Reasoning over Public and Private Data in Retrieval-Based Systems

Simran Arora[‡], **Patrick Lewis**[†], **Angela Fan**[†], **Jacob Kahn**[†*], and **Christopher Ré**[‡*]

[‡]Stanford University
{simran, chrismre}@cs.stanford.edu

[†]Facebook AI Research
{plewis, angelafan, jacobkahn}@fb.com

Abstract

Users and organizations are generating ever-increasing amounts of private data from a wide range of sources. Incorporating private context is important to personalize open-domain tasks such as question-answering, fact-checking, and personal assistants. State-of-the-art systems for these tasks explicitly retrieve information that is relevant to an input question from a background corpus before producing an answer. While today's retrieval systems assume relevant corpora are fully (e.g., publicly) accessible, users are often unable or unwilling to expose their private data to entities hosting public data. We define the SPLIT ITERATIVE RETRIEVAL (SPIRAL) problem involving iterative retrieval over multiple privacy scopes. We introduce a foundational benchmark with which to study SPIRAL, as no existing benchmark includes data from a private distribution. Our dataset, CONCURRENTQA, includes data from distinct public and private distributions and is the first textual QA benchmark requiring concurrent retrieval over multiple distributions. Finally, we show that existing retrieval approaches face significant performance degradations when applied to our proposed retrieval setting and investigate approaches with which these tradeoffs can be mitigated. We release the new benchmark and code to reproduce the results.[1]

1 Introduction

The world's information is split between publicly and privately accessible scopes and the ability to simultaneously reason over both scopes is useful to support personalized tasks. However, retrieval-based machine learning (ML) systems, which first retrieve relevant information to a user input from background knowledge sources before providing an output, do not consider retrieving from private data that organizations and individuals aggregate locally. Neural retrieval systems are achieving impressive performance across applications such as language-modeling (Borgeaud et al., 2022), question answering (Chen et al., 2017), and dialogue (Dinan et al., 2019), and we focus on the under-explored question of how to personalize these systems while preserving privacy.

Consider the following examples that require retrieving information from both public and private scopes. Individuals could ask *"With my GPA and SAT score, which universities should I apply to?"* or *"Is my blood pressure in the normal range for someone 55+?"*. In an organization, an ML engineer could ask: *"How do I fine-tune a language model, based on public StackOverflow and our internal company documentation?"*, or a doctor could ask *"How are COVID-19 vaccinations affecting patients with type-1 diabetes based on our private hospital records and public PubMed reports?"*. To answer such questions, users manually cross-reference public and private information sources. We initiate the study of a retrieval setting that enables using public (global) data to enhance our understanding of private (local) data.

Modern retrieval systems typically collect documents that are most similar to a user's question from a massive corpus, and provide the resulting documents to a separate model, which reasons over the information to output an answer (Chen et al., 2017). Multi-hop reasoning (Welbl et al., 2018) can be used to answer complex queries over information distributed across multiple documents, e.g. news articles and Wikipedia. For such queries, we observe that using multiple rounds of retrieval (i.e., combining the original query with retrieved documents at round i for use in retrieval at round $i+1$) provides over 75% improvement in

[*] Equal contribution.

[1] https://github.com/facebookresearch/concurrentqa

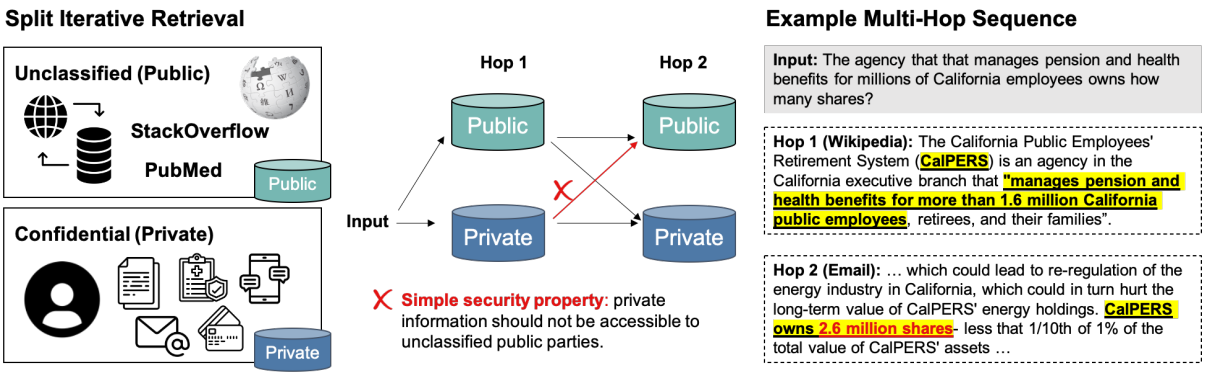

Figure 1: Multi-hop retrieval systems use beam search to collect information from a massive corpus: retrieval in hop_{i+1} is conditioned on the top documents retrieved in hop_i. The setting of retrieving from corpora distributed across multiple privacy scopes is unexplored. Here, the content of a private document retrieved in hop_i is revealed to the entity hosting public data if used to retrieve public documents in hop_{i+1}.

quality versus using one round of retrieval (Section 5). Iterative retrieval is now common in retrieval (Miller et al., 2016; Feldman and El-Yaniv, 2019; Asai et al., 2020; Xiong et al., 2021; Qi et al., 2021; Khattab et al., 2021, inter alia.).

Existing multi-hop systems perform retrieval over a single privacy scope. However, users and organizations often cannot expose data to public entities. Maintaining terabyte-scale and dynamic data is difficult for many private entities, warranting retrieval from *multiple* distributed corpora.

To understand why distributed multi-hop retrieval implicates privacy concerns, consider two illustrative questions an employee may ask. First, to answer "*Of the products our competitors released this month, which are similar to our unreleased upcoming products?*", an existing multi-hop system likely (1) retrieves public documents (e.g., news articles) about competitors, and (2) uses these to find private documents (e.g., company emails) about internal products, leaking no private information. Meanwhile, "*Have any companies ever released similar products to the one we are designing?*" entails (1) retrieving private documents detailing the upcoming product, and (2) performing similarity search for public products *using information from the confidential documents*. The latter reveals private data to an untrusted entity hosting a public corpus. An effective privacy model will minimize leakage.

We introduce the SPLIT ITERATIVE RETRIEVAL (SPIRAL) problem. Public and private document distributions usually differ and our first observation is that *all* existing textual benchmarks require retrieving from one data-distribution. To appropriately evaluate SPIRAL, we create the first textual multi-distribution benchmark, CONCURRENTQA, which spans Wikipedia in the public domain and emails in the private domain, enabling the study of two novel real-world retrieval setups: (1) multi-distribution and (2) privacy-preserving retrieval:

- **Multi-distribution retrieval** The ability for a model to effectively retrieve over multiple distributions, even in the absence of privacy constraints, is a precursor to effective SPIRAL systems since it is unlikely for all private distributions to be reflected at train time. However, the typical retrieval setup requires retrieving over a single document distribution with a single query distribution (Thakur et al., 2021). We initiate the study of the real-world multi-distribution setting. We find that the SoTA multi-hop QA model trained on 90.4k Wikipedia data *underperforms* the same model trained on the 15.2k CONCURRENTQA (Wikipedia and Email) examples by 20.8 F1 points on questions based on Email passages. Further, we find the performance of the model trained on Wikipedia improves by 4.3% if we retrieve the top $\frac{k}{2}$ passages from each distribution vs. retrieving the overall top k passages, which is the standard protocol.

- **Privacy-preserving retrieval** We then propose a framework for reasoning about the privacy tradeoffs required for SoTA models to achieve as good performance on public-private QA as is achieved in public-QA. We evaluate performance when *no* private information is revealed, and models trained only on public data (e.g.

Wikipedia) are utilized. Under this privacy standard, models sacrifice upwards of 19% performance under SPIRAL constraints to protect document privacy and 57% to protect query privacy when compared to a baseline system with standard, non privacy-aware retrieval mechanics. We then study how to manage the privacy-performance tradeoff using selective prediction, a popular approach for improving the reliability of QA systems (Kamath et al., 2020; Lewis et al., 2021; Varshney et al., 2022).

In summary: (1) We are the first to report on problems with applying existing neural retrieval systems to the public and private retrieval setting, (2) We create CONCURRENTQA, the first textual multi-distribution benchmark to study the problems, and (3) We provide extensive evaluations of existing retrieval approaches under the proposed real-world retrieval settings. We hope this work encourages further research on private retrieval.

2 Background & Related Work

2.1 Retrieval-Based Systems

Open-domain applications, such as question answering and personal assistants, must support user inputs across a broad range of topics. *Implicit-memory* approaches for these tasks focus on memorizing the knowledge required to answer questions within model parameters (Roberts et al., 2020). Instead of memorizing massive amounts of knowledge in model parameters, *retrieval-based systems* introduce a step to retrieve information that is relevant to a user input from a massive corpus of documents (e.g., Wikipedia), and then provide this to a separate task model that produces the output. Retrieval-free approaches have not been shown to work convincingly in multi-hop settings (Xiong et al., 2021).

2.2 Multi-hop Retrieval

We focus on open-domain QA (ODQA), a classic application for retrieval-based systems. ODQA entails providing an answer a to a question q, expressed in natural language and without explicitly provided context from which to find the answer (Voorhees, 1999). A *retriever* collects relevant documents to the question from a corpus, then a *reader* model extracts an answer from selected documents.

Our setting is concerned with complex queries where supporting evidence for the answer is distributed across multiple (public and private) documents, termed multi-hop reasoning (Welbl et al., 2018). To collect the distributed evidence, systems use multiple *hops* of retrieval: representations of the top passages retrieved in $\mathrm{hop_i}$ are used to retrieve passages in $\mathrm{hop_{i+1}}$ (Miller et al., 2016; Feldman and El-Yaniv, 2019; Asai et al., 2020; Wolfson et al., 2020; Xiong et al., 2021; Qi et al., 2021; Khattab et al., 2021). [2] Finally, we discuss the applicability of existing multi-hop benchmarks to our problem setting in Section 4.

2.3 Privacy Preserving Retrieval

Information retrieval is a long standing topic spanning the machine learning, databases, and privacy communities. We discuss the prior work and considerations for our setup along three axes: **(1) Levels of privacy**. Prior private retrieval system designs guarantee privacy for different components across both *query* and *document* privacy. *Our setting requires both query and document privacy.* **(2) Relative isolation of document storage and retrieval computation**. The degree to which prior retrieval and database systems store or send private data to centralized machines (with or without encryption) varies. *Our work structures dataflow to eliminate processing of private documents on public retrieval infrastructure.* **(3) Updatability and latency**. Works make different assumptions about how a user will interact with the system. These include (1) tolerance of high-latency responses and (2) whether corpora are static or changing. *Our setting focuses on open-domain questions for interactive applications with massive, temporally changing corpora and requiring low-latency.*

Isolated systems with document and query privacy but poor updatability. To provide the *strongest possible privacy* guarantee, i.e. no information about the user questions or passages is revealed, prior work considers when purely local search is possible (Cao et al., 2019), i.e. search performed on systems controlled exclusively by the user. This guarantee provides no threat opportunities assuming that both data (documents and queries) and computation occur on controlled infrastructure. Scaling the amount of locally hosted data and updating local corpora with quickly changing public data is challenging; we build a system that might meet such demands.

[2] Note that beyond multi-hop QA, retrieval augmented language models and dialogue systems also involve iterative retrieval (Guu et al., 2020).

Public, updatable database systems providing query privacy. A distinct line of work explores how to securely perform retrieval such that the user query is not revealed to a public entity that hosts and updates databases. Private information retrieval (PIR) (Chor et al., 1998) in the cryptography community refers to a setup where users know the entry in a remote database that they want to retrieve (Kogan, 2020). The threat model is directly related to the cryptographic scheme used to protect queries and retrieval computation. Here, the document containing the answer is assumed to be known; leaking the particular corpus containing the answer may implicitly leak information about the query. In contrast, we focus on *open-domain applications*, where users ask about any topic imaginable and do not know which corpus item holds the answer. Our setting also considers document privacy, as discussed in Section 6.

Public, updatable but high-latency secure nearest neighbor search with document and query privacy. The next relevant line of work focuses on secure nearest neighbor search (NNS) (Murugesan et al., 2010; Chen et al., 2020a; Schoppmann et al., 2020; Servan-Schreiber, 2021), where the objective is to securely (1) compute similarity scores between the query and passages, and (2) select the top-k scores. The speed of cryptographic tools (secure multi-party computation, secret sharing) that are used to perform these steps increase as the sparsity of the query and passage representations increases. Performing the secure protocol over dense embeddings can take *hours per query* (Schoppmann et al., 2020). As before, threats in this setting are related to vulnerabilities in cryptographic schemes or in actors gaining access to private document indices if not directly encrypted. Prior work relaxes privacy guarantees and computes approximate NNS; speeds, however, are still several seconds per query (Schoppmann et al., 2020; Chen et al., 2020a). This is prohibitive for iterative open domain retrieval applications.

Partial query privacy via fake query augmentation for high-latency retrieval from public databases. Another class of privacy techniques for hiding the user's intentions is query-obfuscation or k-anonymity. The user's query is combined with fake queries or queries from other users to increase the difficulty of linking a particular query to the user's true intentions (Gervais et al., 2014). This multiplies communication costs since nearest neighbors must be retrieved for each of the k queries; iterative retrieval worsens this cost penalty. Further, the private query is revealed among the full set of k; the threat of identifying the user's true query remains (Haeberlen et al., 2011).

Finally, we note that our primary focus is on *inference-time* privacy concerns and note that during training time, federated learning (FL) with differential privacy (DP) is a popular strategy for training models without exposing training data (McMahan et al., 2016; Dwork et al., 2006).

Overall, despite significant interest in IR, there is limited attention towards characterizing the privacy risks as previously observed (Si and Yang, 2014). Our setting, which focuses on supporting open-domain applications with modern dense retrievers is not well-studied. Further, the prior works do not characterize the privacy concerns associated with *iterative* retrieval. Studying this setting is increasingly important with the prevalence of API-hosted large language models and services. For instance, users may want to incorporate private knowledge into systems that make multiple calls to OpenAI model endpoints (Brown et al., 2020; Khattab et al., 2022). Code assistants, which may be extended to interact with both private repositories and public resources like Stack Overflow, are also seeing widespread use (Chen et al., 2021).

3 Problem Definition

Objective Given a multi-hop input q, a set of private documents $p \in D_P$, and public documents $d \in D_G$, the objective is to provide the user with the correct answer a, which is contained in the documents. Figure 1 (Right) provides an example. Overall, the SPLIT ITERATIVE RETRIEVAL (SPIRAL) problem entails maximizing quality, while protecting query and document privacy.

Standard, Non-Privacy Aware QA Standard non-private multi-hop ODQA involves answering q with the help of passages $d \in D_G$, using beam search. In the first iteration of retrieval, the k passages from the corpus, $d_1, ..., d_k$, that are most relevant to q are retrieved. The text of a retrieved passage is combined with q using function f (e.g., concatenating the query and passages sequences) to produce $q_i = f(q, d_i)$, for $i \in [1..k]$. Each q_i (which contains d_i) is used to retrieve k more passages in the following iteration.

We now introduce the SPIRAL retrieval problem. The user inputs to the QA system are the private corpus D_P and questions q. There are two key properties of the problem setting.

Property 1: Data is likely stored in multiple enclaves and personal documents $p \in \mathbf{D_P}$ can not leave the user's enclave. Users and organizations own private data, and untrustworthy (e.g., cloud) services own public data. First, we assume users likely do not want to publicly expose their data to create a single public corpus nor blindly write personal data to a public location. Next, we also assume it is challenging to store global data locally in many cases. This is because not only are there terabytes of public data and user-searches follow a long tail (Bernstein et al., 2012) (i.e. it is challenging to anticipate all a user's information needs), but public data is also constantly being updated (Zhang and Choi, 2021). Thus, D_P and D_G are hosted as separate corpora.

Now given q, the system must perform one retrieval over D_G and one over D_P rank the results such that the top-k passages will include k_P private and k_G public passages, and use these for the following iteration of retrieval. If the retrieval-system stops after a **single-hop**, there is no document privacy risk since no $p \in D_P$ is publicly exposed and no query privacy risk if the system used to retrieve from D_P is private, as is assumed. However for **multi-hop** questions, if $k_P > 0$ for an initial round of retrieval, meaning there exists some $p_i \in D_P$ which was in the top-k passages, it would sacrifice privacy if $f(q, p_i)$ were to be used to perform the next round of retrieval from D_G. Thus, for the strongest privacy guarantee, public retrievals should precede private document retrievals. For less privacy-sensitive use cases, this strict ordering can be weakened.

Property 2: Inputs that entirely rely on private information should not be revealed publicly. Given the multiple indices, D_P and D_G, q may be entirely answerable using multiple hops over the D_P index, in which case, q would never need to leave the user's device. For example, the query from an employee standpoint, *Does the search team use any infrastructure tools that our personal assistant team does not use?*, is fully answerable with company information. Prior work demonstrates that queries are very revealing of user interests, intents, and backgrounds (Xu et al., 2007; Gervais et al., 2014). There is an observable difference in the search behavior of users with privacy concerns (Zimmerman et al., 2019) and an effective system will protect queries.

4 CONCURRENTQA Benchmark

Here we develop a testbed for studying public-private retrieval. We require questions spanning two corpora, D_P and D_G. First, we consider using existing benchmarks and describe the limitations we encounter, motivating the creation of our new benchmark, CONCURRENTQA. Then we describe the dataset collection process and its contents.

4.1 Adapting Existing Benchmarks

We first adapt the widely used benchmark, HotpotQA (Yang et al., 2018), to study our problem. HotpotQA contains multi-hop questions, which are each answered using two Wikipedia passages. We create HotpotQA-SPIRAL by splitting the Wikipedia corpus into D_G and D_P. This results in questions entirely reliant on $p \in D_P$, entirely on $d \in D_G$, or reliant on a mix of one private and one public document, allowing us to evaluate performance under SPIRAL constraints.

Ultimately however, D_P and D_G come from a single Wikipedia distribution in HotpotQA-SPIRAL. Private and public data will often reflect different linguistic styles, structures, and topics. We observe all existing textual multi-hop benchmarks require retrieving from a single distribution. We cannot combine two existing benchmarks over two corpora because this will not yield questions that rely on both corpora simultaneously. To evaluate with a more realistic setup, we create a new benchmark: CONCURRENTQA. We quantitatively demonstrate the limitations of using HotpotQA-SPIRAL in the experiments and analysis.

4.2 CONCURRENTQA Overview

We create and release a new multi-hop QA dataset, CONCURRENTQA, which is designed to more closely resemble a practical use case for SPIRAL. CONCURRENTQA contains questions spanning Wikipedia documents as D_G and Enron employee emails (Klimt and Yang, 2004) as D_P. [3] We propose two unique evaluation settings for CONCURRENTQA: performance (1) conditioned on the sub-domains in which the question evidence

[3]The Enron Corpus includes emails written by 158 employees of Enron Corporation and are in the public domain.

Question	Hop 1 and Hop 2 Gold Passages
What was the estimated 2016 population of the city that generates power at the Hetch Hetchy hydroelectric dams?	*Hop 1* An email mentions that San Francisco generates power at the Hetch Hetchy dams. *Hop 2* The Wikipedia passage about San Francisco reports the 2016 census-estimated population.
Which firm invested in both the 5th round of funding for Extraprise and first round of funding for JobsOnline.com?	*Hop 1* An email lists 5th round Exraprise investors. *Hop 2* An email lists round-1 investors for JobsOnline.com.

Table 1: Example CONCURRENTQA queries based on Wikipedia passages (D_G) and emails (D_P).

can be found (Section 5), and (2) conditioned on the degree of privacy protection (Section 6).

The corpora contain 47k emails (D_P) and 5.2M Wikipedia passages (D_G), and the benchmark contains 18,439 examples (Table 2). Questions require three main reasoning patterns: (1) *bridge questions* require identifying an entity or fact in Hop_1 on which the second retrieval is dependent, (2) *attribute questions* require identifying the entity that satisfies all attributes in the question, where attributes are distributed across passages, and (3) *comparison questions* require comparing two similar entities, each appearing in a separate passage. We estimate the benchmark is 80% bridge, 12% attribute, and 8% comparison questions. We focus on factoid QA.

Benchmark Design Each benchmark example includes the *question* that requires reasoning over multiple documents, *answer* which is a span of text from the supporting documents, and the specific *supporting sentences* in the documents which are used to arrive at the answer and can serve as supervision signals.

As discussed in Yang et al. (2018), collecting a high quality multi-hop QA dataset is challenging because it is important to provide *reasonable* pairs of supporting context documents to the worker — not all article pairs are conducive to a good multi-hop question. There are four types of pairs we need to collect for the Hop_1 and Hop_2 passages: Private and Private, Private and Public, Public and Private, and Public and Public. We use the insight that we can obtain meaningful passage-pairs by showing workers passages that mention similar or overlapping entities. All crowdworker assignments contain unique passage pairs. A detailed description of how the passage pairs are produced is in Appendix C and we release all our code for creating the passage pairs.

Split	Total	EE	EW	WE	WW
Train	15,239	3762	4002	3431	4044
Dev	1,600	400	400	400	400
Test	1,600	400	400	400	400

Table 2: Size statistics. The evaluation splits are balanced between questions with gold passages as emails (E) vs. Wikipedia (W) passages for Hop_1 and Hop_2.

Benchmark Collection We used Amazon Turk for collection. The question generation stage began with an onboarding process in which we provided training videos, documents with examples and explanations, and a multiple-choice exam. Workers completing the onboarding phase were given access to pilot assignments, which we manually reviewed to identify individuals with high quality submissions. We worked with these individuals to collect the full dataset. We manually reviewed over 2.5k queries in the quality-check process and prioritized including the manually-verified examples in the final evaluation splits.

In the manual review, examples of the criteria that led us to discard queries included: the query could be answered using one passage alone, had multiple plausible answers either in or out of the shown passages, or lacked clarity. During the manual review, we developed a multiple-choice questionnaire to streamline the checks along the identified criteria. We then used this to launch a second Turk task to validate the generated queries that we did not manually review. Assembling the cohort of crowdworkers for the validation task again involved onboarding and pilot steps, in which we manually reviewed performance. We shortlisted $\sim$20 crowdworkers with high quality submissions who collectively validated examples appearing in the final benchmark.

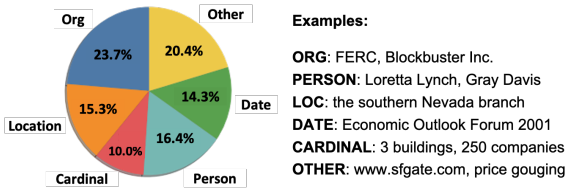

Figure 2: NER types for CONCURRENTQA answers.

4.3 Benchmark Analysis

Emails and Wiki passages differ in several ways. **Format:** Wiki passages for entities of the same type tend to be similarly structured, while emails introduce many formats — for example, certain emails contain portions of forwarded emails, lists of articles, or spam advertisements. **Noise:** Wiki passages tend to be typo-free, while the emails contain several typos, URLs, and inconsistent capitalization. **Entity Distributions:** Wiki passages tend to focus on details about one entity, while a single email can cover multiple (possibly unrelated) topics. Information about email entities is also often distributed across passages, whereas public-entity information tends to be localized to one Wiki passage. We observe that a private entity occurs 9× on average in gold training data passages while a public entity appears 4× on average. There are 22.6k unique private entities in the gold training data passages, and 12.8k unique public entities. **Passage Length:** Finally, emails are 3× longer than Wiki passages on average.[4]

Answer Types CONCURRENTQA is a factoid QA task so answers tend to be short spans of text containing nouns, or entity names and properties. Figure 2 shows the distribution NER tags across answers and examples from each category.

Limitations As in HotpotQA, workers see the gold supporting passages when writing questions, which can result in lexical overlap between the questions and passages. We mitigate these effects through validation task filtering and by limiting the allowed lexical overlap via the Turk interface. Next, our questions are not organic user searches, however existing search and dialogue logs do not contain questions over public and private data to our knowledge. Finally, Enron was a major public corporation; data encountered during pretraining could impact the distinction between public and private data. We investigate this in Section 5.

Ethics Statement The Enron Dataset is already widely-used in NLP research (Heller, 2017). That said, we acknowledge the origin of this data as collected and made public by the U.S. FERC during their investigation of Enron. We note that many of the individuals whose emails appear in the dataset were not involved in wrongdoing. We defer to using inboxes that are frequently used in prior work.

In the next sections, we evaluate CONCURRENTQA in the SPIRAL setting. We first ask how a range of SoTA retrievers perform in the multi-domain retrieval setting in Section 5, then introduce baselines for CONCURRENTQA under a strong privacy guarantee in which *no* private information is revealed whatsoever in Section 6.

5 Evaluating Mixed-Domain Retrieval

Here we study the SoTA multi-hop model performance on CONCURRENTQA in the novel multi-distribution setting. The ability for models trained on public data to generalize to private distributions, with little or no labeled data, is a precursor to solutions for SPIRAL. In the commonly studied zero-shot retrieval setting (Guoa et al., 2021; Thakur et al., 2021), the top k of k passages will be from a single distribution, however users often have diverse questions and documents.

We first evaluate multi-hop retrievers. Then we apply strong single-hop retrievers to the setting, to understand the degree to which iterative retrieval is required in CONCURRENTQA.

5.1 Benchmarking Multi-Hop Retrievers

Retrievers We evaluate the multi-hop dense retrieval model (MDR) (Xiong et al., 2021), which achieves SoTA on multi-hop QA and multi-hop implementation of BM25, a classical bag-of-words method, as prior work indicates its strength in OOD retrieval (Thakur et al., 2021).

MDR is a bi-encoder model consisting of a query encoder and passage encoder. Passage embeddings are stored in an index designed for efficient retrieval (Johnson et al., 2017). In Hop_1, the embedding for query q is used to retrieve the k passages $d_1, ..., d_k$ with the highest *retrieval score* by the maximum inner product between question and passage encodings. For multi-hop MDR, those retrieved passages are each appended to q and encoded, and each of the k resulting embeddings are

[4] Since information density is generally lower in emails vs. Wiki passages, this helps crowdworkers generate meaningful questions. Lengths chosen within model context window.

Retrieval Method	Overall		Domain-Conditioned			
	EM	*F1*	*EE*	*EW*	*WE*	*WW*
ConcurrentQA-MDR	**48.9**	**56.5**	**49.5**	**66.4**	**41.8**	68.3
HotpotQA-MDR	45.0	53.0	28.7	61.7	41.1	**81.3**
Subsampled HotpotQA-MDR	37.2	43.9	23.8	51.1	28.6	72.1
BM25	33.2	40.8	44.2	30.7	50.2	30.5
Oracle	74.1	83.4	66.5	87.5	89.4	90.4

Table 3: ConcurrentQA results using four retrieval approaches, and oracle retrieval. On the right, we show performance (F1 scores) by the domains of the Hop_1 and Hop_2 gold passages for each question (email is "E", Wikipedia is "W", and "EW" indicates the gold passages are email for Hop_1 and Wikipedia for Hop_2).

used to collect k more passages in Hop_2, yielding k^2 passages. The top-k of the passages after the final hop are inputs to the reader, ELECTRA-Large (Clark et al., 2020). The reader selects a candidate answer in each passage.[5] The candidate with the highest *reader score* is outputted.

Baselines We evaluate using four retrieval baselines: **(1) ConcurrentQA-MDR**, a dense retriever trained on the ConcurrentQA train set (15.2k examples), to understand the value of in-domain training data for the task; **(2) HotpotQA-MDR**, trained on HotpotQA (90.4K examples), to understand how well a publicly trained model performs on the multi-distribution benchmark; **(3) Subsampled HotpotQA-MDR**, trained on subsampled HotpotQA data of the same size as the ConcurrentQA train set, to investigate the effect of dataset size; and **(4) BM25** sparse retrieval. Results are in Table 3. Experimental details are in Appendix A.[6]

Training Data Size *Strong dense retrieval performance requires a large amount of training data.* Comparing ConcurrentQA-MDR and Subsampled HotpotQA-MDR, the former outperforms by 12.6 F1 points as it is evaluated in-domain. However, the HotpotQA-MDR baseline, trained on the full HotpotQA training set, performs nearly equal to ConcurrentQA-MDR. Figure 3 shows the performance as training dataset size varies. Next we observe the sparse method

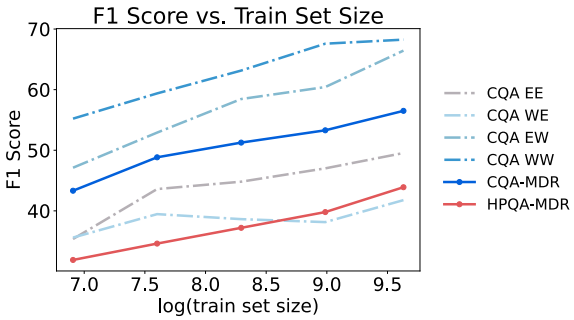

Figure 3: F1 score vs training data size, training MDR on subsampled HotpotQA (HPQA) and subsampled ConcurrentQA (CQA) training data. We also show trends by the question domain for CQA (dotted lines).

matches the zero-shot performance of the Subsampled HotpotQA model on ConcurrentQA. For larger dataset sizes (HotpotQA-MDR) and in-domain training data (ConcurrentQA-MDR), dense outperforms sparse retrieval. Notably, it may be difficult to obtain training data for all private or temporally arising distributions.

Domain Specific Performance *Each retriever excels in a different subdomain of the benchmark.* Table 3 shows the retrieval performance of each method based on whether the gold supporting passages for Hop_1 and Hop_2 are email (E) or Wikipedia (W) passages (EW is Email-Wiki for Hop_1-Hop_2). HotpotQA-MDR performance on WW questions is far better than on questions involving emails. The sparse retriever performs worse than the dense models on questions involving W, but better on questions with E in Hop_2. When training on ConcurrentQA, performance on questions involving E improves significantly, but remains low on W-based questions. Finally, we explicitly provide the gold supporting passages to the reader model (Oracle). EE oracle performance also remains low, indicating room to improve the reader.

[5] Xiong et al. (2021) compares ELECTRA and other readers such as FiD (Izacard and Grave1, 2021), finding similar performance. We follow their approach and use ELECTRA.

[6] We check for dataset leakage stemming from the "public" models potentially viewing "private" email information in pretraining. Using the MDR and ELECTRA models fine-tuned on HotpotQA, we evaluate on ConcurrentQA using a corpus of only Wiki passages. Test scores are 72.0 and 3.3 EM for questions based on two Wiki and two email passages respectively, suggesting explicit access to emails is important.

Method	Recall@10
Two-hop MDR	77.5
One-hop MDR	45.7
Contriever	52.7
Contriever MS-MARCO	64.3

Table 4: Comparing the retrieval quality using one-hop MDR, Contriever, and Contriever fine-tuned on MS-MARCO to the quality of two-hop MDR. Results are over the HotpotQA dataset.

How well does the retriever trained on public data perform in the SPIRAL setting? We observe the HotpotQA-MDR model is biased towards retrieving Wikipedia passages. On examples where the gold Hop_1 passage is an email, 15% of the time, no emails appear in the top-k Hop_1 results; meanwhile, this only occurs 4% of the time when Hop_1 is Wikipedia. On the slice of EE examples, 64% of Hop_2 passages are E, while on the slice of WW examples, 99.9% of Hop_2 passages are W. If we simply *force* equal retrieval ($\frac{k}{2}$) from each domain on each hop, we observe 2.3 F1 points (4.3%) improvement in CONCURRENTQA performance, compared to retrieving the overall top-k. *Optimally* selecting the allocation for each domain is an exciting question for future work.

Performance on WE questions is notably worse than on EW questions. We hypothesize this is because several emails discuss each Wikipedia-entity, which may increase the noise in Hop_2 (i.e., WE is a one-to-many hop, while for EW, W typically contains one valid entity-specific passage). The latter is intuitively because individuals refer to a narrow set of public entities in private discourse.

5.2 Benchmarking Single-Hop Retrieval

In Section 3, we identify that iterative retrieval implicates document privacy. Therefore, an important preliminary question is to what degree multiple hops are actually required? We investigate this question using both HotpotQA and CONCURRENTQA. We evaluate MDR using just the first-hop results and Contriever (Izacard et al., 2021), the SoTA single-hop dense retrieval model.

Results In Table 4, we summarize the retrieval results from using three off-the-shelf models for HotpotQA: (1) the HotpotQA MDR model for one-hop, (2) the pretrained Contriever model, and (3) the MS-MARCO (Nguyen et al., 2016) fine-tuned variant of Contriever. We observe a sizeable gap between the one and two hop baselines. Strong single-hop models trained over more diverse publicly available data may help address the SPIRAL problem as demonstrated by Contriever fine-tuned on MS-MARCO.

However, when evaluating the one-hop baselines on CONCURRENTQA, we find Contriever underperforms the two-hop baseline more significantly as shown in Appendix Table 8. This is consistent with prior work that finds Contriever quality degrades on tasks that increasingly differ from the pretraining distribution (Zhan et al., 2022). By sub-domain, Contriever MS-MARCO returns the gold first-hop passage for 85% of questions where both gold passages are from Wikipedia, but for less than 39% of questions when at least one gold passage (Hop_1 and/or Hop_2) is an email. By hop, we find Contriever MS-MARCO retrieves the first-hop passage 49% of the time and second-hop passage 25% of the time.

Finally, to explore whether a stronger single-hop retriever may further improve the one-hop baseline, we continually fine-tune Contriever on CONCURRENTQA. We follow the training protocol and use the code released in Izacard et al. (2021), and include these details in Appendix A. The fine-tuned model achieves 39.7 Recall@10 and 63.6 Recall@100, while two-hop MDR achieves 55.9 Recall@10 and 73.8 Recall@100 (Table 9 in the Appendix). We observe Contriever's one-hop Recall@100 of 63.6 exceeds the two-hop MDR Recall@10 of 55.9, suggesting a tradeoff space between the number of passages retrieved per hop (which is correlated with cost) and the ability to circumvent iterative retrieval (which we identify implicates privacy concerns).

6 Evaluation under Privacy Constraints

This section provides baselines for CONCURRENTQA under privacy constraints. We concretely study a baseline in which no private information is revealed publicly whatsoever. We believe this is an informative baseline for two reasons:

1. The privacy setting we study is often categorized as an access-control framework — different parties have different degrees of access to different degrees of privileged information. While this setting is quite restrictive, this privacy framework is widely used in practice for instance in the government and medical fields (Bell and LaPadula, 1976; Hu et al., 2006).

Privacy Level	Sample Questions Answered under Each Privacy Level
Answered with **No Privacy**, but *not* under Document Privacy	*Q1* In which region is the **site of a meeting** between Dabhol manager Wade Cline and Ministry of Power Secretary A. K. Basu located? *Q2* What year was the state-owned regulation **board that was in conflict** with Dabhol Power over the DPC project formed?
Answered with **Document Privacy**	*Q1* The U.S. Representative from New York who served from 1983 to 2013 requested a summary of what **order concerning a price cap complaint**? *Q2* **How much of the company** known as DirecTV Group does GM own?
Answered with **Query Privacy**	*Q1* Which CarrierPoint backer has a partner on SupplySolution's board? *Q2* At the end of what year did Enron India's managing director responsible for managing operations for Dabhol Power believe it would go online? ***All evidence is in private emails and not in Wikipedia.**

Table 5: Examples of queries answered under different privacy restrictions. **Bold** indicates private information.

Model	HotpotQA-SPIRAL		ConcurrentQA	
	EM	*F1*	*EM*	*F1*
No Privacy Baseline	62.3	75.3	45.0	53.0
No Privacy Multi-Index	62.3	75.3	45.0	53.0
Document Privacy	56.8	68.8	36.1	43.0
Query Privacy	34.3	43.3	19.1	23.8

Table 6: Multi-hop QA datasets using the dense retrieval baseline (MDR) under each privacy setting.

2. There are many possible privacy constraints as users find different types of information to be sensitive (Xu et al., 2007). Studying these is an exciting direction that we hope is facilitated by this work. Because the appropriate privacy relaxations are subjective, we focus on characterizing the upper (Section 5) and lower-bounds (Section 6) of retrieval quality in our proposed setting.

Setup We use models trained on Wikipedia data, to evaluate performance under privacy restrictions both in the in-distribution multi-hop HotpotQA-SPIRAL (an adaptation of the HotpotQA benchmark to the SPIRAL setting (Yang et al., 2018)) and multi-distribution ConcurrentQA settings. Motivating the latter setup, sufficient training data is seldom available for all private distributions. We use the multi-hop SoTA model, MDR, which is representative of the iterative retrieval procedure that is used across multi-hop solutions (Miller et al., 2016; Feldman and El-Yaniv, 2019; Xiong et al., 2021, inter alia.).

We construct Hotpot-SPIRAL by randomly assigning passages to the private (D_P) and public (D_G) corpora. To enable a clear comparison, we ensure that the sizes of D_P and D_G, and the proportions of questions for which the gold documents are public and private in Hop_1 and Hop_2 match those in ConcurrentQA.

6.1 Evaluation

We evaluate performance when no private information (neither queries nor documents) is revealed whatsoever. We compare four baselines, shown in Table 6. **(1) No Privacy Baseline**: We combine all public and private passages in one corpus, ignoring privacy concerns. **(2) No Privacy Multi-Index**: We create two corpora and retrieve the top k from each index in each hop, and retain the top-k of these $2k$ documents for the next hop, without applying any privacy restriction. Note performance should match single-index performance. **(3) Document Privacy**: We use the process in (2), but cannot use a private passage retrieved in Hop_1 to subsequently retrieve from public D_G. **(4) Query Privacy**: The baseline to keep q entirely private is to only retrieve from D_P.

We can answer many complex questions while revealing *no* private information whatsoever (see Table 5). However, in maintaining document privacy, the end-to-end QA performance degrades by 9% HotpotQA and 19% for ConcurrentQA compared to the quality of the non-private sys-

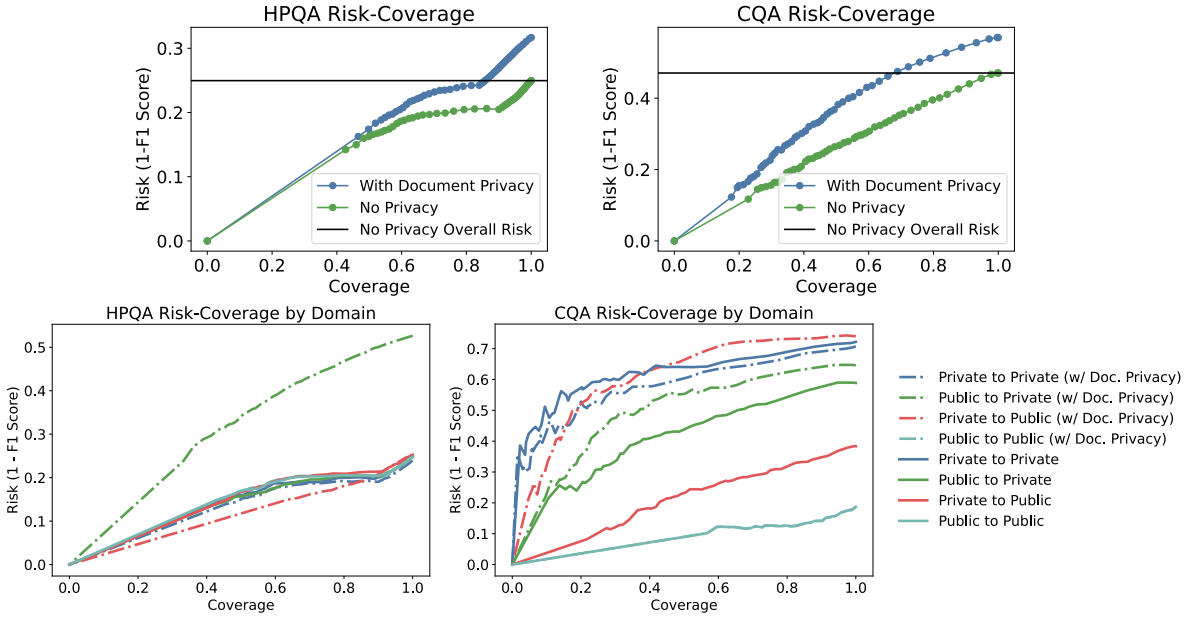

Figure 4: Risk-coverage curves using the model trained on Wikipedia data for HotpotQA-PAIR and multi-distribution CONCURRENTQA retrieval, both under No Privacy and Document Privacy, where privacy is achieved by restricting $Private$ to $Public$ retrieval altogether. The left shows the overall test results, and the right is split by the the domains of the gold supporting passages for the question at hand, for Hop_1 to Hop_2.

tem; degradation is worse under query privacy. We hope the resources we provide facilitate future work under alternate privacy frameworks.

6.2 Managing the Privacy-Quality Tradeoff

Alongside improving the retriever's quality, an important area of research for end-to-end QA systems is to avoid providing users with incorrect predictions, given existing retrievers. Significant work focuses on equipping QA-systems with this *selective-prediction* capability (Chow, 1957; El-Yaniv and Wiener, 2010; Kamath et al., 2020; Jones et al., 2021, inter alia.). Towards improving the reliability of the QA system, we next evaluate selective prediction in our novel retrieval setting.

Setup Selective prediction aims to provide the user with an answer only when the model is confident. The goal is to answer as many questions as possible (*high coverage*) with as high performance as possible (*low risk*). Given query q, and a model which outputs $(\hat{a}, c)$, where $\hat{a}$ is the predicted answer and $c \in \mathbb{R}$ represents the model's confidence in $\hat{a}$, we output $\hat{a}$ if $c \geq \gamma$ for some threshold $\gamma \in \mathbb{R}$, and abstain otherwise. As γ increases, risk and coverage both tend to decrease. The QA model outputs an answer and score for each of the top-k retrieved passages — we compute the softmax over the top-k scores and use the top softmax score as c (Hendrycks and Gimpel, 2017; Varshney et al., 2022). Models are trained on HotpotQA, representing the public domain.

Results Risk-coverage curves for HotpotQA and CONCURRENTQA are in Figure 4. Under Document Privacy, the "No Privacy" score of 75.3 F1 for HotpotQA and 53.0 F1 for CONCURRENTQA are achieved at 85.7% and 67.8% coverage.

In the top plots, in the absence of privacy concerns, the risk-coverage trends are worse for CONCURRENTQA vs. HotpotQA (i.e. quality degrades more quickly as the coverage increases). Out-of-distribution selective prediction is actively studied (Kamath et al., 2020). However, this setting differs from the standard setup. The bottom plots show on CONCURRENTQA that that the risk-coverage trends differ widely based on the sub-domains of the questions; the standard retrieval setup typically has a single distribution (Thakur et al., 2021).

Next, privacy restrictions correlate with degredations in the risk-coverage curves on both CONCURRENTQA and HotpotQA. Critically, HotpotQA is in-distribution for the retriever. Strategies beyond selective prediction via max-prob, the prevailing approach in NLP (Varshney et al., 2022), may be useful for the SPIRAL setting.

7 Conclusion

We ask how to personalize neural retrieval-systems in a privacy-preserving way and report

on how arbitrary retrieval over public and private data poses a privacy concern. We define the SPIRAL retrieval problem, present the first textual multi-distribution benchmark to study the novel setting, and empirically characterize the privacy-quality tradeoffs faced by neural retrieval systems.

We motivated the creation of a new benchmark, as opposed to repurposing existing benchmarks. CONCURRENTQA is multi-distributional — we qualitatively identified differences between the public Wikipedia and private emails in Section 4.3, and quantitatively demonstrated the effects of applying models trained on one distribution (e.g. public) to the mixed-distribution (e.g. public and private) setting in Sections 5 and 6. Private iterative retrieval is underexplored and we hope the benchmark-resource and evaluations we provide inspire further research on this topic, for instance under alternate privacy models.

Acknowledgements

We thank Jack Urbaneck, Wenhan Xiong, and Gautier Izacard for their advice and feedback. We gratefully acknowledge the support of NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); US DEVCOM ARL under No. W911NF-21-2-0251 (Interactive Human-AI Teaming); ONR under No. N000141712266 (Unifying Weak Supervision); ONR N00014-20-1-2480: Understanding and Applying Non-Euclidean Geometry in Machine Learning; N000142012275 (NEPTUNE); NXP, Xilinx, LETI-CEA, Intel, IBM, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, Google Cloud, Salesforce, Total, the HAI-GCP Cloud Credits for Research program, the Stanford Data Science Initiative (SDSI), Stanford Graduate Fellowship, and members of the Stanford DAWN project: Facebook, Google, and VMWare. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of NIH, ONR, or the U.S. Government.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations (ICLR)*.

David E. Bell and Leonard J. LaPadula. 1976. Secure computer system: Unified exposition and multics interpretation. *The MITRE Corporation*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Michael S. Bernstein, Jaime Teevan, Susan Dumais, Daniel Liebling, , and Eric Horvitz. 2012. Direct answers for search queries in the long tail. *SIGCHI*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning (PMLR)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Admodei. 2020. Language models are few-shot learners. *Advances in neural information processing systems (NeurIPS)*, 33:1877–1901.

Qingqing Cao, Noah Weber, Niranjan Balasubramanian, and Aruna Balasubramanian. 2019. Deqa: On-device question answering. In *The 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

Hao Chen, Ilaria Chillotti, Yihe Dong, Oxana Poburinnaya, Ilya Razenshteyn, and M. Sadegh Riazi. 2020a. Sanns: Scaling up secure approximate k-nearest neighbors search. In *USENIX Security Symposium*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. In *arXiv:2107.03374*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. 1998. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981.

Chao-Kong Chow. 1957. An optimum character recognition system using decision functions. In *IRE Transactions on Electronic Computers*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations (ICLR)*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography Conference (TCC)*.

Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. In *Journal of Machine Learning Research (JMLR)*.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Arthur Gervais, Reza Shokri, Adish Singla, Srdjan Capkun, and Vincent Lenders. 2014. Quantifying web-search privacy. In *ACM Conference on Computer and Communications Security (SIGSAC)*.

Mandy Guoa, Yinfei Yanga, Daniel Cera, Qinlan Shenb, and Noah Constant. 2021. Multireqa: A cross-domain evaluation for retrieval question answering models. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm:

Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Andreas Haeberlen, Benjamin C Pierce, and Arjun Narayan. 2011. Differential privacy under fire. In *USENIX Security Symposium*, volume 33, page 236.

Nathan Heller. 2017. What the enron e-mails say about us.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

Vincent C. Hu, David F. Ferraiolo, and Rick D. Kuhn. 2006. Assessment of access control systems. National Institute of Standards and Technology (NIST).

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. In *Transactions on Machine Learning Research (TMLR)*.

Gautier Izacard and Edouard Grave1. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Erik Jones, Shiori Sagawa, Pang Wei Koh, Ananya Kumar, and Percy Liang. 2021. Selective classification can magnify disparities across groups. In *International Conference on Learning Representations (ICLR)*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *35th Conference on Neural Information Processing Systems (NeurIPS)*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

B. Klimt and Y. Yang. 2004. Introducing the enron corpus. In *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS)*.

Henry Corrigan-Gibbs Dmitry Kogan. 2020. Private information retrieval with sublinear online time. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics (TACL)*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. In *Transactions of the Association for Computational Linguistics (TACL)*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mummoorthy Murugesan, Wei Jiang, Chris Clifton, Luo Si, and Jaideep Vaidya. 2010. Efficient privacy-preserving similar document detection. *The International Journal on Very Large Data Bases (VLDB)*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *CoCo@NIPS*.

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. 2021. "KILT: a benchmark for knowledge intensive language tasks". In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2523–2544.

Peng Qi, Haejun Lee, Oghenetegiri Sido, and Christopher D. Manning. 2021. Retrieve, read, rerank, then iterate: Answering open-domain questions of varying reasoning steps from text.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Phillipp Schoppmann, Lennart Vogelsang, Adrià Gascón, and Borja Balle. 2020. Secure and scalable document similarity on distributed databases: Differential privacy to the rescue. *Proceedings on Privacy Enhancing Technologies (PETS)*.

Sacha Servan-Schreiber. 2021. Private nearest neighbor search with sublinear communication and malicious security. In *2022 IEEE Symposium on Security and Privacy (SP)*.

Luo Si and Hui Yang. 2014. Privacy-preserving ir: When information retrieval meets privacy and security. *Proceedings of the 37th international conference on Research development in information retrieval (ACM SIGIR)*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Nandan Thakur, Nils Reimers, Andreas Ruckle, Abhishek Srivastav, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Kho, and Ashish Sabharwal. 2021. Musique: Multi-hop questions via single-hop question composition. In *Transactions of the Association for Computational Linguistics (TACL)*.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022. Investigating selective prediction approaches across several tasks in iid, ood, and adversarial settings. *Findings of the Association for Computational Linguistics (ACL)*.

Ellen M. Voorhees. 1999. The trec-8 question answering track report. In *TREC*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. In *Transactions of the Association for Computational Linguistics (TACL)*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Beran. 2020. Break it down: A question understanding benchmark. In *Transactions of the Association for Computational Linguistics (TACL)*.

Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick Lewis, William Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations (ICLR)*.

Yabo Xu, Benyu Zhang, Zheng Chen, and Ke Wang. 2007. Privacy-enhancing personalized web search. In *Proceedings of the 16th international conference on World Wide Web (WWW)*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen andRuslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wentau Yih, Matthew Richardson, Christopher Meek, Ming-Wei, and Chang Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jingtao Zhan, Qingyao Ai, Yiqun Liu, Jiaxin Mao, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Disentangled modeling of domain and relevance for adaptable dense retrieval. *arXiv:2208.05753*.

Michael J.Q. Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Steven Zimmerman, Alistair Thorpe, Chris Fox, and Udo Kruschwitz. 2019. Investigating the interplay between searchers' privacy concerns and their search behavior. In *Proceedings of the 42nd International ACM Conference on Research and Development in Information Retrieval (SIGIR)*.

Model	Avg-PR
Learning Rate	5e-5
Batch Size	150
Maximum passage length	300
Maximum query length at initial hop	70
Maximum query length at 2nd hop	350
Warmup ratio	0.1
Gradient clipping norm	2.0
Traininig epoch	64
Weight decay	0

Table 7: Retrieval hyperparameters for MDR training on CONCURRENTQA and Subsampled-HotpotQA experiments.

A Experimental Details

The MDR retriever is trained with a contrastive loss as in Karpukhin et al. (2020), where each query is paired with a (gold annotated) positive passage and m negative passages to approximate the softmax over all passages. We consider two methods of collecting negative passages: first, we use random passages from the corpus that do not contain the answer (random), and second, we use one top-ranking passage from BM25 that does not contain the answer as a hard-negative paired with remaining random negatives. We do not observe much difference between the two approaches for CONCURRENTQA-results (also observed in Xiong et al. (2021)), and thus use random negatives for all experiments.

The number of passages retrieved per hop, k, is an important hyperparameter; increasing k tends to increase recall, but sacrifice precision. A larger k is also less efficient at inference time. We use $k = 100$ for all experiments in the paper and Table 9 studies the effect of using different values of k.

We find the hyperparameters in Table 7 in the Appendix work best and train on up to 8 NVidia-A100 GPUs.

Sparse Retrieval For the sparse retrieval baseline, we use Pyserini with default parameters.[7] We consider different values of $k \in \{1, 10, 25, 100\}$ per retrieval, reported in Table 10 in the Appendix.

[7] https://github.com/castorini/pyserini

Model	Recall@10
Two-hop MDR	55.9
Contriever	12.1
Contriever MS-MARCO	36.9

Table 8: Comparison of one-hop baseline models evaluated on the two-hop CONCURRENTQA task without finetuning.

k	Avg-PR	F1
$k = 1$	41.4	33.5
$k = 10$	55.9	44.7
$k = 25$	63.3	48.0
$k = 50$	68.4	50.4
$k = 100$	73.8	53.0

Table 9: Retrieval performance (Average Passage-Recall@k, F1) for $k \in \{1, 10, 25, 50, 100\}$ retrieved passages per hop using the retriever trained on HotpotQA for OOD CONCURRENTQA test data.

k	F1
$k = 1$	22.0
$k = 10$	34.6
$k = 25$	37.8
$k = 50$	39.3
$k = 100$	40.8

Table 10: F1 score on the CONCURRENTQA test data for $k \in \{1, 10, 25, 100\}$ per hop using BM25.

We generate the second hop query by concatenating the text of the initial query and first hop passages.

QA Model We use the provided ELECTRA-Large reader model checkpoint from Xiong et al. (2021) for all experiments. The model was trained on HotpotQA training data. Using the same reader is useful to understand how retrieval quality affects performance, in the absence of reader modifications.

Contriever Model We use the code released by for zero-shot and fine-tuning implementation and evaluation (Izacard et al., 2021).[8] We perform a hyperparamter search for the learning rate $\in \{1e-4, 1e-5\}$, temperature $\in \{0.5, 1\}$, and number of negatives $\in \{5, 10\}$. We found a learning rate of $1e-5$ with a linear schedule and 10 negative passages to be best. These hyperparameters are chosen following the protocol in Izacard et al. (2021).

[8] https://github.com/facebookresearch/contriever

B Additional Analysis

We include two figures to further characterize the differences between the Wikipedia and Enron distributions.

Figure 5 (Left, Middle) in the Appendix shows the UMAP plots of CONCURRENTQA questions using BERT-base representations, split by whether the gold hop passages are both from the same domain (e.g., two Wikipedia or two email passages) or require one passage from each domain. The plots reflect a separation between Wiki-based and email-based questions and passages.

C CONCURRENTQA Details

Here we compare CONCURRENTQA to available textual QA benchmarks and provide additional details on the benchmark collection procedure.

C.1 Overview

CONCURRENTQA is the first multi-distribution textual benchmark. Existing benchmarks in this category are summarized in Table 11 in the Appendix. We note that HybridQA (Chen et al., 2020b) and similar benchmarks also include multi-modal documents. However, these only contain questions that require one passage from each domain for all questions, i.e. one table and one passage. Our benchmark considers text-only documents, where questions can require arbitrary retrieval patterns across the distributions.

C.2 Benchmark Construction

We need to generate passage pairs for $\text{Hop}_1, \text{Hop}_2$ of two Wikipedia documents (Public, Public), an email and a Wikipedia document (Public, Private and Private, Public), and two emails (Private, Private).

Public-Public Pairs For Public-Public Pairs, we use a directed Wikipedia Hyperlink Graph, G where a node is a Wikipedia article and an edge (a, b) represents a hyperlink from the first paragraph of article a to article b. The entity associated with article b, is mentioned in article a and described in article b, so b forms a *bridge*, or commonality, between the two contexts. Crowdworkers are presented the final public document pairs $(a, b) \in G$.

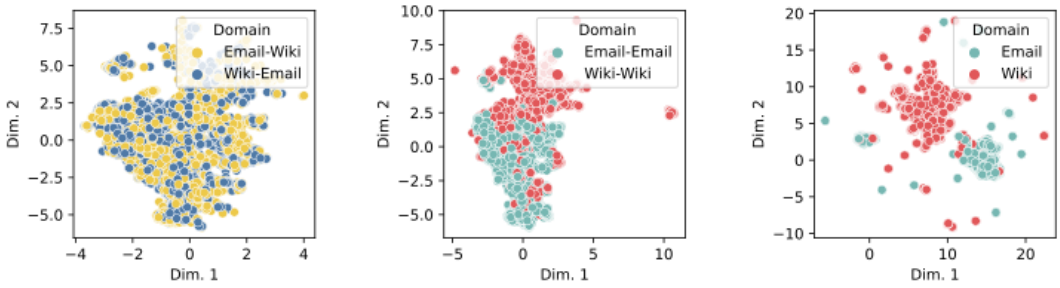

Figure 5: UMAP of BERT-base embeddings, using Reimers and Gurevych (2019), of CONCURRENTQA **questions** based on the domains of the gold passage chain to answer the question (left and middle). I.e., questions that require an Email passage for hop 1 and Wikipedia passage for hop 2 are shown as "Wiki-Email". Embeddings for all gold **passages** are also shown, split by domain (right).

Dataset	Size	Domain
WebQuestions (Berant et al., 2013)	6.6K	Freebase
WebQSP (Yih et al., 2016)	4.7K	Freebase
WebQComplex (Talmor and Berant, 2018)	34K	Freebase
MuSiQue (Trivedi et al., 2021)	25K	Wiki
DROP (Dua et al., 2019)	96K	Wiki
HotpotQA (Yang et al., 2018)	112K	Wiki
2Wiki2MultiHopQA (Ho et al., 2020)	193K	Wiki
Natural-QA (Kwiatkowski et al., 2019)	300K	Wiki
CONCURRENTQA	18.4K	Email & Wiki

Table 11: Existing textual multi-hop benchmarks are designed over a single-domain.

We provide the title of b as a hint to the worker, as a potential anchor for the multi-hop question.

To initialize the Wikipedia hyperlink graph, we use the KILT KnowledgeSource resource (Petroni et al., 2021) to identify hyperlinks in each of the Wikipedia passages. [9] To collect passages that share enough in common, we eliminate entities b which are too specific or vague, having many plausible correspondences across passages. For example, given a representing a "company", it may be challenging to write a question about its connection to the "business psychology" doctrine the company ascribes to (b is too specific) or to the "country" in which the company is located (b is too general). To determine which Wiki entities to permit for a and b pairings shown to the workers, we ensure that the entities come from a restricted set of entity-categories. The Wikidata knowledge base stores type categories associated with entities (e.g., "Barack Obama" is a "politician" and "lawyer"). We compute the frequency of Wikidata types across the 5.2 million entities and permit entities containing any type that occurs at least 1000 times. We also restrict to Wikipedia documents containing a minimum number of sentences and tokens. The intuition for this is that highly specific types entities (e.g., a legal code or scientific fact) and highly general types of entities (e.g. countries) occur less frequently.

Pairs with Private Emails Unlike Wikipedia, hyperlinks are not readily available for many unstructured data sources including the emails, and the non-Wikipedia data contains both private and public (e.g., Wiki) entities. Thus, we design the following approach to annotate the public and private entity occurrences in the email passages:

1. We collect candidate entities with SpaCy. [10]
2. We split the full set into candidate public and candidate private entities by identifying Wikipedia linked entities amongst the spans tagged by the NER model. We annotate

[9] https://github.com/facebookresearch/KILT

[10] https://spacy.io/

the text with the open-source SpaCy entity-linker, which links the text to entities in the Wiki knowledge base, to collect candidate occurrences of global entities. [11] We use heuristic rules to filter remaining noise in the public entity list.

3. We post-process the private entity lists to improve precision. High precision entity-linking is critical for the quality of the benchmark: a query assumed to require the retrieval of private passages a and b should not be unknowingly answerable by public passages. After curating the private entity list, we restrict to candidates which occur at least 5 times in the deduplicated set of passages.

A total of 43.4k unique private entities and 8.8k unique public entities appear in the emails, and 1.6k private and 2.3k public entities occur at least 5 times across passages. We present crowd workers emails containing at least three total entities to ensure there is sufficient information to write the multi-hop question.

Private-Private Pairs are pairs of emails that mention the same private entity e. The Private-Public and Public-Private are pairs of emails mentioning public entity e and the Wikipedia passage for e. In both cases, we provide the hint that e is a potential anchor for the multi-hop question.

Comparison Questions For comparison questions, Wikidata types are readily available for public entities, and we use these to present the crowdworker with two passages describing entities of the same type. For private emails, there is no associated knowledge graph so we heuristically assigned types to private entities, by determining whether type strings occurred frequently alongside the entity in emails (e.g., if "politician" is frequently mentioned in the emails in which an entity occurs, assign the "politician" type).

Finally, crowdworkers are presented with a passage pair and asked to write a question that requires information from both passages. We use separate interfaces for bridge vs. comparison questions and guide the crowdworker to form bridge questions by using the passages in the desired order for Hop_1 and Hop_2.

[11] https://github.com/egerber/spaCy-entity-linker