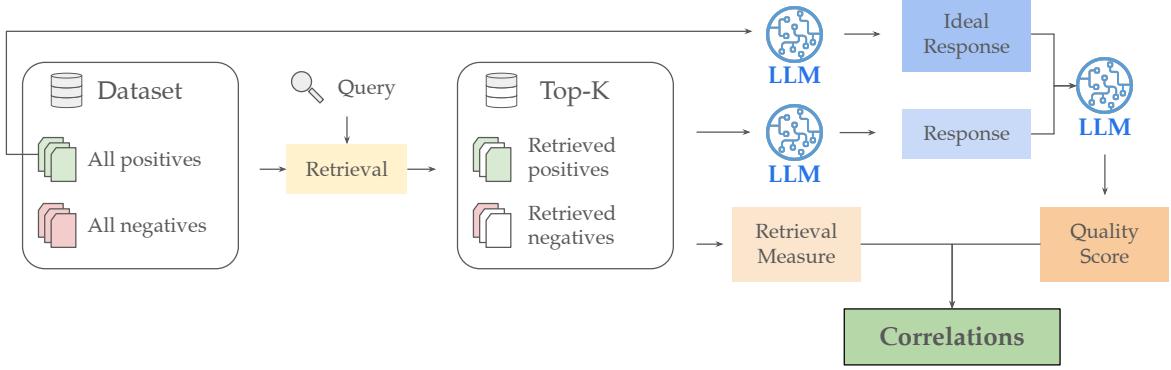


Figure 1: Obtaining a correlation between a measure of quality of retrieval and LLM judgment score.

Mansurova et al., 2024) and the order of the top- K texts can be important (Ma, 2025; Guo et al., 2024; Xia et al., 2025).

Our contribution:

1. We introduce a simple measure “ T ”(as in Top- K selection) independent of the total number N_p of KB documents relevant to the query; we argue the measure is well founded and convenient for real-life evaluations.
2. Taking a diverse selection from well established datasets, we compose a dataset consisting of queries, positives, negatives, LLM-generated answers from the retrieved results and LLM-generated scores of quality of the generated results. The retrieval is done by several popular embedding models.
3. Using the above dataset, we obtain and review the correlations of F , F_e (F with a simple estimate of N_p), $nDCG$ and T measures with the LLM quality score. We gather insights on the relative weaknesses and strengths of each of these measures.

2 Setup

2.1 Measures

As a benchmark measure we use the well known F measure (Van Rijsbergen, 1979)

$$F = \frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}} \quad (1)$$

with a parameter $\alpha \in [0, 1]$. Here $P = \frac{n_p}{K}$ is the precision and $R = \frac{n_p}{N_p}$, the recall where n_p represents the number of positives (relevant documents) within the top K selection. This is equivalent to

$$F = \frac{n_p}{\alpha K + (1 - \alpha) N_p} \quad (2)$$

More commonly F is used with the parametrization $\beta^2 = \frac{1}{\alpha} - 1$.

We also consider “ F_e ” (F -estimated) - equivalent to F , but with N_p restricted to the number of positives in the top $2K$ rather than the total number of positives. This can be considered as a simple estimate of N_p , especially useful for low N_p , when an estimate by an exact extrapolation to large K or by a capture-recapture method may be too crude.

For the case of not knowing (or estimating) N_p , we consider $nDCG$ (Järvelin and Kekäläinen, 2000, 2002; Wang et al., 2013). We also suggest a simple measure

$$T = (1 - \alpha)n_p - \alpha \frac{n_n}{K} \quad (3)$$

where $n_n = K - n_p$ is the number of negative (irrelevant) documents within K . Similarly to F , T includes a dependency on the parameter α . For F , the parameter α may be selected for a desired weighted tradeoff between the precision and recall; for T it gives a tradeoff between importance of positives and negatives within K . Alternatively, α for T can be viewed as weighting a tradeoff between the precision and the absolute number n_p of the selected relevant documents. If n_p is replaced by $\frac{n_p}{N_p}$ in Equation 3, then T becomes an arithmetic weighted average of the precision and recall, similar to F being a harmonic weighted average of the precision and recall. Thus, the motivation for T , as it is defined by Equation 3, is simple removal of unknown N_p .

An unnormalized version of T

$$T_u = (1 - \alpha)n_p - \alpha n_n \quad (4)$$

is simpler and equivalent to T (up to reparameterization by α), but we do not have a motivation to remove the normalization of n_n by K . If K

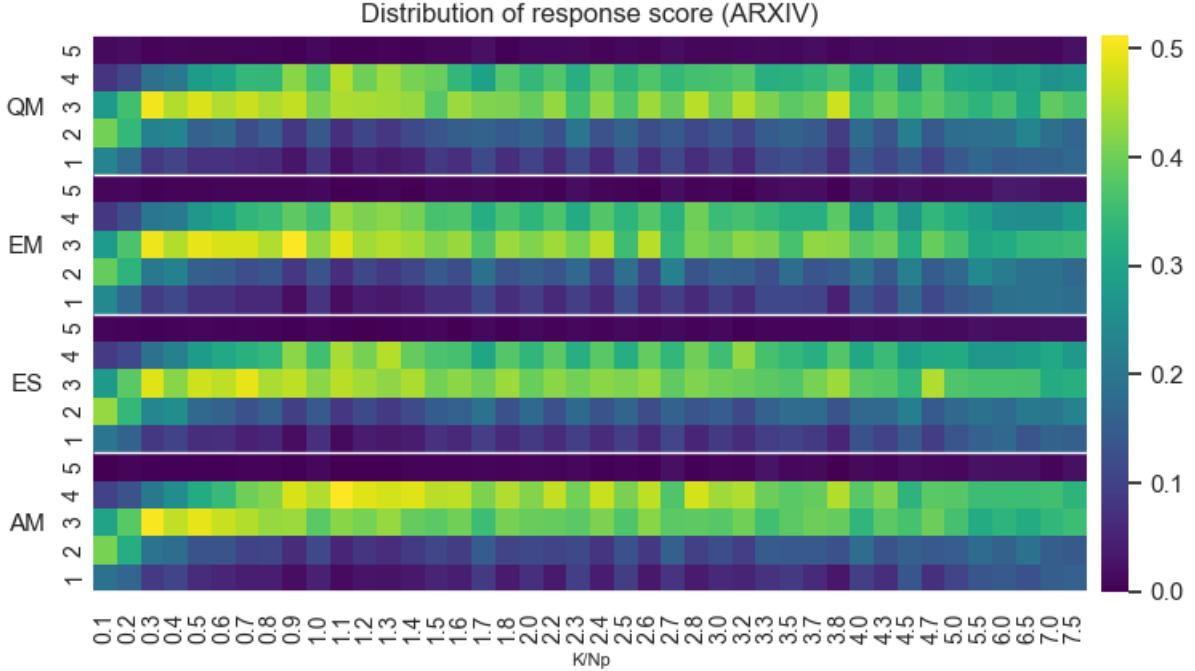


Figure 2: Distribution of the response score (1 to 5) for embedding models shown on Y-axis. On ARXIV; using only segments with minimum 300 samples, the ratio $\frac{K}{N_p}$ is rounded to the first digit.

is changing (for example, in an empirical setting, depending on amount of retrieved texts or their vicinity to the query), the normalization should be helpful. Indeed, in our observations we will see that T_u is not as good a measure as T .

2.2 Embeddings

We use the following embedding models (the abbreviated notations are for our plots):

1. AM = all-MiniLM-L12-v2¹ (Reimers and Gurevych, 2019)
2. EM = multilingual-e5-small² (Wang et al., 2024)
3. ES = e5-small-v2³ (Wang et al., 2022)
4. QM = multi-qa-MiniLM-L6-cos-v1⁴

All these embeddings have size 384.

2.3 Datasets

We sample several well-known datasets to create a single, combine benchmark which we refer to as retrieval-response⁵ (the short notations are for some of our plots):

¹<https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2>

²<https://huggingface.co/intfloat/multilingual-e5-small>

³<https://huggingface.co/intfloat/e5-small-v2>

⁴<https://huggingface.co/sentence-transformers/multi-qa-MiniLM-L6-cos-v1>

⁵<https://huggingface.co/datasets/primer-ai/retrieval-response>

1. A = ARXIV (snapshot 173)⁶ ⁷
2. Hp = HotpotQA paragraphs
3. Hs = HotpotQA sentences
4. M = MSMARCO (Bajaj et al., 2018)
5. N = Natural Questions (Kwiatkowski et al., 2019)

The HotpotQA dataset (Yang et al., 2018) is used with two different granularities: the positives and negatives (by the relevance to a query) are either (1) paragraphs (samples “Hp”) or (2) sentences (samples “Hs”). The full details of the combined dataset are in Appendix A.1.

In practice, it is generally not known which choice of K will be optimal for a given retrieval. For different queries, it may happen that $K = n_p + n_n$ is either less or greater than the total number N_p of positives. In what follows, we attempt to vary K between values above and below N_p when we have sufficient data.

2.4 Quality judgment

We assess the quality of each measure by how well it correlates with an LLM-scored quality of a response generated by LLM from a set of top K documents. For each entry in our dataset, the response

⁶https://huggingface.co/datasets/arxiv-community/arxiv_dataset

⁷<https://www.kaggle.com/datasets/Cornell-University/arxiv>

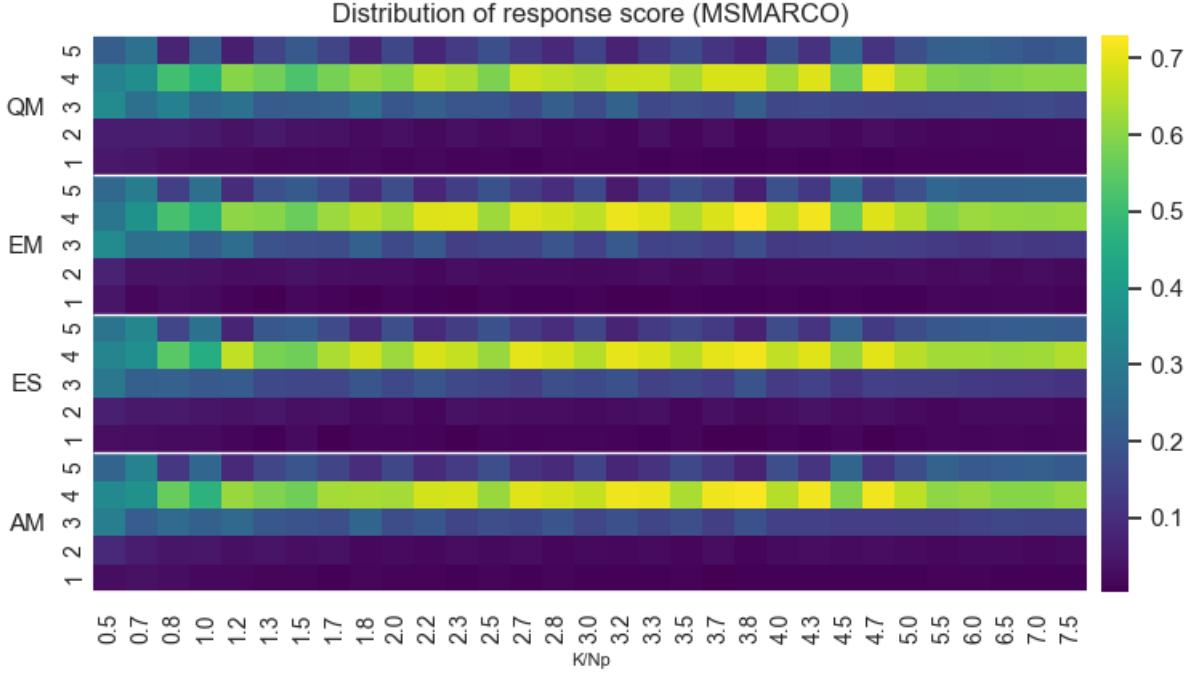


Figure 3: Distribution of the response score (1 to 5) for embedding models shown on Y-axis. On MSMARCO; using only segments with minimum 300 samples, the ratio $\frac{K}{N_p}$ is rounded to the first digit.

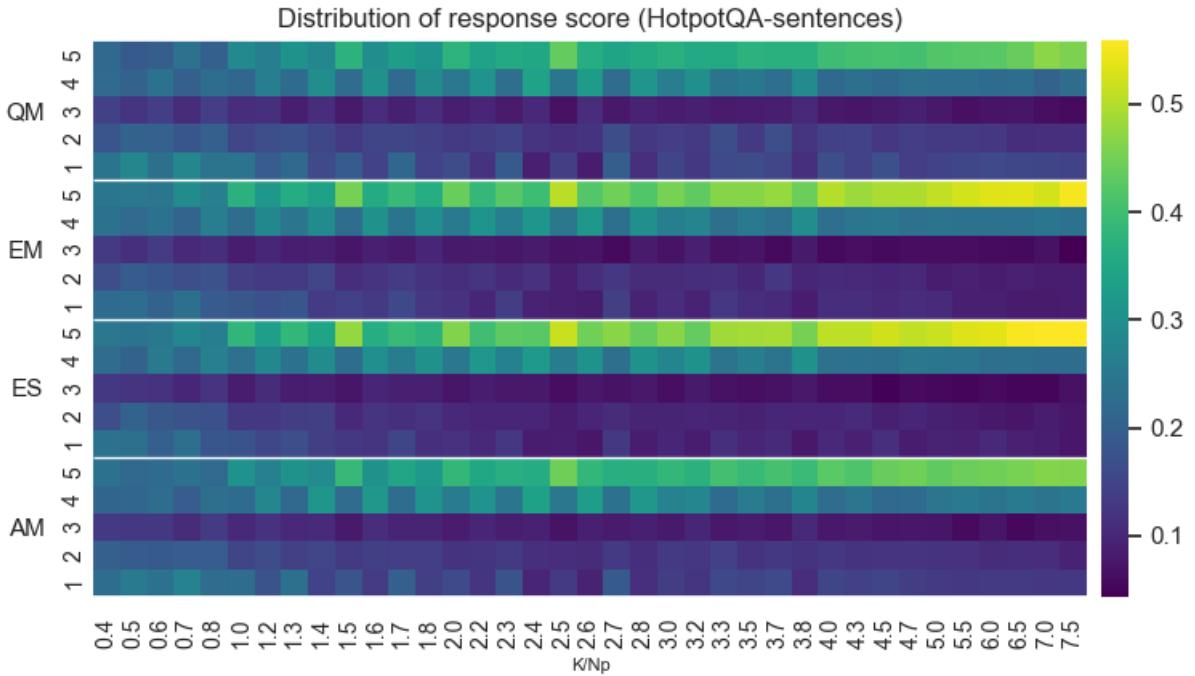


Figure 4: Distribution of the response score (1 to 5) for embedding models shown on Y-axis. On HotpotQA-sentences; using only segments with minimum 300 samples, the ratio $\frac{K}{N_p}$ is rounded to the first digit.

is compared to an “ideal” one, which is based on all the positive documents (see Figure 1). This comparison forms the basis of the quality score (Linkert scale 1 to 5, see Appendices A.2 and A.3). We use GPT-4o-mini for both generating and scoring the

responses. A higher correlation with the score provides better support for using one measure over another. For scores which use α (such as F and T), the correlations are given by the maximal values as α is varied within a subset of the data.

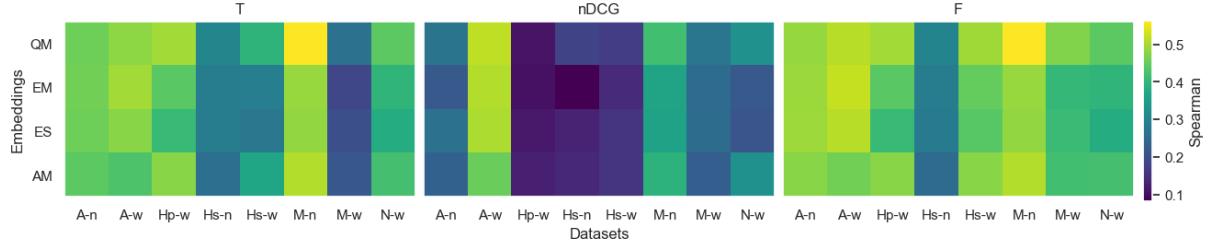


Figure 5: Spearman correlation between the retrieval measures (T , $nDCG$ and F) and the response score.

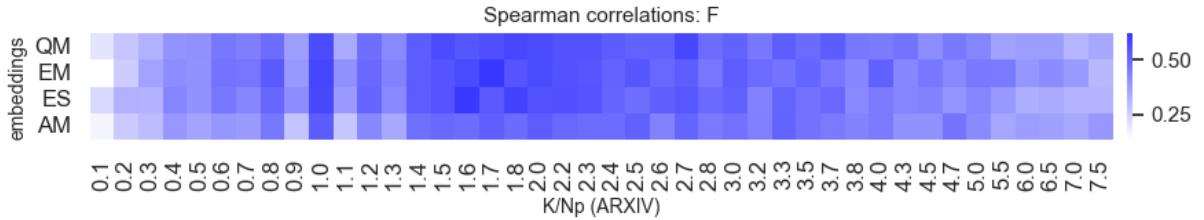


Figure 6: Spearman correlation between F and the response score, on ARXIV.

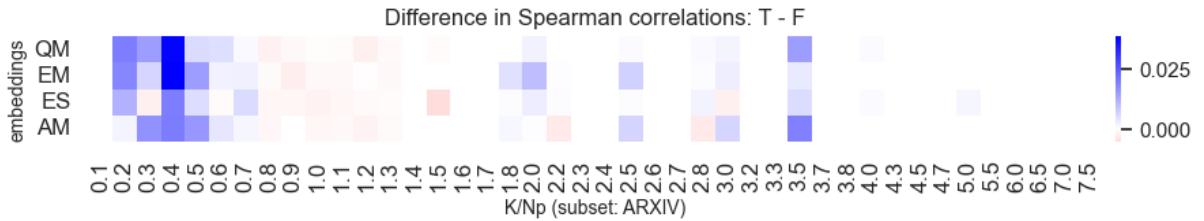


Figure 7: Difference between the Spearman correlations: T -response minus F -response. On ARXIV; using only segments with minimum 300 samples, the ratio $\frac{K}{N_p}$ is rounded to the first digit.

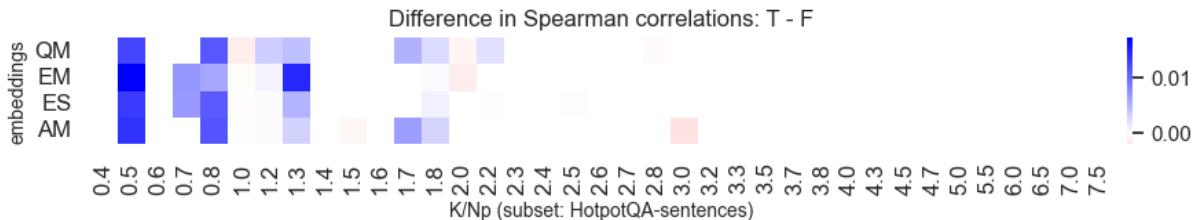


Figure 8: Difference between the Spearman correlations: T -response minus F -response. On HotpotQA-sentences.

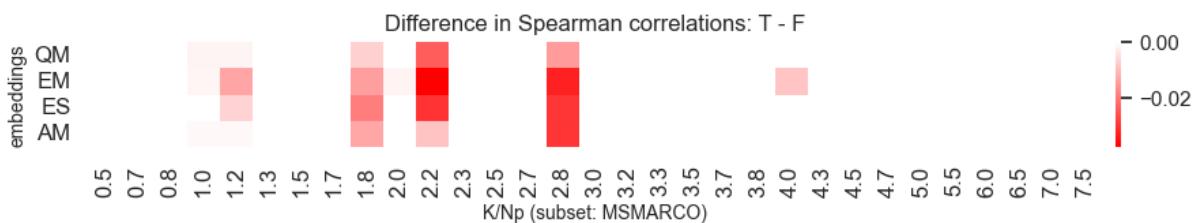


Figure 9: Difference between the Spearman correlations: T -response minus F -response. On MSMARCO.

3 Observations

3.1 Response score distribution

The distribution of the response score is shown for ARXIV in Figure 2. We split the dataset into

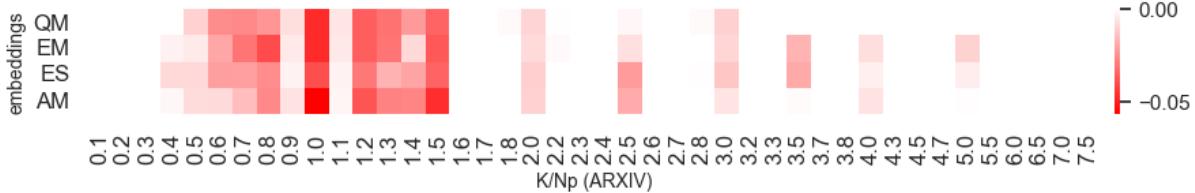


Figure 10: Difference between the Spearman correlations: T_u -response minus T -response. On ARXIV.

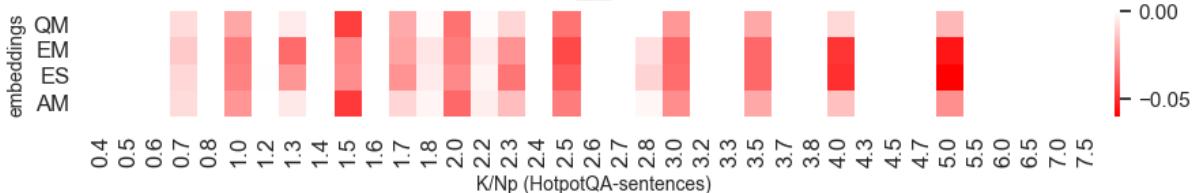


Figure 11: Difference between the Spearman correlations: T_u -response minus T -response. On HotpotQA-sentences.

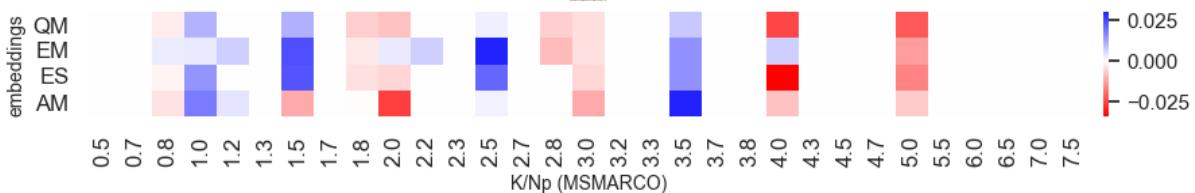


Figure 12: Difference between the Spearman correlations: T_u -response minus T -response. On MSMARCO.

subsets according to the ratio K/N_p , rounded to the first digit, and retain only the subsets which have at least 300 samples. In each subset (and for each embedding model) the sum of the values (5 vertical cells in the heatmap) equals 1.

We see that for ARXIV, the best scored responses occur at values of K which are comparable to N_p . At lower values of K/N_p , there may not be enough positive texts to give a good response. At high K/N_p there may be too many negative texts within K , which can confuse the LLM and lead to lower scores. For ARXIV specifically, these scores may be exacerbated due to the difficulty of the dataset.

Figure 3 shows a similar pattern for MSMARCO, but with a higher “optimal” ratio of $\frac{K}{N_p}$ than for ARXIV. We speculate the reason for this is that MSMARCO contains texts which are simpler to understand relative to ARXIV. Therefore the detriment which comes from adding extra negatives is outweighed by the increasing likelihood of capturing more positives. This is demonstrated to a further extent with HotpotQS-sentences (Figure 4, which must be even simpler for the LLM to understand given the higher “optimal” value of K/N_p .

Our observations for the Natural Questions and HotpotQA-paraphrases datasets are similar to HotpotQA-sentences, but have fewer available segments; they are shown in Appendix B.

3.2 Correlations

Spearman correlations of F , $nDCG$ and T with the response score are shown in Figure 5. The datasets defined in Section 2.3 are split here into “narrow” (suffix “-n”) and “wide” (suffix “-w”), with samples having $K < N_p$ and $K \geq N_p$ respectively. The Pearson and Kendall Tau correlations have similar trends and are given in Appendix C.

From Figure 1 we can see that for many of the segments considered, the T measure does not lose much in terms of correlations compared to F even though it is formulated without N_p .

For a more detailed analysis, we consider more refined segments of data - splitting the datasets by the ratio $\frac{K}{N_p}$, rounded to the first digit, as described in Section 3.1. Examples of the correlations from ARXIV on such segments are shown in Figure 6 for the F measure. The level of correlations between F and the response score is similar for other datasets (Appendix C.2). The correlations between

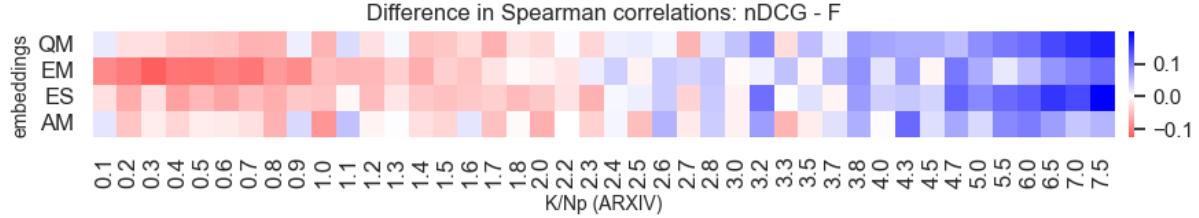


Figure 13: Difference between the Spearman correlations: $nDCG$ -response minus F -response. On ARXIV.

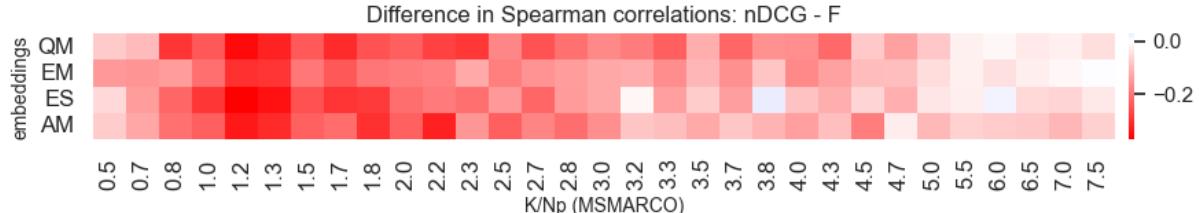


Figure 14: Difference between the Spearman correlations: $nDCG$ -response minus F -response. On MSMARCO.

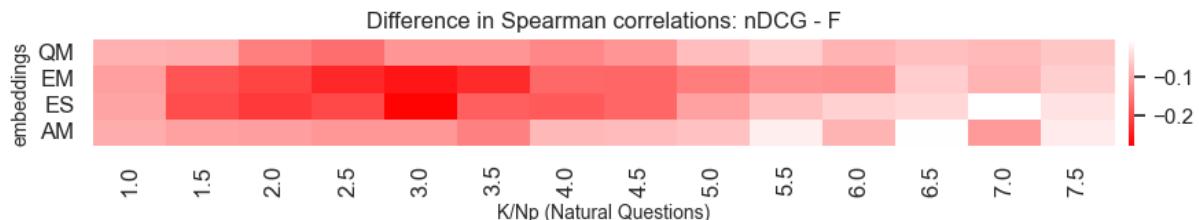


Figure 15: Difference between the Spearman correlations: $nDCG$ -response minus F -response. On Natural Questions.

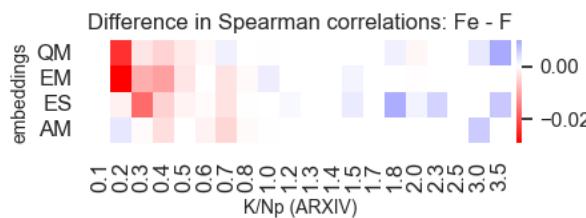


Figure 16: Difference between the Spearman correlations: Fe_e -response minus F -response. On ARXIV.

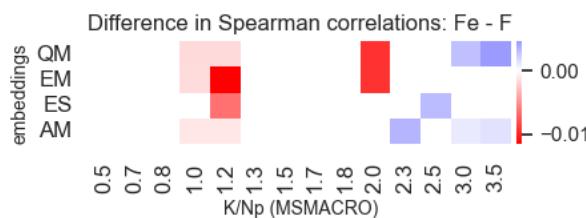


Figure 17: Difference between the Spearman correlations: Fe_e -response minus F -response. On MSMARCO.

T and the response score are also not very different (Appendix C.2). For this reason, in the next Section we show the differences between the correlations.

3.3 Differences between measures

Figures 7, 8 and 9 show the difference between the correlations of T with the response score and F with the response score. The difference depends more on the data than on the embeddings. Specifically, for ARXIV and HotpotQA-sentences, the T measure correlates better with the response score than F on most segments; for MSMARCO it is worse. The results for HotpotQA-paraphrases and Natural Questions are not shown because there is no noticeable difference for them.

The simpler, unnormalized version of T (T_u , Equation 4) has worse correlations with the response quality score on ARXIV (Figure 10) and on HotpotQA-sentences (Figure 11), has mixed and less difference on MSMARCO (Figure 12) and has no noticeable difference on HotpotQA-paraphrases and on Natural Questions.

For the majority of the datasets, $nDCG$ shows worse correlations relative to F , but gets better for high $\frac{K}{N_p}$ in ARXIV (Figure 13).

We suggest the reason lies in the complexity of the ARXIV abstracts, which yields greater benefits

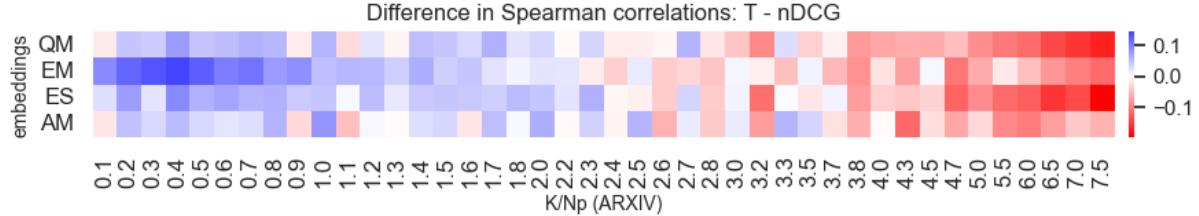


Figure 18: Difference between the Spearman correlations: Q -response minus $nDCG$ -response. On ARXIV.

when the top K findings are presented to the LLM in an easier order: first relevant, then non-relevant. The order becomes especially important for high $\frac{K}{N_p}$ ratio, when there are inevitably many negative (irrelevant) items within top K . A similar trend can be seen for MSMARCO (Figure 14) and Natural Questions (Figure 15); however for these datasets, $nDCG$ is still worse than F . The trend is less pronounced for HotpotQA (see Figures 36 and 37 and Appendix D), possibly because HotpotQA-sentences samples are too simple and because the range of K/N_p for HotpotQA-paragraphs is too narrow.

In Figures 16 and 17, we see similar trend appearing for F_e (F with N_p replaced by a crude estimate $-n_p$ within $2K$) when evaluating on ARXIV and MSMARCO. The reason here, we assume, is that the estimate of N_p , however primitive, is still more reliable at higher ratios $\frac{K}{N_p}$, simply because the top $2K$ documents have more positives. The comparisons on other datasets are more limited, and shown in Figures 38 and 39 in Appendix D (no noticeable difference on HotpotQA-paragraphs, not shown).

It is surprising, however, that F_e at high ratios of $\frac{K}{N_p}$ can become even better than F , meaning that an estimated N_p serves better than the real N_p . We suspect the reason is that the relevant documents falling into the top $2K$ (used in our estimation of N_p) are more important and should “weigh more” than other relevant documents when counting N_p . Because these documents happen to be closer to the query (by embedding similarity), it is likely they also would be easier for the LLM and more useful, especially at high $\frac{K}{N_p}$.

As we already observed (Figures 13, 14, 15, 36, 37), $nDCG$ is mostly worse than F , except at high $\frac{K}{N_p}$ values for ARXIV - when the content is difficult and the order of numerous negatives (irrelevant documents) within top K is important for the LLM. Since T is better or not too different from F (Figures 7, 8 and 9), T is also better than $nDCG$ at not too high $\frac{K}{N_p}$ values for ARXIV (Figure 18); it

is also better than $nDCG$ for other datasets (Figures 41, 40, 42 and 43 in Appendix D).

4 Conclusion

We considered a few measures representing different approaches to using N_p , the total count of documents relevant to a query. Measure F requires knowing N_p , which can be achieved by labeling and fixing part of KB. Measures T (suggested here) and $nDCG$ do not use N_p . Measure F_e uses a crudely estimated value for N_p . We compared how well these measures correlate with a quality score, assigned (by an LLM) to a response generated (by an LLM) from the top K retrieved documents. Our conclusions:

1. At high $\frac{K}{N_p}$ ratios, the order of documents within top- K becomes too important for the LLM, and $nDCG$ becomes better than F or T .
2. At $\frac{K}{N_p}$ ratios comparable to 1 or lower, the T measure is better than $nDCG$. For ratios below 1, it is also better or the same as F , depending on the type of the documents.
3. At high $\frac{K}{N_p}$ ratios, substituting a simple estimate of N_p from a $2K$ selection instead of the real N_p not just approximates F , but even improves it. We suggest the reason is that the relevant documents ranked higher (and therefore falling into the top $2K$) would be more important for the LLM response than the rest of the relevant documents.
4. Within our observations, the document type and the ratio $\frac{K}{N_p}$ appear to be more important than the choice of an embedding model when selecting a retrieval measure (and simply for the response score).

Of course in practice, the range of $\frac{K}{N_p}$ depends on the distribution of N_p in the used data, the choice of K (which may be varied at runtime depending on a query and the retrieved texts) and on a consideration of computation expenses and time lag (growing with K).

Limitations

Our experiments focus on retrieval scenarios with relatively moderate ranges of N_p , which may limit the applicability of our findings to broader retrieval regimes.

We left out a consideration of the additional latency incurred when evaluating the results of a retrieval in realtime. The measures (even not involving a knowledge of N_p) require identifying relevancy of the top- K documents with respect to a query. Outside of using a fixed query with a corresponding set of labeled documents, one direct method of obtaining labels is by using an LLM to assess the relevancy of each document to the query after making the top- K selection. This is particularly useful in a retrieval system monitoring setting, where any additional time spent measuring the quality of a retrieval will not block any downstream tasks—the system simply continues on while the results are recorded in the background.

References

- Zahra Abbasiantaeb, Chuan Meng, Leif Azzopardi, and Mohammad Aliannejadi. 2024. [Can we use large language models to fill relevance judgments?](#) *arXiv*, arXiv:2405.05600.
- Chen Amiraz, Florin Cuconasu, Simone Filice, and Zohar Karnin. 2025. [The distracting effect: Understanding irrelevant passages in RAG](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18228–18258, Vienna, Austria. Association for Computational Linguistics.
- 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*, arXiv:1611.09268.
- Yan Fan, Yang Liu, Yukun Liu, and Jing Qin. 2022. [Nearly optimal capture-recapture sampling and empirical likelihood weighting estimation for M-estimation with big data](#). *arXiv*, arXiv:2209.04569.
- Marco Ferrante, Nicola Ferro, and Norbert Fuhr. 2021. [Towards meaningful statements in IR evaluation: mapping evaluation measures to interval scales](#). *arXiv*, arXiv:2101.02668.
- Rémi Fraysse, Rémi Choquet, Carlo Costantini, and Roger Pradel. 2023. [Population size estimation with capture-recapture in presence of individual misidentification and low recapture](#). *arXiv*, arXiv:2304.00885.
- Debasis Ganguly and Emine Yilmaz. 2023. [Query-specific variable depth pooling via query performance prediction towards reducing relevance assessment effort](#). *arXiv*, arXiv:2304.11752.
- Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. [What makes a good order of examples in in-context learning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904, Bangkok, Thailand. Association for Computational Linguistics.
- Kalervo Järvelin and Jaana Kekäläinen. 2000. [IR evaluation methods for retrieving highly relevant documents](#). In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’00*, page 41–48, New York, NY, USA. Association for Computing Machinery.
- Kalervo Järvelin and Jaana Kekäläinen. 2002. [Cumulated gain-based evaluation of IR techniques](#). *ACM Trans. Inf. Syst.*, 20(4):422–446.
- Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. [Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, Albuquerque, New Mexico. Association for Computational Linguistics.
- Mucahid Kutlu, Tamer Elsayed, and Matthew Lease. 2018. [Intelligent topic selection for low-cost information retrieval evaluation: A new perspective on deep vs. shallow judging](#). *Information Processing & Management*, 54(1):37–59.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Xiangci Li and Jessica Ouyang. 2025. [How does knowledge selection help retrieval augmented generation?](#) *ArXiv*, arXiv:2410.13258.
- Jing Ma. 2025. [Input order shapes LLM semantic alignment in multi-document summarization](#). *arXiv*, arXiv:2512.02665.
- Aigerim Mansurova, Aiganysh Mansurova, and Aliya Nugumanova. 2024. [QA-RAG: Exploring LLM reliance on external knowledge](#). *Big Data and Cognitive Computing*, 8(9).
- Gonçalo Mordido and Christoph Meinel. 2020. [Mark-evaluate: Assessing language generation using population estimation methods](#). *arXiv*, arXiv:2010.04606.

- Phuc Nguyen, Deva Ramanan, and Charless Fowlkes. 2018. [Active testing: An efficient and robust framework for estimating accuracy](#). *arXiv*.
- Filip Radlinski and Nick Craswell. 2010. [Comparing the sensitivity of information retrieval metrics](#). In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’10, page 667–674, New York, NY, USA. Association for Computing Machinery.
- Mandeep Rathee, V Venktesh, Sean MacAvaney, and Avishek Anand. 2025. [Breaking the lens of the telescope: Online relevance estimation over large retrieval sets](#). *arXiv*, arXiv:2504.09353.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Van Rijsbergen. 1977. [A theoretical basis for the use of co-occurrence data in information retrieval](#). *Journal of Documentation*, 33(2):106–119.
- Sebastian Schmidt, Prasanga Dhungel, Christoffer Löfller, Björn Nieth, Stephan Günnemann, and Leo Schwinn. 2025. [Effective data pruning through score extrapolation](#). *arXiv*, arXiv:2506.09010.
- Milad Shokouhi, Justin Zobel, Falk Scholer, and S. M. M. Tahaghoghi. 2006. [Capturing collection size for distributed non-cooperative retrieval](#). In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’06, page 316–323, New York, NY, USA. Association for Computing Machinery.
- Ian Soboroff. 2007. [A comparison of pooled and sampled relevance judgments](#). In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’07, page 785–786, New York, NY, USA. Association for Computing Machinery.
- Rikiya Takehi, Ellen M. Voorhees, Tetsuya Sakai, and Ian Soboroff. 2025. [LLM-assisted relevance assessments: When should we ask LLMs for help?](#) In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’25, page 95–105, New York, NY, USA. Association for Computing Machinery.
- Shubham Upadhyay, Ehsan Kamalloo, and Jimmy Lin. 2024. [LLMs can patch up missing relevance judgments in evaluation](#). *arXiv*, arXiv:2405.04727.
- Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2020. [Assessing ranking metrics in top-N recommendation](#). *Inf. Retr.*, 23(4):411–448.
- C.J. Van Rijsbergen. 1979. *Information Retrieval*, second edition. Butterworths, London.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Dixin Jiang, Rangan Majumder, and Furu Wei. 2022. [Text embeddings by weakly-supervised contrastive pre-training](#). *arXiv*, arXiv:2212.03533.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. [Multilingual E5 text embeddings: A technical report](#). *arXiv*, arXiv:2402.05672.
- Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. 2013. [A theoretical analysis of NDCG type ranking measures](#). *arXiv*, arXiv:1304.6480.
- William Webber. 2012. [Approximate recall confidence intervals](#). *arXiv*, arXiv:1202.2880.
- Yuan Xia, Jingbo Zhou, Zhenhui Shi, Jun Chen, and Haifeng Huang. 2025. [Improving retrieval augmented language model with self-reasoning](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25534–25542.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Emine Yilmaz and Javed A. Aslam. 2006. [Estimating average precision with incomplete and imperfect judgments](#). In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM ’06, page 102–111, New York, NY, USA. Association for Computing Machinery.
- Justin Zobel. 1998. [How reliable are the results of large-scale information retrieval experiments?](#) In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’98, page 307–314, New York, NY, USA. Association for Computing Machinery.

A Dataset Retrieval-Response

A.1 Data

The dataset⁸ consists of three parts:

1. Query-texts samples
2. Ranked samples
3. Graded samples

⁸<https://huggingface.co/datasets/primer-ai/retrieval-response>

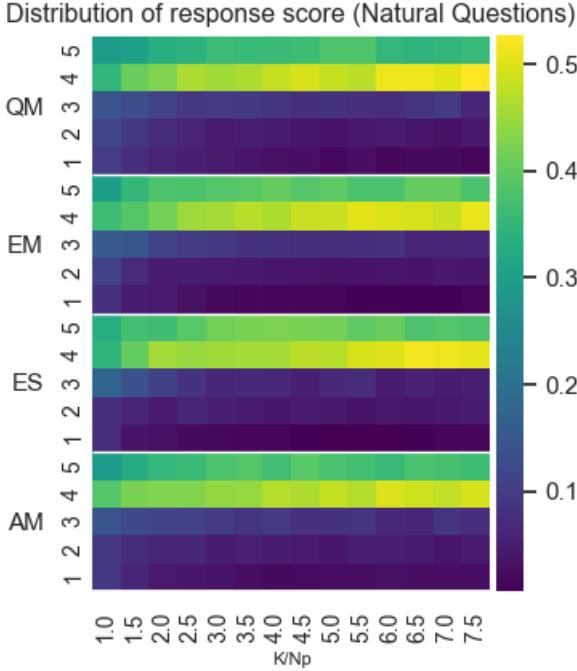


Figure 19: Distribution of the response score (1 to 5) for embedding models shown on Y-axis. On Natural Questions; using only segments with minimum 300 samples, the ratio $\frac{K}{N_p}$ is rounded to the first digit.

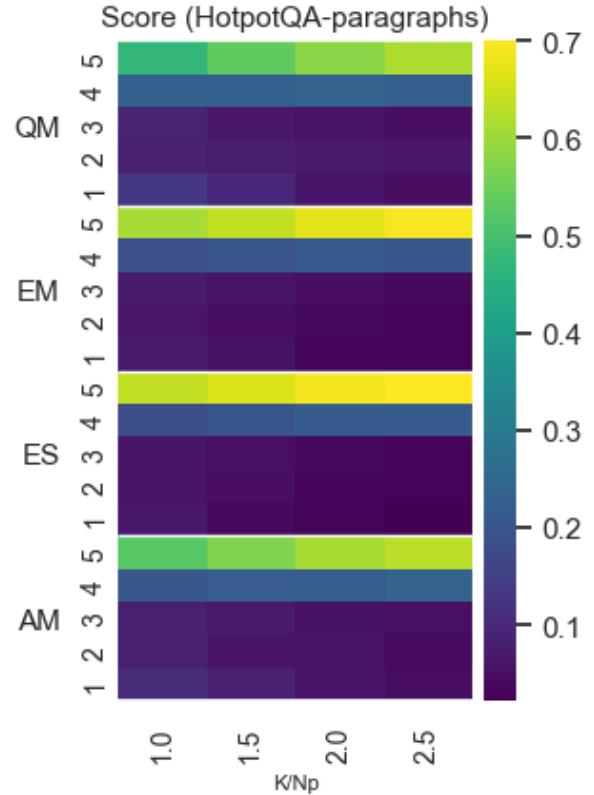


Figure 20: Distribution of the response score (1 to 5) for embedding models shown on Y-axis. On HotpotQA-paragraphs; using only segments with minimum 300 samples, the ratio $\frac{K}{N_p}$ is rounded to the first digit.

A.1.1 Query-texts samples

Query-texts samples consist of 6112 samples, each sample is a dictionary with the following items:

1. “id”: A string Id of the sample. The Id consists of a name of a subset, concatenated by “-” with Id of the item in the subset. For example, Id=“N-5” means that it is sample #5 from the subset Natural Questions. Each sample is uniquely identified by its Id.
2. “q”: The query.
3. “p”: The list of positives.
4. “n”: The list of negatives.

All the subsets in the dataset: “A”, “Hp-e”, “Hp-h”, “Hp-m”, “Hs-e”, “Hs-h”, “Hs-m”, “M”, “N”. The short names, as explained in Section 2, are “A” for ARXIV, “H” for HotpotQA, “M” for MSMARCO and “N” for Natural Questions.

HotpotQA appears with two different granularities: The positives and negatives are (1) paragraphs in “Hp” and (2) sentences in “Hs”. Beyond that, we keep the HotpotQA classification of the queries: easy (suffix “-e”), medium (“-m”) and hard (“-h”).

In the ARXIV sample, a query is a name of an arxiv category. The positives are abstracts of papers in this category and the negatives are abstracts of papers in a related, but different category. We

kept the following additional, not strictly necessary, information, with additional keys:

1. “c1”: The symbolic name of a category for positives. (The name of this category serves as the query and is stored with the key “q”.)
2. “c2”: The symbolic name of a category for negatives.
3. “q2”: The name of the category for negatives (not used).

For example, a sample with id=“A-0” has c1=“math.ca”, c2=“math.pr”, q=“classical analysis and ODEs” and q2=“probability” (it also has a list of positives “p” and a list of negatives “n”).

A.1.2 Ranked samples

Each of the query-texts samples (Appendix A.1.1) can be used as a retrieval example with different number N_c of candidates, number N_p of positives ($N_p < N_c$), number K of top retrieved texts, and the retrieval can be done with different embeddings.

The number of ranked samples is 42992; each of the four embeddings (listed in Section 2) is used in 10748 samples. Each ranked sample is a dictionary

with the following items:

1. “id”: A string Id of the sample, the same as in the query-texts samples.
2. “E”: The embedding’s short notation, as specified in Section 2. The embedding used for ranking of all the candidates and selecting top-K candidates.
3. “Nc”: Total number of candidates (positives and negatives), taken from the corresponding query-texts sample (with the same “id”).
4. “Np”: Total number of positives, taken as the first N_p positives “p” of the corresponding query-texts sample. (Negatives are also taken as first $N_c - N_p$ from the negatives “n”.)
5. “K”: A sorted list of all the K (number of retrieved candidates, “top-K”) used for this sample.
6. “P”: A list of precisions calculated for the top-K specified in the list “K”, in the same order. Has the same length as the list “K”.
7. “R”: A list of recalls calculated for the top-K specified in the list “K”, in the same order.
8. “rank”: A list (length N_c) of indexes of all the candidates, sorted by ranks accordingly to cosine similarities with query, by the embedding “E”.

Each ranked sample is uniquely identified by the tuple (id, E, Nc, Np).

In order to get the ranked texts corresponding to the “rank” list of a ranked sample S_r , its query-texts sample S_q (the sample with the same “id”) can be used as in Figure 21.

To assure an understanding of the data of a ranked sample S_r and of the corresponding query-texts sample S_q , see a few assertions in Figure 22.

A.1.3 Graded samples

The LLM grading is done for each choice of “top-K” in each of the ranked samples. This means that each ranked sample S_r makes $\text{len}(S_r['K'])$ graded samples. In total, this gives 535888 graded samples.

The grading is done as outlined in Section 2; more details and the prompts are in Appendices A.2 and A.3. Each graded sample is a dictionary with the following items, most of which are familiar from the ranked samples:

1. “id”: A string Id of the sample, the same as in the query-texts samples.
2. “E”: The embedding’s short notation.
3. “Nc”: Total number of candidates (positives and negatives).

4. “Np”: Total number of positives.
5. “K”: A value of K (“top-K”) taken from the list of “K” in the corresponding ranked sample.
6. “rank”: A list equal to the first K elements of the list “rank” of the corresponding ranked sample S_r , i.e. $S_r['rank'][0:K]$.
7. “inK”: A list created from the “rank” (the item above), by replacement of each index by 1 (if positive) or 0 (if negative).
8. “answer_ideal”: LLM-generated answer to the query, obtained by using all the positives from the corresponding query-texts sample.
9. “answer_topK”: LLN-generated answer to the query, obtained by using the retrieved K candidates, given to LLM in their ranking order.
10. “grade”: The LLM-generated score (on Linkert scale from 1 to 5), obtained by comparing the top-K answer to the ideal answer, with the knowledge of the query.
11. “P”: A value of precision corresponding to the selected K; given here for convenience.
12. “R”: A value of recall corresponding to the selected K; given here for convenience.

Each graded sample is uniquely identified by the tuple (id, E, Nc, Np, K) and it is related to its ranked sample by the tuple (id, E, Nc, Np). To assure an understanding of the data of a graded sample S_g and of the corresponding ranked sample S_r , see a few assertions in Figure 30.

A.1.4 Dataset content

Each subset of the retrieval-response dataset has comparable amounts of samples with a different number of candidates N_c , total number of positives N_p and choice of K , subject to the availability in the sources we used. Table 1 gives a summary of the dataset content: the number of ranked samples for each embedding, subset, N_c , N_p and a range K_r of the selection (“top-K”) choices K . The embedding model is not specified, because the numbers of samples are the same for each embedding.

Table 2 shows how the samples are split between the narrow ($K < N_p$) and wide ($K \geq N_p$) parts of the subsets.

A.2 Response generation

The following prompt is used during the *Response Generation* phase. Given a query and a set of reference documents, the model must generate a response based only on the provided references, without introducing external knowledge.

System Message

```

Np = Sr['Np']
texts = [(Sq['p'][i] if i<Np else Sq['n'][i-Np]) for i in Sr['rank']]

```

Figure 21: Snippet for getting ranked texts for a ranked sample S_r from the corresponding query-sample S_q .

```

len(Sr['P']) == len(Sr['K'])
len(Sr['R']) == len(Sr['K'])
len(Sr['rank']) == Sr['Nc']
Sr['Nc'] <= len(Sq['p']) + len(Sq['n'])
Sr['Np'] <= len(Sq['p'])
Sr['Np'] < Sr['Nc']
Sr['Nc'] >= 2
Sr['Np'] >= 2

```

Figure 22: Snippet of assertions that can be made for some of the data from a ranked sample S_r and from the corresponding query-texts sample S_q data.

You are an AI assistant that uses reference documents to respond to a given query.

User Message

Please respond to the following query according to the information provided in the reference documents. Be sure to only use what is in the reference documents to respond to the query and nothing else.

Query:

<QUERY>

Reference documents:

<REFERENCES>

A.3 Quality score generation

In the *Quality Score Generation* phase, an LLM evaluator compares a generated response to its ideal reference response. The model returns a discrete score from 1–5, following the rubric below, to measure content completeness and alignment.

System Message

You are an AI assistant who compares a response to its ideal response. Given a query, a response, and an ideal response, determine how close the response is to the ideal response. Return only a single digit (1, 2, 3, 4, or 5) with no explanations.

User Message

RUBRIC

- 1 – The response includes substantially less of the relevant information than the ideal response.
- 2 – The response includes about half of the relevant information present in the ideal response.

	subset	N_c	N_p	K_r	n_r
A	50	2	[2,15]	300	
	50	3	[2,15]	300	
	50	4	[2,15]	300	
	50	5	[2,15]	300	
	50	10	[2,15]	300	
	50	15	[2,20]	300	
	100	5	[2,15]	300	
	100	10	[2,15]	300	
	100	15	[2,20]	300	
	Hp	10	2	[2,5]	900
Hs	10	2	[2,5]	900	
	30	5	[2,15]	719	
	50	2	[2,15]	900	
	50	3	[2,15]	900	
	50	4	[2,15]	803	
	50	5	[2,15]	298	
	30	6	[2,15]	231	
	30	2	[2,15]	300	
	30	3	[2,15]	300	
	30	4	[2,15]	300	
M	30	5	[2,15]	62	
	50	2	[2,15]	300	
	50	3	[2,15]	300	
	50	4	[2,15]	199	
	50	5	[2,15]	36	
	N	20	2	[2,10]	300
	30	2	[2,15]	300	

Table 1: Number n_r of ranked samples for each embedding, subset, N_c , N_p and a range of K . The number of graded samples equals to n_r multiplied by the number of different K in the range K_r . For example, the number of the graded samples in the first row equals $300*(15-2+1)=4200$.

- 3 – The response includes most of the relevant information present in the ideal response.
- 4 – The response includes nearly all relevant information present in the ideal response.
- 5 – The response includes all relevant information present in the ideal response.

Query: [query]

Response: [response]

Ideal Response: [ideal response]

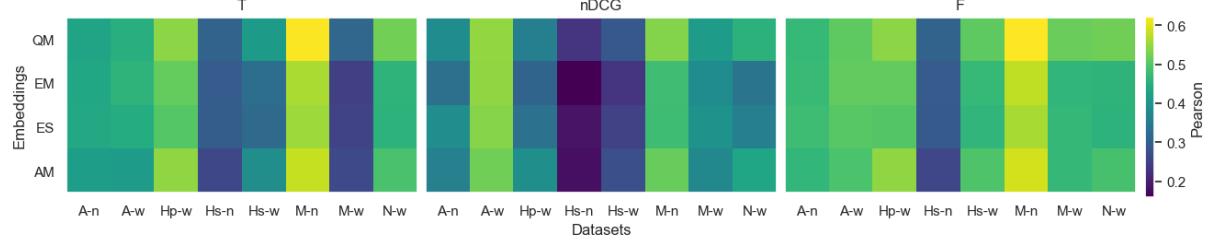


Figure 23: Pearson correlation between the retrieval measures (T , $nDCG$ and F) and the response score.

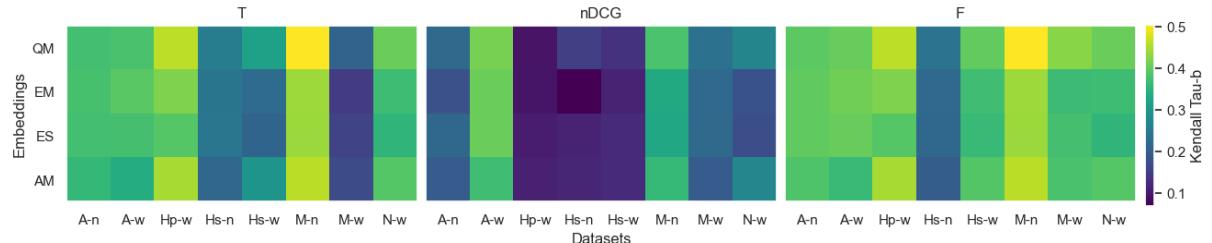


Figure 24: Kendall Tau-b correlation between the retrieval measures (T , $nDCG$ and F) and the response score.

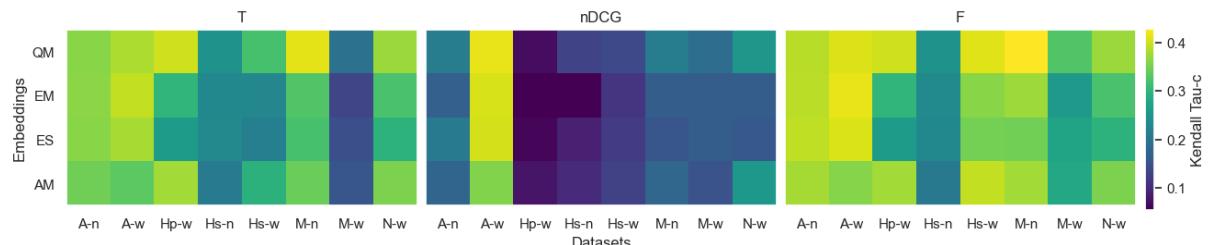


Figure 25: Kendall Tau-c correlation between the retrieval measures (T , $nDCG$ and F) and the response score.

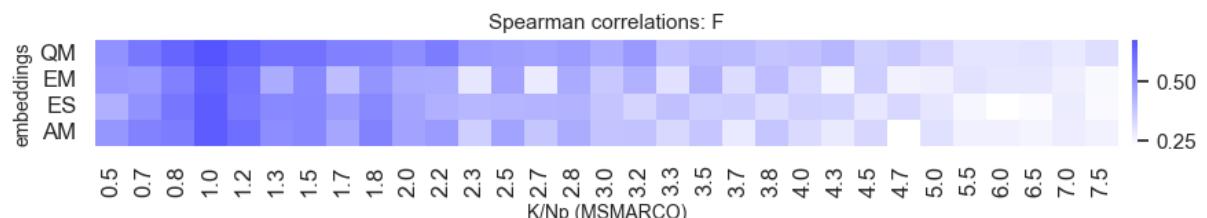


Figure 26: Spearman correlation between F and the response score, on MSMARCO.

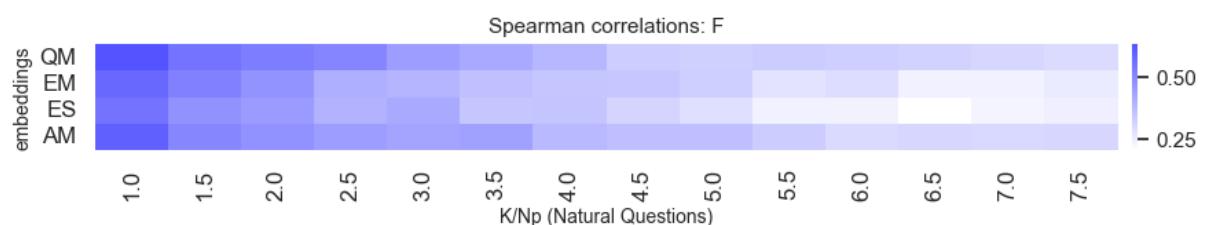


Figure 27: Spearman correlation between F and the response score, on Natural Questions.

B Response Score

In Section 3.1 we have shown the distribution of the response score for ARXIV, MSMARCO and HotpotQA-sentences. Here we show the dis-

tributions for Natural Questions (Figure 19) and HotpotQA-paraphrases (Figure 20). They show the same simple pattern as HotpotQA-sentences: the larger the ratio $\frac{K}{N_p}$, the better the response score,

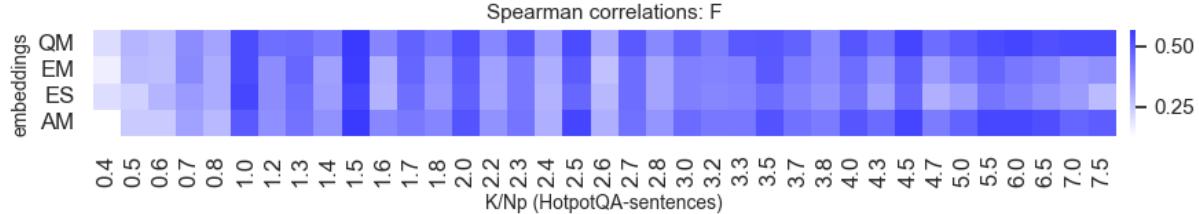


Figure 28: Spearman correlation between F and the response score, on HotpotQA-sentences.

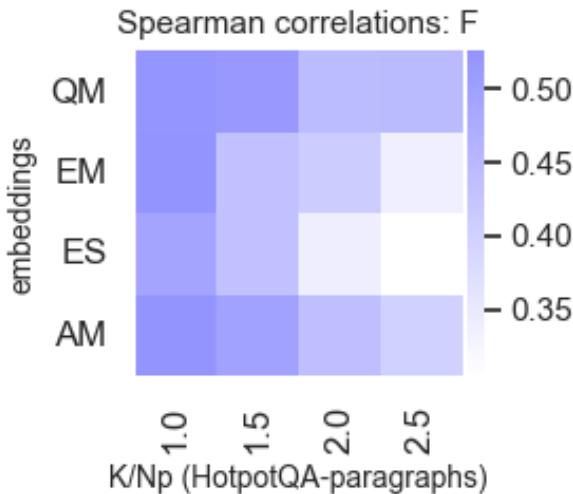


Figure 29: Spearman correlation between F and the response score, on HotpotQA-paraphrases.

```

len(Sg['rank']) == Sg['K']
len(Sg['ink']) == Sg['K']
sum(Sg['ink']) <= Sg['Np']
Sg['K'] in Sr['K']
Sg['rank'] == Sr['rank'][Sg['K']]
    
```

Figure 30: Snippet of assertions that can be made for some of the data from a graded sample S_g and from the corresponding ranked sample S_r data.

meaning that the texts are simple enough for LLM and negative samples are not as important as catching more positives within the range we consider.

Both for generating (Appendix A.2) and scoring the responses we used LLM GPT-4o-mini.

C Correlations with response score

C.1 Narrow and wide subsets

In Section 3.2 the Spearman correlations of the measures T , $nDCG$ and F are shown in Figure 5. Here we show the corresponding Pearson and Kendall Tau correlations in Figures 23, 24, 25.

subset	n_r	n_g	n_{gn}	n_{gw}
A	2700	40800	15300	25500
Hp	900	3600	0	3600
Hs	4751	57514	6481	51033
M	1797	25158	1892	23266
N	600	6900	0	6900

Table 2: For each embedding and subset, the counts: Number n_r of ranked samples; number n_g of graded samples; number n_{gn} of narrow graded samples ($K < N_p$); number n_{gw} of wide graded samples ($K \geq N_p$).

C.2 Subsets by $\frac{K}{N_p}$

In Section 3.2 the Spearman correlations of F with the response score on ARXIV data were shown in Figure 6. Here in Figures 26, 27, 28 and 29 we show the correlations on the other datasets; this level of Spearman correlations is usually considered as moderate.

Figures 31, 32, 33, 34 and 35 show that correlations of T with the response score are similar across the datasets and the ratios $\frac{K}{N_p}$.

D Differences between measures

In this Section we show more heatmaps illustrating the differences between the measures - for the comparisons that were not decisive enough to be put in the main body of the paper.

Comparison of $nDCG$ and F for measure-response correlations on HotpotQA is shown in Figures 36 and 37.

Comparison of F_e and F on HotpotQA-sentences is shown in Figure 38, and on Natural Questions in Figure 39. There is no noticeable difference on HotpotQA-paraphrases (not shown).

Comparisons between T and $nDCG$ was shown for ARXIV in Figure 18. Here it is shown for other datasets in Figures 40, 41, 42, 43.

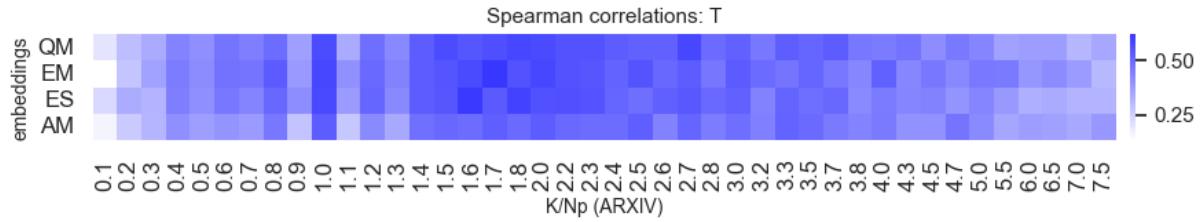


Figure 31: Spearman correlation between T and the response score, on ARXIV.

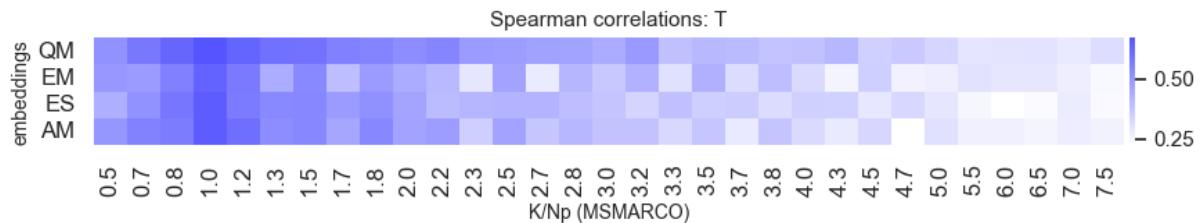


Figure 32: Spearman correlation between T and the response score, on MSMARCO.

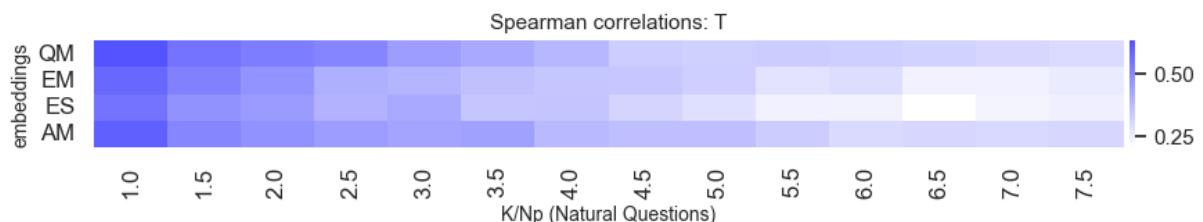


Figure 33: Spearman correlation between T and the response score, on Natural Questions.

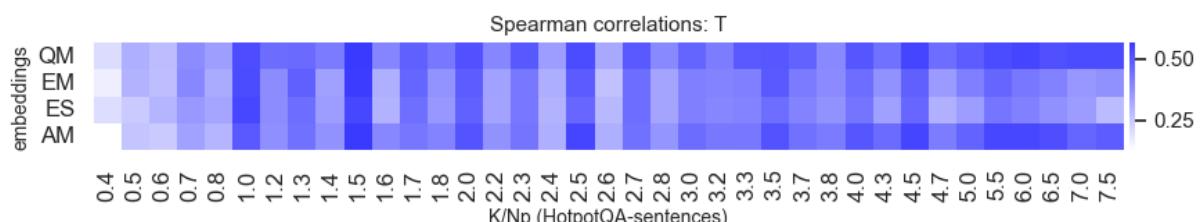


Figure 34: Spearman correlation between T and the response score, on HotpotQA-sentences.

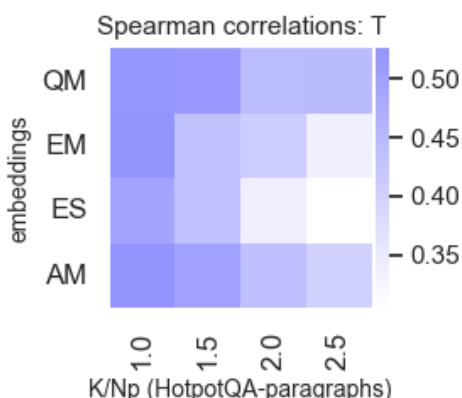


Figure 35: Spearman correlation between T and the response score, on HotpotQA-paragraphs.

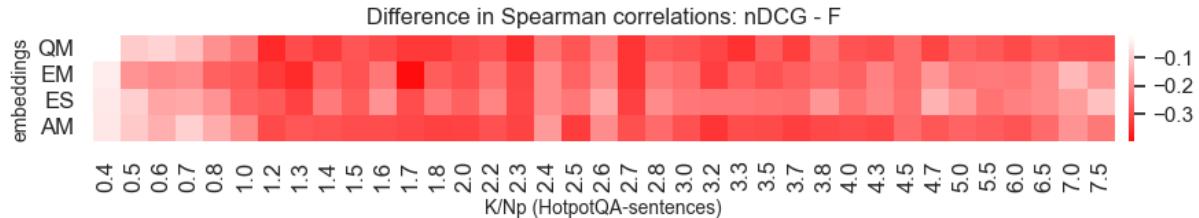


Figure 36: Difference between the Spearman correlations: $nDCG$ -response minus F -response. On HotpotQA-sentences.

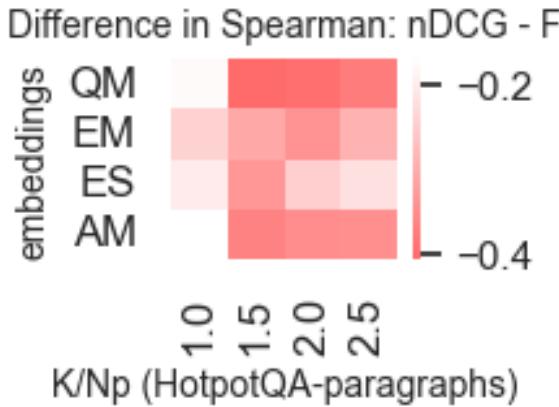


Figure 37: Difference between the Spearman correlations: $nDCG$ -response minus F -response. On HotpotQA-paraphrases.

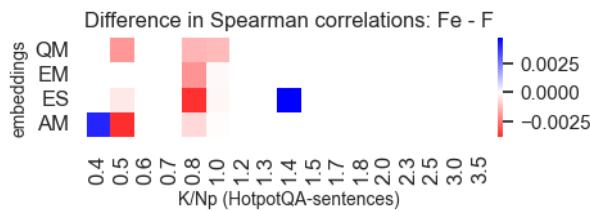


Figure 38: Difference between the Spearman correlations: F_e -response minus F -response. On HotpotQA-sentences.

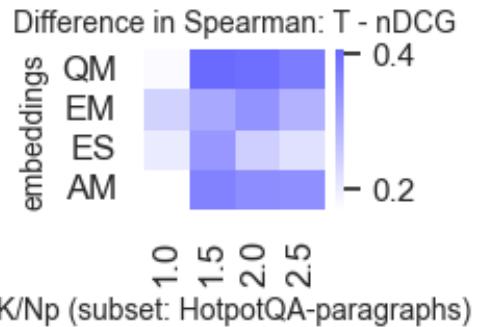


Figure 40: Difference between the Spearman correlations: T -response minus $nDCG$ -response. On HotpotQA-paraphrases.

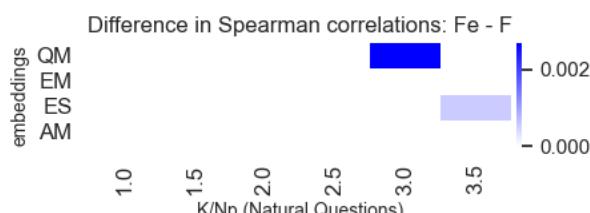


Figure 39: Difference between the Spearman correlations: F_e -response minus F -response. On Natural Questions.

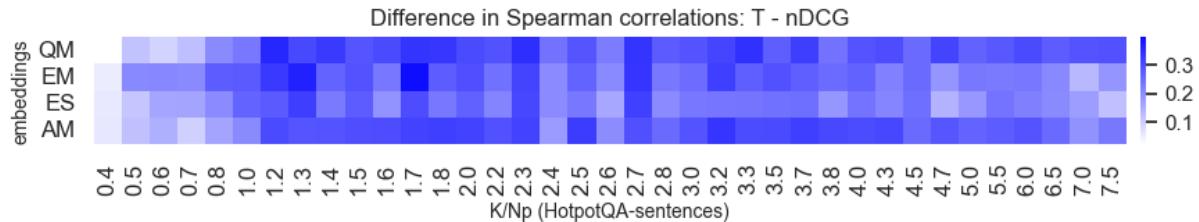


Figure 41: Difference between the Spearman correlations: T -response minus $nDCG$ -response. On HotpotQA-sentences.

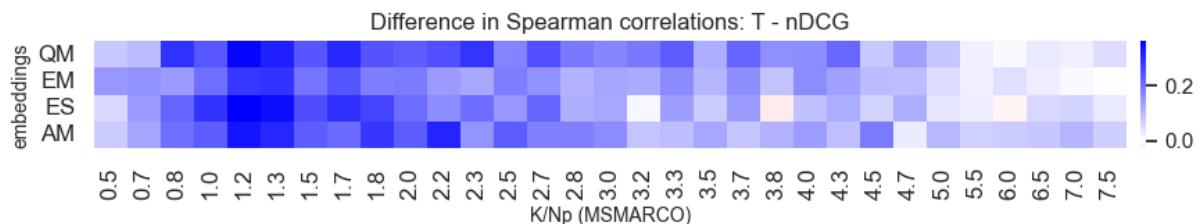


Figure 42: Difference between the Spearman correlations: T -response minus $nDCG$ -response. On MSMARCO.

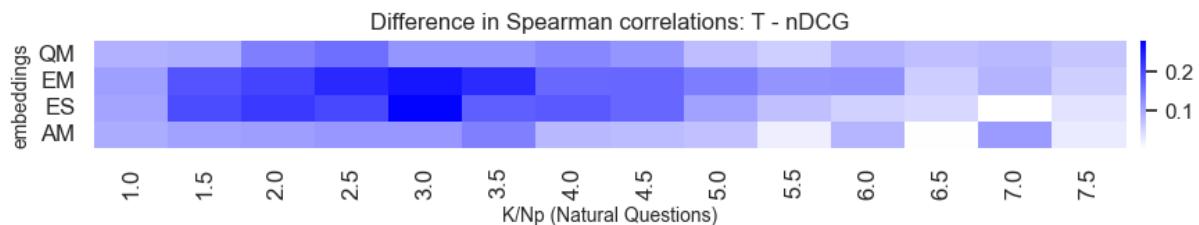


Figure 43: Difference between the Spearman correlations: T -response minus $nDCG$ -response. On Natural Questions.