WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED

Bachelor Thesis

Analysis of Biases, Failures, and Limitations in N-gram-based Generative Retrieval

Richard Takacs

Date of Birth: 24.05.2005
Student ID: 12313036

Subject Area: Generative Retrieval

Studienkennzahl: UJ033560

Supervisor: Adrian Bracher, Svitlana Vakulenko

Date of Submission: TBD

Department of Information Systems & Operations Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

DEPARTMENT FÜR WIRTSCHAFTSINFORMATIK UND OPERATIONS MANAGEMENT DEPARTMENT OF INFORMATION SYSTEMS AND OPERATIONS MANAGEMENT

Contents

1 Introduction

IR (Information Retrieval), as defined by Wikipedia is "the task of identifying and retrieving information system resources that are relevant to an information need." Traditionally, IR (Information Retrieval) has long been structured around the "index-retrieve-then-rank" pipeline [1, 2], with the most popular (and best performing) methods using sparse retrieval, such as BM25 [3] and SPLADE [4]. These rely on lexical representations but suffer when terms do not overlap exactly [5, 6, 7]. A newer method called DR (Dense Retrieval), popularized by frameworks such as DPR [8] utilize the bidirectional-encoder architectures (typically from BERT [9]) to address this issue by encoding queries and documents into semantic vector representations. However, DR systems still depend on separate vector index structures [10, 11], and queries and documents are encoded independently with limited semantic interaction [12].

GR (Generative Retrieval) represents an emerging framework in IR (Information Retrieval) which aims to directly generate document identifiers through autoregressive language models such as BART [13] and T5 [14], consolidating the retrieval process into a single end-to-end model [15, 16, 17]. using the LM to encode the entire document corpus into the parameters of a single transformer. This architecture offers several advantages: it eliminates the need for separate dense vector indexes, reduces storage needs [18] reduces memory footprint [19], enables the model to learn document representations jointly with the retrieval objective [20, 21] , and enables the optimization of document representations and retrieval objectives [22] [18, 19, 21]. Despite these promising characteristics, generative retrieval systems exhibit distinct failure modes that differ fundamentally from those observed in traditional retrieval architectures. While GR failures typically manifest as low similarity scores in vector space, generative retrieval failures can include hallucination of non-existent document identifiers, incorrect pruning during autoregressive generation, poor performance on dynamically expanding corpora, and challenges with scalability [2, 23, 19, 24]. It is also worth noting that some GR implementations employ auxiliary data structures for constrained generation, making them not fully end-to-end differentiable.**recheck**

Existing literature on GR documents various challenges and failure cases. However, these failures are typically discussed in isolation with individual paper proposing new methods or architectures. There is no systematic taxonomy that classifies the types of failures that occur in generative retrieval systems, their relative frequencies in practice, or their underlying causes.

This thesis addresses this gap by developing a taxonomy of failures, biases, and limitations for GR systems by synthesizing documented failure cases

from the literature and conducting analysis of retrieval outputs from the SEAL and MINDER models [18, 25] on the NQ (Natural Questions) dataset. The NQ dataset [26] is a benchmark of Google search queries paired with full Wikipedia pages, annotated with the exact section and phrases of the page containing the answer.

1.1 Research Question

This thesis addresses two interconnected research questions:

RQ1: What is a systematic classification of the failures, behavioral biases, and structural limitations observed in generative retrieval systems?

RQ2: What are the most frequent failure modes observed in SEAL's and MINDER's n-gram-based retrieval, and to which mechanisms in the retrieval process can these failures be attributed?

1.2 Contributions

TODO REWORK This thesis makes the following contributions:

1. A synthesis of failure modes documented across generative retrieval research, organized into five categories: inference and decoding failures, memorization and representation issues, scalability limitations, dynamic corpus problems, and architectural constraints.

2. A mapping from abstract failure modes to observable indicators measurable from inference outputs.

3. Quantitative characterization of failure patterns via the SEAL and MINDER framework, including correlation analysis between failure indicators and retrieval success.

The code used for this thesis is available on Github.

2 Background

In this section we introduce the concept of GR (Generative Retrieval) and discuss the important details of the SEAL and MINDER architecture.

2.1 Information Retrieval

Systems like SEAL achieves 7x less storage than DPR while showing comparable accuracy@k and exact match metrics on the NQ (Natural Questions) 320k dataset [18]. Another GR system, PAG [19] required approximately 7x less memory to index the entire corpus compared to single-vector dense retrieval models and comparable memory needs to BM25. It achieved comparable or better MRR@10 results on the MS-MARCO Dev set compared to sparse and dense retrieval models. Notably, PAG achieved a 70.1% improvement over BM25 and an 11.2% improvement over TAS-B [27]. While some GR implementations employ auxiliary data structures for constrained generation (such as SEAL's use of the FM-Index), these differ fundamentally from the separate encoding and indexing stages required by traditional systems. Additionally, the pre-trained language models used in GR can perform effectively in zero- and few-shot scenarios when limited labeled data is available [2].

2.2 DocID Design

docid (document identifier) design is important for retrieval effectiveness, as it replaces traditional external indices with the models internal parametric memory. A docid serves as a unique "index" that the model must autoregressively generate to point toward a specific document or passage. Initial foundational works categorized these identifiers into three main types: unstructured atomic identifiers (such as unique random integers), naively structured string identifiers (such as titles), and semantically structured, numeric identifiers (such hierarchical cluster paths). While atomic IDs require the model to memorize random associations, semantic IDs group similar documents together under shared prefix tokens, improving retrieval quality [24].

String identifiers use natural language such as document titles, n-grams, URLs, or synthetic pseudo-queries, which allow the model to use its pretrained linguistic knowledge rather than learning a new symbolic system from scratch. Substring or n-gram identifiers allow any span of text within a document to act as its identifier. In SEAL and MINDER, the documents aren't assigned a single identifier per document, unlike in other frameworks such as DSI [28] or GENRE [20]. A single document can have multiple

possible identifiers, as evidenced in SEAL, which treats all n-grams within the passage as valid identifiers. Additionally, MINDER also generates pseudo-queries for each document, which it can occasionally use as "n-grams". While string-based IDs are more interpretable and generalizable to dynamic corpora where document collections expand, they often suffer from "false pruning", where the sequence-based generation discards a relevant document early in the beam search if the model fails to predict the correct prefix token. We will focus on string-based docids in this thesis.

Other docids types include semantically structured identifiers are designed to capture the relationships between documents using hierarchical categorization methods, such as by recursively applying k-means clustering to document embeddings, creating a tree where each root-to-leaf path serves as a DocID. Because they share prefixes with similar documents, they facilitate semantic matching, but their sequential nature makes them, just like string-based docids, susceptible to false pruning. Unstructured atomic identifiers treat each document as a single, unique token added directly to the model's vocabulary. Unlike sequential designs, retrieval involves only a single decoding step, where the model sorts the logits of all docid tokens to produce a ranked list. While this makes them immune to false pruning and very FLOP efficient during inference, they require a large increase in model parameters as the corpus scales [24]. Because they are often randomly initialized, they lack inherent semantic connections to the text, forcing the model to rely entirely on memorizing the association between document content and its identifier during the training phase.

docid design is consistently identified as a core challenge affecting these representation issues. Numerical docids offer efficiency but suffer from limited generalization and overfitting to initial training sets, while string-based docids maintain better semantic alignment but are more susceptible to false pruning [15, 29, 1].

2.3 SEAL Architecture

SEAL (Search Engines with Autoregressive LMs (Bevilacqua et al. 2022)) [18], addresses docid design challenges by using existing substrings as its retrieval targets. Such challenges specifically addressed are the unavailability of unique metadata (like titles) for all documents, the insufficient granularity of page- and document-level identifiers, as well as the need to force a structure onto the search space, such as hierarchical cluster trees, which can be difficult to construct for large-scale benchmarks. SEAL chunks documents into passages and defines identifiers as any n-gram within those passages. An BART model is trained to generate distinctive, query-relevant n-grams

that identify documents containing them. SEAL constrains generation during decoding so that the model only generates n-grams actually existing in the corpus. For ranking, SEAL aggregates information from multiple generated n-grams while downweighing overlaps if an n-gram overlaps with a higher-scoring one already selected. This is implemented so that the final score does not overscore based on repetitive or redundant document content.

Retrieval Mechanism

During the generation phase, SEAL enforces that at each decoding step, the language model can only generate tokens that would extend the current partial n-gram into a sequence that exists somewhere in the corpus. As an example, when SEAL has generated the partial sequence "earthquakes can be,", "earthquakes can be predicted" can only be generated if this complete 4-gram exists as a contiguous substring in at least one document [18]. "earthquakes can be predicted" exists in the corpus in the sentence "discussing why the claim that earthquakes can be predicted is false."

After n-gram generation, in the retrieval phase, for each n-gram, SEAL identifies every passage containing that n-gram and assign a relevance score based on the n-gram's uniqueness and frequency in the corpus. Passages are then ranked by additively aggregating the scores from all n-grams they contain. The FM-index, which SEAL uses, provides a range of rows in the Burrows-Wheeler Transform matrix, and each row corresponds to an occurrence of that n-gram in the corpus. SEALthen maps these occurrences back to their respective document/passage id-s.This means that a document scores highly not because any single n-gram uniquely identifies it, but because it contains multiple high-scoring n-grams. This provides robustness, since even if some highly ranked n-grams point to an irrelevant document, other n-grams can still guide retrieval toward relevant documents.

This architecture helps explain why SEAL can overcome vocabulary mismatch issues. While the model cannot generate n-grams that do not exist in the corpus, it can find paraphrases and related terms as long as those appear in corpus documents using BART's parametric knowledge.

Scoring Mechanism

SEAL's failure modes are directly influenced by its scoring function. SEAL computes n-gram scores by combining the language model's conditional probability of an n-gram given a query with corpus frequency to favor distinctive n-grams.

The unconditional n-gram probability is computed from corpus statistics:

$$P(n) = \frac{F(n, R)}{\sum_{d \in R} |d|} \tag{1}$$

where $F(n, R)$ is the frequency of n-gram n in corpus R, and $|d|$ is the length of document d.

The n-gram score combines conditional and unconditional probabilities:

$$w(n, q) = \max\left(0, \log \frac{P(n|q)(1 - P(n))}{P(n)(1 - P(n|q))}\right) \tag{2}$$

This formulation promotes n-grams that have high conditional probability given the query ($P(n|q)$), computed by BART, but low unconditional probability in the corpus ($P(n)$). This means that distinctive, query-relevant n-grams are prioritized.

It is important to highlight that $P(n|q)$ is computed autoregressively by BART and depends on both the n-gram n and the query q. Consequently, the same n-gram string can receive different scores for different queries. Therefore n-gram scores are query-dependent.

Finally, the document score aggregates contributions from multiple non-overlapping n-grams:

$$W(d, q) = \sum_{n \in K^{(d)}} w(n, q)^{\alpha} \cdot \text{cover}(n, K^{(d)}) \tag{3}$$

where $K^{(d)}$ is the subset of generated n-grams n matching document d. An n-gram is included in $K^{(d)}$ only if it has at least one occurrence in d that does not positionally overlap with a higher-scoring n-gram's occurrence. The parameter α is a hyperparameter. The coverage weight $\text{cover}(n, K^{(d)})$ downweights individual n-grams whose tokens overlap with tokens of higher-scoring n-grams in the same document, preventing passages from receiving inflated scores when they contain many lexically similar n-grams.[18].

Here it is worth noting that SEAL, generates fixed token-length ngrams (by default $k = 10$, except for titles). However, they save all partially-decoded sequences from the beam-search [18]. Because the system records this history, the final score of a passage is always the sum of strings of different lengths. As expected, n-gram length is strongly correlated with frequency ($\rho = -0.835, p < 0.001$). Consequentially,

To illustrate SEAL's n-gram based scoring, lets examine the query "who sings does he love me with reba". The top-ranked passage (titled "Linda Davis") matched the following keys (excerpt):

N-gram	Corpus Freq. $F(n, R)$	Score $w(n, q)$
`</s> Linda Davis @@`	9	216.33
`"Does He Love`	23	196.06
`Reba`	1,851	103.56
`country singles`	777	83.47
`country music singer`	4,024	39.20

Table 1: Example n-gram keys for query "who sings does he love me with reba". Lower corpus frequency correlates with higher scores for query-relevant n-grams.

It is also worth noting that while the additive nature of the scoring mechanism could theoretically bias retrieval toward longer documents in a corpus of varying document lengths, this effect is neutralized by dividing $F(n, R)$ by the total number of tokens in the corpus R.

2.4 MINDER Architecture

The MINDER (Multiview Identifiers Enhanced Generative Retrieval (Li et al. 2024)) framework [25] extends the n-gram based approach of SEAL by introducing synthetic and multiview identifiers. MINDER is motivated by the observation that single-view identifiers, such as document titles or extractive n-grams, often lack the contextualized information necessary to satisfy complex queries, especially if those require rephrasing.

To address this, MINDER assigns three distinct "views" of identifiers to each passage:

1. Title of the document
2. N-grams, similar to SEAL
3. Pseudo-queries, i.e. synthetic identifiers generated via a query-generation model that reflect potential queries

The inclusion of pseudo-queries is intended to bridge the gap between user queries and document content. While SEAL is limited to the exact word orderings present in the corpus, MINDER's thus allows the model to match queries against rephrased or summarized versions of the document via pseudo-queries.

MINDER relies on the conditional probability $P(i|q)$ of the identifier i given query q, computed by the autoregressive language model. Because pseudo-queries are typically much longer than n-grams, their raw log-probabilities

are naturally lower due to the multiplicative nature of autoregressive decoding. To ensure that pseudo-queries can compete with shorter identifiers, MINDER introduces a biased score to offset this length penalty. The final relevance score for a passage p is an aggregation of the scores from all predicted identifiers that appear in that passage across all three views:

$$S(q, p) = \sum_{i \in I_p} s(i, q) \tag{4}$$

where I_p is the set of identifiers generated by the model for passage p, and $s(i, q)$ is the language model score for identifier i.

To manage multiple identifier types, MINDER identifier-specific prefix tokens (e.g., `<TS>` for titles, `<QS>` for queries). During inference, the model is prompted with an identifier prefix and the query, and constrains the generation to valid sequences within the respective view.

2.5 Evolution of SEAL/MINDER

While this thesis focuses primarily on the foundational SEAL and MINDER architectures, it is essential to mention their successors like LTRGR and DGR.

All four systems share a common a BART-based autoregressive language model paired with an FM-Index for constrained decoding. MINDER builds upon SEAL by adding "multiview identifiers", such as pseudoqueries. LTRGR introduced a second training phase using a supervised Rank Loss, to minimize the score of negative passages relative to positive ones. DGR uses a cross-encoder model as a to imrpove the ranking list.

This thesis intentionally focuses its failure analysis on the core of such systems, as LTRGR and DGR function as "optimization layers" built upon the MINDER. Therefore, many of the fundamental behavioral biases are inherited from the underlying generative mechanics. In case that does not hold, we will specifically mention the other systems.

3 Literature Review of Failure Modes

Despite the advantages, GR systems exhibit several distinct failure modes, behavioral biases, and structural limitations that motivate this thesis. As explained by Li et al. [16], GR processes can be roughly segmented into two components, document retrieval and response generation. In this thesis, we will focus on the failure modes of the former, with briefly discussing the failure modes of the latter.

False Pruning

Prefix pruning, also referred to as false pruning occurs during autoregressive generation of docids when the beam search prematurely discards the prefix of a relevant docid because its cumulative probability at a specific step is lower than that of competing candidates [15, 19, 20, 30]. This occurs as beam search tends to get stuck in the local optima [31], as it perceives only preceding tokens and lacks visibility into subsequent ones, and thus, once a prefix is pruned, the system cannot recover the correct document regardless of subsequent token probabilities. There have been efforts to combat this, such as implementing backtracking [32] and replacing beam-search with a set-based sampling [15]. Non-autoregressive generation does not exhibit this issue [33, 34]. Empirical analysis by Zeng et al. [19] indicate that even with large beam sizes, constrained beam search often fails to match brute-force decoding performance, suggesting that relevant docids are frequently eliminated during search.

The autoregressive nature of decoding inherently favors shorter n-grams over longer ones, [35] drowning out highly specific and discriminative n-grams. Additionally, longer n-grams naturally introduce more chances of false pruning. These indicate that the global optimum itself could be a short or irrelevant prefix, making the model unable to accurately pinpoint correct passages by preferring the shortest valid docid path or the one with the most frequent initial tokens, regardless of their actual relevance to the query [30].

Contiguous n-grams

TODO!!!! As discussed in the previous paragraph 3, a fundamental structural constraint of autoregressive, beam-search and n-gram-based retrieval is that such n-grams must be strictly contiguous substrings within the corpus. This means the model cannot match n-grams that would semantically connect a query to a passage if the relevant terms are separated by intervening words or are cut off between passages.

Consider a passage stating "Tom likes pizza and tomatoes" and a query "does Tom like tomatoes?" The relevant n-gram "likes tomatoes" does not exist as a contiguous substring in the passage due to the intervening words "pizza" and "and." This issue is partially mitigated by the fact that n-grams "and tomatoes," "pizza and tomatoes," "tomatoes." are still generated and scored, however this is not a robust solution. Introducing skip-grams [22], using methods such as permutation-invariant decoding [15], or forcing overlap between passages [22] could theoretically be solutions to this problem.

Hallucination

Related is the problem of hallucination, in which the model generates invalid (with respect to the pre-defined document index) identifiers. Constrained decoding is often used to eliminate this problem with solutions such as constraining the docid generation through an FM-index [18] or a trie [36]. Using Bloom filters [37] could be another method **TODO THIS ISNT VALIDATED**. There have been several implementations of constrained decoding, such as SEAL [18] and MINDER [25].

Several GR solutions use intentional hallucination, such as MINDER[25], where the generation of additional identifiers, such as pseudoqueries is achieved through intentional, constrained hallucination of docid generation.

Forgetfulness

One of the problems in GR is the apparent forgetting of fine-grained features [35]. While generative models capture coarse-grained semantic clusters effectively, they struggle to accurately memorize and decode fine-grained document distinctions [38]. According to Yuan et al. [35], error rates (defined as the % of tokens predicted at a position that does not match the GTtoken in a docid) for NCI [39] increase significantly at later token positions in docid sequences, rising from approximately 1% at the first position to over 12% by the sixth position.

Indistinct Identifiers

GR systems also suffer from cases where the mapping from document content to identifiers is not sufficiently distinct [40]. When using synthetic queries [41] or semantic clustering [28] as identifiers, multiple documents may share quasi-identical or highly similar representations, causing the model to conflate distinct documents [42, 20, 16].

Scalability

Scalability limitations become evident as corpus size increases, as GR models exhibit limitations in terms of capacity [43, 30]. The "parameter budget" of the transformer can become insufficient to encode nuances of a massive corpus. Empirical studies published by Pradeep et al. [24] show sharp performance decline on T5 [14] as corpora scale from 100,000 to 8.8 million passages. For example, the T5-XL model achieves only around 30% of the MRR@10 score on the full MS MARCO corpus compared to the base corpus.

Associated with scalability is the issue that oftentimes increasing model size does not yield proportional improvements. According to the aforementioned study [24], scaling T5 from XL (3B parameters) to XXL (11B parameters) degraded retrieval performance when using naive numeric identifiers on the full MS MARCO dataset, despite identical training configurations.

Additionally, unstructured atomic identifiers, while considered non-mainstream assign each document a single unique integer identifier [28], requiring the model's output vocabulary to equal the corpus size. Since the output projection layer has dimensions of hidden size $\times$ vocabulary size [44, 28], it makes this approach prohibitively expensive for large corpora. As an example, 8.8 million documents (the amount of passages in MsMarcoFull [45]) with a hidden size of 768 (the hidden dimension of the T5-base model [14]) adds approximately 6.8 billion parameters to the output layer alone, making this approach prohibitively expensive for large corpora. In contrast, sequential identifiers reuse a fixed vocabulary across multiple decoding steps, keeping parameter count constant regardless of corpus size [17, 24]. **recheck this!**

Dynamic Corpus

In environments where corpora change, GR models exhibit two opposing failure modes [46, 29, 35, 47]. Forgetting occurs when updating a model with new documents degrades its ability to retrieve previously indexed documents. Experiments demonstrate that sequential indexing of new batches can cause indexing accuracy for the initial corpus to drop by more than 25 points, necessitating retraining and increasing costs [47]. The opposing failure is an excessive bias against recency, where models fail to retrieve newly added content, favoring documents from the original training corpus [29].

GR models also exhibit forgetting during initial training as well. Analysis of training dynamics shows models frequently "forget" and "re-learn" document-to-identifier mappings, with approximately 88% of documents undergoing at least one forgetting event during training [47].

An additional issue identified is the problem that newly added documents are practically inaccessible until model weights are updated and the model retrained, unlike dense retrieval systems which can index new embeddings instantly without retraining the encoder [17].

Interpretability

Lastly, the lack of interpretability in generation processes presents challenges for understanding and debugging failures. Unlike sparse retrieval where term matching is transparent [3, 4], or dense retrieval where embedding similarity

can be analyzed [8], the internal decision processes of generative retrieval is significantly more opaque [1, 48].

4 Proposed Failure Taxonomy

TODO NOT READY

Table 2: Proposed Taxonomy of Generative Retrieval Failures.

Category	Specific Failure Mode
Data-Level	Annotation Errors Answer Cut-off between Passages
Representational ?	Metadata Overreliance Cultural Bias Model Capacity
Algorithmic ?	DocID Discriminativeness ? Short DocID Preference False Pruning

OUTDATED!!!

5 Methodology

Our methodology is threefold. First, we ran SEAL and MINDER with their default settings on the NQ (Natural Questions) [26], analyzing their outputs. Second, we will run the models on the MSMARCO dataset [45] as well.Third, we will perform ablation studies based on our identifier failure modes from the literature review in Section 3 and the failure modes identified during our analysis of the outputs.

We chose NQ (Natural Questions) as it is an academic standard for retrieval benchmarking. The NQ retrieval corpus consists of a Wikipedia dump split into approximately 21 million passages of 100 tokens each.

The SEAL and MINDER output dataset contains the results of 6,515 queries from the Natural Questions benchmark. For each query, the following relevant information is available:

- The original natural language query

Table 3: Classification of potential retrieval failure modes for SEAL and MINDER.

Type	Potential Failure Mode	Applicability
Measurable	Nonspecific n-grams	Both
	Too many unigrams	Both
	Redundant n-grams	Both
	Query–n-gram not overlapping	Both
	Answer not in n-gram	Both
	Single n-gram dominating	Both
	Title repetition in top-k	Both
	No pseudo-queries as n-gram	MINDER
Non-measurable	False pruning	Both
	Answer cut-off btwn. passages	Both
	Optima not found in beam search	Both
	Scalability issues	Both
	Handling of dynamic corpora	Both
	Training learning bottlenecks	Both

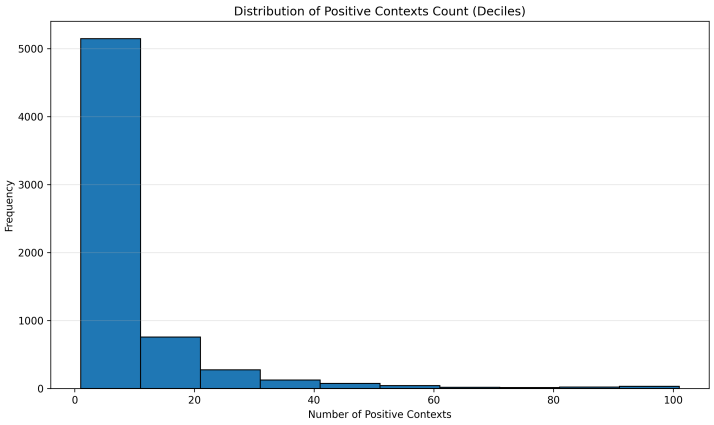
- GT answer string(s)
- Annotated GTpassages containing the answer
- Negative and hard negative passages
- The model's top-100 retrieved passages, ranked by passage score, each containing:
  - Passage title and text
  - Aggregate document score
  - The set of matched identifiers and their respective scores

From the output data available, the n-gram keys are deemed most useful to understand SEAL's andMINDER's retrieval decisions. Each key entry contains three parts:

1. The n-gram string (e.g., `‘‘ Reba’’`, `‘‘</s> Linda Davis @@’’`)
2. The corpus frequency of how many times this n-gram appears across all documents
3. The n-gram score computed via the scoring function

The MINDER output is structurally identical to that of SEAL's, including being ran on the same 6,515 queries. The main difference arises from the pseudoqueries, which are appended to the *text*.

The output dataset (of both systems) have an average per-query annotated correct passage count of 8.46, with the minimum being 1 and maximum being 101. This is important to analyze to accurately identify evaluation metrics. To illustrate the distribution, please consult the histogram below.



Evaluation Metrics

We use R-precision as the overall retrieval quality of both SEAL and MINDER and hits@1 (being identical to precision@1) to quantify whether the model successfully retrieved a positive context in its top results for a query. We also use hits@10, since retrieval is rarely only about the first result only. We chose $k = 10$ somewhat arbitrarily, as it provides a balance between being too strict by only measuring the top results and being too lax by allowing lower-scored retrieved passages to be included in the analysis. We avoid using Precision@k and Recall@k with $k > 1$ to avoid misrepresentation of queries where the amount of ground truths is less than k.

For the analysis, results are calculated side-by-side for both systems analyzed when appropriate. To analyze distribution, we bin the results into deciles and report them. Additionally, when appropriate, we report the Spearman *rho* values to analyze monotonic correlation between variables.

6 Failure Analysis

Metadata Dependency

Both SEAL and MINDER rely passage metadata heavily to retrieve the correct passages. For SEAL, in the top-100 retrieved passages among the 6515 queries analyzed, titles account for 39.85% of total score. For MINDER, titles make up 29.6% and pseudoqueries 19.67% of total score. In cases where the answer string is found exactly both as a "title n-gram" and "nontitle n-gram", the title n-gram receieves on average 2866% more score in SEAL and 2499% inMINDER. Interestingly, this overreliance drops when we only account for successfully retrieved passages, with titles accounting for 29.1% for SEAL and 20.6% for MINDER(11,27% for pseudoqueries). The scoring differential between the answer string being a "title n-gram" and "nontitle n-gram" also drops to 1876% for SEAL and 1439% For MINDER. This can indicate that when the model is "correct", it is using more of the body-textThis can be considered a "feature" on cases where there is clear metadata in the dataset (e.g. NQ) and cases where retrieving the correct document is more important than retrieving the current passage. However, if our aim is to create a generalized GR model, this should be considered a "failure".

Both Bevilacqua et al. [18] and Li et al. [25, 49, 50] note that performance drops when titles are unavailable [18, 25, 49, 50]. Bevilacqua et al. [18], however, does not report SEAL ablation results without titles as identifiers. MINDERperforms 10.3% worse recall@5 and 7.4% worse recall@20 on NQ without titles and pseudoqueries. On MSMARCO [45], SEAL had a 30.7% worse Recall@5 than BM25 on MSMARCO, with MINDER only marignally (3.1%) better than BM25, likely due to pseudoqueries. Neither SEALnor MINDER achieved a Recall@5 larger than 30% on MSMARCO. LTRGR and DGR achieved 40.2% and 42.9%, respectively. It is also worth noting that NQ is based on highly structured Wikipedia passages, while MSMarco is less cleanly structured with messier or lacking metadata.

Our analysis shows that For SEAL, title score is higher by 280% on average for positive "ground truth" retrieved passages than negative ones. However, the median is only 41% and the σ 1113%, indicating that there is a subset of queries where the title is massively scored, indicating that the model uses it as quasi the only identifier, while for others the title is much less helpful. For MINDER, accordingly, the values are 325%, 57.5%, 939%. An extreme example was found inSEAL, where for the query "who sang this is how we do it", correct passages have 545x more title score than incorrect passages (in passages where a title-as-ngram is present). In these cases, title matching is extremely discriminative, which makes it well suited for datasets

with clear metadata, but potentially bad for ones without.

Cultural Bias

English content in public corpora is itself not culturally neutral, as a substantial portion reflects Western (particularly US and Eurocentric) perspectives and framing simply because such sources dominate English public corpora. As BART (which SEAL and MINDER uses) was trained on the English Wikipedia and on BookCorpus [51, 52], it can be presumed that BART would implicitly default to Western interpretations in ambiguous cases. Other commonly used language models, such as T5 and LLaMa-3 [14, 53] were also trained on overwhelmingly English data. Similar bias was also demonstrated for LLMs by multiple papers [54, 55, 56].

Similarly, the NQ dataset is also based on the English Wikipedia [26], adding onto the issue. Such a "Western-centricity" can be also be shown in other commonly used datasets, such as MSMarco citebajaj2018msmarcohumangenerated, Trivia QA [57] and the KILT datasets [58].

For SEAL, this is best illustrated by the query: "where did the southern song have their capital".

Although the correct answer is Lin'an (the query referring to the Southern Song Dynasty), SEAL focused entirely on passages about the southern United States, especially about about Tennessee and Virginia. The model likely found a quasi-perfect "query keyword match" for such passages and connected "southern" with the southern US, "song" with country music, the 'Dixie' song and the 'Music City' sobriquet of Nashville, Tennessee, and, building onto these, "capital" with Nashville being the capital of Tennesse and Richmond being the capital of Virginia and the former capital of the confederate South. These substrings, while common, were likely propped up by their implicit connection to the US in the dataset and BART, even though they possess different meanings in a US geography context versus a 12th-century Chinese history context.

There is no ground truth passages retrieved in the *top-100* retrieved passages, which consists entirely of passages about the southern United States. This is also an example of *false pruning*, as SEALseemingly discarded the tokens related to the Song Dynasty early on.

An important observation is that MINDER successfully retrieved the relevant passages. All queries in the *top-100* retrieved passages were in some way relevant to Ancient China, with the first ground-truth passage appearing at rank-4. Both systems use BART-large as their autoregressive language model and the same dataset to build their FM-index. The observed success is therefore likely due to the use of pseudoqueries. Since BART is capa-

ble of generating the correct terms, failure in SEAL happened likely due to failing to disambiguate the context, in which pseudoqueries assisted. For the highest-ranked retrieved GT passage, MINDER generated pseudoqueries such as *what was the capital of the song dynasty in north china* and *who took over the capital of the song dynasty*.

Annotation Issues

The reliability of GR evaluation depends on the quality of GT annotations in benchmark datasets. When those contain annotation errors, such as relevant passages incorrectly labeled as negative or vice versa, the measured performance metrics no longer accurately reflect a system's true retrieval capabilities. These can lead to misleading conclusions being published that do not generalize to real-world scenarios or accurately represent comparisons.

For QA tasks, multiple passages may contain semantically equivalent answers but only a subset are annotated as ground truth. The authors of MS MARCO, TriviaQA, and NQ each acknowledge this concern [45, 57, 26].

TODO BELOW EITHER REWRITE OR IDK or maybe add the tables as appendix or idk. this below isnt really relevant to actual failure modes, similarly how most correl analysis isnt relevant either.

During our analysis, we identified such cases where the annotation can cause misleading evaluation results. Here we present 3 such examples: **Example 1: "which apostle had a thorn in his side" (answer: Paul).** The ground-truth passage is from the Wikipedia article "Thorn in the flesh," which explains that Paul the Apostle used this phrase in 2 Corinthians 12:7–9. SEAL's top result was from Galatians 3, which mentions "Paul the Apostle" as the author of that epistle but contains nothing about a thorn. Even though SEAL seemed to be able to identify "Paul" as a relevant term, potentially from BART's parametric knowledge, Galatians 3 is the wrong chapter entirely. SEAL matched on generic terms ("Paul," "apostle") but never generated n-grams containing "thorn."

Example 2: "how many judges currently serve on the supreme court" (answer: nine). The ground-truth passage directly states the current composition of the Court. SEAL's top result is a historical passage stating that in 1869, "the Circuit Judges Act returned the number of justices to nine, where it has since remained." This is borderline, because the reader could infer the answer, but the passage is not specifically about the current number of judges.

Example 3: "who is the youngest judge currently sitting on the u.s. supreme court" (answer: Neil Gorsuch). The ground-truth pas-

sage discusses Gorsuch's nomination process and him being the youngest sitting justice. SEAL's top result directly states: "Neil Gorsuch is the youngest justice sitting, at 50 years of age." This is a success, because SEAL found a passage that answers the question, even if not annotated.

These examples show that these cases are a mix of such cases. To determine how many fall into each category, in section A, we apply LLM-as-a-judge to classify whether each retrieved passage genuinely answers the question or not.

TODO FROM HERE If most of the 349 cases are genuine retrievals, then SEAL's effective top-10 recall is closer to 81.5% rather than 76.1%. If most are coincidental matches, then 76.1% remains the accurate figure. Similar calculations can be done for MINDER.

For each case, we present a large language model (`claude-sonnet-4-5-20250929`) with the original question, the expected answer, and the retrieved passage text, asking it to determine whether the passage genuinely answers the question or whether the answer string's presence is coincidental. The model was chosen based on expected reasoning capabilities and cost of operation.

The judge classifies each case into one of three categories:

- **YES**: The passage directly and unambiguously answers the question.
- **NO**: The answer string appears coincidentally and the passage does not answer the question.
- **PARTIAL**: The passage provides relevant information from which the answer could be inferred, but does not state it directly.

Overfitting - Underfitting

UNVALIDATED If the model is trained through too few epochs, it can underfit and retrieval can be noisy. If it it trained on too many, the model memorizes the training set too well and loses the ability to generalize to new queries. This is especially problematic if the model is expected to "zero-shot" retrieve or when the model did not see the documents beforehand

Model capacity

UNVALIDATED

Smaller models might not have capacity to memorize enough docids.

Length Bias

UNVALIDATED The model might prefer short, common ngrams vs specific, long identifiers.

Latency

UNVALIDATED Beam search with constraints is probably significantly slower than dot-product (or similar) searches

Vocabulary - Tokenization

UNVALIDATED

Subword UNVALIDATED
If an important identifier (e.g. a proper noun like Lin'an from the TODO example) is split into many meaningless subwords, the model might struggle to generate it consistently compared to common words. This can happen even through BART uses BPE [22], which allows it to to handle "out-of-vocabulary" words by breaking them down into known subunits.

Coverage UNVALIDATED If BART doesn't have good representations for domain-specific jargon, e.g. for rare medical terms, retrieval can fail.

Zero-Shot Retrieval

UNVALIDATED Supervised GR models usually perform significantly worse on datasets they weren't explicitly trained on [59, 29]. Since they usually lack robust matching capabilities like that of BM25 or Contriever [22] in zero-shot settings because they tend to overfit the training query distribution

Unseen Documents

UNVALIDATED
GR by default cannot retrieve documents added post-training . While the probability is not infinite, as documents added ex-post still get indexed by the FM-index, they are not represented in the training distribution
Othertimes, a full retraining is required to index new information.

Sampling Failures

UNVALIDATED If the model is trained on irrelevant substring, it confuses the model's query-document mapping

6.1 Generative Failures

Using the identified failure modes in Table 3, we present the analysis results pertaining to generation below.

Low Lexical Specificity

N-gram corpus frequency reflects how distinctive a generated n-gram is. Rare n-grams provide stronger evidence for specific documents, while common n-grams match many topically-related passages. We analyze the average corpus frequency of the top-scored n-grams to determine whether the frequency (and thus score) of n-grams contribute to retrieval outcomes.

Across 6,515 queries, the average mean corpus frequency of the top-5 scored n-grams for the top-1 retrieved passage is 28753 for SEAL and 21310 for MINDER, while the same value using all-ngrams for such passage is 382,136 and 497,399, respectively. $k = 5$ was arbitrarily selected as a representation of the n-grams with the strongest influence on the overall score of each passage.

Table 4 summarizes the results. The decimals of the frequencies have been truncated for readability.

Table 4: Hits@k success rate by n-gram frequency deciles.

	SEAL		MINDER	
Decile	**Freq. Range**	**H@1 / H@10**	**Freq. Range**	**H@1 / H@10**
D1	2.0–91	71.0% / 92.8%	1–16	76.5% / 93.9%
D2	91–356	51.5% / 85.1%	16–55	65.9% / 89.4%
D5	2,940–5,698	43.0% / 76.3%	513–1,359	45.8% / 80.2%
D8	18,581–35,824	31.4% / 70.4%	7,434–16,582	36.7% / 72.1%
D10	81,049–719,139	25.9% / 65.8%	44,813–1.7M	26.8% / 60.7%

We can see that in D1, where the top-1 passage's top-5 scored n-grams average frequency is low, both hits@1 and hits@10 is surprisingly high for both systems, with MINDER performing marginally better, while in D10, where the same average frequency is considerably high (implying generic n-grams), the retrieval performance drops significantly for both systems. The

Spearman's ρ between such frequency and hits@1 is -0.238 and -0.290 respectively, both being $p < 0.001$, indicating that average n-gram frequency is inversely correlated with retrieval success.

Notably, using average frequency across all n-grams (instead of top-5) shows no significant correlation with performance (SEAL: $\rho = 0.029, p = 0.02$; MINDER: $\rho = -0.008, p = 0.51$), suggesting that the presence of highly-ranked generic terms, rather than overall genericity, drives performance degradation. This happens because such high-frequency n-grams match many topically-related passages, diluting discriminative power among passages. While SEAL's scoring function theoretically downweights common n-grams through the $P(n)$ term in Equation 2, this analysis reveals that high-frequency n-grams can still dominate the top-ranked positions. MINDER does not penalize high frequency n-grams. Instead, it artifically boosts the score of pseudoqueries [25].

Preference for Unigrams

Here we examine whether the analyzed systems disproportionately generate short n-grams (unigrams and bigrams) rather than longer, more distinctive multi-word sequences. Longer n-grams generally provide stronger discriminative power, as multi-word phrases are far less likely to appear across multiple passages than individual words.

We quantify this generation bias by measuring the fraction of generated n-grams that consist of a single token in the top-1 retrieved passage. Across 6,515 queries, the mean unigram proportion is 0.869 ± 0.066 (median 0.878) for SEAL and 0.853 ± 0.071 (median 0.863) for MINDER. The average n-gram length is 1.28 ± 0.18 words (median 1.24) for SEAL, and 1.48 ± 0.39 (median 1.35) for MINDER. This indicates that, on average, more than four-fifths of generated n-grams are unigrams for both systems. MINDER shows slightly higher variance in length ($\sigma = 0.39$ vs. 0.18).

We categorize queries into deciles based on the unigram proportion. Queries in D1 (lowest unigram proportion) achieve considerably higher hit rates than those in D10, as evidenced by Table 5.

The results are interesting for MINDER specifically, as MINDER is able to use its generated pseudoqueries (which tend to be longer) as n-grams. This correlates with the results presented below at Section 6.2, indicating that pseudoqueries appearing in n-grams provides the retrieval model with a higher hitrate. MINDER also shows a higher hit rate at lower unigram proportion deciles.

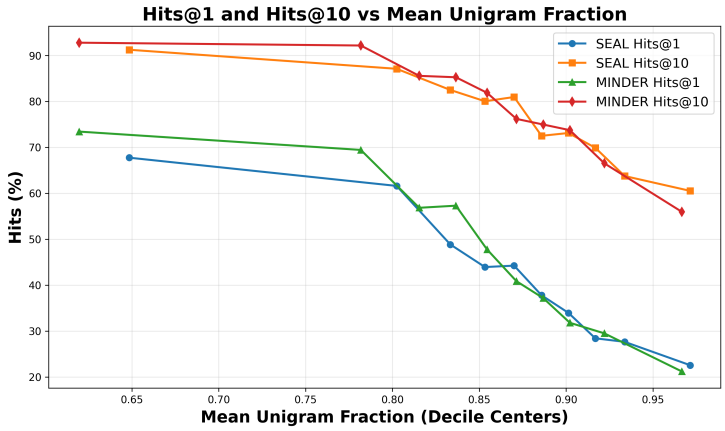
The Spearman ρ between unigram proportion and hits@k for SEAL is $\rho - 0.278$ for $k = 1$ and $\rho - 0.220$ for $k = 10$, while for MINDER is $\rho - 0.330$

Table 5: Hits@k success rate by unigram proportion deciles.

Decile	SEAL Div. Range	SEAL H@1 / H@10	MINDER Div. Range	MINDER H@1 / H@10
D1	0.51–0.78	67.7% / 91.2%	0.48–0.76	73.4% / 92.8%
D2	0.78–0.82	61.6% / 87.1%	0.76–0.80	69.4% / 92.2%
D3	0.82–0.84	48.8% / 82.5%	0.80–0.83	56.8% / 85.6%
D5	0.86–0.88	44.2% / 81.0%	0.85–0.86	47.8% / 81.9%
D8	0.91–0.92	28.4% / 69.9%	0.89–0.91	31.8% / 73.7%
D10	0.94–1.00	22.6% / 60.5%	0.93–1.00	21.2% / 55.9%

for $k = 1$ and $\rho - 0.260$ for $k = 10$. Each value is statistically significant at $p < 0.001$.

To understand the results more illustratively, we summarized the table in the following chart, clearly showing MINDER's marginally higher success rate at low unigram proportions. The hit rates converge at higher unigram proportions.



Query-to-n-gram Overlap

Here we measure the lexical alignment between query terms and the n-grams. We use `GPT2TokenizerFast` for this. High overlap indicates that the model successfully generated n-grams containing query-relevant terms, while low overlap suggests a mismatch where generated n-grams, though potentially topically related, do not directly address the query's specific terms. We quantify this using the fraction of query tokens that appear in any generated n-gram in the highest-scored retrieved passage, denoted by $C(Q, G_n)$, where Q denotes the query tokens and G_n denotes the tokens of the generated

n-grams.

Our analysis across 6,515 queries shows that the token-level lexical overlap varies substantially across the top-1 retrieved passages. As visible in Table 6, there is a positive correlation between lexican overlap and retrieval success. For SEAL, $\rho = 0.323$ for $k = 1$ and $\rho = 0.190$ for $k = 10$. For MINDER, the Spearman correlations are $\rho = 0.380$ and $\rho = 0.202$, respectively. All values are statistically significant at $p < 0.001$.

Table 6: Hits@k by $C(Q, G_n)$ deciles.

	SEAL		MINDER	
Decile	$C(Q, G_n)$	**H@1 / H@10**	$C(Q, G_n)$	**H@1 / H@10**
D1	0.00–0.27	12.4% / 55.1%	0.00–0.36	14.4% / 61.5%
D2	0.29–0.36	24.5% / 70.2%	0.37–0.44	25.7% / 67.9%
D3	0.36–0.40	29.6% / 74.5%	0.45–0.50	35.1% / 71.2%
D5	0.46–0.50	42.3% / 76.9%	0.58–0.63	44.2% / 79.7%
D8	0.61–0.67	53.4% / 78.1%	0.76–0.80	62.8% / 85.7%
D10	0.73–1.00	70.0% / 88.4%	0.90–1.00	77.4% / 88.6%

As visible from Table 6, the hit rate of SEAL and MINDER are relatively similar, and are positively correlated with $C(Q, G_n)$. Similar to other failure modes, low query coverage typically arises when the model generates semantically related but lexically distinct n-grams (e.g., for a query containing "automobile," the model could generate n-grams with "car") or when the model generates generic topical n-grams that lack the specific query terms needed to isolate the answer (e.g., a query about "Victorian architecture in London" generates only "architecture" and "London").

Answer as N-gram

We analyze whether retrieval success correlates with the model's ability to generate the specific answer string as an identifier, or whether it relies on the surrounding context. While generating the answer string provides a valuable anchor for a passage, the primary goal of GR is document identification, not direct question answering.

Table 7 illustrates the results of our analysis. We looked at whether among the top-1 and top-10 scored retrieved passages, at least one of them contained the answer string or not and what the Hits@1 value is for these cases. The results are mostly similar between SEAL and MINDER, with most top-1 ranked retrieved passages not containing the answer string, yet when

they do, the respective query's Hits@1 rate doubles and Hits@10 increases significantly.

Table 7: Performance metrics conditioned on answer string presence.

	SEAL		MINDER	
Metric	**Pres.**	**Abs.**	**Pres.**	**Abs.**
Queries (N)	1,096	5,419	1,457	5,058
Hits@1	75.0%	34.9%	75.4%	38.2%
Hits@10	93.5%	72.6%	93.0%	74.3%

We can see that in cases where the answer string is identified among the n-grams, the success rate jumps significantly for both systems. This indicates that direct identification is a strong predictor of retrieval success. Both SEAL and MINDER are able to retrieve (at least) one correct passage in around 35% of cases at the first place and 73% of cases among the top-10 retrieved, suggesting that the model is able to identify the semantic environment in which an answer is likely to be in relatively convincingly. That said, the performance gap between the two cases indicates that when the model fails to generate the specific answer as an identifier, the remaining contextual cues are oftentimes too generic to distinguish the correct passage from topically similar alternatives.

6.2 Scoring Failures

Just like in Section 6.1, we present the analysis results of the identified failure modes (Table 3) pertaining to scoring below.

Overreliance on a Single Identifier

Overreliance on a single identifier measures the extent to which the top-1 ranked passage's total score is concentrated in one high-scoring n-gram versus distributed across multiple n-grams. We define this as the ratio of the highest scored n-gram score and the sum of all matched n-gram scores for the passage.

Our analysis across 6,515 queries reveals that lower such ratio correlates with higher retrieval success, as shown in Table 8. For SEAL, we observe a Spearman correlation of $\rho = -0.115$ ($p < 0.001$), with success rates declining from 48.6% in the lowest decile (D1) to 29.6% in the highest decile (D10).

This trend is also noticable in MINDER ($\rho = -0.169$, $p < 0.001$), which also exhibits a lower mean ratio of 0.36 compared to SEAL's 0.44 and a

Table 8: Hits@1 by top n-gram ratio deciles (max score / sum scores).

Decile	SEAL (Mean: 0.44) Range	SEAL (Mean: 0.44) Hits@1	MINDER (Mean: 0.36) Range	MINDER (Mean: 0.36) Hits@1
D1	0.12–0.30	48.6%	0.12–0.25	58.0%
D2	0.30–0.35	45.8%	0.25–0.28	59.3%
D3	0.35–0.38	45.4%	0.28–0.31	55.8%
D5	0.41–0.44	44.3%	0.33–0.35	45.1%
D8	0.50–0.53	35.4%	0.40–0.43	40.2%
D10	0.57–0.85	29.6%	0.48–0.94	32.5%

higher hit rate in all deciles. This shift toward more distributed evidence is assumed to be a consequence of MINDER's multiview identifiers, since by aggregating scores from titles, pseudo-queries, and substrings, MINDER theoretically reduces reliance on any single identifier.

This finding aligns with the additive scoring design, which benefits from distributed evidence by providing redundant confirmation across multiple n-grams. Conversely, if the top-scoring n-gram matches many passages or identifies the wrong one, the remaining low-scoring n-grams may provide insufficient corrective evidence to ensure correct ranking.

Pseudo-Query Contribution inMINDER

Here we analyze the contribution of synthetic identifiers in the MINDER framework. As detailed in Section 2.4, MINDER incorporates pseudo-queries alongside titles and substrings to capture high-level semantic information. Our analysis of 6,515 queries reveals an interesting divergence between the quantity of generated pseudo-queries and their influence on the final ranking.

Numerically, pseudo-queries constitute a negligible fraction of the generated identifiers, accounting for only 0.32% of the total n-grams (87714 out of 26.9 million). However, their contribution to the total accumulated score is disproportionately high, representing 4.52% of the total score mass (14.8 million out of 328.5 million). This confirms that MINDER's scoring mechanism successfully uses sparse but highly discriminative synthetic identifiers.

Interestingly, titles still dominate the picture the scoring disproportionately, as only 1.08% of ngrams are titles yet contribute 23.2% of the total score. If we combine pseudoqueries and titles, the resulting numbers are 1.4% and 27.72%, respectively.

7 Discussion

TODO

7.1 Key Findings

TODO

7.2 Generalizability

TODO

8 Conclusion

TODO

AI Use Disclosure

Some portions of code for this thesis were generated with the assistance of *Claude by Antropic* and *ChatGPT by OpenAI* large language models. All AI-generated code was reviewed, edited, and verified by the author.

A LLM-as-Judge prompt

The prompt used for the LLM-as-Ludge component in Section A is pasted below:

```
Question: [QUESTION]
Expected answer: [ANSWER]
Retrieved passage title: [PASSAGE TITLE]
Retrieved passage text: [PASSAGE TEXT]

Does this passage answer the question?
– YES: The passage directly answers the question.
– NO: The answer string appears coincidentally but does
    not answer the question.
– PARTIAL: Contains relevant information but requires
   inference.

The answer explanation should not exceed 100 words.
```

References

[1] Yubao Tang, Ruqing Zhang, Weiwei Sun, Jiafeng Guo, and Maarten De Rijke. Recent advances in generative information retrieval. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1238–1241, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3641239. URL https://doi.org/10.1145/3589335.3641239.

[2] Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. Rethinking search: making domain experts out of dilettantes. *ACM SIGIR Forum*, 55(1):1–27, June 2021. ISSN 0163-5840. doi: 10.1145/3476415.3476428. URL http://dx.doi.org/10.1145/3476415.3476428.

[3] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389, 01 2009. doi: 10.1561/1500000019.

[4] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3463098. URL https://doi.org/10.1145/3404835.3463098.

[5] Eunseong Choi, Sunkyung Lee, Minjin Choi, Hyeseon Ko, Young-In Song, and Jongwuk Lee. Spade: Improving sparse representations using a dual document encoder for first-stage retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 272–282. ACM, October 2022. doi: 10.1145/3511808.3557456. URL http://dx.doi.org/10.1145/3511808.3557456.

[6] Biplob Biswas and Rajiv Ramnath. Efficient and interpretable information retrieval for product question answering with heterogeneous data, 2024. URL https://arxiv.org/abs/2405.13173.

[7] Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list, 2021. URL https://arxiv.org/abs/2104.07186.

[8] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage

retrieval for open-domain question answering, 2020. URL https://arxiv.org/abs/2004.04906.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

[10] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval, 2020. URL https://arxiv.org/abs/2007.00808.

[11] Haitao Li, Qingyao Ai, Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Zheng Liu, and Zhao Cao. Constructing tree-based index for efficient and effective dense retrieval, 2023. URL https://arxiv.org/abs/2304.11943.

[12] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers, 2021. URL https://arxiv.org/abs/2112.07899.

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

[14] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[15] Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, Fangchao Liu, and Zhao Cao. Generative retrieval via term set generation, 2024. URL https://arxiv.org/abs/2305.13859.

[16] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From matching to generation: A survey on generative information retrieval, 2025. URL https://arxiv.org/abs/2404.14851.

[17] Tzu-Lin Kuo, Tzu-Wei Chiu, Tzung-Sheng Lin, Sheng-Yang Wu, Chao-Wei Huang, and Yun-Nung Chen. A survey of generative information retrieval, 2024. URL https://arxiv.org/abs/2406.01197.

[18] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. Autoregressive search engines: Generating substrings as document identifiers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Z4kZxAjg8Y.

[19] Hansi Zeng, Chen Luo, and Hamed Zamani. Planning ahead in generative retrieval: Guiding autoregressive generation through simultaneous decoding, 2024. URL https://arxiv.org/abs/2404.14600.

[20] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Learning to tokenize for generative retrieval, 2023. URL https://arxiv.org/abs/2304.04171.

[21] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. Listwise generative retrieval models via a sequential learning process, 2024. URL https://arxiv.org/abs/2403.12499.

[22] TODO. Todo. In TODO, editor, *TODO*, TODO. URL TODO.

[23] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023. ISSN 1557-7341. doi: 10.1145/3571730. URL http://dx.doi.org/10.1145/3571730.

[24] Ronak Pradeep, Kai Hui, Jai Gupta, Adam D. Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q. Tran. How does generative

retrieval scale to millions of passages?, 2023. URL https://arxiv.org/abs/2305.11841.

[25] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. Multiview identifiers enhanced generative retrieval, 2023. URL https://arxiv.org/abs/2305.16675.

[26] 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, Jakob Dai, Andrew M.and Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

[27] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling, 2021. URL https://arxiv.org/abs/2104.06967.

[28] Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index, 2022. URL https://arxiv.org/abs/2202.06991.

[29] Zhen Zhang, Xinyu Ma, Weiwei Sun, Pengjie Ren, Zhumin Chen, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Replication and exploration of generative retrieval over dynamic corpora, 2025. URL https://arxiv.org/abs/2504.17519.

[30] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. Scalable and effective generative information retrieval, 2023. URL https://arxiv.org/abs/2311.09134.

[31] Felix Stahlberg and Bill Byrne. On nmt search errors and model errors: Cat got your tongue?, 2019. URL https://arxiv.org/abs/1908.10090.

[32] R. Zhou and Eric A. Hansen. Beam-stack search: Integrating backtracking with beam search. In *International Conference on Automated Planning and Scheduling*, 2005. URL https://api.semanticscholar.org/CorpusID:11314454.

[33] Ravisri Valluri, Akash Kumar Mohankumar, Kushal Dave, Amit Singh, Jian Jiao, Manik Varma, and Gaurav Sinha. Scaling the vocabulary of non-autoregressive models for efficient generative retrieval, 2024. URL https://arxiv.org/abs/2406.06739.

[34] Xinpeng Zhao, Zhaochun Ren, Yukun Zhao, Zhenyang Li, Mengqi Zhang, Jun Feng, Ran Chen, Ying Zhou, Zhumin Chen, Shuaiqiang Wang, Dawei Yin, and Xin Xin. Diffugr: Generative document retrieval with diffusion language models, 2026. URL https://arxiv.org/abs/2511.08150.

[35] Peiwen Yuan, Xinglin Wang, Shaoxiong Feng, Boyuan Pan, Yiwei Li, Heda Wang, Xupeng Miao, and Kan Li. Generative dense retrieval: Memory can be a burden, 2024. URL https://arxiv.org/abs/2401.10487.

[36] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval, 2021. URL https://arxiv.org/abs/2010.00904.

[37] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970. ISSN 0001-0782. doi: 10.1145/362686.362692. URL https://doi.org/10.1145/362686.362692.

[38] Ye Wang, Xinrun Xu, and Zhiming Ding. Mindref: Mimicking human memory for hierarchical reference retrieval with fine-grained location awareness, 2025. URL https://arxiv.org/abs/2402.17010.

[39] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, Xing Xie, Hao Allen Sun, Weiwei Deng, Qi Zhang, and Mao Yang. A neural corpus indexer for document retrieval, 2023. URL https://arxiv.org/abs/2206.02743.

[40] Fuwei Zhang, Xiaoyu Liu, Xinyu Jia, Yingfei Zhang, Shuai Zhang, Xiang Li, Fuzhen Zhuang, Wei Lin, and Zhao Zhang. Multi-level relevance document identifier learning for generative retrieval. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10066–10080, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.497. URL https://aclanthology.org/2025.acl-long.497/.

[41] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. Semantic-enhanced differentiable search index inspired by learning strategies, 2023. URL https://arxiv.org/abs/2305.15115.

[42] Jiehan Cheng, Zhicheng Dou, Yutao Zhu, and Xiaoxi Li. Descriptive and discriminative document identifiers for generative retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(11):11518–11526, Apr. 2025. doi: 10.1609/aaai.v39i11.33253. URL https://ojs.aaai.org/index.php/AAAI/article/view/33253.

[43] Thong Nguyen and Andrew Yates. Generative retrieval as dense retrieval, 2023. URL https://arxiv.org/abs/2306.11397.

[44] Anja Reusch and Yonatan Belinkov. Reverse-engineering the retrieval process in genir models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 668–677, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3730076. URL https://doi.org/10.1145/3726302.3730076.

[45] 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. Ms marco: A human generated machine reading comprehension dataset, 2018. URL https://arxiv.org/abs/1611.09268.

[46] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. Continual learning for generative retrieval over dynamic corpora. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 306–315. ACM, October 2023. doi: 10.1145/3583780.3614821. URL http://dx.doi.org/10.1145/3583780.3614821.

[47] Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. Dsi++: Updating transformer memory with new documents, 2023. URL https://arxiv.org/abs/2212.09744.

[48] Andrew Slavin Ross, Nina Chen, Elisa Zhao Hang, Elena L. Glassman, and Finale Doshi-Velez. Evaluating the interpretability of generative models by interactive reconstruction, 2021. URL https://arxiv.org/abs/2102.01264.

[49] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. Learning to rank in generative retrieval, 2023. URL https://arxiv.org/abs/2306.15222.

[50] Yongqi Li, Zhen Zhang, Wenjie Wang, Liqiang Nie, Wenjie Li, and Tat-Seng Chua. Distillation enhanced generative retrieval. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11119–11129, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.662. URL https://aclanthology.org/2024.findings-acl.662/.

[51] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books . In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Los Alamitos, CA, USA, December 2015. IEEE Computer Society. doi: 10.1109/ICCV.2015.11. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.11.

[52] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

[53] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen,

Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldst, , Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie

Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holl, , Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Z, , Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White,

Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[54] Tarek Naous, Michael J. Ryan, Alan Ritter, and Wei Xu. Having beer after prayer? measuring cultural bias in large language models, 2024. URL https://arxiv.org/abs/2305.14456.

[55] Md Abdul Aowal, Maliha T Islam, Priyanka Mary Mammen, and Sandesh Shetty. Detecting natural language biases with prompt-based learning, 2023. URL https://arxiv.org/abs/2309.05227.

[56] Roberto Navigli, Simone Conia, and Björn Ross. Biases in large language models: Origins, inventory, and discussion. *J. Data and Information*

Quality, 15(2), June 2023. ISSN 1936-1955. doi: 10.1145/3597307. URL https://doi.org/10.1145/3597307.

[57] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

[58] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. Kilt: a benchmark for knowledge intensive language tasks, 2021. URL https://arxiv.org/abs/2009.02252.

[59] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Changjiang Zhou, Maarten de Rijke, and Xueqi Cheng. On the robustness of generative information retrieval models, 2024. URL https://arxiv.org/abs/2412.18768.

List of Acronyms

DGR Distillation Enhanced Generative Retrieval (Li et al. 2024) 8

docid document identifier 3, 4, 9, 10

DR Dense Retrieval 1

GR Generative Retrieval 1, 3, 8, 10, 11, 23

GT Ground Truth 10, 13, 17

IR Information Retrieval 1

LM Language Model 1

LTRGR Learning to Rank in Generative Retrieval (Li et al. 2023) 8

MINDER Multiview Identifiers Enhanced Generative Retrieval (Li et al. 2024) 2–4, 7, 8, 10, 12–18, 20–25

MSMARCO Microsoft Machine Reading Comprehension (Bajaj et al. 2018) 12, 15

NQ Natural Questions 2, 3, 12

PAG Planning Ahead in Generative Retrieval (Zeng et.al., 2024) 3

SEAL Search Engines with Autoregressive LMs (Bevilacqua et al. 2022) 2–8, 10, 12–18, 20–25

A Annotation Issues

TODO REWRITE To identify such cases, we check whether the answer string appears in any top-k retrieved passage based on token-level overlap using `GPT2TokenizerFast`. We use $k = 10$, as discussed above. Table 9 presents the results.

Table 9: Retrieval outcomes based on answer string matching ($k = 10$).

Outcome	SEAL		MINDER	
	n	%	n	%
Answer in top-k (Hits@k)	4,961	76.1%	5,113	78.5%
Answer in unannotated passage	349	5.4%	362	5.6%
Answer string not in top-k	1,205	18.5%	1,040	16.0%

For SEAL in 5.4% of the queries, for MINDER in 5.6% of the queries, the answer string appears in (at least) one of the top-10 retrieved passages that is not the annotated ground truth. However, string matching alone does not determine whether the passage genuinely answers the question.

To understand what the 349 "answer in different passage" cases actually look like, we examine three examples in detail extracted form the SEAL outputs. Similar cases can be found in MINDER's output too.

A.1 LLM-as-a-Judge Results

According to the methodology explained in Section A, we adjust the estimate of SEAL's and MINDER's retrieval performance. Table 11 presents the adjusted hits@10 under different inclusion criteria. **MINDER DATA IS TBD. TODO**

Table 10 presents the classification results.

Table 10: LLM-as-a-judge classification of answer-in-different-passage cases.

Verdict	SEAL		MINDER	
	Count	Percentage	Count	Percentage
YES	TBD	TBD	TBD	TBD
NO	TBD	TBD	TBD	TBD
PARTIAL	TBD	TBD	TBD	TBD

The results reveal that **TBD**

Table 11: Adjusted hits@10 estimates based on LLM-as-a-judge results.

Inclusion Criterion	SEAL Hits@10	SEAL %*	MINDER Hits@10	MINDER %*
Hit@10	4961	76.1%	5113	78.4%
+ YES verdicts	TBD	TBD	TBD	TBD
+ YES and PARTIAL verdicts	TBD	TBD	TBD	TBD

*Percentage of the total 6,515 queries.

TODO ANALYSIS TBD....

Please note that we will refrain from "updating the dataset", as that would make our results incomparable with other papers' results on the standard NQ dataset.

A.2 Query Terms Corpus Coverage

We analyzed the relationship between query term availability and retrieval success using SEAL's FM-Index. For each of 6,515 Natural Questions queries, we measured the percentage of query words present in the Wikipedia corpus and correlated this with retrieval performance (Hits@1 and Hits@10). Queries were stratified into deciles based on coverage scores. The mean coverage was 85.2% with 90.5% of all queries having $\geq 70\%$ of their terms present in the corpus.

Results revealed no meaningful correlation between corpus coverage and retrieval success (Pearson's $r = 0.017$, $p = 0.165$ for Hits@1; $r = 0.001$, $p = 0.962$ for Hits@10). As shown in Table 12, queries in D1 achieved 42.1% Hits@1 and 77.1% Hits@10, while queries with complete coverage in D1 achieved 44.8% Hits@1 and 77.4% Hits@10, effectively identical performance. This null result demonstrates that SEAL's retrieval failures are not primarily driven by out-of-vocabulary terms or lexical gaps.

Table 12: Retrieval performance across corpus coverage deciles. Coverage represents the percentage of query words present in the Wikipedia corpus.

Decile	Coverage	Hits@1	Hits@10	N
D1 (lowest)	0.38–0.70	42.1%	77.1%	651
D2	0.70–0.75	37.6%	74.7%	651
D3	0.75–0.78	41.9%	76.5%	651
D4	0.78–0.88	39.2%	73.4%	651
D5	0.88–0.88	40.2%	77.1%	651
D6	0.88–0.89	43.6%	76.7%	651
D7	0.89–0.90	41.5%	75.7%	651
D8	0.90–1.00	41.6%	79.3%	651
D9	1.00–1.00	43.9%	73.6%	651
D10 (highest)	1.00–1.00	44.8%	77.4%	656

Impact of N-gram Quantity

While not a necessarily failure mode, given that SEAL's scoring mechanism additively aggregates evidence from multiple n-grams, it is essential to examine whether the quantity of matched n-grams correlates with retrieval success. Intuitively, a higher number of matched n-grams should provide stronger cumulative evidence for a passage and a passage with more matched n-grams has a mathematical advantage over a passage with fewer matched n-grams. However, excessive generation may also introduce noise if additional n-grams are redundant or non-discriminative.

Across 6,515 queries, the number of n-grams matched to the top-1 retrieved passage has a mean of 41.5 (median: 42.0, σ: 6.3), with a range of 11 to 65. Table 13 presents retrieval performance stratified by n-gram count deciles.

The relationship between n-gram count and retrieval success is notably weak (Spearman $\rho = 0.040$, $p = 0.001$), exhibiting a non-monotonic, inverted-U pattern. Hits@1 increases from 36.0% for passages with low identifier counts (D1, <34) to a peak of 46.1% in the fourth decile, subsequently plateauing and slightly declining to 41.6% for passages in the highest decile (D10, >50). This trend suggests that while a minimum amount of evidence is required to identify a passage, accumulating n-grams beyond a moderate threshold yields diminishing returns and does not significantly influence hit-rate.

Notably, the mean n-gram corpus frequency ($\bar{F}_c$) remains stable across all deciles. This stability indicates that the hit-rate plateau is not caused by the model reverting to generic "filler" tokens at higher counts. Instead,

Table 13: Hits@1 and mean n-gram corpus frequency $\bar{F}_c$ by n-gram count deciles for the top-1 retrieved passage.

Decile	**Range**	N	**Hits@1**	$\bar{F}_c$
D1	11–34	828	36.0%	384,785
D2	35–36	495	37.4%	378,346
D3	37–39	1,017	39.5%	380,174
D4	40–40	425	46.1%	392,012
D5	41–42	861	45.2%	382,931
D6	43–43	448	43.8%	382,506
D7	44–45	795	45.0%	385,779
D8	46–47	622	42.4%	382,767
D9	48–49	432	41.7%	380,999
D10	50–65	592	41.6%	371,719

the it likely stems from the scoring mechanism's inability to disambiguate between a small set of highly discriminative n-grams and the large aggregate score from multiple, less specific identifiers.

Overlapping N-gram Tokens

We initially hypothesized that cases where the model produces semantically redundant n-grams with high token overlap would constitute a failure mode by inflating scores without providing new evidence. To test this, we measure a ratio defined as the number of unique tokens divided by the total number of tokens across all generated n-grams for the top-1 retrieved query, which we will refer to as "diversity". We split up the n-grams into tokens using `GPT2TokenizerFast`.

Table 14 illustrates the results. Contrary to our initial hypothesis, the results reveal a negative correlation between diversity and hits@k. For SEAL, $\rho = -0.111$ for $k = 1$ and $\rho = -0.101$ for $k = 10$. For MINDER, the Spearman correlations are $\rho = -0.289$ and $\rho = -0.242$, respectively. All values are statistically significant at $p < 0.001$.

Table 14: Hits@k by token-based diversity ratio deciles (unique / total tokens).

	SEAL		MINDER	
Decile	**Diversity**	**H@1 / H@10**	**Diversity**	**H@1 / H@10**
D1	0.36–0.48	52.6% / 84.4%	0.28–0.56	70.9% / 91.4%
D2	0.48–0.49	46.3% / 81.4%	0.56–0.64	64.3% / 88.6%
D3	0.49–0.50	43.1% / 78.2%	0.64–0.71	53.2% / 85.6%
D5	0.51–0.52	40.4% / 73.9%	0.76–0.79	53.7% / 84.1%
D8	0.53–0.54	34.9% / 68.1%	0.86–0.88	32.2% / 73.2%
D10	0.56–0.63	32.4% / 69.8%	0.92–1.00	22.0% / 55.6%

It is interesting to note that MINDER has a considerably higher success rate at low diversity scenarios. This is expected, as MINDER's generates titles, pseudo-queries, and substrings simultaneously, meaning that the model frequently repeats key tokens across these different identifier views. The highest success rate is found in the lowest diversity decile, which suggests that the model is able to reinforce a specific semantic cluster through multiview identifiers.

These findings suggest that token repetition is not a redundancy failure, but rather a signal of model certainty. When SEAL and MINDER are highly confident in a target document, it generates multiple overlapping n-grams that concentrate score on a specific semantic cluster. In contrast, high identifier diversity indicates a lack of model confidence, where the model produces a wide variety of topically related but lexically distinct tokens that fail to converge.

Unique Articles among Passages

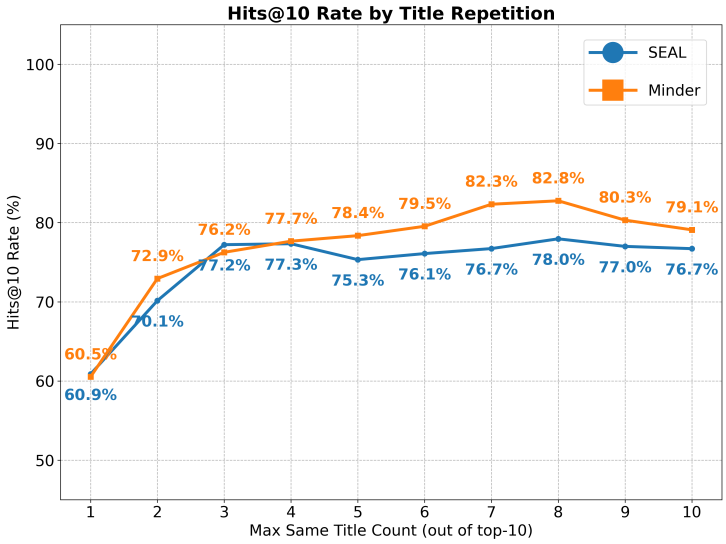
Firstly, we analyze whether there is a correlation with retrieval failure and the amoung of unique titles retrieved in the top-k passages. For this, according to the other analyses, we use $k = 10$. Thie analysis is possible, as Wikipedia articles have unique titles.

The results are illustrated in the following table and line chart:

Table 15: Hits@k by title repetition count.

Max Repetition*	SEAL N	SEAL H@1 / H@10	MINDER N	MINDER H@1 / H@10
1	23	30.4% / 60.9%	38	28.9% / 60.5%
2	556	45.3% / 70.1%	665	46.9% / 72.9%
4	653	42.1% / 77.3%	819	46.8% / 77.7%
6	573	41.5% / 76.1%	650	45.1% / 79.5%
8	567	44.3% / 78.0%	609	49.8% / 82.8%
10	1,606	38.0% / 76.7%	990	42.1% / 79.1%

*Maximum number of passages sharing the same title in the top-10 results.



We can see that when stratified by maximum title repetition, i.e. the maximum number of passages from the same Wikipedia article per query, MINDER marginally outperforms SEAL, especially on queries where the system is relatively sure of the answer, as evidenced by a specific title appearing in several of the top-10 results. We can see that in cases where the max repetition is very low, i.e. the model couldn't confidently pinpoint a single document and is likely guessing, retrieval success suffers. Based on this analysis, there are significantly more cases with high title repetition than low title repetition, indicating that the models are fairly certain about the correct document (or, alternatively, are confidently incorrect about one).

A.3 Preliminary Analysis

We analyze the distribution of generated n-gram keys across retrieved documents. We categorize each query based on Precision@2: PP (both correct), PN (rank-1 correct, rank-2 incorrect), NP (rank-1 incorrect, rank-2 correct), and NN (both incorrect). Table 16 presents the distribution of queries across retrieval outcome categories. In our definition, positive means the retrieved passage corresponds to a ground truth passage, therefore "correct", while negative accordingly means "incorrect".

Table 16: Distribution of queries across retrieval outcome categories ($N = 6,515$).

	SEAL		MINDER	
Category	**Count**	**Percentage**	**Count**	**Percentage**
PP	1,287	19.8%	1,345	20.6%
PN	1,427	21.9%	1,685	25.9%
NP	702	10.8%	729	11.2%
NN	3,099	47.6%	2,756	42.3%

For both architectures analyzed, nearly half of all queries result in NN cases. Combined with the NP cases, this indicates that for SEAL 58.4% of queries, for MINDER 53.5% fail to retrieve a relevant passage at rank 1. Conversely, 41.7% and 46.5% of queries achieve successful Precision@1 retrieval (PP + PN), respectively.

We also categorize keys into three sets: keys appearing exclusively in positive (ground-truth) passages, keys appearing exclusively in negative passages, and keys shared by both. Table 17 presents the aggregate statistics. μ % Keys Unique to P defines the average percentage of keys that are unique to positive passages only across the top-k retrieved queries, μ % Keys Unique to N defined accordingly. μ % Keys Shared ($P \cap N$) denotes the average percentage of keys that are shared between both negative and positive passages across the top-k retrieved queries. μ Score accordingly refers to the average n-gram score across queries for each categories.

Although the results between the two systems are fairly similar, with MINDER performing marginally better, it is noticable that both SEAL and MINDER generates substantially more keys matching negative passages than positive passages. Second, keys unique to negative passages receive marginally higher average scores than keys unique to positive passages, with keys shared having he highest average score. This pattern indicates that both systems assign higher scores to n-grams that are not discriminative enough,

Table 17: Aggregate n-gram key statistics across all queries ($N = 6,515, K = 10$).

Metric	**SEAL**	**MINDER**
μ % Keys Unique to P	11.6%	12.3%
μ % Keys Unique to N	78.2%	75.3%
μ % Keys Shared ($P \cap N$)	10.2%	12.5%
μ Score (Unique P)	5.67 ($\pm$6.69)	6.12 ($\pm$7.01)
μ Score (Unique N)	6.66 ($\pm$2.90)	6.94 ($\pm$2.89)
μ Score (Shared)	17.28 ($\pm$12.20)	17.34 ($\pm$11.49)

to either way.

The high variance in scores for unique positive keys compared to unique negative keys suggests that the models' ability to identify relevant n-grams is highly inconsistent. While some "Unique P" keys receive very high scores, many others likeyly receive negligible scores, dragging down the mean. Furthermore, the high variance in the 'Shared' category indicates a lack of discriminative power and their influence on the final ranking is unpredictable.