The Probabilistic Relevance Framework: BM25 and Beyond

Domain: RAG

Relevance Score: 4

Abstract

The Probabilistic Relevance Framework (PRF) is a formal framework for document retrieval, grounded in work done in the 1970–1980s, which led to the development of one of the most successful text-retrieval algorithms, BM25. In recent years, research in the PRF has yielded new retrieval models capable of taking into account document meta-data (especially structure and link-graph information). Again, this has led to one of the most successful Web-search and corporate-search algorithms, BM25F. This work presents the PRF from a conceptual point of view, describing the probabilistic modelling assumptions behind the framework and the different ranking algorithms that result from its application: the binary independence model, relevance feedback models, BM25 and BM25F. It also discusses the relation between the PRF and other statistical models for IR, and covers some related topics, such as the use of non-textual features, and parameter optimisation for models with free parameters.

Summary

explores the theoretical underpinnings, development, and extensions of the Probabilistic Relevance Framework (PRF) used in information retrieval systems. Central to this framework is the idea of estimating the probability of relevance between a query and a document, which serves as the foundation for ranking algorithms like BM25.

Limitations

- **Assumptions of Relevance:** The paper assumes relevance is a binary property, which may not capture the nuances of user needs where relevance can be graded or context-dependent.
- **Independence Assumptions:** The model relies on conditional independence between terms, which is often not true in practice. This can lead to oversimplifications and inaccuracies in relevance estimation.
- **Lack of Explicit Probability Estimates:** While the model focuses on ranking documents, it does not provide a mechanism for estimating the actual probability of relevance for each document, which can be crucial in certain retrieval scenarios.

Dense Passage Retrieval for Open-Domain Question Answering

Domain: RAG

Relevance Score: 6

Abstract

Open-domain question answering relies on ef- ficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto method. In this work, we show that retrieval can be practically implemented us- ing dense representations alone, where embeddings are learned from a small number of questions and passages by a simple dual- encoder framework. When evaluated on a wide range of open-domain QA datasets, our dense retriever outperforms a strong Lucene- BM25 system greatly by 9%-19% absolute in terms of top-20 passage retrieval accuracy, and helps our end-to-end QA system establish new state-of-the-art on multiple open-domain QA benchmarks.

Summary

The paper presents an innovative approach to passage retrieval for answering open-domain questions. It addresses limitations in traditional sparse vector models like TF-IDF and BM25 by introducing dense representations trained using a dual-encoder framework. This framework uses embeddings learned from question-passage pairs to improve retrieval accuracy. Dense Passage Retrieval (DPR) is shown to significantly outperform BM25, achieving superior performance on top-20 and top-100 passage retrieval accuracy across multiple benchmarks. The study's key contributions include the effective use of a dual-encoder architecture optimized for inner product similarity between questions and passages, without requiring extensive pretraining. DPR's robustness is demonstrated through strong empirical results on datasets like Natural Questions and TriviaQA, where it achieves state-of-the-art results in passage retrieval and end-to-end question answering. Additionally, the research highlights that dense retrieval methods benefit from careful training setups, such as in-batch negatives, which improve retrieval precision.

Limitations

- **Dependence on Pre-trained Models**: The Dense Passage Retriever (DPR) relies on the BERT pre-trained model, which may limit its performance on domains or tasks that differ significantly from the data used for pre-training.
- **Training Data Requirements**: Although the paper claims that a small number of question-passage pairs can yield good results, the need for

labeled pairs still poses a challenge, especially in domains where such data is scarce.

- **Computational Intensity**: The training process, particularly for the dense representations, is computationally intensive. While the retrieval process is efficient, the initial indexing of passages requires significant resources and time.
- **Generalization Issues**: The model's performance may degrade when applied to datasets that differ from those used during training, indicating potential overfitting to the training data.
- **Evaluation Metrics**: The reliance on specific evaluation metrics (e.g., exact match) may not fully capture the model's performance in real-world applications, where nuanced understanding and flexibility are required.

Learning Transferable Visual Models From Natural Language Supervision

Domain: OCR

Relevance Score: 4

Abstract

State-of-the-art computer vision systems are trained to predict a fixed set of predetermined object categories. This restricted form of supervision limits their generality and usability since additional labeled data is needed to specify any other visual concept. Learning directly from raw text about images is a promising alternative which leverages a much broader source of supervision. We demonstrate that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study the performance of this approach by benchmarking on over 30 different existing computer vision datasets, spanning tasks such as OCR, action recognition in videos, geo-localization, and many types of fine-grained object classification. The model transfers non-trivially to most tasks and is often competitive with a fully supervised baseline without the need for any dataset specific training. For instance, we match the accuracy of the original ResNet-50 on ImageNet zero-shot without needing to use any of the 1.28 million training examples it was trained on. We release our code and pre-trained model weights at https://github.com/OpenAI/CLIP.

Summary

The paper introduces CLIP (Contrastive Language-Image Pretraining), a scalable framework for training visual models directly from raw text-image pairs. Unlike traditional methods that rely on pre-defined object categories, CLIP uses a dataset of 400 million image-text pairs to train both an image and a text encoder jointly, predicting the alignment of an image with its corresponding text. This contrastive approach allows the model to generalize without task-specific training, enabling zero-shot transfer to various computer vision tasks. The study demonstrates CLIP's efficacy by testing it across more than 30 datasets, including tasks like image classification, OCR, and action recognition. CLIP achieves competitive results against state-of-the-art supervised models, matching ImageNet accuracy of ResNet50 in a zero-shot setting. Furthermore, CLIP exhibits robustness to distribution shifts, outperforming standard models under such conditions. Despite its strengths, the authors highlight limitations, such as computational demands and challenges in handling complex or abstract tasks.

Limitations

- **Performance Comparison**:
  - Zero-shot CLIP's performance is often only competitive with a linear classifier on ResNet-50 features, which is below the overall state-of-the-art (SOTA).
  - Significant improvements are needed for CLIP to reach SOTA performance across evaluation suites, estimated to require around a 1000x increase in compute.
- **Evaluation Methodology**:
  - The reliance on validation sets during development may not reflect true zero-shot scenarios, as it introduces a form of bias.
  - The selection of evaluation datasets may be co-adapted with CLIP's capabilities, potentially skewing results.

C-RAG: Certified Generation Risks for Retrieval-Augmented Language Models

Domain: RAG

Relevance Score: 6.5

Abstract

Despite the impressive capabilities of large language models (LLMs) across diverse applications, they still suffer from trustworthiness issues, such as hallucinations and misalignments. Retrieval-augmented language models (RAG) have been proposed to enhance the credibility of generations by grounding external knowledge, but the theoretical understandings of their generation risks remains

unexplored. In this paper, we answer: 1) whether RAG can indeed lead to low generation risks, 2) how to provide provable guarantees on the generation risks of RAG and vanilla LLMs, and 3) what sufficient conditions enable RAG models to reduce generation risks. We propose C-RAG, the first framework to certify generation risks for RAG models. Specifically, we provide conformal risk analysis for RAG models and certify an upper confidence bound of generation risks, which we refer to as conformal generation risk. We also provide theoretical guarantees on conformal generation risks for general bounded risk functions under test distribution shifts. We prove that RAG achieves a lower conformal generation risk than that of a single LLM when the quality of the retrieval model and transformer is non-trivial. Our intensive empirical results demonstrate the soundness and tightness of our conformal generation risk guarantees across four widely-used NLP datasets on four state-of-the-art retrieval models.

Summary

The paper introduces C-RAG, a framework designed to certify and provide theoretical guarantees for the generation risks associated with retrieval-augmented language models (RAG). The authors address critical issues like hallucinations and reliability in large language models (LLMs), focusing on whether RAG models can effectively minimize generation risks compared to standard LLMs.

Limitations

- **Probability of Guarantee**: The C-RAG framework provides high-confidence risk bounds, but there is still a possibility of generations with excessive risks. More calibration samples may be needed to achieve a higher confidence level and mitigate outlier occurrences.
- **Trade-off with External Knowledge Base Size**: While a larger external knowledge base can reduce conformal generation risk, it may also increase the time complexity of KNN searching and the space complexity for storing examples, leading to a trade-off between generalization/utility and inference efficiency.

Atlas: Few-shot Learning with Retrieval Augmented Language Models

Domain: RAG

Relevance Score: 6.5

Abstract

Large language models have shown impressive few-shot results on a wide range of tasks. However, when knowledge is key for such results, as is the case for

tasks such as question answering and fact checking, massive parameter counts to store knowledge seem to be needed. Retrieval-augmented models are known to excel at knowledge intensive tasks without the need for as many parameters, but it is unclear whether they work in few-shot settings. In this work we present Atlas, a carefully designed and pre-trained retrieval-augmented language model able to learn knowledge intensive tasks with very few training examples. We perform evaluations on a wide range of tasks, including MMLU, KILT and Natural Questions, and study the impact of the content of the document index, showing that it can easily be updated. Notably, Atlas reaches over 42% accuracy on Natural Questions using only 64 examples, outperforming a 540B parameter model by 3% despite having 50x fewer parameters.

Summary

The paper presents Atlas, a retrieval-augmented language model designed to excel in few-shot learning tasks, particularly those requiring extensive knowledge, such as question answering and fact-checking. Unlike traditional large language models that rely heavily on vast parameter counts to store knowledge, Atlas utilizes a dual-encoder architecture for document retrieval, allowing it to achieve impressive performance with significantly fewer parameters (11B) compared to models like PaLM (540B). The authors demonstrate that Atlas can achieve over 42% accuracy on the Natural Questions dataset using only 64 training examples, outperforming larger models by 3% while being 50 times smaller. The study emphasizes the importance of joint pre-training of the retriever and language model components, exploring various training objectives and pretext tasks to enhance few-shot performance. Through extensive experiments across multiple benchmarks, including MMLU, KILT, and TriviaQA, Atlas establishes new state-of-the-art results in several tasks, showcasing its adaptability, interpretability, and efficiency. The findings suggest that retrieval-augmented models like Atlas can effectively decouple memorization from generalization, making them a promising approach for knowledge-intensive natural language processing tasks.

Limitations

- **Complexity of Fine-tuning**
  - The fine-tuning process may require careful tuning of hyperparameters, which can be resource-intensive and may not be straightforward for all users.
  - The need for joint training of the retriever and language model adds complexity to the training process.
- **Scalability Issues**: As the size of the document index increases, the computational resources required for retrieval and processing may become a bottleneck, limiting scalability in real-world applications.

REST: Retrieval-Based Speculative Decoding

Domain: RAG

Relevance Score: 7

Abstract

We introduce Retrieval-Based Speculative Decoding (REST), a novel algorithm designed to speed up language model generation. The key insight driving the development of REST is the observation that the process of text generation often includes certain common phases and patterns. Unlike previous methods that rely on a draft language model for speculative decoding, REST harnesses the power of retrieval to generate draft tokens. This method draws from the reservoir of existing knowledge, retrieving and employing relevant tokens based on the current context. Its plug-and-play nature allows for seamless integration and acceleration of any language models, all without necessitating additional training. When benchmarked on 7B and 13B language models in a single-batch setting, REST achieves a significant speedup of 1.62X to 2.36X on code or text generation. The code of REST is available at https://github.com/FasterDecoding/REST.

Summary

The paper introduces Retrieval-Based Speculative Decoding (REST), a novel algorithm aimed at enhancing the efficiency of language model generation. Unlike traditional speculative decoding methods that rely on a smaller draft language model, REST utilizes a retrieval mechanism to generate draft tokens from a datastore of existing knowledge. This approach allows for significant speed improvements in text and code generation, achieving speedups of 1.62x to 2.36x on 7B and 13B language models without requiring additional training. The method constructs a Trie from retrieved candidates and employs a tree attention mechanism for verification, ensuring that the generated sequences maintain high quality while minimizing computational overhead.

Limitations

- **Dependence on Datastore Quality**
  - The performance of REST is directly influenced by the accuracy and completeness of the datastore.
  - A higher quality datastore may be required for better alignment with the LLM, potentially necessitating the use of content generated by the LLM itself.
- **Lack of In-Context Abilities**: REST may struggle with tasks that require understanding of context, such as retrieving personalized variable names in code generation. This limitation raises questions about how retrieval methodologies can effectively handle complex contextual requirements.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Domain: RAG

Relevance Score: 7

Abstract

Large pre-trained language models have been shown to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks. However, their ability to access and precisely manipulate knowledge is still limited, and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures. Additionally, providing provenance for their decisions and updating their world knowledge remain open research problems. Pre-trained models with a differentiable access mechanism to explicit non-parametric memory can overcome this issue, but have so far been only investigated for extractive downstream tasks. We explore a general-purpose fine-tuning recipe for retrieval-augmented generation (RAG) – models which combine pre-trained parametric and non-parametric memory for language generation. We introduce RAG models where the parametric memory is a pre-trained seq2seq model and the non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever. We compare two RAG formulations, one which conditions on the same retrieved passages across the whole generated sequence, the other can use different passages per token. We fine-tune and evaluate our models on a wide range of knowledge-intensive NLP tasks and set the state-of-the-art on three open domain QA tasks, outperforming parametric seq2seq models and task-specific retrieve-and-extract architectures. For language generation tasks, we find that RAG models generate more specific, diverse and factual language than a state-of-the-art parametric-only seq2seq baseline.

Summary

The paper presents Retrieval-Augmented Generation (RAG), a novel approach that combines pre-trained parametric memory (a sequence-to-sequence model) with non-parametric memory (a dense vector index of Wikipedia) to enhance knowledge-intensive natural language processing (NLP) tasks. RAG models utilize a retriever to access relevant documents based on input queries and a generator to produce outputs conditioned on both the input and the retrieved documents. The authors explore two formulations of RAG: RAG-Sequence, which uses the same retrieved document for the entire output sequence, and RAG-Token, which allows different documents for each token generated.

Limitations

- **Performance on Specific Tasks**: Although RAG models set state-of-the-art results on certain open-domain QA tasks, their performance may not generalize across all knowledge-intensive tasks.
- **Scalability Concerns**: The approach may face scalability issues when dealing with larger datasets or more complex tasks, particularly in terms of retrieval efficiency.
- **Potential for Misuse**: The ability to generate factual content raises concerns about the potential misuse of the technology for generating misleading or harmful information.

REALM: retrieval-augmented language model pre-training

Domain: Foundation model + RAG

Relevance Score: 7

Abstract

Language model pre-training has been shown to capture a surprising amount of world knowledge, crucial for NLP tasks such as question answering. However, this knowledge is stored implicitly in the parameters of a neural network, requiring everlarger networks to cover more facts. To capture knowledge in a more modular and interpretable way, we augment language model pretraining with a latent knowledge retriever, which allows the model to retrieve and attend over documents from a large corpus such as Wikipedia, used during pre-training, fine-tuning and inference. For the first time, we show how to pre-train such a knowledge retriever in an unsupervised manner, using masked language modeling as the learning signal and backpropagating through a retrieval step that considers millions of documents. We demonstrate the effectiveness of Retrieval-Augmented Language Model pretraining (REALM) by fine-tuning on the challenging task of Open-domain Question Answering (Open-QA). We compare against state-of-the-art models for both explicit and implicit knowledge storage on three popular Open-QA benchmarks, and find that we outperform all previous methods by a significant margin (4-16% absolute accuracy), while also providing qualitative benefits such as interpretability and modularity.

Summary

The paper presents REALM (Retrieval-Augmented Language Model), a novel framework that enhances language model pre-training by integrating a learned knowledge retriever. Unlike traditional language models that store knowledge implicitly within their parameters, REALM allows the model to explicitly retrieve

and utilize information from a large corpus, such as Wikipedia, during both pre-training and inference. This approach not only improves the model's ability to access relevant knowledge but also enhances interpretability and modularity. The authors demonstrate the effectiveness of REALM by fine-tuning it on the challenging task of Open-domain Question Answering (Open-QA), achieving state-of-the-art results across multiple benchmarks and outperforming existing models by a significant margin.

The paper details the architecture of REALM, which consists of a neural knowledge retriever and a knowledge-augmented encoder, and describes the training process that involves backpropagating through the retrieval step. The authors also address computational challenges associated with large-scale document retrieval and propose strategies to optimize performance. Through extensive experiments, REALM shows substantial improvements in accuracy and retrieval effectiveness, highlighting its potential for advancing natural language processing tasks that require extensive world knowledge.

Limitations

- **Dependence on Quality of Knowledge Corpus**:
  - The effectiveness of REALM is heavily reliant on the quality and comprehensiveness of the knowledge corpus (e.g., Wikipedia).
  - If the corpus lacks relevant information, the model's performance may degrade.
- **Limited Generalization to Other Domains**:
  - The experiments primarily focus on Open-domain Question Answering (Open-QA), which may not generalize well to other NLP tasks or domains.
  - The model's performance in specialized domains or with domain-specific knowledge is not thoroughly evaluated.

Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering

Domain: RAG

Relevance Score: 7

Abstract

Generative models for open domain question answering have proven to be competitive, without resorting to external knowledge. While promising, this approach requires to use models with billions of parameters, which are expensive to train and query. In this paper, we investigate how much these models can benefit from retrieving text passages, potentially containing evidence. We obtain state-of-the-art results on the Natural Questions and TriviaQA open

benchmarks. Interestingly, we observe that the performance of this method significantly improves when increasing the number of retrieved passages. This is evidence that sequence-to-sequence models offers a flexible framework to efficiently aggregate and combine evidence from multiple passages.

Summary

The paper titled investigates the integration of generative models with passage retrieval techniques to enhance open domain question answering (QA) systems. The authors highlight that while generative models have shown competitive performance without external knowledge, they often require extensive resources due to their large parameter sizes. By incorporating passage retrieval, particularly from sources like Wikipedia, the authors demonstrate that their approach, termed Fusion-in-Decoder, significantly improves performance on benchmarks such as Natural Questions and TriviaQA. The method retrieves multiple passages and utilizes a sequence-to-sequence model to generate answers, effectively aggregating evidence from these passages.

Limitations

- **Limited Integration of Retrieval and Generation**:
  - The proposed method processes passages independently in the encoder, which may limit the model's ability to leverage inter-passage relationships effectively.
  - Future work could explore more integrated approaches that combine retrieval and generation more seamlessly.
- **Scalability Concerns**:
  - Although the method scales well with the number of retrieved passages, there may be practical limits to how many passages can be effectively processed, especially in real-time applications.
  - The computational cost may increase significantly with larger numbers of passages, potentially leading to latency issues.

Retrieval Augmented Code Generation and Summarization

Domain: RAG

Relevance Score: 8

Abstract

oftware developers write a lot of source code and documentation during software development. Intrinsically, developers often recall parts of source code or code summaries that they had written in the past while implementing software or documenting them. To mimic developers' code or summary generation behavior, we

propose a retrieval augmented framework, REDCODER, that retrieves relevant code or summaries from a retrieval database and provides them as a supplement to code generation or summarization models. REDCODER has a couple of uniqueness. First, it extends the state-of-the-art dense retrieval technique to search for relevant code or summaries. Second, it can work with retrieval databases that include unimodal (only code or natural language description) or bimodal instances (code-description pairs). We conduct experiments and extensive analysis on two benchmark datasets of code generation and summarization in Java and Python, and the promising results endorse the effectiveness of our proposed retrieval augmented framework.

Summary

The paper presents REDCODER, a retrieval-augmented framework designed to enhance code generation and summarization tasks for software developers. By mimicking the behavior of developers who often recall and adapt previously written code or summaries, REDCODER retrieves relevant code snippets or summaries from a database and incorporates them into the generation process. The framework employs a two-step approach: first, a retriever module identifies relevant code or summaries, and then a generator module uses this augmented input to produce the desired output. The authors conducted extensive experiments on benchmark datasets in Java and Python, demonstrating that REDCODER significantly improves the quality of generated code and summaries compared to existing models, achieving notable increases in Exact Match and BLEU scores. The uniqueness of REDCODER lies in its ability to work with both unimodal and bimodal retrieval databases, allowing it to leverage high-quality source code and natural language descriptions effectively. The results indicate that the integration of retrieved information enhances the performance of code generation and summarization tasks, validating the framework's approach to automating software development processes. The paper concludes with a discussion on the potential for extending REDCODER to other code automation tasks, highlighting its contributions to the field of software engineering and natural language processing.

Limitations

- **Limited to Specific Programming Languages**: The experiments were primarily conducted on Java and Python, which may limit the generalizability of the findings to other programming languages.
- **Performance on Long Code**: The performance of the generator (PLBART) decreases with increasing code length, indicating challenges in handling longer code snippets effectively.

DocPrompting: Generating Code by Retrieving the Docs

Domain: RAG

Relevance Score: 8

Abstract

Publicly available source-code libraries are continuously growing and changing. This makes it impossible for models of code to keep current with all available APIs by simply training these models on existing code repositories. Thus, existing models inherently cannot generalize to using unseen functions and libraries, because these would never appear in the training data. In contrast, when human programmers use functions and libraries for the first time, they frequently refer to textual resources such as code manuals and documentation, to explore and understand the available functionality. Inspired by this observation, we introduce DocPrompting: a natural-language-to-code generation approach that explicitly leverages documentation by (1) retrieving the relevant documentation pieces given an NL intent, and (2) generating code based on the NL intent and the retrieved documentation. DocPrompting is general: it can be applied to any programming language and is agnostic to the underlying neural model. We demonstrate that DocPrompting consistently improves NL-to-code models: DocPrompting improves strong base models such as CodeT5 by 2.85% in pass@1 (52% relative gain) and 4.39% in pass@10 (30% relative gain) in execution-based evaluation on the popular Python CoNaLa benchmark; on a new Bash dataset tldr, DocPrompting improves CodeT5 and GPT-Neo1.3B by up to absolute 6.9% exact match.

Summary

The paper introduces a novel approach called DocPrompting, which enhances natural language to code generation (NL $\rightarrow$ code) by leveraging code documentation. Traditional code generation models struggle to generalize to unseen functions and libraries due to their reliance on training data that may not include all available APIs. In contrast, human programmers often consult documentation when encountering new functions. DocPrompting addresses this gap by first retrieving relevant documentation based on a natural language intent and then generating code using both the intent and the retrieved documentation. The authors demonstrate that this method significantly improves the performance of existing models, such as CodeT5 and GPT-Neo, across various benchmarks, including Python and Bash, achieving notable gains in execution-based evaluations. The paper also details the implementation of DocPrompting, which consists of a retriever that selects relevant documents and a generator that produces code snippets based on these documents. The experiments conducted show that DocPrompting consistently outperforms baseline models that do not utilize doc-

umentation, highlighting its effectiveness in enabling models to generate code for previously unseen functions. The authors provide new benchmarks for retrieval-based code generation and emphasize the potential for further improvements through better retriever and generator designs.

Limitations

- **Dependence on Documentation Quality**:
  - The effectiveness of DocPrompting heavily relies on the quality and comprehensiveness of the retrieved documentation.
  - If the documentation is outdated, incomplete, or poorly written, it may lead to inaccurate code generation.
- **Generalization to New Libraries**:
  - While DocPrompting aims to generalize to unseen functions and libraries, it may still struggle with entirely new libraries that lack sufficient documentation.
  - The approach assumes that relevant documentation is available for all potential new functions, which may not always be the case.
- **Retrieval Performance Variability**:
  - The performance of the retrieval component can vary significantly based on the chosen retriever (sparse vs. dense).
  - The paper indicates that BM25 performs well for the tldr dataset but not as effectively for CoNaLa, suggesting that the choice of retriever is critical and context-dependent.

Retrieval-Augmented Generation for Large Language Models: A Survey

Domain: RAG

Abstract

Large Language Models (LLMs) showcase impressive capabilities but encounter challenges like hallucination, outdated knowledge, and non-transparent, untraceable reasoning processes. Retrieval-Augmented Generation (RAG) has emerged as a promising solution by incorporating knowledge from external databases. This enhances the accuracy and credibility of the generation, particularly for knowledge-intensive tasks, and allows for continuous knowledge updates and integration of domain-specific information. RAG synergistically merges LLMs' intrinsic knowledge with the vast, dynamic repositories of external databases. This comprehensive review paper offers a detailed examination of the progression of RAG paradigms, encompassing the Naive RAG, the Advanced RAG, and the Modular RAG. It meticulously scrutinizes the tripartite foundation of RAG frameworks, which includes the retrieval, the generation and the augmentation techniques. The paper highlights the state-of-the-art technologies embedded in

each of these critical components, providing a profound understanding of the advancements in RAG systems. Furthermore, this paper introduces up-to-date evaluation framework and benchmark. At the end, this article delineates the challenges currently faced and points out prospective avenues for research and development.

Summary

The paper provides a comprehensive survey of Retrieval-Augmented Generation (RAG) techniques for enhancing Large Language Models (LLMs). It discusses the limitations of LLMs, such as hallucination and outdated knowledge, and presents RAG as a solution that integrates external knowledge sources to improve the accuracy and credibility of generated content. The authors categorize RAG into three paradigms: Naive RAG, Advanced RAG, and Modular RAG, each representing a progression in methodology and effectiveness. The paper meticulously examines the core components of RAG, including retrieval, generation, and augmentation techniques, while also highlighting state-of-the-art technologies and evaluation frameworks.

Document Language Models, Query Models, and Risk Minimization for Information Retrieval

Domain: RAG

Relevance Score: 7

Abstract

We present a framework for information retrieval that combines document models and query models using a probabilistic ranking function based on Bayesian decision theory. The framework suggests an operational retrieval model that extends recent developments in the language modeling approach to information retrieval. A language model for each document is estimated, as well as a language model for each query, and the retrieval problem is cast in terms of risk minimization. The query language model can be exploited to model user preferences, the context of a query, synonomy and word senses. While recent work has incorporated word translation models for this purpose, we introduce a new method using Markov chains defined on a set of documents to estimate the query models. The Markov chain method has connections to algorithms from link analysis and social networks. The new approach is evaluated on TREC collections and compared to the basic language modeling approach and vector space models together with query expansion using Rocchio. Significant improvements are obtained over standard query expansion methods for strong baseline TF-IDF systems, with the greatest improvements attained for short queries on Web data.

Summary

The paper presents a novel framework for information retrieval that integrates document and query models through a probabilistic ranking function grounded in Bayesian decision theory. This approach enhances the traditional language modeling methods by estimating both document and query language models, allowing for a more nuanced understanding of user preferences, context, and word semantics. A key innovation introduced is the use of Markov chains to estimate query models, which improves upon previous translation models by addressing issues related to data sparsity and context independence. The authors evaluate their methods using TREC collections, demonstrating significant performance improvements over standard query expansion techniques, particularly for short queries in web data. The framework emphasizes risk minimization in the retrieval process, allowing for a flexible and general approach to ranking documents based on their relevance to user queries. By leveraging the strengths of language modeling and incorporating user-specific knowledge, the proposed methods show promise in enhancing the effectiveness of information retrieval systems. The experiments conducted validate the efficacy of the Markov chain method for query expansion, highlighting its potential to outperform traditional models like TF-IDF and Rocchio in various retrieval scenarios.

Limitations

- **Potential Overfitting**: The models may be prone to overfitting, especially when using a limited number of documents for feedback in the Markov chain method.
- **Context-Independence of Translation Models**: The translation probabilities used in the models are context-independent, which limits their ability to handle word-sense ambiguity and contextual nuances.

A Neural Corpus Indexer for Document Retrieval

Domain: RAG

Relevance Score: 7

Abstract

Current state-of-the-art document retrieval solutions mainly follow an indexretrieve paradigm, where the index is hard to be directly optimized for the final retrieval target. In this paper, we aim to show that an end-to-end deep neural network unifying training and indexing stages can significantly improve the recall performance of traditional methods. To this end, we propose Neural Corpus Indexer (NCI), a sequence-to-sequence network that generates relevant document identifiers directly for a designated query. To optimize the recall performance of NCI, we invent a prefix-aware weight-adaptive decoder architecture, and

leverage tailored techniques including query generation, semantic document identifiers, and consistency-based regularization. Empirical studies demonstrated the superiority of NCI on two commonly used academic benchmarks, achieving +21.4% and +16.8% relative enhancement for Recall@1 on NQ320k dataset and R-Precision on TriviaQA dataset, respectively, compared to the best baseline method.

Summary

The paper presents the Neural Corpus Indexer (NCI), an innovative end-to-end deep neural network designed to enhance document retrieval performance by directly generating relevant document identifiers for specific queries. Traditional document retrieval methods often rely on separate indexing and retrieval stages, which can limit optimization for final retrieval targets. NCI addresses this limitation by employing a sequence-to-sequence architecture that integrates training and indexing, utilizing techniques such as a prefix-aware weight-adaptive decoder, query generation, and semantic document identifiers. Empirical results demonstrate that NCI significantly outperforms existing methods, achieving notable improvements in recall metrics on benchmark datasets like NQ320k and TriviaQA. The authors highlight the advantages of NCI, including its ability to capture deep interactions between queries and documents, and its potential to serve as a comprehensive solution for next-generation information retrieval systems. By optimizing the entire retrieval process within a unified framework, NCI reduces the dependency on traditional indexing methods and enhances the efficiency of document retrieval, making it a promising approach for future research and applications in the field.

Limitations

- **Model Capacity Requirements**: The current implementation of the Neural Corpus Indexer (NCI) requires a larger model capacity to effectively scale to web-scale applications.
- **Dependency on Augmented Queries**: The performance of NCI heavily relies on the quality and diversity of augmented queries generated during training.
- **Limited Generalization**: The model may struggle to generalize well to unseen queries or documents that differ significantly from the training data.

TIARA: Multi-grained Retrieval for Robust Question Answering over Large Knowledge Base

Domain: RAG

Relevance Score: 6

Abstract

Pre-trained language models (PLMs) have shown their effectiveness in multiple scenarios. However, KBQA remains challenging, especially regarding coverage and generalization settings. This is due to two main factors: i) understanding the semantics of both questions and relevant knowledge from the KB; ii) generating executable logical forms with both semantic and syntactic correctness. In this paper, we present a new KBQA model, TIARA, which addresses those issues by applying multi-grained retrieval to help the PLM focus on the most relevant KB contexts, viz., entities, exemplary logical forms, and schema items. Moreover, constrained decoding is used to control the output space and reduce generation errors. Experiments over important benchmarks demonstrate the effectiveness of our approach. TIARA outperforms previous SOTA, including those using PLMs or oracle entity annotations, by at least 4.1 and 1.1 F1 points on GrailQA and WebQuestionsSP, respectively. Specifically on GrailQA, TIARA outperforms previous models in all categories, with an improvement of 4.7 F1 points in zero-shot generalization.

Summary

The experimental results demonstrate that TIARA significantly outperforms previous state-of-the-art models on benchmark datasets like GrailQA and WebQuestionsSP, achieving improvements of at least 4.1 and 1.1 F1 points, respectively. Notably, TIARA excels in zero-shot generalization scenarios, showcasing its robustness in handling unseen queries. The paper highlights the importance of contextual retrieval and constrained decoding in enhancing the capabilities of PLMs for KBQA, ultimately contributing to a more effective and reliable system for querying large-scale knowledge bases.

Limitations

- **Retrieval Efficiency**:
  - The retrieval efficiency of the proposed method needs further optimization.
  - Logical form enumeration takes more than 7 seconds per question without caching, which may not meet practical requirements.

Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection

Domain: RAG

Relevance Score: 9

Abstract

Despite their remarkable capabilities, large language models (LLMs) often produce responses containing factual inaccuracies due to their sole reliance on the parametric knowledge they encapsulate. Retrieval-Augmented Generation (RAG), an ad hoc approach that augments LMs with retrieval of relevant knowledge, decreases such issues. However, indiscriminately retrieving and incorporating a fixed number of retrieved passages, regardless of whether retrieval is necessary, or passages are relevant, diminishes LM versatility or can lead to unhelpful response generation. We introduce a new framework called Self-Reflective Retrieval-Augmented Generation (Self-RAG) that enhances an LM's quality and factuality through retrieval and self-reflection. Our framework trains a single arbitrary LM that adaptively retrieves passages on-demand, and generates and reflects on retrieved passages and its own generations using special tokens, called reflection tokens. Generating reflection tokens makes the LM controllable during the inference phase, enabling it to tailor its behavior to diverse task requirements. Experiments show that Self-RAG (7B and 13B parameters) significantly outperforms state-of-the-art LLMs and retrieval-augmented models on a diverse set of tasks. Specifically, Self-RAG outperforms ChatGPT and retrieval-augmented Llama2-chat on Open-domain QA, reasoning and fact verification tasks, and it shows significant gains in improving factuality and citation accuracy for long-form generations relative to these models.

Summary

The paper introduces a novel framework called Self-Reflective Retrieval-Augmented Generation (Self-RAG), designed to enhance the quality and factual accuracy of large language models (LLMs) through on-demand retrieval and self-reflection. Traditional Retrieval-Augmented Generation (RAG) methods often retrieve fixed passages indiscriminately, which can lead to irrelevant or low-quality outputs. In contrast, Self-RAG employs a mechanism where the model generates special reflection tokens to determine the necessity of retrieval, evaluate the relevance of retrieved passages, and critique its own outputs. This adaptive approach allows the model to tailor its responses based on the specific requirements of the task, significantly improving factual accuracy and citation precision across various tasks, including open-domain question answering and long-form generation. Experimental results demonstrate that Self-RAG outperforms state-of-the-art LLMs and retrieval-augmented models, including ChatGPT and Llama2-chat, across multiple benchmarks. The framework not only enhances the model's ability to generate accurate and verifiable information but also allows for customizable behavior during inference, enabling users to adjust the model's focus on factual accuracy versus creativity based on the task at hand. Overall, Self-RAG represents a significant advancement in the field of LLMs, addressing the persistent issue of factual inaccuracies in generated content.

Limitations

- **Dependence on Retrieved Passages**:
  - The effectiveness of Self-RAG heavily relies on the quality and relevance of the retrieved passages.
  - If the retrieval model fails to provide relevant information, the output quality may degrade significantly.
- **Potential for Factual Inaccuracies**:
  - Despite improvements in factual accuracy, Self-RAG can still generate outputs that are not fully supported by the citations.
  - The model may produce plausible-sounding but incorrect information if the retrieved passages are misleading or incorrect.
- **Complexity of Implementation**:
  - The framework introduces additional complexity in terms of training and inference due to the integration of reflection tokens and the need for a retriever model.
  - This complexity may hinder practical deployment in real-world applications where simplicity and efficiency are crucial.
- **Generalization to New Tasks**: The ability of Self-RAG to generalize to new, unseen tasks or domains remains uncertain, particularly if those tasks differ significantly from the training data.

Precise Zero-Shot Dense Retrieval without Relevance Labels

Domain: RAG

Relevance Score: 9

Abstract

While dense retrieval has been shown effective and efficient across tasks and languages, it remains difficult to create effective fully zero-shot dense retrieval systems when no relevance label is available. In this paper, we recognize the difficulty of zero-shot learning and encoding relevance. Instead, we propose to pivot through Hypothetical Document Embeddings~(HyDE). Given a query, HyDE first zero-shot instructs an instruction-following language model (e.g. InstructGPT) to generate a hypothetical document. The document captures relevance patterns but is unreal and may contain false details. Then, an unsupervised contrastively learned encoder~(e.g. Contriever) encodes the document into an embedding vector. This vector identifies a neighborhood in the corpus embedding space, where similar real documents are retrieved based on vector similarity. This second step ground the generated document to the actual corpus, with the encoder's dense bottleneck filtering out the incorrect details. Our experiments show that HyDE significantly outperforms the state-of-the-art unsupervised

dense retriever Contriever and shows strong performance comparable to fine-tuned retrievers, across various tasks (e.g. web search, QA, fact verification) and languages~(e.g. sw, ko, ja).

Summary

The paper presents a novel approach called HyDE (Hypothetical Document Embeddings) aimed at improving zero-shot dense retrieval systems, which traditionally struggle without relevance labels. The authors propose a two-step process where an instruction-following language model, such as InstructGPT, generates a hypothetical document based on a given query. This document, although potentially containing inaccuracies or hallucinations, captures relevance patterns. Subsequently, an unsupervised contrastively learned encoder, like Contriever, encodes this hypothetical document into an embedding vector, which is then used to retrieve similar real documents from a corpus based on vector similarity. The experimental results demonstrate that HyDE significantly outperforms the state-of-the-art unsupervised dense retriever Contriever and achieves performance comparable to fine-tuned models across various tasks, including web search, question answering, and fact verification, as well as in multiple languages. The authors emphasize that their method requires no supervision and can be implemented using existing models without any modifications, making it a practical solution for emerging search tasks that lack relevance data.

Limitations

- **Dependence on Language Models**: The HyDE method relies heavily on real-time generation from large language models (LLMs), which may not be suitable for tasks requiring high throughput or low latency.
- **Potential Bias**: The generated documents may reflect biases present in the LLMs, potentially skewing the search results.

Corrective Retrieval Augmented Generation

Domain: Corrective Retrieval Augmented Generation

Relevance Score: 8.5

Abstract

Large language models (LLMs) inevitably exhibit hallucinations since the accuracy of generated texts cannot be secured solely by the parametric knowledge they encapsulate. Although retrieval-augmented generation (RAG) is a practicable complement to LLMs, it relies heavily on the relevance of retrieved documents, raising concerns about how the model behaves if retrieval goes wrong. To this end, we propose the Corrective Retrieval Augmented Generation (CRAG) to improve the robustness of generation. Specifically, a lightweight retrieval

evaluator is designed to assess the overall quality of retrieved documents for a query, returning a confidence degree based on which different knowledge retrieval actions can be triggered. Since retrieval from static and limited corpora can only return sub-optimal documents, large-scale web searches are utilized as an extension for augmenting the retrieval results. Besides, a decompose-then-recompose algorithm is designed for retrieved documents to selectively focus on key information and filter out irrelevant information in them. CRAG is plug-and-play and can be seamlessly coupled with various RAG-based approaches. Experiments on four datasets covering short- and long-form generation tasks show that CRAG can significantly improve the performance of RAG-based approaches.

Summary

The paper introduces Corrective Retrieval Augmented Generation (CRAG), a novel approach designed to enhance the robustness of large language models (LLMs) by addressing the issue of hallucinations and inaccuracies that arise from reliance on retrieved documents. CRAG incorporates a lightweight retrieval evaluator that assesses the quality of retrieved documents and triggers corrective actions based on their relevance, categorized as Correct, Incorrect, or Ambiguous. When the retrieved documents are deemed correct, they undergo a knowledge refinement process to extract essential information. Conversely, if they are incorrect, CRAG resorts to large-scale web searches for supplementary knowledge. The method is designed to be plug-and-play, allowing it to be integrated seamlessly with existing retrieval-augmented generation frameworks. Experimental results across four diverse datasets demonstrate that CRAG significantly improves the performance of standard retrieval-augmented generation (RAG) and state-of-the-art approaches like Self-RAG. The findings highlight CRAG's adaptability and generalizability in both short- and long-form generation tasks, showcasing its effectiveness in mitigating the challenges posed by inaccurate retrievals. The paper concludes by emphasizing the importance of self-correction mechanisms in enhancing the reliability of generative models while acknowledging the need for further advancements in retrieval evaluation capabilities.

Limitations

- **Dependence on External Evaluator**:
  - The CRAG framework relies on a lightweight retrieval evaluator to assess the quality of retrieved documents.
  - Fine-tuning this external evaluator is necessary, which may limit the system's scalability and adaptability.
- **Computational Overhead**:
  - Although the self-correction mechanism is designed to be lightweight, it still incurs some computational overhead.
  - The execution time may increase, especially when processing multiple retrieval actions.

Re2G: Retrieve, Rerank, Generate

Domain: RAG

Relevance Score: 7

Abstract

As demonstrated by GPT-3 and T5, transformers grow in capability as parameter spaces become larger and larger. However, for tasks that require a large amount of knowledge, non-parametric memory allows models to grow dramatically with a sub-linear increase in computational cost and GPU memory requirements. Recent models such as RAG and REALM have introduced retrieval into conditional generation. These models incorporate neural initial retrieval from a corpus of passages. We build on this line of research, proposing Re2G, which combines both neural initial retrieval and reranking into a BART-based sequence-to-sequence generation. Our reranking approach also permits merging retrieval results from sources with incomparable scores, enabling an ensemble of BM25 and neural initial retrieval. To train our system end-to-end, we introduce a novel variation of knowledge distillation to train the initial retrieval, reranker, and generation using only ground truth on the target sequence output. We find large gains in four diverse tasks: zero-shot slot filling, question answering, fact-checking, and dialog, with relative gains of 9% to 34% over the previous state-of-the-art on the KILT leaderboard. We make our code available as open source at https://github.com/IBM/kgi-slot-filling/tree/re2g.

Summary

The paper presents a novel approach called Re2G (Retrieve, Rerank, Generate), which enhances the performance of generative language models by integrating retrieval and reranking mechanisms into a BART-based sequence-to-sequence generation framework. The authors argue that while large transformer models like GPT-3 and T5 have shown impressive capabilities, they can be further improved by leveraging non-parametric memory through retrieval from a corpus of passages. Re2G combines neural initial retrieval with a reranking process that allows for the merging of results from different retrieval methods, such as BM25 and neural approaches, thereby improving the quality of the generated outputs. The system is trained end-to-end using a novel variation of knowledge distillation, which utilizes only the ground truth of the target sequence output. The experimental results demonstrate significant improvements across four diverse tasks—zero-shot slot filling, question answering, fact checking, and dialog—achieving relative gains of 9% to 34% over previous state-of-the-art models on the KILT leaderboard. The paper highlights the effectiveness of the reranking mechanism and the benefits of ensembling retrieval methods, ultimately establishing Re2G as a leading approach in knowledge-intensive natural language processing tasks. The authors have made their code available as open source to facilitate further research and development in this area.

Limitations

- **Dependence on Ground Truth Completeness**:
  - The model's performance is significantly affected by the completeness of the ground truth data.
  - Instances of ambiguity in head entities and multiple possible fillers for relations can lead to errors in output.
- **Challenges in End-to-End Training**:
  - The end-to-end training process presents challenges, particularly in ensuring that the query encoder's gradients are effectively utilized.
  - The proposed solutions to address this issue (combining scores, freezing the query encoder, and online knowledge distillation) may not universally apply across all datasets.

Active Retrieval Augmented Generation

Domain: RAG

Relevance Score: 9

Abstract

Despite the remarkable ability of large language models (LMs) to comprehend and generate language, they have a tendency to hallucinate and create factually inaccurate output. Augmenting LMs by retrieving information from external knowledge resources is one promising solution. Most existing retrieval augmented LMs employ a retrieve-and-generate setup that only retrieves information once based on the input. This is limiting, however, in more general scenarios involving generation of long texts, where continually gathering information throughout generation is essential. In this work, we provide a generalized view of active retrieval augmented generation, methods that actively decide when and what to retrieve across the course of the generation. We propose Forward-Looking Active REtrieval augmented generation (FLARE), a generic method which iteratively uses a prediction of the upcoming sentence to anticipate future content, which is then utilized as a query to retrieve relevant documents to regenerate the sentence if it contains low-confidence tokens. We test FLARE along with baselines comprehensively over 4 long-form knowledge-intensive generation tasks/datasets. FLARE achieves superior or competitive performance on all tasks, demonstrating the effectiveness of our method. Code and datasets are available at https://github.com/jzbjyb/FLARE.

Summary

The paper presents a novel approach called Forward-Looking Active Retrieval Augmented Generation (FLARE), which enhances the capabilities of large language models (LMs) by integrating an active retrieval mechanism during the text

generation process. Traditional retrieval-augmented LMs typically retrieve information only once based on the initial input, which can be limiting for long-form generation tasks that require ongoing access to relevant information. FLARE addresses this limitation by allowing the model to actively decide when and what to retrieve based on the confidence of the generated content. By predicting the upcoming sentence and using it as a query for retrieval, FLARE can gather additional information dynamically, thereby improving the accuracy and relevance of the generated text. The authors conducted comprehensive experiments across four knowledge-intensive long-form generation tasks, demonstrating that FLARE outperforms existing retrieval methods, including both single-time and multi-time retrieval baselines. The results indicate that FLARE's active retrieval strategy significantly enhances the model's performance, particularly in tasks requiring complex reasoning and information synthesis. The paper concludes by highlighting the effectiveness and generalizability of FLARE, suggesting future directions for improving active retrieval strategies and developing efficient architectures for integrating information retrieval with language generation.

Limitations

- **Increased Overheads**:
  - Interleaving generation and retrieval can increase computational overhead and costs.
  - Each retrieval requires activating the language model multiple times, which can be inefficient.
- **Performance in Specific Datasets**:
  - FLARE did not provide significant gains on certain datasets like Wizard of Wikipedia and ELI5.
  - The Wizard of Wikipedia dataset involves relatively short outputs, making multiple retrievals unnecessary.
  - ELI5 requires in-depth answers to open-ended questions, which presents challenges in grounding generation in retrieval.