66. Yu, X., Cheng, H., Liu, X., Roth, D., Gao, J.: ReEval: Automatic hallucination evaluation for retrieval-augmented large language models via transferable adversarial attacks. In: Duh, K., Gomez, H., Bethard, S. (eds.) Findings of the Association for Computational Linguistics: NAACL 2024. pp. 1333–1351. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024), `https://aclanthology.org/2024.findings-naacl.85`
67. Zhang, K., Liu, Q., Qian, H., Xiang, B., Cui, Q., Zhou, J., Chen, E.: Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **35**(1), 377–389 (2021)
68. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5773–5780 (2019)
69. Zhang, K., Zhang, K., Zhang, M., Zhao, H., Liu, Q., Wu, W., Chen, E.: Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. arXiv preprint arXiv:2203.16369 (2022)
70. Zhang, Q., Chen, S., Xu, D., Cao, Q., Chen, X., Cohn, T., Fang, M.: A Survey for Efficient Open Domain Question Answering. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 14447–14465. Association for Computational Linguistics, Toronto, Canada (Jul 2023). `https://doi.org/10.18653/v1/2023.acl-long.808`, `https://aclanthology.org/2023.acl-long.808`
71. Zhang, S., Liu, X., Liu, J., Gao, J., Duh, K., Van Durme, B.: Record: Bridging the gap between human and machine commonsense reading comprehension (Oct 2018). `https://doi.org/10.48550/ARXIV.1810.12885`
72. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating Text Generation with BERT. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), `https://openreview.net/forum?id=SkeHuCVFDr`
73. Zhang, Y., Khalifa, M., Logeswaran, L., Lee, M., Lee, H., Wang, L.: Merging Generated and Retrieved Knowledge for Open-Domain QA. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 4710–4728. Association for Computational Linguistics, Singapore (Dec 2023). `https://doi.org/10.18653/v1/2023.emnlp-main.286`, `https://aclanthology.org/2023.emnlp-main.286`
74. Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Cui, B.: Retrieval-augmented generation for ai-generated content: A survey (Feb 2024). `https://doi.org/10.48550/ARXIV.2402.19473`
75. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena (Jun 2023). `https://doi.org/10.48550/ARXIV.2306.05685`
76. Zhou, Y., Lin, X., Zhang, X., Wang, M., Jiang, G., Lu, H., Wu, Y., Zhang, K., Yang, Z., Wang, K., Sui, Y., Jia, F., Tang, Z., Zhao, Y., Zhang, H., Yang, T., Chen, W., Mao, Y., Li, Y., Bao, D., Li, Y., Liao, H., Liu, T., Liu, J., Guo, J., Zhao, X., WEI, Y., Qian, H., Liu, Q., Wang, X., Kin, W., Chan, Li, C., Li, Y., Yang, S., Yan, J., Mou, C., Han, S., Jin, W., Zhang, G., Zeng, X.: On the opportunities of green computing: A survey (Nov 2023)
77. Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., Chua, T.S.: Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. Tech. rep. (May 2021), `http://arxiv.org/abs/2101.00774`, arXiv:2101.00774 [cs] type: article

A Structure of RAG System

A.1 Retrieval Component

The retrieval component of RAG systems in Figure 1 can be categorized into three types: sparse retrieval, dense retrieval [77], and web search engine. The standard for evaluation is the output of *relevant documents* with numerical scores or rankings.

Before the introduction of neural networks, *sparse retrievals* are widely used for retrieving relative text content. Methods like TF-IDF [46] and BM25 [47] rely on keyword matching and word frequency but may miss semantically relevant documents without keyword overlap.

By leveraging deep learning models such as BERT [9], *dense retrieval* can capture the semantic meaning of texts, which allows them to find relevant documents even when keyword overlap is minimal. This is crucial for complex queries that require a contextual understanding to retrieve accurate information. With advanced fusion structure for queries and documents [28] and the more efficient implementation of K-Nearest Neighbors (KNN) [51], Approximate Nearest Neighbor (ANN) [12,25] search techniques, dense retrieval methods have become practical for large-scale use.

Web search engine employs the complex online search engine to provide relevant documents, such as Google Search [18], Bing Search [40], DuckDuckGo [13]. RAG systems can traverse the web's extensive information, potentially returning a more diverse and semantically relevant set of documents via the API of the search provider. The black box of the search engine and the expense of large-scale search are not affordable sometimes.

It is observed that dense retrieval techniques, particularly those leveraging embeddings, stand out as the preferred choice within the RAG ecosystem. These methods are frequently employed in tandem with sparse retrieval strategies, creating a hybrid approach that balances precision and breadth in information retrieval. Moreover, the adoption of sophisticated web search engines for benchmark assessment underscores their growing significance in enhancing the robustness and comprehensiveness of evaluations.

Indexing The indexing component processes and indexes document collections, such as HuggingFace datasets or Wikipedia pages. Chunking before indexing can improve retrieval by limiting similarity scores to individual chunks, as semantic embedding is less accurate for long articles, and desired content is often brief [32]. Index creation is designed for fast and efficient search. For example, the inverted index for sparse retrieval and the ANN index for dense retrieval.

Sparse Retrieval involves calculating IDF for each term and storing values in a database for quick look-up and scoring when queried.

Dense Retrieval encodes documents into dense vectors using a pre-trained language model like BERT. These vectors are then indexed using an Approximate Nearest Neighbor (ANN) search technique, like graph-based Hierarchical Navigable Small World (HNSW) or Inverted File Index (IVF) [12]. This process allows for the efficient retrieval of "closed" items by given predefined distance metrics.