# Decomposing Complex Queries for Tip-of-the-tongue Retrieval

**Kevin Lin**[♠] **Kyle Lo**[♡] **Joseph E. Gonzalez**[♠] **Dan Klein**[♠]

[♠]University of California Berkeley [♡]Allen Institute for AI

{k-lin,jegonzal,klein}@berkeley.edu kylel@allenai.org

## Abstract

When re-finding items, users who forget or are uncertain about identifying details often rely on creative strategies for expressing their information needs—*complex* queries that describe content elements (e.g., book characters or events), information beyond the document text (e.g., descriptions of book covers), or personal context (e.g., when they read a book). This retrieval setting, called *tip of the tongue* (TOT), is especially challenging for models heavily reliant on lexical and semantic overlap between query and document text.

In this work, we introduce a simple yet effective framework for handling such complex queries by decomposing the query into individual *clues*, routing those as sub-queries to specialized retrievers, and ensembling the results. This approach allows us to take advantage of off-the-shelf retrievers (e.g., CLIP for retrieving images of book covers) or incorporate retriever-specific logic (e.g., date constraints). We show that our framework incorporating query decompositions into retrievers can improve gold book recall up to 7% relative again for Recall@5 on a new collection of 14,441 real-world query-book pairs from an online community for resolving TOT inquiries.[1]

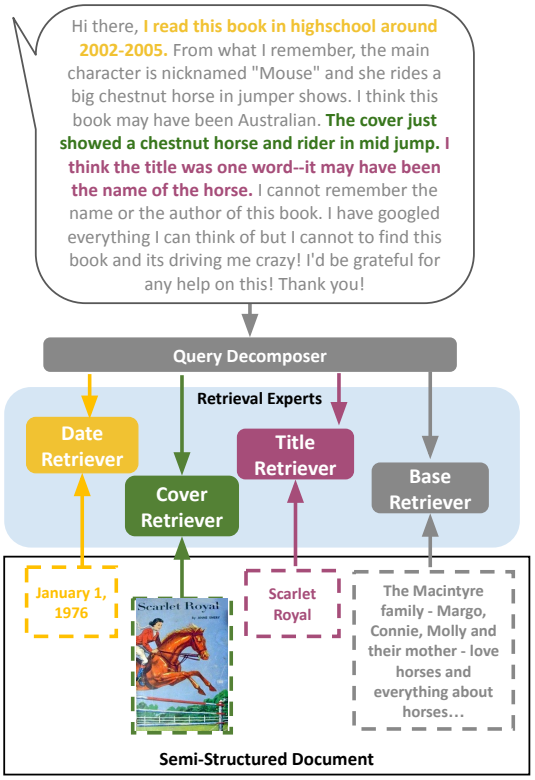



Figure 1: MOREL decomposes complex queries into subqueries routed to specific retrieval experts.

## 1 Introduction

Tip of the tongue (TOT) refers to the retrieval setting in which a user is unable to formulate a precise query that identifies a sought item, even if the user knows they've encountered this item before. For example, users searching for movies they watched or books they read long ago often resort to complex and creative queries that employ a diverse set of strategies to express information relevant to the sought item—high-level categories (e.g., topic, genre), content details from the movie or book (e.g., events, characters), references to personal context (e.g., when they last read the book), descriptions of extratextual elements (e.g., movie promotional posters, book covers), and more. In fact, in an annotation study of TOT queries for movies, Arguello et al. (2021) found over 30 types of informational facets that users may include when crafting queries. Figure 1 shows a TOT query and its corresponding gold book.

A key challenge in TOT retrieval is that queries are not just longer and more complex than those in popular retrieval datasets, but resolving them requires an enriched document collection since query-document relevance can't necessarily be established from document content alone (see Ta-

[1]Code and data at https://github.com/kl2806/whatsthatbook

ble 1). For example, in Figure 1, the query's description of the book cover—*a chestnut horse and rider in mid jump*—can be highly useful for identifying the book, but necessitates the book's representation to contain that information.

In this work, we present a simple yet effective technique for improving TOT retrieval: First, we augment document representations with additional embeddings derived from additionally linked information (images, metadata). Next, we decompose queries into individual sub-queries or *clues* that each capture a single aspect of the target document. Finally, we route these sub-queries to expert retrievers and combine their results with those from a base retriever that receives the original query. Experiments show improvement in gold book recall over description-only retrieval baselines on a set of 14,441 real-world query-book pairs collected from an online forum for resolving TOT inquiries, complete with cover images and metadata.

## 2 Method

Given a collection of documents $d_1, \ldots, d_n$ and a textual query $q$, the TOT retrieval task aims to identify the sought document $d^*$. The input (raw) documents are semi-structured; each document $d$ contains fields $d^{(1)}, \ldots, d^{(k)}$. In the case of books, the fields can correspond to a title, its description, its publication year, an image of its book cover, etc. Missing elements take on a default value (e.g., blank image, earliest publish date in overall book collection). We consider the original document text as one of these fields, which we denote $d^{(o)}$.

### 2.1 Query Decomposition

First, the query decomposer takes a query $q$ and outputs a set of subqueries $q^{(1)}, \cdots, q^{(k)}$. To do this, we use in-context learning with a large language model (LLM) to extract the part of the text from $q$ that is relevant to that field or output the string `"N/A"` if the $q$ does not contain any relevant information to the field; this is repeated for each field.

In practice, we use GPT 3.5 `gpt-3.5-turbo` few-shot prompting with in-context 8 examples. An example prompt template (for book covers) is:

```
You are a utility that extracts
text related to the cover from
a complex query

Query : { X1 }
Cover : { Y1 }

Query : { X2 }
Cover : { Y2 }

...

Query : {X'}
Cover :
```

where `X1` through `X8` are the original query examples for few-shot $q_1, \ldots, q_8$, `Y1` through `Y8` are gold sub-queries $q_1^{(j)}, \ldots, q_8^{(j)}$ (assuming $j$ is the field corresponding to book covers), and the final `X'` is the query intended for sub-query extraction.[2] Each field has its own prompt template. Sub-queries for different fields can be generated in parallel, as the they are independent of each other.

A key implementation detail is that sub-queries need not be pure extractions from the original query. Using LLMs to generate sub-queries affords us the ability to set the few-shot prompt generation targets to be *predictions*. This is important as the information in queries are rarely presented in a form amenable for matching with the corresponding document field. For example, books have publish dates, but queries will rarely mention these dates; instead, users may articulate personal context (e.g., "*I read this book in highschool around 2002-2005*"). Then to simplify the learning task for a date-focused retrieval expert, we might ask the LLM to predict a "latest possible publish date" (e.g., 2005). See Table 2 for examples of generated sub-queries.

### 2.2 Retrieval Experts

We have retriever models, or experts, that specialize to specific field types. Let $R_1, \ldots, R_k$ represent these retrievers. Retrievers can be implemented as dense, sparse, or symbolic logic.

If a retriever requires training, we run the query decomposer over all query-document pairs $(q, d)$ in the training set. This produces effectively $k$ training datasets, where each dataset is comprised of a subquery and document-field pair. For example, field $j$ would have training dataset of examples $(q^{(j)}, d^{(j)})$.

At indexing time, each document's field is indexed according to the specifications of its retriever expert. For example, if the retriever is implemented as an embedding model, then that specific field is

[2] https://platform.openai.com/docs/models/gpt-3-5

| Dataset | Query Length | Lexical Overlap |
|---|---|---|
| MSMarco (Campos et al., 2016) | 7.68 | 0.55 |
| Natural Questions (Kwiatkowski et al., 2019) | 10.35 | 0.52 |
| BioASQ (Tsatsaronis et al., 2015) | 14.82 | 0.58 |
| TREC-COVID (Roberts et al., 2020) | 15.94 | 0.41 |
| SciFact (Wadden et al., 2022) | 19.52 | 0.50 |
| HotPotQA (Yang et al., 2018b) | 22.78 | 0.45 |
| TOMT (Bhargav et al., 2022) | 136.50 | 0.25 |
| *WhatsThatBook* | 156.20 | 0.19 |

Table 1: Tip of the tongue (TOT) queries are significantly longer while also having less lexical overlap with the gold document, compared with queries in popular retrieval datasets. Query length is number of BPE (Sennrich et al., 2016) pieces, averaged across examples. Lexical overlap is fraction of whole words in query that occur in gold passage(s), averaged across examples.

converted into an embedding. On the other hand, if the retriever is a sparse model, then a sparse index would be built using just that specific field's text.

At inference time, each retriever takes a sub-query $q^{(j)}$ and retrieves a document from its associated index of fields.

In practice, for titles and the original book descriptions $x^{(o)}$, we use Contriever (Izacard et al., 2021), a state-of-the-art dense retriever.[3] For both models, we train for a total of 10,000 steps with a batch size of 16, learning rate of 1e-4. For titles, we finetune with 3,327 extracted sub-queries. For our base retriever, we use the full training set of original book descriptions.

For cover images, we use CLIP (Radford et al., 2021), a state-of-the-art retriever that can score matches between embedded images and their textual descriptions. Specifically, we finetune ViT-B/32[4] on 2,220 extracted sub-queries using cross-entropy loss with batch size of 4, learning rate of 5e-5 and weight decay of 0.2 for 10 epochs with the Adam optimizer (Kingma and Ba, 2014). We select the model with the best top 1 retrieval accuracy on a validation set.

For publish dates, we use a symbolic function that heuristically scores 0 if a book was published after the sub-query date (i.e. predicted latest publish date) and 1 otherwise. If necessary, we heuristically resolve the sub-query to a year.

### 2.3 Combining retrieved results

In this work, we restrict to a simple strategy of using a weighted sum of all $k$ retrieval scores across the $(q^{(j)}, d^{(j)})$. That is, the final score is:

$$s(q, d) = \sum_{j=1}^{n} w^{(j)} R_j(q^{(j)}, d^{(j)})$$

All documents are scored in this manner, which induces a document ranking for a given query $q$.

## 3 Datasets

We introduce the *WhatsThatBook* dataset consisting of query-book pairs collected from a public online forum on GoodReads for resolving TOT inquiries about books.[5] On this forum, users post inquiries describing their sought book and community members reply with links to books on GoodReads as proposed answers.[6] If the searcher accepts a book as the correct answer, the post is manually tagged as `SOLVED` and a link to the found book is pinned to the thread. For these solved threads, we take the original inquiry as our query $q$ and the linked book as gold $d^*$. At the end, *WhatsThatBook* contains 14,441 query-book pairs. Each query corresponds to a unique book. Finally, these books are associated with pages on GoodReads, which we used to obtain publication year metadata and images of book covers.

For the experiments in the rest of this paper, we split *WhatsThatBook* into train ($n$=11,552), validation ($n$=1,444) and test ($n$=1,445) sets. By the nature of our dataset construction, the number of queries and books is equal. We use all 14,441 books, which are gold targets with respect to some query, as our full document collection for indexing.

[3] https://huggingface.co/facebook/contriever

[4] https://huggingface.co/sentence-transformers/clip-ViT-B-32

[5] https://www.goodreads.com/group/show/185-what-s-the-name-of-that-book. We scraped data from February 2022.

[6] This is a simplification of community interactions. Threads also may include dialogue between original poster and members but this is beyond the scope of our work.

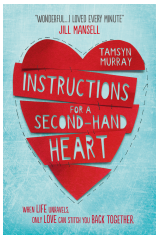
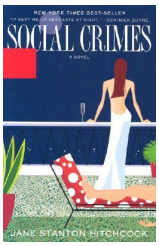
| Query-Document | Title | Date | Cover |
|---|---|---|---|
| *Query:* I think I saw this in a used store once and I remember saying to my new husband m̈y daughter use to read that book to her little brotheränd it's funny because on the outside cover is a little girl reading a book to her little brother. It's called.....my book, or my story, or something simple like that. It would be about 15 or more years old. The girl was blond and the boy brunet....I think!!!! Inside was the cutest little sentences and my kids use to do what each page said. T̈his is my nose..... T̈hese are my eyes..... Things like that. I'd love to see that book again....thank you!! | *Clue:* It's called.....my book, or my story, or something simple like that. | *Clue: 15 years old (2006 or earlier)* | *Clue:* The outside cover is a little girl reading a book to her little brother. |
| *Description: Glossy pictorial hardcover no dust jacket. 2001 7.75x9.13x25. GUIDE FOR PARENTS WITH PICTURES, HOW TO TEACHING CHILDREN READING.* | *Field:* My First Book | *Field:* First published September 1, 1984 | *Field:*   |
| *Query:* BOOK, SPOILERS: i read the book just last year (2019) around september or october. i don't think the book is older than maybe 2010. if i remember correctly the girls brother (donor of the heart) dies at a beach when he falls off a cliff during a race they had. also the boy who received the heart has a friend who's a girl and she has cancer. he really likes drawing comic strips and he always drew her as a superhero with blue hair. i believe the cover had a pink heart on it and i think it was broken with a white background and the title of the book on or above the heart. | *Clue:* The cover had a pink heart on it and i think it was broken with a white background and the title of the book on or above the heart. | *Clue:* 2019 | *Clue:* The cover had a pink heart on it and i think it was broken with a white background and the title of the book on or above the heart. |
| *Description:* Jonny knows better than anyone that life is full of cruel ironies. He's spent every day in a hospital hooked up to machines to keep his heart ticking. Then when a donor match is found for Jonny's heart, that turns out to be the cruellest irony of all. Because for Jonny's life to finally start, someone else's had to end.... | *Field:* Instructions for a Second-hand Heart | *Field:* First published December 1, 2017 | *Field:*   |
| *Query:* The books is probably 11-20 years old. Written by a former journalist. Takes place in NYC. Involves a necklace by Marie Antionette. Something like SOCIAL GRACES, or SOCIETY GRACES. I read this probably in 2000? Thank you for your help. It's a great beach read. | *Clue:* Something like SOCIAL GRACES, or SOCIETY GRACES. | *Clue:* 2000 | *Clue:* n/a |
| *Description:* When her husband of twenty years dies under mysterious circumstances, leaving his fortune–and Jo's position in society–to a mysterious French countess, Jo Slater, once one of New York's leading grande dames, comes up with an ingenious scheme to seek revenge designed to recoup her fortune and reclaim her "throne," with only a little murder standing in her way. Reprint. 75,000 first printing. | *Field:* Social Crimes | *Field:* First published June 12, 2002 | *Field:*   |

Table 2: Query-document pairs, their generated sub-queries or *clues*, and corresponding gold document fields.

## 4 Experiments

### 4.1 Baseline models

We evaluate our approach against several popular retrieval models that have been used as baselines for a range of other retrieval datsets (see Table 1). For text-only models—BM25 (Robertson and Walker, 1997; Robertson and Zaragoza, 2009), Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), and Contriever (Izacard et al., 2021)—the document representation is simply the concatenation of all available document fields into a single text field. For our image-only baseline—CLIP (Radford et al., 2021)—the document represenntation is only the embedded book cover. All baselines receive the same input (full) query. As well, all baselines are finetuned with the same hyperaparameters as described in §2.2 except using the full training set instead of just examples with a successful sub-query extraction.

### 4.2 Results

Table 3 shows the test results on *WhatsThatBook*. We use Recall@$K$ metric as our primary metric since each query has exactly one correct item.

**Baselines.** In this setting with low lexical overlap, we see that dense retrievers like DPR and Contriever outperform sparse retrievers like BM25. Without extracting clues about the book cover, using CLIP on its own is not effective, likely due to its limited context window.[7] Contriever is the overall best-performing baseline model.

**Our method.** Our approach to decompose queries and route clues to specialized retrievers improves performance (ranging from +2 to +3 Recall@$K$ across all cutoffs) over the next best baseline retriever. Looking into the limited expert ablation results, we find that incorporating titles and images that often have more precise descriptions improve the Recall@$K$ for lower values of $K$. Query decomposition improves Recall@5 for images and titles for 3% to 5% relative gain respectively. In constrast, the date retriever does not improve recall for lower values of $K$, and instead is more helpful at higher values of $K$. This may be due to the fact that the descriptions are less precise. Incorporating all dates, covers, and titles together provides further gains, indicating the the benefits from each specialized retrieval is somewhat orthogonal and adding additional expert retrievers could be helpful.

[7]We pass the full query into CLIP and allow for truncation to happen naturally. This is a big issue with CLIP, which supports a narrow query length; hence, motivating our approach to extract *clues* about book covers from the full query.

## 5 Related Work

**Dense methods for document retrieval.** Document retrieval has a long history of study in fields like machine learning, information retrieval, natural language processing, library and information sciences, and others. Recent years has seen the rise in adoption of dense, neural network-based methods, such as DPR (Karpukhin et al., 2020) and Contriever (Izacard et al., 2022), which have been shown can outperform sparse methods like BM25 (Robertson and Walker, 1997; Robertson and Zaragoza, 2009) in retrieval settings in which query-document relevance cannot solely be determined by lexical overlap. Researchers have studied these models using large datasets of query-document pairs in web, Wikipedia, and scientific literature-scale retrieval settings (Campos et al., 2016; Sciavolino et al., 2021; Roberts et al., 2020). Many retrieval datasets have adopted particular task formats such as question answering (Kwiatkowski et al., 2019; Yang et al., 2018b; Tsatsaronis et al., 2015) or claim verification (Thorne et al., 2018; Wadden et al., 2022). We direct readers to Zhao et al. (2022) for a comprehensive, up-to-date survey of methods, tasks, and datasets.

**Known-item and TOT retrieval.** Tip of the tongue (TOT) is a form of known-item retrieval (Buckland, 1979; Lee et al., 2006), a long-studied area in the library and information sciences. Yet, lack of large-scale public datasets has made development of retrieval methods for this task difficult. Prior work on known-item retrieval focused on constructing synthetic datasets (Azzopardi et al., 2007; Kim and Croft, 2009; Elsweiler et al., 2011). For example, Hagen et al. (2015) released a dataset of 2,755 query-item pairs from *Yahoo!* answers and injected query inaccuracies via hired annotators to simulate the phenomenon of *false memories* Hauff and Houben (2011); Hauff et al. (2012), a common property of TOT settings.

The emergence of large, online communities for resolving TOT queries has enabled the curation of realistic datasets. Arguello et al. (2021) categorized the types of information referred to in TOT queries

| Model | Top 5 | Top 10 | Top 20 | Top 100 |
|---|---|---|---|---|
| BM25 (Robertson and Walker, 1997) | 8.3 | 12.5 | 16.2 | 22.5 |
| DPR (Karpukhin et al., 2020) | 13.8 | 31.9 | 39.8 | 57.2 |
| CLIP (Radford et al., 2021) | 1.9 | 2.8 | 3.5 | 5.7 |
| Contriever (Izacard et al., 2022) | 26.5 | 33.5 | 40.3 | 61.3 |
| Ours (Contriever+Date Expert only) | 26.4 | 33.2 | 40.5 | 62.2 |
| Ours (Contriever+Image Expert only) | 27.2 | 34.9 | 42.0 | 61.9 |
| Ours (Contriever+Title Expert only) | 27.8 | 34.0 | 42.2 | 61.2 |
| **Ours (Contriever+All Experts)** | **28.4** | **35.5** | **43.5** | **63.1** |

Table 3: Results on the test set of *WhatsThatBook*. Metrics are Recall@$K$. The top half of models are single-retriever baselines; BM25, DPR, and Contriever all operate over book descriptions only, while CLIP operates over book covers only. All these baselines receive as input the full query. The bottom half of models make use of our query decomposition to obtain sub-queries and trained expert retrievers that operate over richer document representations.

from the website *I Remember This Movie*.[8] Most recently, Bhargav et al. (2022) collected queries from the *Tip Of The Tongue* community on Reddit[9] and evaluated BM25 and DPR baselines. Our work expands on their work in a key way: We introduce a new method for retrieval inspired by long, complex TOT queries. In order to test our method on a large dataset of TOT queries, we collected a new dataset of resolved TOT queries such that we also had access to metadata and book cover images, which were not part of Bhargav et al. (2022)'s dataset.

**Query Understanding and Decomposition.** Our work on understanding complex information-seeking queries by decomposition is related to a line of work breaking down language tasks into modular subtasks (Andreas et al., 2016). More recently, LLMs have been used for decomposing complex tasks such as multi-hop questions into a sequence of simpler subtasks (Khot et al., 2022) or smaller language steps handled by simpler models (Jhamtani et al., 2023).

Related to decomposition of long, complex queries for retrieval is literature on document similarity (Mysore et al., 2022) or query-by-document (QBD) (Yang et al., 2018a). In these works, a common approach is decomposing documents into sub-passages (e.g. sentences) and performing retrieval on those textual units. The key differentiator between these works and ours is that document similarity or QBD are inherently symmetric retrieval operations, whereas our setting requires designing approaches to handle asymmetry in available information (and thus choice of modeling approach or representation) between queries and documents. In this vein, one can also draw parallels to Lewis et al. (2021), which demonstrates that retrieving over model-generated question-answering pairs instead of their originating documents can improve retrieval, likely due to improved query-document form alignment. In a way, this is similar to our use of LLMs to generate clues that better align with extratextual document fields, though our work is focused on query-side decomposition rather than document-side enrichment.

## 6 Conclusion

We study a real-world information-seeking setting—tip of the tongue retrieval—in which users issue long, complex queries for re-finding items despite being unable to articulate identifying details about those items. We introduce a simple but effective approach to handling these complex queries that decomposes them into sub-queries or clues that are routed to expert retrievers for specialized scoring. Our simple framework allows for modular composition of different retrievers and leveraging of pretrained models for specific modalities such as CLIP for document images. We observe improvements of up to 7% relative gain for Recall@5 when incorporating query decomposition into existing retrievers on our newly-introduced *WhatsThatBook*, a large challenging dataset of real-world, tip-of-the-tongue queries for books.

## References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48. IEEE Computer Society.

Jaime Arguello, Adam Ferguson, Emery Fine, Bhaskar Mitra, Hamed Zamani, and Fernando Diaz. 2021.

[8] https://iremember thismovie.com/

[9] https://www.reddit.com/r/tipofmytongue/

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*, CHIIR '21, page 5–14, New York, NY, USA. Association for Computing Machinery.

Leif Azzopardi, Maarten de Rijke, and Krisztian Balog. 2007. Building simulated queries for known-item topics: An analysis using six european languages. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 455–462, New York, NY, USA. Association for Computing Machinery.

Samarth Bhargav, Georgios Sidiropoulos, and Evangelos Kanoulas. 2022. 'it's on the tip of my tongue': A new dataset for known-item retrieval. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 48–56, New York, NY, USA. Association for Computing Machinery.

Michael K. Buckland. 1979. On types of search and the allocation of library resources. *Journal of the American Society for Information Science*, 30(3):143–147.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268.

David Elsweiler, David E. Losada, José C. Toucedo, and Ronald T. Fernandez. 2011. Seeding simulated queries with user-study data for personal search evaluation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 25–34, New York, NY, USA. Association for Computing Machinery.

Matthias Hagen, Daniel Wägner, and Benno Stein. 2015. A corpus of realistic known-item topics with associated web pages in the clueweb09. In *Advances in Information Retrieval*, pages 513–525, Cham. Springer International Publishing.

Claudia Hauff, Matthias Hagen, Anna Beyer, and Benno Stein. 2012. Towards realistic known-item topics for the clueweb. In *Proceedings of the 4th Information Interaction in Context Symposium*, IIIX '12, page 274–277, New York, NY, USA. Association for Computing Machinery.

Claudia Hauff and Geert-Jan Houben. 2011. Cognitive processes in query generation. In *Advances in Information Retrieval Theory*, pages 176–187, Berlin, Heidelberg. Springer Berlin Heidelberg.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Harsh Jhamtani, Hao Fang, Patrick Xia, Eran Levy, Jacob Andreas, and Ben Van Durme. 2023. Natural language decomposition and interpretation of complex utterances. *arXiv preprint arXiv:2305.08677*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Jinyoung Kim and W. Bruce Croft. 2009. Retrieval experiments using pseudo-desktop collections. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1297–1306, New York, NY, USA. Association for Computing Machinery.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jin Ha Lee, Allen Renear, and Linda C. Smith. 2006. Known-item search: Variations on a concept. *Proceedings of the American Society for Information Science and Technology*, 43(1):1–17.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Sheshera Mysore, Arman Cohan, and Tom Hope. 2022. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4453–4470, Seattle, United States. Association for Computational Linguistics.

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*, pages 8748–8763. PMLR.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen Voorhees, Lucy Lu Wang, and William R Hersh. 2020. TREC-COVID: rationale and structure of an information retrieval shared task for COVID-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436.

S. E. Robertson and S. Walker. 1997. On relevance weights with little relevance information. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '97, page 16–24, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artières, Axel-Cyrille Ngonga Ngomo, Norman Heino, Éric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. SciFact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Eugene Yang, David D. Lewis, Ophir Frieder, David A. Grossman, and Roman Yurchak. 2018a. Retrieval and richness when querying by document. In *Biennial Conference on Design of Experimental Search & Information Retrieval Systems*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey. *ArXiv*, abs/2211.14876.