



## On Monotonic Aggregation for Open-domain QA

Sang-eun Han<sup>1,2\*</sup>, Yeonseok Jeong<sup>1\*</sup>, Seung-won Hwang<sup>1,2†</sup>, Kyungjae Lee<sup>3</sup>

<sup>1</sup>Seoul National University, Republic of Korea

<sup>2</sup>SNU-LG AI Research Center, Republic of Korea

<sup>3</sup>LG AI Research, Republic of Korea

{gkstkddms00, jys3136, seungwonh}@snu.ac.kr, kyungjae.lee@lgresearch.ai

### Abstract

Question answering (QA) is a critical task for speech-based retrieval from knowledge sources, by sifting only the answers without requiring to read supporting documents. Specifically, open-domain QA aims to answer user questions on unrestricted knowledge sources. Ideally, adding a source should not decrease the accuracy, but we find this property (denoted as “monotonicity”) does not hold for current state-of-the-art methods. We identify the cause, and based on that we propose Judge-Specialist framework. Our framework consists of (1) specialist retrievers/readers to cover individual sources, and (2) judge, a dedicated language model to select the final answer. Our experiments show that our framework not only ensures monotonicity, but also outperforms state-of-the-art multi-source QA methods on Natural Questions. Additionally, we show that our models robustly preserve the monotonicity against noise from speech recognition. We publicly release our code and setting.

**Index Terms:** open domain QA, QA from speech

### 1. Introduction

Open domain question answering (ODQA) aims to answer user questions on unrestricted topics and knowledge sources. ODQA systems are especially useful when information is dispersed across sources, which makes it difficult for users to find relevant information quickly. The goal of this paper is to improve ODQA techniques to enhance the efficiency and effectiveness of information retrieval through speech.

A major challenge in ODQA is to synergize multiple knowledge sources such as unstructured text [1], structured or semi-structured sources such as knowledge graph [2] and tables [3] by leveraging their complementary strengths: structured sources are highly precise but answer limited questions, while unstructured text answers a wider range of questions due to its widespread availability. A similar challenge is aggregated search [4] for multiple vertical search engines, *e.g.*, image or table search. Aggregated search consists of individual verticals, and the aggregator which ranks the vertical results for user satisfaction.

To illustrate, consider a query from Natural Questions (NQ) [5]: “*Who is the director of the film Avatar?*”. This query can be answered by both text search and table search in the aggregated search, and the aggregator would decide to rank the result with a higher expected quality first.

Meanwhile, prior work on multi-source ODQA, UniK-QA [6] takes a different approach. UniK-QA unifies the representation of sources by linearizing graphs, tables, and lists into unstructured text, such that a unified retriever-reader can identify

\* First two authors equally contributed to this work.

† Corresponding Author

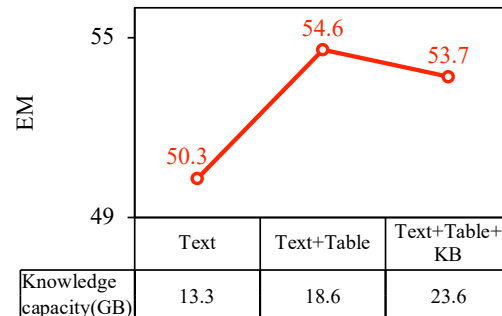


Figure 1: Result in [6]. Added sources in prior work increase the required storage for knowledge, but EM accuracy decreases.

an answer. For example, a table row containing the director of Avatar can be transformed into the text (*e.g.*, “*Director of Avatar is James Cameron*”), such that information from diverse sources can be jointly processed by retriever and reader. UDT-QA [7] inherits this approach, and unifies the sources with a pretrained language model (PLM). This line of work has shown a significant improvement over single-source QA system, demonstrating the benefit of incorporating multiple knowledge sources in ODQA. However, we argue that previous work fails to exploit the increased capacity and diversity of knowledge sources. To motivate, Figure 1 shows that the performance (measured as Exact Match, or EM) does not monotonically improve with increased sources, as the addition of knowledge base (KB) results in performance degradation.

We refer to this phenomenon as “non-monotonicity”, and pursue the multi-source ODQA with “monotonicity” where the addition of knowledge sources does not damage the performance. Our analysis in Section 4.2 shows that the unified retriever fails to cover multiple sources effectively, thereby causing non-monotonicity. To address this issue, we revisit and explore the framework of aggregated search in ODQA. We keep source-specific retriever-readers as vertical ‘specialists’, and propose ‘judge’ to aggregate results for non-decreasing performance. We denote this model as JS-HR, for Judge-Specialist model for High-Resource scenarios. For low-resource scenarios where JS-HR is not affordable, we employ multitask learning (MTL) to keep resource requirement near constant over the increase of sources. We denote this model as JS-LR in Figure 2, from which we confirm that monotonicity holds for both JS-HR and JS-LR: as the sources are added from T (text), TT (text and table), and TTK (text, table and knowledge base), the performance increases accordingly.

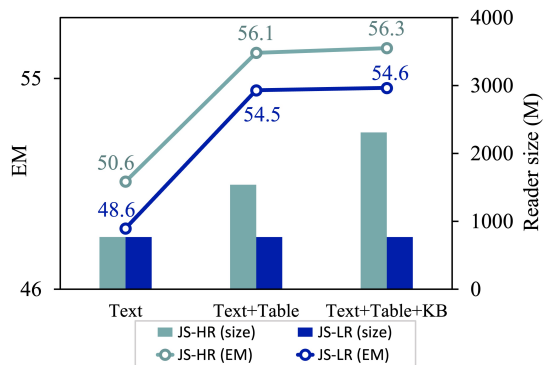


Figure 2: Performance (line) and total parameter size for readers (bar) of our methods in NQ.

Our empirical results also suggest that our proposed approach ensures monotonicity and outperforms the prior method in NQ-speech dataset as well, representing realistic noises from speech recognition. Our code and results are available at [github.com/YeonseokJeong/Judge-Specialist](https://github.com/YeonseokJeong/Judge-Specialist).

## 2. Background & Related Work

This section overviews preliminary knowledge and existing work on ODQA, the task of answering general domain questions from one or more knowledge sources.

### 2.1. ODQA with multiple knowledge sources

UniK-QA [6] proposed to unify knowledge sources into textual format, such that simple transformer-based language models (LM) can support all sources. In UniK-QA, structured data such as table and KB are unified with unstructured text by *linearization*; cell values on the same row are concatenated into text by predefined rules. The unified knowledge is then used in DPR-FiD pipeline (explained in 2.2), where the retriever collects passages from multiple sources.

UDT-QA [7] unifies knowledge sources into unstructured text as well, but with LM-based method known as *verbalization*. In verbalization [8], PLM generates the unstructured text which describes the content of given structured data.<sup>1</sup> The unified knowledge is then utilized in DPR-UnitedQA [9] pipeline. We adopt the same verbalization approach for our work, since the authors report that the verbalized knowledge is favored by readers over the linearized one. UDT-QA outperforms UniK-QA on NQ, but the issue of non-monotonicity persists.

### 2.2. Retrievers and Readers

Retriever-reader architecture is conventionally adopted approach for ODQA [1, 6, 9]. We build our model on DPR-FiD architecture, since it is widely adopted by state-of-the-art multi-source ODQA. FiD models are trained from T5-large huggingface checkpoint.

DPR, or dense passage retriever, is widely used for the document retrieval, such as in [1, 6, 7]. DPR demonstrated significant improvement over the previous methods such as TF-IDF or BM25 [10]. DPR retriever has the biencoder structure, which

<sup>1</sup>In UDT-QA, T5-large is used to verbalize structured data into text. To keep the faithfulness of generation against hallucination / missing information, training samples are filtered with ROUGE-1 score.

consists of a document encoder and a query encoder. The encoders are trained such that the embeddings of positive documents have greater dot product with the question embedding.

FiD, or fusion-in-decoder, is a commonly used reader component in ODQA, due to its ability to effectively fuse information from multiple passages [6, 11]. Implemented on encoder-decoder model such as T5 [12] or BART [13], FiD encodes retrieved passages independently, then the decoder computes full attention on the concatenated representation of passages.

**Our distinction** is to separate retrievers and readers to obtain answer candidates from each specific source, instead of relying on unified retriever-reader pipeline. The judge then filters out incorrect candidates to preserve the monotonicity. Our approach is comparable to Watson [14], where multiple expert systems produce candidates then aggregate them to determine the final answer. However, while Watson relies on multiple different low-level algorithms such as relation extraction, we leverage the ability of transformer-based LM (T5-large) to evaluate the candidates with unified knowledge.

## 3. Proposed Method

This section overviews JS-HR (Judge-Specialist for High Resource) optimizing for monotonicity (Section 3.1) and JS-LR (Judge-Specialist for Low Resource) for a higher scalability (Section 3.2).

### 3.1. Goal 1: Monotonicity

To preserve monotonicity, QA process is broken down into two phases: **V-step** to collect enough answer candidates from vertical specialists, and **A-step** to aggregate the candidates and determine the final answer with the judge model.

#### 3.1.1. V-step: Candidates from Specialists

The V-step aims to collect candidates from each vertical specialist, thereby improving the recall of the system. We use beam search to generate candidates, and invest a larger beam size to high-performing specialists, such that a more useful source can provide more candidates. Specifically, beam size is set to be proportional to the dev set performance (EM) of the specialist reader representing a source.

We first determine the proportion of beam size for each source, then allocate the beam size according to the manually chosen total budget (total number of candidates). Formally, the proportion  $p_k$  for  $k$ -th source among  $m$  sources is calculated as below:

$$p_k = \frac{EM_k}{\sum_{j=1}^m EM_j} \quad (1)$$

where  $EM_x$  is the dev set EM of the specialist reader representing  $x$ -th source. Beam size  $b_k$  is then calculated with regard to the given total budget  $B$ .

$$b_k = \lceil p_k \cdot B \rceil \quad (2)$$

We round  $b_k$  to the nearest positive integer.

#### 3.1.2. A-step: Selection with the Judge

In A-step, we aggregate candidates  $\langle A_1, A_2, \dots, A_n \rangle$  from verticals and select the final answer. We frame the selection process as a problem of calibration, quantifying the trustworthiness of candidates with probability  $P$  from LM. Formally, candidate

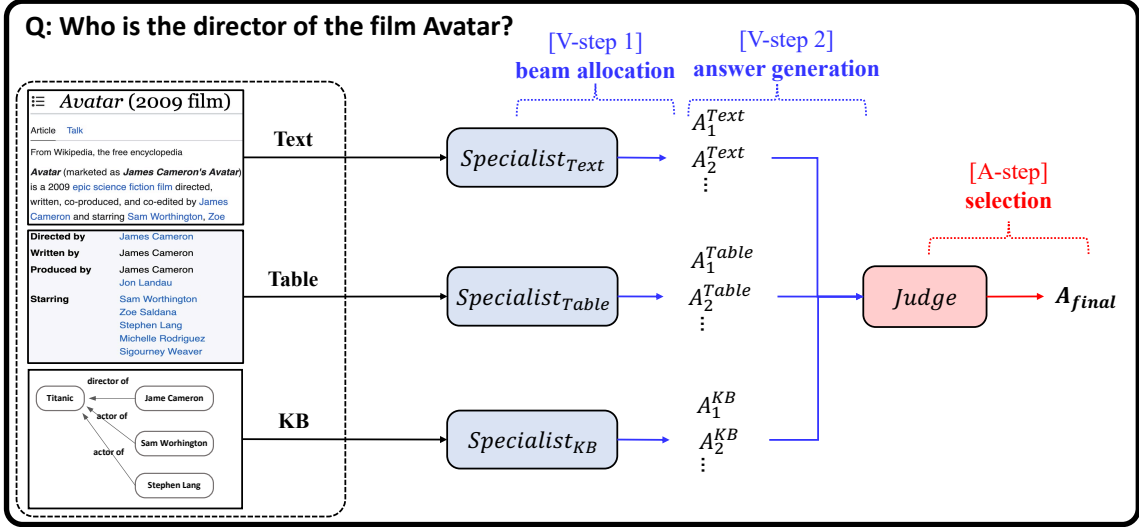


Figure 3: Overview of our method (JS-HR). Multiple specialists generate candidates, and one judge selects the final answer. For judge model, we use fine-tuned T5-large.

answer  $A$  which consists of token  $a_1, a_2, \dots, a_n$  can be evaluated by LM probability  $P$ .

$$P(A|Q, C) = \left\{ \prod_{t=1}^n P(a_t | a_{<t}, Q, C) \right\}^{\frac{1}{n}} \quad (3)$$

where  $Q$  is question and  $C$  is the context, which can be obtained from both existing unified retriever<sup>2</sup> and our proposed specialist retriever.

The two types of retrievers have complementary strength. If the answer is unanimously supported by all sources, the unified retriever can provide high-quality context. To illustrate, in Figure 3, specialist retriever for text would bring relevant passage “Avatar is a SF movie directed by James Cameron”, and similarly specialists for table/KB would retrieve verbalized evidences separately. Alternatively, unified retriever would integrate evidences from diverse sources: “Avatar is a SF movie directed by James Cameron. (text) ... Director of Avatar is James Cameron. (verbalized table)”

Meanwhile, we can imagine another extreme where only one of the sources supports the answer. In such case, passages from non-supporting sources in the unified retriever would only work as a noise. For such cases, using context from specific source has the effect of denoising unrelated information.

We evaluate candidates with both types of context, and combine two LM probabilities for the final selection. We provide the context from unified retriever to the judge, and calculate the probability  $P_J$  for  $A_1, A_2, \dots, A_n$  as Equation 3. Similarly, context from specialist retriever is provided to the specialist reader, to calculate the probability  $P_S$  for each candidate. We then select the candidate with the highest average value of  $P_J$  and  $P_S$  from the candidate set. While there are multiple ways to combine the probabilities, we empirically found that a simple average gives strong results.

<sup>2</sup>We use the same unified retriever across all multi-source settings (TT, TTK). Specifically, we use unified retriever trained / inferred on TT data for our main experiments.

### 3.2. Goal 2: Graceful Scaling

We discuss how to implement the Judge-Specialist framework, for both high- and low-resource scenarios.

**JS-HR** (Judge-Specialist for High-Resource): By default, we utilize the FiD [1] models for both the specialist readers and the judge, which are trained on the passages of respective sources. Additionally, we assume that each specialist operates independently and is trained in isolation. This setting is referred to as JS-HR, which is the original version of our Judge-Specialist framework.

**JS-LR** (Judge-Specialist for Low-Resource): In an effort to reduce the size of specialists, we can consider diverse MTL techniques. We stress that JS-LR is orthogonal to any such technique, and implement JS-LR with a widely adopted baseline: we simultaneously feed single-source contexts to the model during training, and backpropagate the sum of losses from each source. This integrated model requires 3 times fewer parameters than an original design while still serving as a specialist for any source.

## 4. Experimental Results

### 4.1. Main Results

Table 1 shows the result of our models on NQ,<sup>3</sup> compared to existing methods. We also compare ours to the baseline model which consists of UDT-QA retriever and FiD reader (referred to as ‘Baseline’ in Table 1), to better observe the effect of the separated retrievers. The result shows that (1) JS-HR outperforms existing methods (including baseline) in multi-source settings and (2) JS-HR and JS-LR both preserve the monotonicity, whereas other methods fail to benefit from KB.<sup>4</sup>

<sup>3</sup>We use open-domain split of NQ, which contains 79K examples for train set, 8.7K for dev set, and 3.6K for test set. Following [7], we use the knowledge-answerable data augmented train set for TT / TTK reader, which consists of 80K examples. We report single-run result (both training and evaluation) on this dataset.

<sup>4</sup>We use 8 A6000 GPUs for the beam search, and 8 RTX 3090 GPUs for other experiments. Each reader is trained with learning rate of 5e-5

Model	Text (T)	Text+Table (TT)	TT+KB (TTK)
<i>Prior methods</i>			
UniK-QA (multi-data)	50.3	54.6	53.7
UniK-QA (per-data)	49.0	54.1	54.0
UDT-QA	<b>51.8</b>	55.2	55.1
Baseline	50.9	53.7	53.0
<i>Ours</i>			
JS-HR	50.6	<b>56.1</b>	<b>56.3</b>
JS-LR	48.7	54.5	54.6

Table 1: Experimental result on the test sets of NQ using EM scores (with budget B=9)

Method	JS-HR	JS-LR
$P_S$	51.0	50.9
$P_J$	55.3	54.5
$P_{JS}$ (ours)	<b>56.3</b>	<b>54.6</b>

Table 2: Ablation study on the selection in NQ using EM score (TTK).

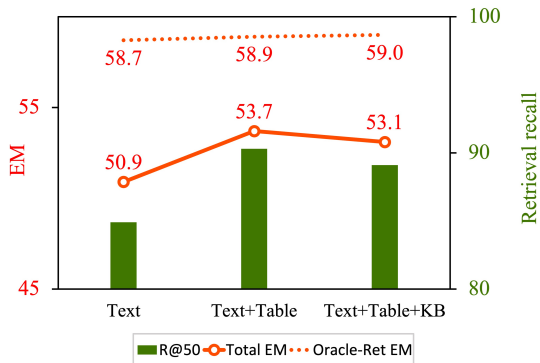


Figure 4: Analysis of non-monotonicity of a baseline in NQ (TTK).

#### 4.2. Cause of non-monotonicity

Figure 4 shows our analysis on the unified retriever-reader pair, using the same retriever setting as [7] (baseline method discussed in Section 4.1). Our analysis reveals that the non-monotonicity results from the failure of the unified retriever.

We observe that the monotonicity holds in NQ subset, where the retriever successfully obtains the oracle context that contains the ground truth. (shown by the ‘Oracle-Ret EM’, dotted line in the figure) However, without such condition, the retrieval performance (‘R@50’, bar graph) decreases as KB is added, leading to the non-monotonicity in overall QA performance (‘Total EM’, solid line). This result suggests that the unified retriever has a limited generalizability towards multiple sources, which motivates us to use the specialist retrievers.

and 10 epoch for 6 days, and the best checkpoint is selected by the corresponding dev EM. We apply early stopping when the dev EM does not increase within two consecutive epochs.

<sup>5</sup>R@50 indicates the retrieval recall within the top-50 passages.

#### 4.3. Effects of Judge

Table 2 shows the ablation study on the selection. We observe dramatic performance drops in all settings when  $P_J$  is removed from the selection criterion, showing the critical role of the judge. Meanwhile, the performance drop is limited when  $P_S$  is removed.

Model	T	TT	TTK
Baseline	29.8	32.2	31.5
JS-LR	29.3	33.1	33.7
JS-HR	<b>30.4</b>	<b>33.8</b>	<b>34.0</b>

Table 3: Experimental result on the test sets of NQ-speech. (with budget B=9)

#### 4.4. Robustness to Speech Noises

To deploy QA models with speech interface, it is important to consider whether the models can generalize to inputs with real-life noises. To examine the robustness to such noises, we tested our models in NQ-speech where questions contain errors from automatic speech recognition (ASR) [15].<sup>6</sup> Table 3 shows that our models preserve monotonicity with speech noises, and generalize better than the baseline method discussed in Section 4.1.

Additionally, we made a consistent observation in another realistic noisy dataset, NQ-typo, where the questions have random typo errors [16]. Our models perform closely to clean NQ, achieving 39.1, 44.0, 44.2 in EM scores for JS-HR in T, TT, TTK respectively. These results suggest that our models show a higher generalizability in both settings, while their relative performance may depend on the noise type – and ASR noise is more challenging. A promising future direction is thus to close such a gap using speech-targeted debiasing [15].

## 5. Conclusion

In this paper, we proposed a simple yet effective method to utilize the complementarity in heterogeneous sources, and thereby achieve the monotonicity, unlike the current state-of-the-art methods. Our proposed approach keeps readers and retrievers as specialists to each source and adds a dedicated LM (the ‘judge’) to aggregate candidates for the final answer. We validated the effectiveness and monotonicity in both clean and speech settings.

## 6. Limitation

The focus of this work is to show the potential of Judge-Specialist framework, not to optimize the judge/specialist model. Hence we did not further explore for the optimal practice in this paper, but any such effort can be orthogonally combined with our framework.

## 7. Acknowledgement

This work was supported by LG AI Research. This work was partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)].

<sup>6</sup>ASR error is obtained by synthetic TTS system (Google TTS) and transcription system (wav2vec 2.0).

## 8. References

- [1] G. Izacard and É. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 874–880.
- [2] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou, “Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 3554–3565. [Online]. Available: <https://aclanthology.org/2021.naacl-main.278>
- [3] L. Nan, C. Hsieh, Z. Mao, X. V. Lin, N. Verma, R. Zhang, W. Kryściński, N. Schoelkopf, R. Kong, X. Tang *et al.*, “Fetaqa: Free-form table question answering,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 35–49, 2022.
- [4] M. Lalmas, “Aggregated search,” *Advanced Topics in Information Retrieval*, pp. 109–123, 2011.
- [5] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee *et al.*, “Natural questions: A benchmark for question answering research,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019.
- [6] B. Oguz, X. Chen, V. Karpukhin, S. Peshterliev, D. Okhonko, M. Schlichtkrull, S. Gupta, Y. Mehdad, and S. Yih, “Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering,” *arXiv preprint arXiv:2012.14610*, 2020.
- [7] K. Ma, H. Cheng, X. Liu, E. Nyberg, and J. Gao, “Open domain question answering with a unified knowledge interface,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1605–1620.
- [8] L. F. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, “Investigating pretrained language models for graph-to-text generation,” *arXiv preprint arXiv:2007.08426*, 2020.
- [9] H. Cheng, Y. Shen, X. Liu, P. He, W. Chen, and J. Gao, “Unit-edqa: A hybrid approach for open domain question answering,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 3080–3090.
- [10] S. Robertson, H. Zaragoza *et al.*, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [11] P. Lewis, Y. Wu, L. Liu, P. Minervini, H. Küttler, A. Piktus, P. Stenetorp, and S. Riedel, “Paq: 65 million probably-asked questions and what you can do with them,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1098–1115, 2021.
- [12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.
- [13] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv preprint arXiv:1910.13461*, 2019.
- [14] D. A. Ferrucci, “Introduction to “this is watson”,” *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 1:1–1:15, 2012.
- [15] G. Sidiropoulos, S. Vakulenko, and E. Kanoulas, “On the impact of speech recognition errors in passage retrieval for spoken question answering,” in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4485–4489.
- [16] G. Sidiropoulos and E. Kanoulas, “Analysing the robustness of dual encoders for dense retrieval against misspellings,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 2132–2136.