Retrieval-Enhanced Machine Learning

Hamed Zamani*
University of Massachusetts Amherst
zamani@cs.umass.edu

Fernando Diaz*
Google Research
diazf@acm.org

Mostafa Dehghani
Google Research
dehghani@google.com

Donald Metzler
Google Research
metzler@google.com

Michael Bendersky
Google Research
bemike@google.com

ABSTRACT

Although information access systems have long supported *people* in accomplishing a wide range of tasks, we propose broadening the scope of users of information access systems to include task-driven *machines*, such as machine learning models. In this way, the core principles of indexing, representation, retrieval, and ranking can be applied and extended to substantially improve model generalization, scalability, robustness, and interpretability. We describe a generic retrieval-enhanced machine learning (REML) framework, which includes a number of existing models as special cases. REML challenges information retrieval conventions, presenting opportunities for novel advances in core areas, including optimization. The REML research agenda lays a foundation for a new style of information access research and paves a path towards advancing machine learning and artificial intelligence.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Machine learning**.

KEYWORDS

Retrieval Augmentation; Memory Augmentation; Knowledge Grounding

1 INTRODUCTION

The vast majority of existing machine learning (ML) systems are designed to be self-contained, with both knowledge and reasoning encoded in model parameters. Consequently, increasing the capacity of machine learning models by increasing their parameter size generally leads to higher accuracy [17]. For example, the number of parameters used in state-of-the-art language models has increased

---

*Both authors contributed equally to the paper.

from 94 million in ELMo [45] to 1.6 trillion in Switch Transformers [13], an over 16× increase in just three years (2018 – 2021). Despite these successes, improving performance by increasing the number of model parameters can incur significant cost and limit access to a handful of organizations that have the resources to train them [4]. As such, focusing model development on the number of parameters is neither scalable nor sustainable in the long run.

Motivated by recent work demonstrating both that high capacity models memorize training data [6] and that using retrieval-style methods can offload memorization to storage [5], we propose the augmenting ML models with access to stored information through information retrieval (IR) techniques. Whereas IR has proven an effective tool to support people accessing large text corpora, we believe that IR can be extended to support machines accessing not just large text corpora but more abstractly-represented knowledge stores. By designing machine learning architectures that have explicit access to an information retrieval system, we can decouple reasoning from memory, reducing the required model parameters and leveraging the efficiency, scalability, and effectiveness of IR techniques. We refer to this class of approaches as *retrieval-enhanced machine learning* (REML). In this paper, we describe how core principles of indexing, representation, retrieval, and ranking can be used to develop REML models.

Using retrieval to improve model accuracy is not without precedent. Predating modern machine learning methods, the IR community developed some of the earliest known retrieval-enhanced machine learning models. For example, pseudo-relevance feedback [2, 8] leverages a retrieval system to analyze results of an 'initial' search query before producing a final ranking. This purely algorithmic use of a retrieval system in order to improve ranking model performance foreshadows its usefulness in modern applications. More recently, natural language processing models that incorporate retrieval capabilities have been shown to improve model performance [21, 39]. Although leveraging rather basic retrieval models, these approaches present an opportunity for ML systems to be further improved with more sophisticated IR methods.

We introduce a generic framework that enables ML models to be augmented with IR capabilities that support querying a corpus for useful information, utilizing retrieved results, providing feedback to the retrieval model, and, if necessary, storing information for future access. This framework is flexible enough to both represent several existing ML models and scaffold future models.

This paper is organized in order to motivate, describe, and ground REML as a research program. We begin in Section 2 by describing

the motivation for REML, specifically demonstrating why IR techniques provide a unique opportunity for ML. In Section 3, we discuss the challenges in developing each component of the proposed framework and suggest three categories of optimization approaches for REML models: (1) independent optimization of prediction and retrieval models, (2) their conditional optimization, and (3) their joint end-to-end optimization. Using this framework, in Section 4, we review several existing ML models in order to draw connections to REML. And, although these related models suggest the potential benefit of REML, substantial open research questions limit the applicability and effectiveness of contemporary IR methods. In Section 5, we conclude with a broad research program in REML, touching on the opportunity for the different subareas of IR research to contribute to the advancement of ML model performance.

2 MOTIVATION

Despite the success of modern high capacity models, focusing on the number of parameters as a primary mechanism to improve performance can be brittle, unsustainable, and opaque [4]. We argue that these concerns can be addressed by developing ML models that, instead of encoding knowledge in parameters, can access large collections of information items using efficient, effective, and robust retrieval technologies. Some of the major applications of REML is presented below:

Generalization. Recent work has shown that many ML models can significantly benefit from simple retrieval augmentation approaches. For instance, KNN-LM [32] linearly interpolates large language model predictions with the nearest neighbors of the given context input. This approach does not even require further training or fine-tuning. The authors showed substantial improvements in terms of language model perplexity in both in-distribution and out-of-distribution test sets, demonstrating the generalizability of this approach. KNN-LM together with several other examples reviewed in Section 4 suggest that enhancing ML models using retrieval models will have a large impact on the generalizability of the models. Retrieval enhancement is expected to have large impact on domain adaptation, zero-shot, and few-shot learning tasks.

Scalability. ML models compress information from training data to support accurate prediction at inference time. Although increasing model capacity by adding parameters often translates into an improvement in predictive power, recent studies demonstrate that large deep learning models often memorize training instances and concepts associated with them in their model parameters [6]. As an alternative to such implicit memorization, retrieval systems can explicitly store information either directly from the training set or from concepts derived during the learning process. Because retrieval architectures are often designed to scale, a retrieval system can provide efficient access to this information, substantially reducing the need for high capacity models and increasing throughput.

Collection Updates and the Temporal Aspect. Current ML models make predictions solely based on the data observed during training. Although effective in stationary domains, this approach can be brittle in nonstationary domains, such as news, where new information constantly emerges. And, while periodic retraining is possible in some slowly-changing domains, for quickly-changing domains, this solution is impractical. An information access system can decouple reasoning from knowledge, allowing it to be maintained and updated independent of model parameters at a cadence aligned with the corpus.

Interpretability and Explainability. Because the knowledge in training data is encoded in learned model parameters, explanations of model predictions often appeal to abstract and difficult-to-interpret distributed representations. By grounding inference on retrieved information, predictions can more easily be traced specific data, often stored in a human-readable format such as text.

On-Device Machine Learning. State-of-the-art ML models require significant computational power and memory availability, which are not available on devices such as smartphones. Retrieval-enhanced ML models can potentially decouple memorization from generalization and store a large collection (memory) of information items on a remote server. Thus, a small, efficient ML model can be hosted on-device. By minimizing the interactions between the retrieval component and the ML model, this can potentially revolutionize the applications of on-device machine learning. If privacy is an issue, the information items stored on the remote server can be encrypted and methods, such as the recently developed distance-preserving encryption schemes for nearest neighbor search [16], can be adopted for privacy-preserving retrieval.

Collectively, these properties of IR techniques suggest the development of REML, which we pursue in the subsequent sections.

3 RETRIEVAL-ENHANCED MACHINE LEARNING

This paper focuses on predictive ML models. Let $\mathcal{X}$ be the input (feature) space for the task and $\mathcal{Y}$ be the output (prediction) space. Given an input $x \in \mathcal{X}$, a ML model produces a prediction in the output space $\widehat{y} \in \mathcal{Y}$. Supervised learning models are often trained by minimizing an empirical prediction loss (error) over instances in a training set $T = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$.

Retrieval-enhanced machine learning (REML) refers to models composed of two coupled components: one model that makes predictions by communicating with N models each mediating access to a repository of information or knowledge. A REML model is defined as $f_\theta(x; R_{\omega_1}, R_{\omega_2}, \cdots, R_{\omega_N})$. The model f_θ parameterized by θ is called the *prediction model* and R_{ω_i} denotes the ith *information access model* parameterized by ω_i. Thus, to produce $\widehat{y}$, the prediction model can interface with N information access models. Each R_{ω_i} includes a collection or repository C_i that is available—through an information access model—to the prediction model. This repository could be composed of natural language documents—as with text retrieval—or some other indexed representation. As such, C_is reflect a large set of parameters available to the model that can be leveraged *ad hoc*, as with many non-parametric and lazy learning techniques. The goal of retrieval-enhanced supervised learning models is to minimize the empirical risk,

$$\frac{1}{|T|} \sum_{(x,y) \in T} \mathcal{L}\left(f_\theta(x; R_{\omega_1}, R_{\omega_2}, \cdots, R_{\omega_N}), y\right) \tag{1}$$

where $\mathcal{L}$ is a loss function for each training instance.

3.1 Overview

We define the following *necessary* requirements (Reqs) for REML:

(a) Cat 1: Retrieval-only (b) Cat 2: Retrieval with memory (c) Cat 3: Retrieval with feedback (d) Cat 4: Retrieval with memory & feedback
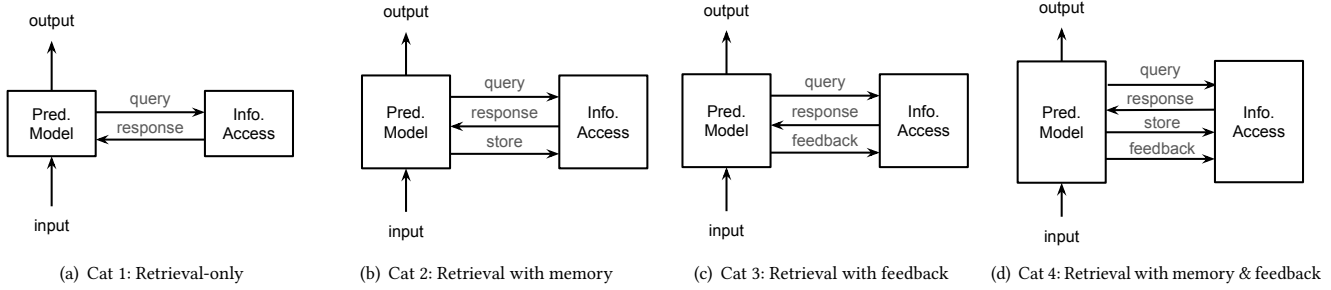
Figure 1: Retrieval-enhanced machine learning models should implement three necessary requirements (querying, retrieval, and response utilization) and may implement two optional properties (storing information and providing feedback to the information access model). This results in four categories of REML models presented above.
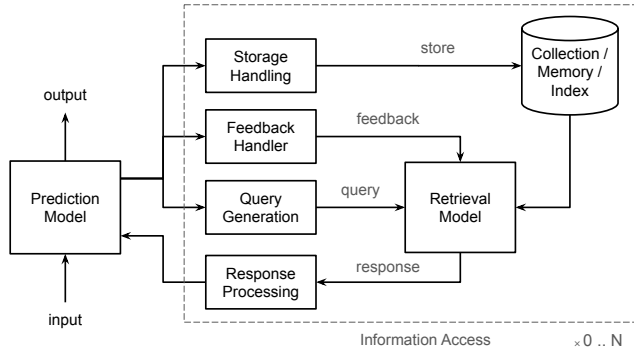

Figure 2: A generic framework for REML.

Req 1 **Querying**: the prediction model f_θ should be able to submit input-dependent queries to the information access models, i.e., R_{ω_i}s.

Req 2 **Retrieval**: each information access model R_{ω_i} should be able to efficiently process the prediction model's queries and retrieve relevant information items from a memory or collection C_i.

Req 3 **Response Utilization**: the prediction model f_θ should utilize the response returned by the information access models for making predictions.

Considering these three requirements, we can envision the first category of REML models. A high-level overview of models in this category is presented in Figure 1(a). Most existing retrieval-enhanced ML models, such as REALM [21], belong to this category.

REML may also benefit from two additional *optional* properties:

Opt 1 **Storing**: the prediction model may store some information items in a memory for future access during both training and inference. Such information items will be accessible to the model through querying (Req 1).

Opt 2 **Feedback**: the prediction model may be able to provide feedback to the information access models. This enables the information access models to improve based on the feedback.

Figure 1(b) depicts the second category of REML models that take advantage of Opt 1 by storing information in a memory and accessing the information later. On the other hand, Figure 1(c) demonstrates a high-level overview of the third category of REML models that can provide feedback (Opt 2) to the information access systems. The last category (Figure 1(d)) implements both of these optional properties and supports querying, utilizing retrieval

responses, storing information, and providing feedback to the information access systems.

Based on these requirements and optional properties, Figure 2 envisions a generic framework for REML. The framework consists of two major parts: the prediction model f_θ and the information access models R_{ω_i}s. For each input x, the model f_θ may decide to run multiple retrieval processes by either submitting multiple queries, accessing multiple data repositories and/or memories, providing feedback to the information access component, or a combination of the above. The number of retrieval processes can be zero for some inputs, and thus REML generalizes typical predictive modeling.

3.2 Information Access in REML

In its most generic form, each information access system in the proposed REML framework consists of five components: (1) Query Generation, (2) Retrieval Model, (3) Response Processing, (4) Feedback Handler, and (5) Storage Handler. In the following subsections, we discuss potential implementations for each component.

3.2.1 Query Generation. In current information access systems, queries mostly take the form of unstructured text (e.g., keyword queries or natural language questions), structured query language (e.g., SQL), or multi-media items (e.g., images). Such query languages and formats can be also adopted by retrieval-enhanced ML models. The Query Generation component is responsible for generating one of these query formats. Note that depending on the application and due to efficiency or effectiveness requirements, one may simply cast the query generation problem to query selection from a set of pre-defined queries. In either case, the Query Generation (or Selection) component should be able to translate the information need of the prediction model f_θ to a query language or format that can be efficiently processed by the information access model R_{ω_i}. Since retrieval models accessible by f_θ may accept different query languages, the Query Generation component may be unique to each retrieval model.

Existing information access systems are designed for people and, therefore, existing query formats (mentioned above) are understandable by people. In the context of REML, we can relax the requirement of an interpretable query language. Besides the common query languages and formats, the prediction models can produce any *latent representation* (e.g., a high-dimensional dense vector) as a query. For instance, any hidden layer representation produced by the prediction model f_θ may be used as a query for retrieval. That

being said, queries may also be generated from the input x itself without any involvement of the prediction model parameters.

Under REML, prediction models do not have restrictions on the number of queries that can be submitted for each input x. As a result, a model may generate multiple, sequential queries produced for each input x, resulting in a *query session* analogous to human search sessions. While current search engines base sessions on temporally-adjacent user queries, REML prediction models can, when querying, explicitly indicate a unique session ID associated with the input x.

3.2.2 Retrieval Model. The retrieval model component aims at retrieving information items from the given collection, repository, or memory in response to each query produced by the Query Generation component. Existing retrieval models are mostly designed based on the probability ranking principle (PRP) [49], in which documents are ranked based on their probability of relevance to the query. In the IR literature, relevance can be defined in five levels [52]: (1) systematic or algorithmic relevance, (2) topical relevance, (3) cognitive relevance or pertinence, (4) situational relevance or utility, and (5) motivational or affective relevance. However, these definitions assume that the retrieved documents are consumed by humans. This assumption no longer holds for REML models, thus the notion of relevance needs to be revisited for REML.

When designing retrieval models for REML, relevance can be thought of as the utility that the prediction model obtains by consuming the results produced by the retrieval model; this is similar to task-based perspectives on (human) information retrieval [30]. For simplicity and without loss of generality, assume for each input x, the prediction model f_θ only submits a single query q to a retrieval model that returns a result list $L_q = \{(d_1, \phi(d_1)), (d_2, \phi(d_2)), \cdots, (d_k, \phi(d_k))\}$, where each d_i is a document[1] in the collection and $\phi(d_i)$ encodes a list of features and properties associated with document d_i. For instance, $\phi(d_i)$ may contain the document score produced by the retrieval model in addition to a number of features used by the retrieval model to compute the score. With a slight abuse of notation, let $f(x; L_q)$ denote the prediction function that submits the query q to a retrieval model and uses its response (i.e., L_q) to make a prediction. Then, the utility gain can be defined as:

$$\text{UtilityGain}(q, L_q; f_\theta, x) = U(f_\theta(x; L_q), y) - U(f_\theta(x; \emptyset), y) \quad (2)$$

where $U(\cdot, \cdot)$ represents some desired utility function. This definition assumes that data points (x, y) are i.i.d. samples. Utility gain depends on how the prediction model f_θ consumes L_q for producing $\widehat{y}$. Utility gain can take on both positive and negative values. A negative gain means that the retrieval results L_q have negative impact on predicting the ground truth label. This definition can be extended to multiple queries per x.

The implementation of retrieval models for REML depends on the nature of documents in the collection. For instance, one can use the vector space model and employ the inner product as the similarity function between query and document vectors. Section 3.3 provides more information on the optimization of retrieval models in REML.

---

[1] In this paper, we refer to retrievable items, e.g., unstructured text, image, or even latent vectors, as documents.

3.2.3 Response Processing. The way the prediction models consume the retrieved items has a substantial impact on their end-to-end performance. The Response Processing component takes the results returned by the retrieval models for each query q (i.e., L_qs) and prepares it for consumption by the prediction model.

This component can be implemented by returning the content of the retrieved documents, synthesizing a summary of their content, producing one or more semantic representations of their content, combining all the information presented in L_q in some way, and so on. There are many design choices here and the best choice will largely depend on the nature of the machine learning model and the task it is being applied to.

3.2.4 Feedback Handler. When training retrieval models, it is often desirable to get feedback from the machine learning model. Such feedback can then be used as a signal for optimizing the retrieval model. We can imagine various forms of feedback in this context. For example, the model can compute the utility gain of documents returned by the retrieval model using Equation (2). As another example, the feedback may be computed based on the gradients of the prediction loss with respect to the retrieved information. Section 3.3 discusses how the model's feedback can be used for optimizing retrieval models in REML.

3.2.5 Storage Handler. If the prediction model has the ability to store information in the repository (or memory), the Storage Handler can expand the collection by storing the information item into the memory. However, for efficient storage and access of a large number of items, careful consideration of memory management techniques, hardware requirements, and storage data structures beyond existing technologies (e.g., inverted indexes) is required. Besides information storage, this component is also responsible for storage management. Thus, it should implement caching, compression, access controls, and time-to-live requirements as necessary.

3.3 REML Optimization

We envision three optimization approaches for REML: (1) independent optimization of prediction and information access models, (2) conditional optimization of these models such that the quality of one impacts the optimization of the other, and (3) joint end-to-end optimization of both models. Without loss of generality, here we assume that there only exists one information access model.

3.3.1 Independent Optimization of Prediction and Information Access Models. In independent optimization, the training process of the prediction model f_θ is independent of the retrieval performance. For example, we can assume that the retrieval model is optimal. Formally, we can optimize the prediction model of REML as:

$$\theta^* = \arg\min_\theta \frac{1}{|T|} \sum_{(x,y) \in T} \mathcal{L}(f_\theta(x; R_{\text{opt}}), y) \quad (3)$$

where R_{opt} denotes an optimal retrieval model and can be modeled using ground truth relevance information, if available. Similar to [72], we can also model imperfect retrieval models by introducing noise to an optimal ranking behavior. The retrieval model can be trained using typical learning-to-rank (LTR) formulation, independent of f_θ. For the same of space, we refer the reader to Liu [42] for more information on LTR models.

3.3.2 Conditional Optimization of Prediction and Information Access Models. In conditional optimization, the prediction model parameters get updated conditioned on the retrieval model's performance and vice versa. This process can be done iteratively until a stopping criterion is met (e.g., convergence or early stopping based on performance on a held-out validation set). Therefore, the prediction model can be optimized as:

$$\theta^{(t)} = \arg\min_{\theta} \frac{1}{|T|} \sum_{(x,y) \in T} \mathcal{L}(f_\theta(x; R_{\omega^{(t)}}), y) \tag{4}$$

$$\omega^{(t+1)} = \arg\min_{\omega} \frac{1}{|T|} \sum_{(x,y) \in T} \mathcal{L}(f_{\theta^{(t)}}(x; R_\omega), y) \tag{5}$$

where $\theta^{(t)}$ and $\omega^{(t)}$ denote the parameters of the prediction model and the information access model at the tth iteration, respectively. These equations assume that both models are being optimized. In case of using unsupervised retrieval models, the second optimization process would be skipped (i.e., $\omega_{t+1} = \omega_t$).

3.3.3 Joint End-to-End Optimization. In end-to-end optimization of REML, both ML and information access models are trained jointly by optimizing a single objective function. Formally, it is defined as:

$$\theta^*, \omega^* = \arg\min_{\theta,\omega} \frac{1}{|T|} \sum_{(x,y) \in T} \mathcal{L}(f_\theta(x; R_\omega), y) \tag{6}$$

For optimizing this objective via gradient descent-based optimizers, the whole REML process (both models and their interactions) is required to be differentiable. End-to-end optimization is expected to perform better than the last two optimization approaches, but given the complexity of retrieval from large collections, this requirement may be difficult to satisfy in some cases.

3.4 Extending REML to Multiple ML Models

Previous sections consider only a single prediction model that interacts with multiple retrieval processes (see Figure 2). This section extends the REML framework to multiple prediction models. Similar to current search engines that provide service to many users, retrieval models can be also employed by multiple ML models.

Assume there are M prediction models $f_{\theta_1}, f_{\theta_2}, \cdots, f_{\theta_M}$ that use N information access models denoted by $R_{\omega_1}, R_{\omega_2}, \cdots, R_{\omega_N}$. Each R_{ω_i} should provide service to multiple prediction models. This introduces the following challenges:

Shared Query Language: All prediction models may need to share the same query language for interacting with retrieval systems.
Shared Response Formats: The responses produced by each retrieval system will be used by all prediction models. Therefore, the prediction models should be able to utilize the response format used by each retrieval model.
Shared Storage: The storage used by each retrieval model is shared between all prediction models. Storage is a limited resource, thus a policy may be required to regulate storage usage for each prediction model. Moreover, the data stored by each prediction model may not be interpretable by other models or may not be shared due to privacy restrictions. The Storage Handling component should develop memory management and access restriction policies and functionalities for each storage request.

Personalization:[2] The prediction models have special needs and they utilize the retrieval responses differently. Therefore, in response to a query q submitted by two prediction models f_{θ_i} and f_{θ_j}, the retrieval models may want to respond differently. In this case, retrieval models would need to implement models and techniques for personalizing the search results.
Comparable Feedback Across Prediction Models: Comparable feedback across prediction models enables us to easily aggregate the obtained feedback. Otherwise, the feedback can be used for each individual prediction model as a form of personalization.
Optimizing Retrieval Models: In case of dealing with trainable retrieval models, the optimization solutions introduced in Section 3.3 need further adjustments. Let $\mathcal{L}_i$ denote the loss function associated with the ith prediction model. Thus, the joint end-to-end optimization of models can be achieved as follows:

$$\arg\min_{\theta,\omega} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|T_i|} \sum_{(x,y) \in T_i} \alpha_i \mathcal{L}_i(f_{\theta_i}(x; R_\omega), y) \tag{7}$$

where T_i denotes the training data for the ith prediction task. This formulation assumes that the loss values are comparable across prediction models. The hyper-parameter α_is control the weight of each loss function. The conditional optimization formulation can be adjusted, similarly.

3.5 Information Access Evaluation in REML

The prediction model should be evaluated based on its performance on the downstream task, and appropriate evaluation methodologies and metrics should be chosen considering the downstream task. This evaluation is the same for any predictive model designed for that task. Therefore, we skip the evaluation of prediction models and discuss approaches for evaluating the information access models. Evaluating information access in REML is particularly important for diagnosing the retrieval process and designing retrieval systems that provide service to multiple prediction models (see Section 3.4). The retrieval component in REML can be evaluated either extrinsically or intrinsically:

Extrinsic Evaluation: The information access quality can be quantified by measuring its impact on the prediction model for the downstream task. This is perhaps the most important factor in evaluating information access in REML. Note that in case of having multiple prediction models, extrinsic evaluation is defined for each prediction model independently. However, aggregating the downstream performances for different prediction models is challenging, because prediction models may be evaluated based on various metrics and methodologies and they may not aggregate easily. Extrinsic evaluation can be done both through offline and online evaluation.
Intrinsic Evaluation: In intrinsic evaluation, the retrieval model is evaluated independent of the prediction models. To do so, one may define *relevance* based on the desired documents expected to be retrieved for a prediction model. This definition may be obtained from experts or by analyzing observations from prediction models' behavior. Then presumably an annotation process, e.g., through pooling, may be employed for creating data collections for intrinsic evaluation of the information access model. Metrics used in

---

[2]Personalization is often used for humans. We stick to the same terminology to be consistent with the IR literature.

intrinsic evaluation are expected to have high correlations with the downstream performance of the prediction models. We highlight that most metrics used in the IR literature have been developed based on user behaviors with search engines. For instance, many of them assume that users assess documents sequentially. However, such assumptions may not hold for many ML models. Thus, new evaluation metrics may need to be developed.

4 CASE STUDIES

Since REML is a general framework, we can discuss related approaches as special cases of REML. This exercise helps us understand how and when REML might work and suggests opportunities for extending existing work.

4.1 Knowledge Grounding

Fully data-driven ML models, despite demonstrating success across a wide number of tasks, still lack grounding in the real world. Access to external knowledge, via *knowledge grounding*, may help with this issue [11, 22, 34, 39, 76]. Knowledge grounding models make predictions based on the results returned by a retrieval model.

In the context of language modeling, one class of methods uses retrieval results as evidence to support *reasoning*. For example, the knowledge retriever module in REALM [21] accesses information from an encoded Wikipedia corpus during pre-training. In text generation, RetGen [75] combines a grounded text generator with a document retriever. Grounding the generation helps with the issue of hallucinated facts, and the retrieval component makes the grounding effective and efficient. Lewis et al. [39] highlighted the importance of retrieval in knowledge-intensive NLP tasks and introduced retrieval-augmented generation (RAG) by augmenting a generator with the output of a non-parametric retriever that uses maximum inner product search.

Entities as Experts (EaE) [15] introduces an entity memory that can be accessed by the model and the retrieved representations of entities are combined with the input representation for entity linking, mention detection, and masked language modeling tasks. Similarly, Fact as Experts (FaE) [61] incorporates a fact memory for language modeling. Such a mechanism gives access to factual information, that may expand or change over time, while there is no need for additional training or fine-tuning.

In open-domain QA, a common approach is to retrieve documents or passages from Wikipedia or even the Web and then extract answers [27, 47]. Lee et al. [37] used an encoded Wikipedia corpus to train a retrieval model and then fine-tune the prediction model for a QA objective. Khattab et al. [33] used a retrieval component for multi-hop reasoning, where the retrieved facts from each hop are summarized into a short context and becomes a part of the query for the subsequent hops. Similarly, Das et al. [10] performed iterative retrieval for expanding and rewriting multi-hop questions. This is also the case for task-oriented dialogues. For instance, LaMDA [58] shows the benefit of granting dialogue systems access to external knowledge for reducing unsourced statement hallucination [53].

The approaches presented in this subsection mostly use simple retrieval models, e.g., TF-IDF or inner product of learned representations, for finding factual information from external knowledge

bases. Therefore, one can look at knowledge grounding as an implementation of REML, mostly based on Category 1: Retrieval-only (Figure 1(a)) or Category 3: Retrieval with feedback (Figure 1(c)).

4.2 Memory-Augmented Machine Learning

Using a memory component where the model can read from and/or write into is one of the most common ways of implementing REML in neural networks. The main motivation is to use an explicit storage buffer to make it easier for the network to rapidly incorporate new information and not to forget in the future.

A model may use an internal memory where it compresses and accumulates information to access them in later stages of the process. This has been the base of several neural architecture classes. For instance, Long Short-Term Memory networks (LSTMs) [25] or Gated Recurrent Networks [7] that use a latent state as a memory to collect information from previous time steps. Attention-based models [3, 60] also treat different parts of the input as memories and use soft access as the retrieval mechanism to manage the interaction between them. However, memory-augmented neural networks refers to cases of using an external memory [51]. Among main works in this area, memory networks [55] explicitly store information in a form that is element-wise addressable. Neural Turing machines [18, 19] are well-known examples of ML models that can read from and write into an external memory matrix in order to represent and manipulate complex data structures.

The common target property of memory-augmented neural networks is incorporating an external memory that is trained end-to-end with the objective and data from downstream tasks. This most resonates with the fourth category of REML: Retrieval with memory and feedback (Figure 1(d)). However, the memory size in existing models is relatively small and extending the memory size is an exciting and challenging research direction.

4.3 Retrieval-Enhanced Input Representation

A number of retrieval-enhanced models use the retrieved items to update the representations of the model's input. This is different from knowledge grounding in the sense that the information items do not necessary include the knowledge required for accomplishing the task. Instead, the retrieved information contains patterns that can help the model to learn more expressive representations.

Pseudo relevance feedback (PRF) is an example of such models. It uses the top retrieved documents for updating the query representation through query expansion. It has shown successful results in a wide range of retrieval tasks [2, 8, 14, 28, 36, 68, 73], demonstrating the quality of the produced query representations for retrieval. Recently, Hashemi et al. [22] proposed Guided Transformer, an extension to the Transformer network that includes cross attention for contextualizing inputs with retrieved information from multiple information sources to learn more accurate representations of the model's input. In their subsequent work [23], the authors proposed an approach for learning multiple representations for query intents by utilizing the retrieval results and taking advantage of the Guided Transformer network for representation adjustment. More recently, Borgeaud et al. [5] proposed RETRO for language modeling and showed that by using networks like Guided Transformer one can enable access to a trillion-scale database for a relatively small model.

Related approaches have been also used in computer vision [20, 35, 40, 54]. For example, Xu et al. [69] studied the task of image inpainting whose goal is to restore missing regions of an image. They introduced a "texture memory" that augments a neural network with access to patches extracted from unmasked regions of the input image. For the task of 3D scene reconstruction, Siddiqui et al. [54] used retrieval for creating multiple approximate reconstructions and then fusing them with an attention-based blending module to generate the output. For object detection, Kuo et al. [35] used retrieval from a large-scale dataset of 3D models to understand the underlying 3D structure of objects seen in a 2D image.

Similar to knowledge grounding, retrieval-enhanced representation learning can take advantage of information items that are similar to the input by learning from patterns observed in the retrieved results. Thus, the first (retrieval-only) and the third (retrieval with feedback) REML categories are often used for this purpose.

4.4 Generalization through Memorization

Combining retrieval-based and generative approaches has been explored in a number of applications. In this case, the retrieval component can contribute by producing accurate responses when memorization is sufficient.

Motivated by the goal of memorizing rare patterns explicitly, Khandelwal et al. [32] introduced KNN-LM, where a retrieval mechanism is used to find the nearest neighbor tokens given the prefix as query. KNN-LM linearly interpolates the predicted distribution for the next token using distance information from the retrieval mechanism. BERT-KNN [29] employs a similar nearest neighbor algorithm to augment a BERT model to learn better representations for rare facts. This idea has also been extended to machine translation [31]. It is shown that retrieval augmentation improves domain adaptation by using a domain-specific datastore for retrieval. Tay et al. [57] proposed training a large model that memorizes the mapping of document content to document ids, which can be used to retrieve relevant document ids given a query at inference time. This model could be an alternative to KNN based models we discussed above to serve a REML system as a differential index.

In dialogue systems, given a dialogue history as a query, a retrieval unit can be used to return the top ranked candidate response as the next dialogue utterance [50]. Such retrieval-based approaches can also be combined with response generation models and form a hybrid solution for dialogue systems [70].

Another approach to improve generalization through memorization is through updating retrieval results. In some cases, editing an existing candidate output is easier than generating it from scratch, especially in complex structured output generation tasks, like code generation. Hashimoto et al. [24] proposed to retrieve a training example given the input and edit it to the desired output. The retriever and the editing modules are trained jointly. Pasupat et al. [44] proposed using exemplar retrieval for semantic parsing. In their setup, given a query, the parser retrieves a set of related exemplars, augments the query using the retrieved information, and then incorporates a seq2seq model [56] to produce an output parse.

The aforementioned methods try to use a retrieval component to handle memorization cases. It is found useful, especially for cases where sufficient training data is not available. Many existing models are based on a retrieval-only implementation of REML.

4.5 Efficient Access to Longer Context

Due to memory constraints as well as efficiency and effectiveness reservations, consuming and representing large inputs, e.g., long text documents or videos, are challenging. REML offers a solution to address this issue by giving access to the context of any size via a retrieval mechanism. Here we mention a few examples of studies that exploit this idea.

Wu et al. [63] proposed using a long-term feature bank for detailed video understanding. The long-term feature bank stores a rich, time-indexed representation of a long video. Then the video understanding model consults with the bank through a retrieval module to get features that encode information about past and future scenes, objects, and actions. Similarly, MemViT [64], proposes to process videos in an online fashion and store information in memory at each iteration. The model can retrieve prior context from the memory to enable long-term modeling for the recognition task. Similar approaches have also been used for video object segmentation [43] and video summarization [38].

For processing long documents, researchers often split the documents into passages. For instance, Dai and Callan [9] only used the first passage of each document for document retrieval. Xiong et al. [66] used the passage with the maximum similarity score with the query. The end-to-end intra-document cascading model [26] can be seen as a REML model with feedback. It first selects (retrieves) a number of passages from the document and then consumes the selected passages for scoring the document.

The methods presented in this subsection are perhaps the simplest implementations of REML: the retrieval collection is not large, and some of them do not use feedback.

4.6 Retrieval-Enhanced Optimization

All the methods mentioned above use a retrieval component at the inference time for making accurate predictions. Some approaches use retrieval components solely for the purpose of optimization, e.g., for producing training data and/or computing loss functions. Thus, the retrieval model will not be used during inference.

A natural application of retrieval-enhanced optimization is for retrieval tasks. Dehghani et al. [12] introduced a weak supervision approach for IR by producing large-scale training data through BM25 and training ML models for document ranking. Zamani and Croft [71] used the top retrieved documents to produce a relevance model distribution for training queries and learn relevance-based word embedding. Producing hard negative instances for training learning-to-rank models is another application of REML. For instance, ANCE [66] and its extensions [41, 46] are dense retrieval models that iteratively use the model parameters to retrieve documents for producing 'hard' negative samples for training the model.

Wu et al. [65] used a retrieval unit to enable unsupervised training of machine translations, i.e., using two monolingual corpora in the source and target languages with no alignment. As an alternative to back translation, they proposed retrieving a sentence from the target corpus using the source sentence and applying some

changes using an editing mechanism to the retrieved target to generate source-target pairs and train the MT model. Triantafillou et al. [59] proposed an approach for few-shot learning through retrieval. This approach retrieves items for each input and uses them for making predictions. Via this approach, a model can adapt to a new domain without additional training or new data.

An interesting use case of REML is the pre-training task of CLIP [48] and VideoCLIP [67] which are practically optimizing for text-image and text-video retrieval, respectively. They are in fact capturing cross-modal relevance that led to learning representations that are effective in various setups, like zero-shot classification.

5 A RESEARCH AGENDA

While Section 4 provides evidence of the efficacy and broad applicability of REML, there remain significant open research challenges in fully realizing the general REML vision, some of which are already mentioned in previous sections.

5.1 Querying

In developing a prediction model that supports retrieval, understanding how to query becomes a core research question. First, this involves knowing when to query. In some situations, a prediction model may not benefit from a retrieval operation (even if it benefits on average). Although current retrieval-enhanced systems issue the equivalent of queries for *every instance*, when querying incurs some cost, be it in the form of latency or financial expense, developing models that "know when they don't know" would allow the prediction algorithm to explicitly trade off cost and benefit. A prediction model that has access to multiple information access services can make this decision for individual corpora, perhaps select the appropriate source for the instance. Second, at a more granular level, how retrieval might benefit a model may vary by instance x. For example, retrieval may support uncertainty in one part of the θ for one instance and uncertainty in another part of θ for another instance. This self-interrogation can be explicitly designed or implicitly learned. Nevertheless, even learnable behavior requires an architecture and parameters to adapt. Finally, many retrieval situations can benefit from the searcher conveying non-semantic meta-retrieval information such as uncertainty in (aspects of) the query or context of the retrieval itself. People often convey similar information to human intermediaries [1] and we suspect that more expressive querying can also emerge in REML.

In developing an information access model to support a prediction model, similar questions arise. First, developing or learning a query language requires expressiveness that captures the breadth of model needs. At the same time, it should allow for communication of meta-retrieval or structured properties of the retrieved set. Moreover, these properties need to be explored within the effectiveness and efficiency constraints. Second, although a query may be effective and efficient in general, it may be ambiguous or imprecise for a particular retrieval scenario. This is especially likely in situations where multiple models may develop inconsistent uses of the query language (Section 3.4).

5.2 Storing

The ability of the prediction model to store items presents unique problems not encountered in traditional retrieval or ML research. Although architectures like memory networks [62] provide modestly sized storage, we anticipate models storing or serializing on a larger scale with more permanence. In situations with multiple models (Section 3.4), we anticipate the corpus operating as a means to share derived knowledge (to avoid re-computation during inference) or prediction model parameters (to support learning).

In developing a prediction model that supports storage, understanding how to store becomes a core research question. Just as with querying, a model needs to reason about when to store, what to store from its parameters or reasoning, and how to represent that information. Each of these questions is relevant both to sharing derived knowledge as well as model parameters. Like queries, stored items may include auxiliary information such as the model's confidence in the derived data or parameter values, the prediction task, and other information that may be valuable for an information access system to make retrieval decisions. More so than with queries, a model might need to be more judicious in storage operations, since injecting irrelevant or erroneous content into the corpus can significantly degrade its usefulness.

In developing an information access model to support storage, classic problems related to indexing arise. First, as with queries, the language, schema, or representation of an item requires careful construction to optimize for effectiveness and efficiency. Second, in accepting a storage request from a prediction model, the information access system needs to model the value of the content. Redundant items can either add noise or improve coverage, depending on the task. Or, an item may require processing to make indexing and retrieval more effective. These decisions can be based on the content of the item or meta-data about the item, such as the confidence of the model or, in the case of multiple models, confidence in the prediction model itself. Third, if an item *should* be stored, there is the question of *how* to store it. This includes questions of item compression and representation, both of which need to occur incrementally but improve with batch, corpus-wide computation. Finally, in the case of limited capacity in the retrieval index, storage operations may necessitate purging less effective content. This requires that the information access model reason about how collection management decisions impact prediction models.

5.3 Searching

Ranking functions, a fundamental property of traditional information access systems, influence design decisions about how to store content compactly, how to search that content quickly, and how to return results effectively. In moving toward REML, several fundamental research questions need to be addressed in order to satisfy these properties for machines. First, items in REML indexes are likely to be differently structured than existing text documents (see Section 5.2). Although representations like dense, fixed dimensional vectors are amenable to efficient storage and retrieval, structures that include uncertainty and other attributes may require embedding as a representation amenable to fast retrieval (e.g., vectors) or different indexing schemes altogether. Second, the representations of items in the index themselves should be selected for effectiveness

in supporting prediction models, as well as the space and runtime efficiency. In some cases, this means accurate and fast score computation. When a retrieval involves more elaborate post-processing before returning results, this may mean decomposing items before indexing (as is often done when retrieving passages, as opposed to documents). Third, in situations where there are multiple prediction models, the information access system can use the identity of the model in order to 'personalize' results for that model. Similarly, we can interpret the feedback from prediction models based on where it comes from; some models may not provide actionable feedback early in learning, others may be quite reliable, while others yet might be adversarial. Third, these representations and their associated ranking functions themselves should be tunable given feedback from prediction models (see Section 5.5). Adjustments to representations and model weights should be sensitive to confidence in the feedback signal in situations where feedback includes a confidence estimate or if the information access model can estimate the reliability of the feedback.

5.4 Information Presentation & Consumption

Representing the retrieval results in traditional information access involves returning a ranked list of items. Although items include scores, these are often only used to sort items and are rarely presented to the user. In the context of REML, we can consider more elaborate representations of retrieval results because they are being consumed by machines. This introduces a number of exciting research directions. First, system designers will need to understand the appropriate information to communicate to prediction models, be it an item ranking, a scored set, a set where each item is associated with a score distribution, a graph of inter-item relationships, or some other object derived from the retrieval. Each of these choices needs to satisfy improving the prediction model's effectiveness, within any cost constraints (e.g., bandwidth, compute). Moreover, in situations with multiple prediction models, the consistency, interpretability, and maintainability of this representation language become extremely important. Second, from an efficiency perspective, just as computing a top k ranking can suggest fast document scoring, information about the representation can introduce opportunities for more efficient computations of objects like graphs and score distributions. Third, a prediction model with access to multiple information access models needs to reason over multiple sets of results. Information encoded in the results–explicitly or not–can allow the prediction model to consider the reliability of results before incorporating them into inference. Finally, from a machine learning perspective, *how* to incorporate results into inference will become an important area of work. Current approaches based on neighbors provide a simple approach, although more sophisticated techniques are likely to improve performance.

5.5 Feedback

Modern information access systems use implicit user feedback in order to optimize model parameters. Although we can imagine a prediction model providing loss information in its feedback similar to how users might provide slate-level feedback, machines may be able to convey more granular and expressive feedback to the information access model. As such, the first area of research centers

on forms of feedback, including scalar values, vectors of values, and more expressive data with goal of helping the information access model improve. While single scalar feedback values seem simplest, even modern search engines exploit implicit item-level feedback. We can imagine more targeted and attributed feedback provided by the prediction model. This structured feedback can include attribution to different components of the retrieval structure (Section 5.4). Of course, this requires the prediction model being able to identify the relationship between prediction error and different parts of the retrieval result; in the case of multiple information access services, attribution to individual corpus results. The second area of research focuses on how an information access model might adjust model parameters given rich feedback from the prediction model. Current ranking models, with appropriate treatment of different biases, can interpret user feedback as attributed to individual items in the ranking. A machine may be able to provide feedback that has fewer biases and better calibration than human feedback. This includes exploring a new space of feedback beyond scalar item-level values. This also calls for novel approaches for optimizing information access models based on the provided feedback.

5.6 Evaluation

The objective of REML is to support machines. As such, standard methods of evaluating modeling performance (e.g., Equation 1) can be adopted to assess prediction model performance. Nevertheless, REML introduces several research directions around model evaluation. First, because of the large, flexible storage capacity, REML can memorize training data or cache previous predictions, resulting in performance metrics (e.g., accuracy) conflating a model's ability to reason (i.e., the prediction model) and its ability to remember (i.e., the information access model). Methods of selecting evaluation instances or ablation experiments can isolate the contribution of each component. Second, in situations with multiple prediction models, we need methods to assess performance changes for a group of models with a shared information access service. Although these per-model losses can be aggregated into a simple average, this may obscure model- or task-specific under-performance. That said, in some situations, storage operations might result in sharing information, boosting collective performance, and necessitating an evaluation method that decouples reasoning from memorization. Finally, efficiency metrics that capture the cost of query and response operations (e.g., latency, financial) will need to be developed.

In some cases, we are interested in evaluating the information access model in isolation to make a claim about generalizability of a specific retrieval model to new prediction models, just as we traditionally consider evaluation queries as a *sample* from the full set of queries we would like to apply a system to. Although we can evaluate information access models using the existing information access evaluation methods (e.g., Cranfield-style offline evaluation, click feedback), we anticipate the opportunity—and sometimes *need*—to develop entirely new evaluation schemes. First, although a prediction model can be evaluated by its loss function, an information access model can be evaluated by its adoption. Indeed, if a retrieval component is not used, then perhaps it can be removed altogether. To see why retrieval systems may be more or less valuable over time, consider the situation where a prediction model can store

items such as partial inference or complete inference; in this case, the storage can act like a cache, with queries likely to grow with time, depending on the data. Or, if there are multiple information access services, the usefulness of some may increase or decrease over time. Nonstationarity can also arise when instances have serial dependencies, such as when a retrieval system is repeatedly queried during a dialog or multi-hop task. Second, estimating an information access model's performance on out of sample domains or tasks requires careful selection of training and evaluation tasks. Third, in developing offline or batch evaluation methods, although we can avoid some issues, labeling items for relevance and designing metrics reflective of model use becomes difficult, since existing ranking metrics are unlikely to approximate how a machine would consume results (see Section 5.4). Finally, REML presents a tremendous opportunity to study these questions *in silico*. This means that experimentation and analysis, although more complicated, will be much faster than systems serving people, without safety concerns, since experiments can be run isolated from people.

6 CONCLUSION

Although the large number of parameters in models such as deep neural networks has, in part, resulted in impressive improvements in performance across a wide range of tasks, evidence suggests that these successes may be partially due to the increased capacity to store information in model parameters [74]. We claim that, if model capacity is being used to store information, then we should decouple reasoning from memory and expand the scope of information retrieval to also support ML models. Starting from this claim, we have presented a general framework, its relation to existing methods, and its ability to substantially advance how we think about information retrieval and how we do machine learning.

ACKNOWLEDGMENTS

REFERENCES

[1] Jaime Arguello, Adam Ferguson, Emery Fine, Bhaskar Mitra, Hamed Zamani, and Fernando Diaz. 2021. Tip of the Tongue Known-Item Retrieval: A Case Study in Movie Identification. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. Association for Computing Machinery, New York, NY, USA, 5–14.

[2] R. Attar and A. S. Fraenkel. 1977. Local Feedback in Full-Text Retrieval Systems. *J. ACM* 24, 3 (jul 1977), 397–417. https://doi.org/10.1145/322017.322021

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 610–623. https://doi.org/10.1145/3442188.3445922

[5] 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. 2021. Improving language models by retrieving from trillions of tokens. arXiv:2112.04426 [cs.CL]

[6] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. arXiv preprint arXiv:2012.07805. In *USENIX Security Symposium*. https://arxiv.org/abs/2012.07805

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] W. B. Croft and D. J. Harper. 1979. Using Probabilistic Models of Document Retrieval Without Relevance Information. *J. of Documentation* 35, 4 (1979), 285–295.

[9] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 985–988. https://doi.org/10.1145/3331184.3331303

[10] Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step Entity-centric Information Retrieval for Multi-Hop Question Answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. Association for Computational Linguistics, Hong Kong, China, 113–118. https://doi.org/10.18653/v1/D19-5816

[11] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke. 2019. Learning to transform, combine, and reason in open-domain question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 681–689.

[12] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 65–74. https://doi.org/10.1145/3077136.3080832

[13] William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *arXiv:2101.03961* (2021).

[14] S. L. Feng, R. Manmatha, and V. Lavrenko. 2004. Multiple Bernoulli Relevance Models for Image and Video Annotation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, D.C., USA) *(CVPR'04)*. IEEE Computer Society, USA, 1002–1009.

[15] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as Experts: Sparse Memory Access with Entity Supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 4937–4951. https://doi.org/10.18653/v1/2020.emnlp-main.400

[16] Georg Fuchsbauer, Riddhi Ghosal, Nathan Hauke, and Adam O'Neill. 2021. Approximate Distance-Comparison-Preserving Symmetric Encryption. Cryptology ePrint Archive, Report 2021/1666. https://ia.cr/2021/1666.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[18] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* (2014).

[19] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 7626 (2016), 471–476.

[20] Shir Gur, Natalia Neverova, Chris Stauffer, Ser-Nam Lim, Douwe Kiela, and Austin Reiter. 2021. Cross-Modal Retrieval Augmentation for Multi-Modal Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 111–123. https://doi.org/10.18653/v1/2021.findings-emnlp.11

[21] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 3929–3938. http://proceedings.mlr.press/v119/guu20a.html

[22] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1131–1140. https://doi.org/10.1145/3397271.3401061

[23] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2021. *Learning Multiple Intent Representations for Search Queries*. Association for Computing Machinery, New York, NY, USA, 669–679. https://doi.org/10.1145/3459637.3482445

[24] Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A Retrieve-and-Edit Framework for Predicting Structured Outputs. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/cd17d3ce3b64f227987cd92cd701cc58-Paper.pdf

[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[26] Sebastian Hofstätter, Bhaskar Mitra, Hamed Zamani, Nick Craswell, and Allan Hanbury. 2021. *Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking*. Association for Computing Machinery, New York, NY, USA, 1349–1358. https://doi.org/10.1145/3404835.3462889

[27] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[28] J. Jeon, V. Lavrenko, and R. Manmatha. 2003. Automatic Image Annotation and Retrieval Using Cross-Media Relevance Models *(SIGIR '03)*. Association for Computing Machinery, New York, NY, USA, 119–126. https://doi.org/10.1145/860435.860459

[29] Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN Search Component to Pretrained Language Models for Better QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 3424–3430. https://doi.org/10.18653/v1/2020.findings-emnlp.307

[30] Diane Kelly, Jaime Arguello, and Robert Capra. 2013. NSF Workshop on Task-based Information Search Systems. *SIGIR Forum* 47, 2 (2013).

[31] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest Neighbor Machine Translation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=7wCBOfJ8hJM

[32] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HklBjCEKvH

[33] Omar Khattab, Christopher Potts, and Matei Zaharia. 2020. Baleen: Robust Multi-Hop Reasoning at Scale via Condensed Retrieval. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[34] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-Augmented Dialogue Generation. *CoRR* abs/2107.07566 (2021). arXiv:2107.07566 https://arxiv.org/abs/2107.07566

[35] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. 2020. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *European Conference on Computer Vision*. Springer, 260–277.

[36] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New Orleans, Louisiana, USA) (SIGIR '01)*. Association for Computing Machinery, New York, NY, USA, 120–127. https://doi.org/10.1145/383952.383972

[37] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300* (2019).

[38] Sangho Lee, Jinyoung Sung, Youngjae Yu, and Gunhee Kim. 2018. A memory network approach for story-based temporal summarization of 360 videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1410–1419.

[39] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[40] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. 2015. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer graphics forum*, Vol. 34. Wiley Online Library, 435–446.

[41] Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. *More Robust Dense Retrieval with Contrastive Dual Learning*. Association for Computing Machinery, New York, NY, USA, 287–296. https://doi.org/10.1145/3471158.3472245

[42] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (mar 2009), 225–331. https://doi.org/10.1561/1500000016

[43] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. 2019. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9226–9235.

[44] Panupong Pasupat, Yuan Zhang, and Kelvin Guu. 2021. Controllable Semantic Parsing via Retrieval Augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 7683–7698. https://aclanthology.org/2021.emnlp-main.607

[45] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[46] Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. *Learning Robust Dense Retrieval Models from Incomplete Relevance Labels*. Association for Computing Machinery, New York, NY, USA, 1728–1732. https://doi.org/10.1145/3404835.3463106

[47] Chen Qu, Hamed Zamani, Liu Yang, W. Bruce Croft, and Erik Learned-Miller. 2021. *Passage Retrieval for Outside-Knowledge Visual Question Answering*. Association for Computing Machinery, New York, NY, USA, 1753–1757. https://doi.org/10.1145/3404835.3462987

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[49] Stephen E. Robertson. 1997. *The Probability Ranking Principle in IR*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 281–286.

[50] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for Building an Open-Domain Chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 300–325. https://doi.org/10.18653/v1/2021.eacl-main.24

[51] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*. PMLR, 1842–1850.

[52] Tefko Saracevic. 1996. Relevance reconsidered. In *Proceedings of the Second Conference on Conceptions of Library and Information Science* (Copenhagen, Denmark).

[53] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567* (2021).

[54] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. 2021. RetrievalFuse: Neural 3D Scene Reconstruction with a Database. arXiv:2104.00024 [cs.CV]

[55] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. *Advances in neural information processing systems* 28 (2015).

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc.

[57] Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991* (2022).

[58] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. arXiv:2201.08239 [cs.CL]

[59] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. 2017. Few-Shot Learning through an Information Retrieval Lens. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 2252–2262.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008.

[61] Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and Interpretable Neural MemoryOver Symbolic Knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3678–3691. https://doi.org/10.18653/v1/2021.naacl-main.288

[62] Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[63] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. 2019. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 284–293.

[64] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. 2022. MeMViT: Memory-Augmented Multiscale Vision Transformer for Efficient Long-Term Video Recognition. *arXiv preprint arXiv:2201.08383* (2022).

[65] Jiawei Wu, Xin Wang, and William Yang Wang. 2019. Extract and Edit: An Alternative to Back-Translation for Unsupervised Neural Machine Translation. In *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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 1173–1183. https://doi.org/10.18653/v1/N19-1120

[66] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations (ICLR'21)*.

[67] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6787–6800. https://doi.org/10.18653/v1/2021.emnlp-main.544

[68] Jinxi Xu and W. Bruce Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Zurich, Switzerland) *(SIGIR '96)*. Association for Computing Machinery, New York, NY, USA, 4–11. https://doi.org/10.1145/243199.243202

[69] Rui Xu, Minghao Guo, Jiaqi Wang, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. 2021. Texture memory-augmented deep patch-based image inpainting. *IEEE Transactions on Image Processing* 30 (2021), 9112–9124.

[70] Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W. Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A Hybrid Retrieval-Generation Neural Conversation Model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1341–1350. https://doi.org/10.1145/3357384.3357881

[71] Hamed Zamani and W. Bruce Croft. 2017. Relevance-Based Word Embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 505–514. https://doi.org/10.1145/3077136.3080831

[72] Hamed Zamani and W. Bruce Croft. 2018. On the Theory of Weak Supervision for Information Retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* (Tianjin, China) *(ICTIR '18)*. Association for Computing Machinery, New York, NY, USA, 147–154. https://doi.org/10.1145/3234944.3234968

[73] Chengxiang Zhai and John Lafferty. 2001. Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management* (Atlanta, Georgia, USA) *(CIKM '01)*. Association for Computing Machinery, New York, NY, USA, 403–410. https://doi.org/10.1145/502585.502654

[74] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Sy8gdB9xx

[75] Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2022. Joint Retrieval and Generation Training for Grounded Text Generation. In *AAAI*.

[76] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774* (2021).