Evaluation of Intermediate Steps in Retrieval Augmented Generation

ARAM AYTEKIN, Universität Kassel, Germany
MORITZ DIETRICH, Universität Kassel, Germany
MAXIMILIAN BENNEDIK, Universität Kassel, Germany
DOMENIC BREMMER, Universität Kassel, Germany

This work investigates the research question: Does a RAG pipeline that expands the initial prompt into an LLM-generated query pool produce answers that humans prefer (on correctness, conciseness, and relevance) over a baseline that directly queries Elasticsearch once? To explore this, we implement two retrieval-augmented generation (RAG) pipelines that operate exclusively on search engine result page (SERP) snippets, which are often incomplete and inconsistent. The baseline pipeline (P1) retrieves snippets from a single query and directly conditions an LLM on this evidence. The advanced pipeline (P4), in contrast, employs LLM-based query expansion to generate a pool of reformulated queries, filters the resulting snippets, and integrates them for answer inference. Human evaluation indicates that P4 produces answers that are slightly more coherent, contextually appropriate, and preferred overall, though the improvements remain moderate. However, these gains come at the cost of significantly longer execution time, highlighting a trade-off between answer quality and system efficiency. These findings demonstrate the potential of query-expansion-based RAG pipelines for enhancing answer quality from fragmented snippet collections, while also pointing to open challenges in efficiency, evidence integration, and domain adaptation.

CCS Concepts: • **Information systems** → **Retrieval models and ranking**; **Evaluation of retrieval results**; • **Computing methodologies** → *Natural language processing*.

Additional Key Words and Phrases: RAG, Summary, Pooling, Query

ACM Reference Format:
Aram Aytekin, Moritz Dietrich, Maximilian Bennedik, and Domenic Bremmer. 2025. Evaluation of Intermediate Steps in Retrieval Augmented Generation. 1, 1 (August 2025), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Information retrieval and answer generation have been transformed in recent years by large language models (LLMs) and retrieval-augmented generation (RAG). While these approaches typically rely on access to the full underlying documents, many real-world scenarios such as search engines only provide access to fragmented snippets contained in search engine result pages (SERPs). These snippets are often incomplete, inconsistent, and heterogeneous in style, making coherent answer generation particularly challenging. Nevertheless, being able to synthesize high-quality answers from such fragments is of both practical and theoretical interest.

A central obstacle in this setting lies in how queries are handled. Traditional retrieval methods issue a single query and directly return associated results, which often fails to capture the full breadth of relevant information. Recent advances in LLM prompting, however, allow for automatic query reformulation and expansion, potentially improving coverage and evidence diversity. Whether such query pooling strategies can indeed enhance answer generation quality from SERP snippets remains an open question.

Authors' Contact Information: Aram Aytekin, Universität Kassel, Germany, uk097201@student.uni-kassel.de; Moritz Dietrich, Universität Kassel, Germany, uk097299@student.uni-kassel.de; Maximilian Bennedik, Universität Kassel, Germany, ukxxx@student.uni-kassel.de; Domenic Bremmer, Universität Kassel, Germany, uk095482@student.uni-kassel.de.

2025. Manuscript submitted to ACM

1.1 Research Question

Against this background, our study investigates the following research question: *Does a RAG pipeline that expands the initial prompt into an LLM-generated query pool produce answers that humans prefer (on correctness, conciseness, and relevance) over a baseline that directly queries Elasticsearch once?*

1.2 Contribution

To answer this question, we implement and evaluate two RAG pipelines that operate solely on SERP snippets. The baseline pipeline (P1) issues a single query and generates answers from the retrieved snippets without further reformulation. The advanced pipeline (P4), in contrast, expands the initial query into a pool of reformulated queries using an LLM, retrieves and filters a broader snippet set, and conditions answer generation on this richer evidence. We evaluate both pipelines through human preference judgments along correctness, conciseness, and relevance, and further measure computational cost. Our findings show that while P4 produces answers that are slightly more coherent and preferred overall, it incurs substantially higher runtime costs, highlighting a trade-off between answer quality and efficiency. This contribution provides insight into the potential and the limitations of query-expansion-based RAG pipelines when applied to fragmented snippet data.

2 Background & Related Work

2.1 Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm for combining the generative capabilities of large language models (LLMs) with external retrieval mechanisms to produce factually grounded answers. In typical RAG setups, queries are used to retrieve relevant documents or passages from a large corpus, which are then used to condition the LLM's response [4, 6]. This approach mitigates hallucinations often observed in standalone LLMs and enhances factual accuracy. Previous studies have demonstrated the effectiveness of RAG in settings such as open-domain question answering, summarization, and knowledge-intensive tasks [3, 5]. However, most prior work assumes access to complete and well-structured documents. In contrast, the present study operates under the more constrained setting of search engine result page (SERP) snippets, which are often fragmented, inconsistent, and incomplete. This constraint introduces unique challenges for coherent answer generation, motivating the exploration of advanced retrieval strategies.

2.2 Query Expansion vs. Single-Query Baselines

Traditional information retrieval systems often rely on a single-query approach, issuing one query to retrieve the top-k documents or passages. While computationally efficient, this strategy may fail to capture the full spectrum of relevant evidence, particularly in domains with heterogeneous or incomplete sources. Query expansion techniques aim to address this limitation by reformulating the initial query or generating multiple variant queries to improve coverage. Classical methods include pseudo-relevance feedback and relevance-based models [11], whereas more recent approaches leverage LLMs to generate paraphrases, sub-questions, or expanded queries tailored to the information need [9, 14]. The trade-off is evident: query expansion can increase evidence diversity and retrieval effectiveness, but at the cost of additional computation and potential introduction of irrelevant information. In our study, the baseline pipeline (P1) represents the single-query approach, while the advanced pipeline (P4) operationalizes LLM-driven query expansion within a RAG framework, allowing for a systematic evaluation of this trade-off.

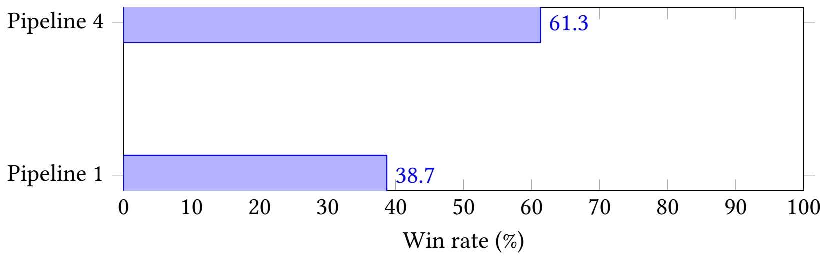

Fig. 1. Win rate comparison between Pipeline 1 and Pipeline 4.

2.3 Human Preference Evaluation

Automatic evaluation metrics such as BLEU [10], ROUGE [7], and BERTScore [15] are limited in their ability to capture the quality of open-ended answers, especially in terms of correctness, conciseness, and relevance [8]. Human evaluation remains the gold standard for assessing answer quality in generative tasks [1, 13]. Pairwise preference judgments, where annotators compare outputs from different systems, have been widely adopted to obtain robust insights into model performance [12].

Recent work has shown that crowdsourcing can be a viable approach for evaluating RAG systems, enabling structured assessments while controlling for annotator variability [2]. Despite the advantages, human evaluation remains resource-intensive. In our study, we conduct a structured human preference evaluation comparing answers generated by P1 and P4 along three dimensions: correctness, conciseness, and relevance. This provides direct evidence of the practical impact of query pooling on perceived answer quality, complementing quantitative retrieval metrics and highlighting the cost-quality trade-offs inherent in LLM-driven RAG pipelines.

3 Methods

xxx

3.1 Pipelines

xxx

3.2 Data & Task Setup

xxx

3.3 Retrieval & LLM Configuration

xxx

3.4 Evaluation Protocol

xxx

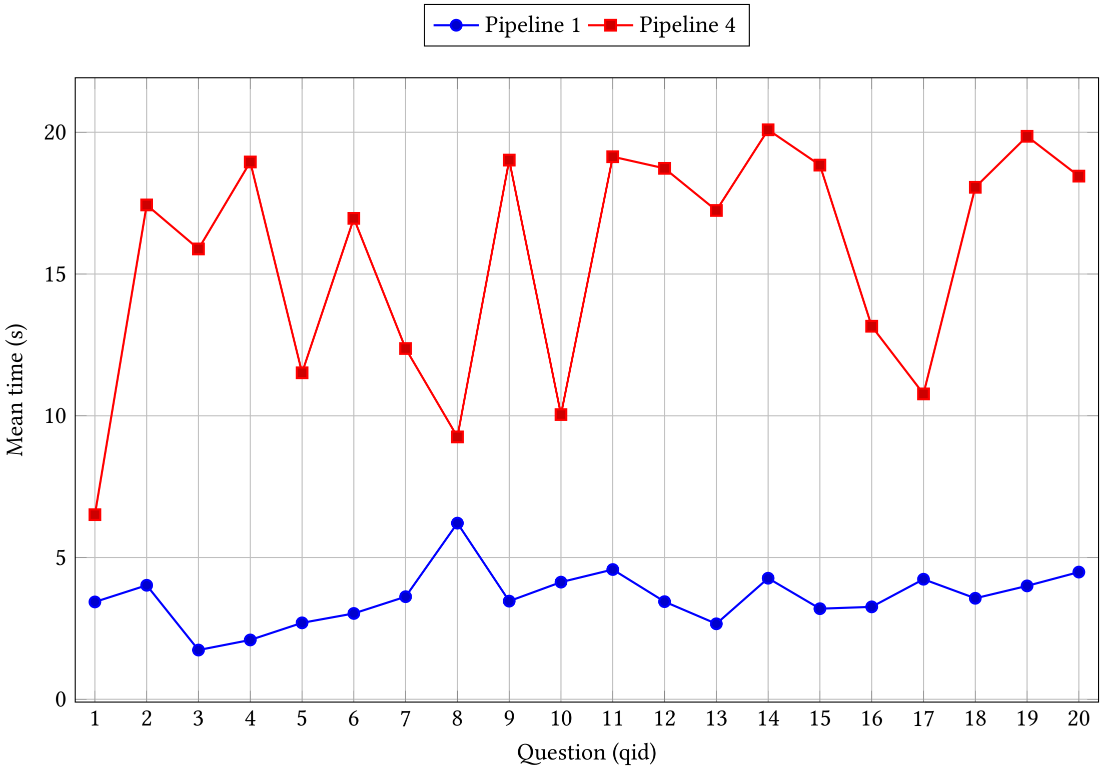

Fig. 2. Per-question mean response time (s) across 20 questions.

3.5 Metrics & Hypotheses

3.6 Cost Measurement

4 Results

4.1 Overall Preference

xxx

4.2 Annotator Remarks

xxx

4.3 Cost-Quality Tradeoff

xxx

5 Conclusion

xxx

6 sample xxx

The primary parameter given to the "acmart" document class is the *template style* which corresponds to the kind of publication or SIG publishing the work. This parameter is enclosed in square brackets and is a part of the documentclass command:

```
\documentclass[STYLE]{acmart}
```

Journals use one of three template styles. All but three ACM journals use the acmsmall template style:

- acmsmall: The default journal template style.
- acmlarge: Used by JOCCH and TAP.
- acmtog: Used by TOG.

The majority of conference proceedings documentation will use the acmconf template style.

- sigconf: The default proceedings template style.
- sigchi: Used for SIGCHI conference articles.
- sigplan: Used for SIGPLAN conference articles.

6.1 Template Parameters

In addition to specifying the *template style* to be used in formatting your work, there are a number of *template parameters* which modify some part of the applied template style. A complete list of these parameters can be found in the *LaTeX User's Guide.*

Frequently-used parameters, or combinations of parameters, include:

- anonymous,review: Suitable for a "double-anonymous" conference submission. Anonymizes the work and includes line numbers. Use with the \acmSubmissionID command to print the submission's unique ID on each page of the work.
- authorversion: Produces a version of the work suitable for posting by the author.
- screen: Produces colored hyperlinks.

This document uses the following string as the first command in the source file:

```
\documentclass[manuscript,screen,review]{acmart}
```

7 Modifications

Modifying the template — including but not limited to: adjusting margins, typeface sizes, line spacing, paragraph and list definitions, and the use of the \vspace command to manually adjust the vertical spacing between elements of your work — is not allowed.

Your document will be returned to you for revision if modifications are discovered.

8 Typefaces

The "acmart" document class requires the use of the "Libertine" typeface family. Your TeX installation should include this set of packages. Please do not substitute other typefaces. The "lmodern" and "ltimes" packages should not be used, as they will override the built-in typeface families.

9 Title Information

The title of your work should use capital letters appropriately - https://capitalizemytitle.com/ has useful rules for capitalization. Use the `title` command to define the title of your work. If your work has a subtitle, define it with the `subtitle` command. Do not insert line breaks in your title.

If your title is lengthy, you must define a short version to be used in the page headers, to prevent overlapping text. The `title` command has a "short title" parameter:

```
\title[short title]{full title}
```

10 Authors and Affiliations

Each author must be defined separately for accurate metadata identification. As an exception, multiple authors may share one affiliation. Authors' names should not be abbreviated; use full first names wherever possible. Include authors' e-mail addresses whenever possible.

Grouping authors' names or e-mail addresses, or providing an "e-mail alias," as shown below, is not acceptable:

```
\author{Brooke Aster, David Mehldau}
\email{dave,judy,steve@university.edu}
\email{firstname.lastname@phillips.org}
```

The `authornote` and `authornotemark` commands allow a note to apply to multiple authors — for example, if the first two authors of an article contributed equally to the work.

If your author list is lengthy, you must define a shortened version of the list of authors to be used in the page headers, to prevent overlapping text. The following command should be placed just after the last `\author{}` definition:

```
\renewcommand{\shortauthors}{McCartney, et al.}
```

Omitting this command will force the use of a concatenated list of all of the authors' names, which may result in overlapping text in the page headers.

The article template's documentation, available at https://www.acm.org/publications/proceedings-template, has a complete explanation of these commands and tips for their effective use.

Note that authors' addresses are mandatory for journal articles.

11 Rights Information

Authors of any work published by ACM will need to complete a rights form. Depending on the kind of work, and the rights management choice made by the author, this may be copyright transfer, permission, license, or an OA (open access) agreement.

Regardless of the rights management choice, the author will receive a copy of the completed rights form once it has been submitted. This form contains LaTeX commands that must be copied into the source document. When the document source is compiled, these commands and their parameters add formatted text to several areas of the final document:

- the "ACM Reference Format" text on the first page.
- the "rights management" text on the first page.
- the conference information in the page header(s).

Rights information is unique to the work; if you are preparing several works for an event, make sure to use the correct set of commands with each of the works.

The ACM Reference Format text is required for all articles over one page in length, and is optional for one-page articles (abstracts).

12 CCS Concepts and User-Defined Keywords

Two elements of the “acmart” document class provide powerful taxonomic tools for you to help readers find your work in an online search.

The ACM Computing Classification System — https://www.acm.org/publications/class-2012 — is a set of classifiers and concepts that describe the computing discipline. Authors can select entries from this classification system, via https://dl.acm.org/ccs/ccs.cfm, and generate the commands to be included in the LaTeX source.

User-defined keywords are a comma-separated list of words and phrases of the authors’ choosing, providing a more flexible way of describing the research being presented.

CCS concepts and user-defined keywords are required for for all articles over two pages in length, and are optional for one- and two-page articles (or abstracts).

13 Sectioning Commands

Your work should use standard LaTeX sectioning commands: `\section`, `\subsection`, `\subsubsection`, `\paragraph`, and `\subparagraph`. The sectioning levels up to `\subsusection` should be numbered; do not remove the numbering from the commands.

Simulating a sectioning command by setting the first word or words of a paragraph in boldface or italicized text is **not allowed.**

Below are examples of sectioning commands.

13.1 Subsection

This is a subsection.

13.1.1 Subsubsection. This is a subsubsection.

Paragraph. This is a paragraph.

Subparagraph This is a subparagraph.

14 Tables

The “`acmart`” document class includes the “`booktabs`” package — https://ctan.org/pkg/booktabs — for preparing high-quality tables.

Table captions are placed *above* the table.

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper “floating” placement of tables, use the environment **table** to enclose the table’s contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material are found in the *LaTeX User’s Guide*.

Table 1. Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

Table 2. Some Typical Commands

Command	A Number	Comments
`\author`	100	Author
`\table`	300	For tables
`\table*`	400	For wider tables

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

Always use midrule to separate table header rows from data rows, and use it only for this purpose. This enables assistive technologies to recognise table headers and support their users in navigating tables more easily.

15 Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

15.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin`…`\end` construction or with the short form `$`…`$`. You can use any of the symbols and structures, from α to ω, available in LaTeX [?]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\rightarrow\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

15.2 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\rightarrow \infty}x=0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty}x_i=\int_{0}^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

16 Figures

The "`figure`" environment should be used for figures. One or more images can be placed within a figure. If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.

Your figures should contain a caption which describes the figure to the reader.

Figure captions are placed *below* the figure.

Every figure should also have a figure description unless it is purely decorative. These descriptions convey whatâs in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than 2000 characters long (including spaces). **Figure descriptions should not repeat the figure caption â their purpose is to capture important information that is not already provided in the caption or the main text of the paper.** For figures that convey important and complex new information, a short text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph. For additional information regarding how best to write figure descriptions and why doing this is so important, please see https://www.acm.org/publications/taps/describing-figures/.

16.1 The "Teaser Figure"

A "teaser figure" is an image, or set of images in one figure, that are placed after all author and affiliation information, and before the body of the article, spanning the page. If you wish to have such a figure in your article, place the command immediately before the `\maketitle` command:

```
\begin{teaserfigure}
  \includegraphics[width=\textwidth]{sampleteaser}
  \caption{figure caption}
  \Description{figure description}
\end{teaserfigure}
```

Fig. 3. 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (https://goo.gl/VLCRBB).

17 Citations and Bibliographies

The use of BibTEX for the preparation and formatting of one's references is strongly recommended. Authors' names should be complete — use full first names ("Donald E. Knuth") not initials ("D. E. Knuth") — and the salient identifying features of a reference should be included: title, year, volume, number, pages, article DOI, etc.

The bibliography is included in your source document with these two commands, placed just before the `\end{document}` command:

```
\bibliographystyle{ACM-Reference-Format}
\bibliography{bibfile}
```

where "`bibfile`" is the name, without the "`.bib`" suffix, of the BibTEX file.

Citations and references are numbered by default. A small number of ACM publications have citations and references formatted in the "author year" style; for these exceptions, please include this command in the **preamble** (before the command "`\begin{document}`") of your LATEX source:

```
\citestyle{acmauthoryear}
```

Some examples. A paginated journal article [?], an enumerated journal article [?], a reference to an entire issue [?], a monograph (whole book) [?], a monograph/whole book in a series (see 2a in spec. document) [?], a divisible-book such as an anthology or compilation [?] followed by the same example, however we only output the series if the volume number is given [?] (so Editor00a's series should NOT be present since it has no vol. no.), a chapter in a divisible book [?], a chapter in a divisible book in a series [?], a multi-volume work as book [?], a couple of articles in a proceedings (of a conference, symposium, workshop for example) (paginated proceedings article) [? ?], a proceedings article with all possible elements [?], an example of an enumerated proceedings article [?], an informally published work [?], a couple of preprints [? ?], a doctoral dissertation [?], a master's thesis: [?], an online document / world wide web resource [? ? ?], a video game (Case 1) [?] and (Case 2) [?] and [?] and (Case 3) a patent [?], work accepted for publication [?], 'YYYYb'-test for prolific author [?] and [?]. Other cites might contain 'duplicate' DOI and URLs (some SIAM articles) [?]. Boris / Barbara Beeton: multi-volume works as books [?] and [?]. A presentation [?]. An article under review [?]. A couple of citations with DOIs: [? ?]. Online citations: [? ? ?]. Artifacts: [?] and [?].

18 Acknowledgments

Identification of funding sources and other support, and thanks to individuals and groups that assisted in the research and the preparation of the work should be included in an acknowledgment section, which is placed just before the reference section in your document.

This section has a special environment:

```
\begin{acks}
...
\end{acks}
```

so that the information contained therein can be more easily collected during the article metadata extraction phase, and to ensure consistency in the spelling of the section heading.

Authors should not prepare this section as a numbered or unnumbered `\section`; please use the "`acks`" environment.

19 Appendices

If your work needs an appendix, add it before the "`\end{document}`" command at the conclusion of your source document.

Start the appendix with the "`appendix`" command:

```
\appendix
```

and note that in the appendix, sections are lettered, not numbered. This document has two appendices, demonstrating the section and subsection identification method.

20 Multi-language papers

Papers may be written in languages other than English or include titles, subtitles, keywords and abstracts in different languages (as a rule, a paper in a language other than English should include an English title and an English abstract). Use `language=...` for every language used in the paper. The last language indicated is the main language of the paper.

For example, a French paper with additional titles and abstracts in English and German may start with the following command

```
\documentclass[sigconf, language=english, language=german,
               language=french]{acmart}
```

The title, subtitle, keywords and abstract will be typeset in the main language of the paper. The commands `\translatedXXX`, `XXX` begin title, subtitle and keywords, can be used to set these elements in the other languages. The environment `translatedabstract` is used to set the translation of the abstract. These commands and environment have a mandatory first argument: the language of the second argument. See `sample-sigconf-i13n.tex` file for examples of their usage.

21 SIGCHI Extended Abstracts

The "`sigchi-a`" template style (available only in LaTeX and not in Word) produces a landscape-orientation formatted article, with a wide left margin. Three environments are available for use with the "`sigchi-a`" template style, and produce formatted output in the margin:

`sidebar`: Place formatted text in the margin.
`marginfigure`: Place a figure in the margin.
`margintable`: Place a table in the margin.

Acknowledgments

To Robert, for the bagels and explaining CMYK and color spaces.

References

[1] Yuntao Bai, Daniel M. Ziegler, Nicholas Chen, and et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. In *arXiv preprint arXiv:2204.05862*.

[2] Lukas Gienapp, Benno Stein, Maik Fröbe, Martin Potthast, and Harrisen Scells. 2025. The Viability of Crowdsourcing for RAG Evaluation. In *arXiv preprint*. Leipzig University, University of Kassel, Friedrich-Schiller-Universität Jena, Bauhaus-Universität Weimar.

[3] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. In *ICLR*.

[4] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *arXiv preprint arXiv:2102.00050*.

[5] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of EMNLP*. 6769–6781.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Holger Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and Sebastian Riedel. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[7] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81.

[8] Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. Not All BLEU Scores Are Created Equal: Evaluating Metrics for Open-Ended Generation. *arXiv preprint arXiv:1603.08023* (2016).

[9] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. In *arXiv preprint arXiv:1901.04085*.

[10] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018*. Association for Computational Linguistics, Brussels, Belgium, 186–191. https://doi.org/10.18653/V1/W18-6319

[11] J. J. Rocchio. 1971. Relevance Feedback in Information Retrieval. *The SMART Retrieval System—Experiments in Automatic Document Processing* (1971).

[12] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, and et al. 2020. Learning to Summarize with Human Feedback. In *NeurIPS*.

[13] Rishi Thoppilan and et al. 2022. LaMDA: Language Models for Dialog Applications. In *arXiv preprint arXiv:2201.08239*.

[14] Author1 Wang and Author2. 2023. Query Expansion with Large Language Models for Improved Retrieval. In *arXiv preprint arXiv:2301.XXXXX.*

[15] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations (ICLR).* Addis Ababa, Ethiopia. OpenReview.net.

A Research Methods

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B Online Resources

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.