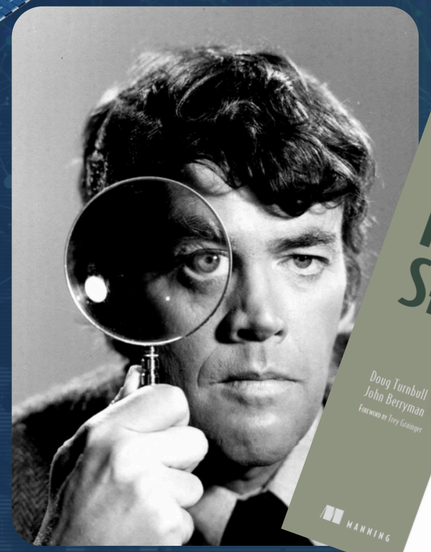
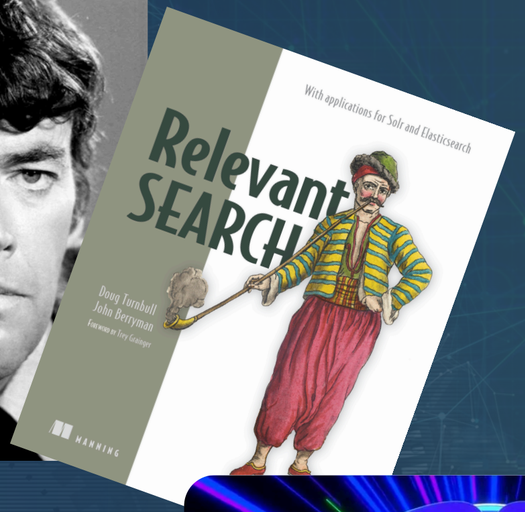
Arcturus Labs

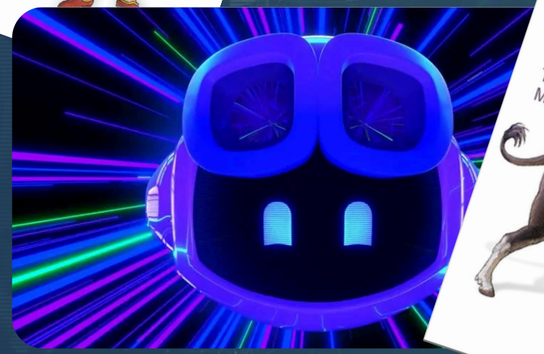
Lexical Love: Rediscovering the Power of Lexical Search in RAG

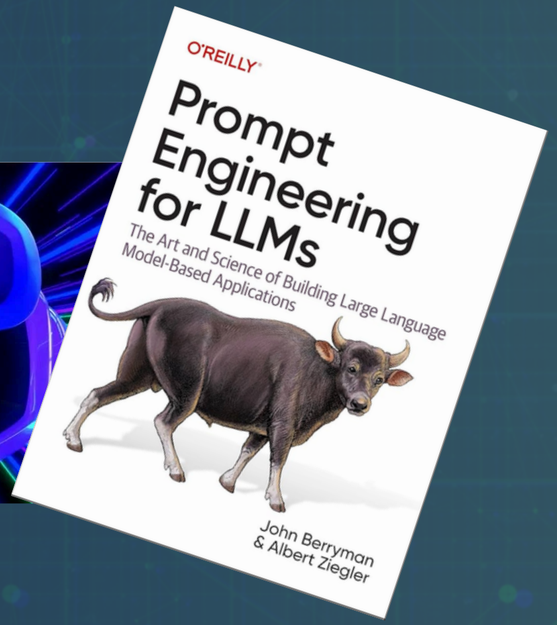
Arcturus Labs

Hi, I'm John Berryman









Arcturus Labs

www.arcturus-labs.com

Caveat:

Lexical search is my hammer, and the world is my nail.

Arcturus Labs

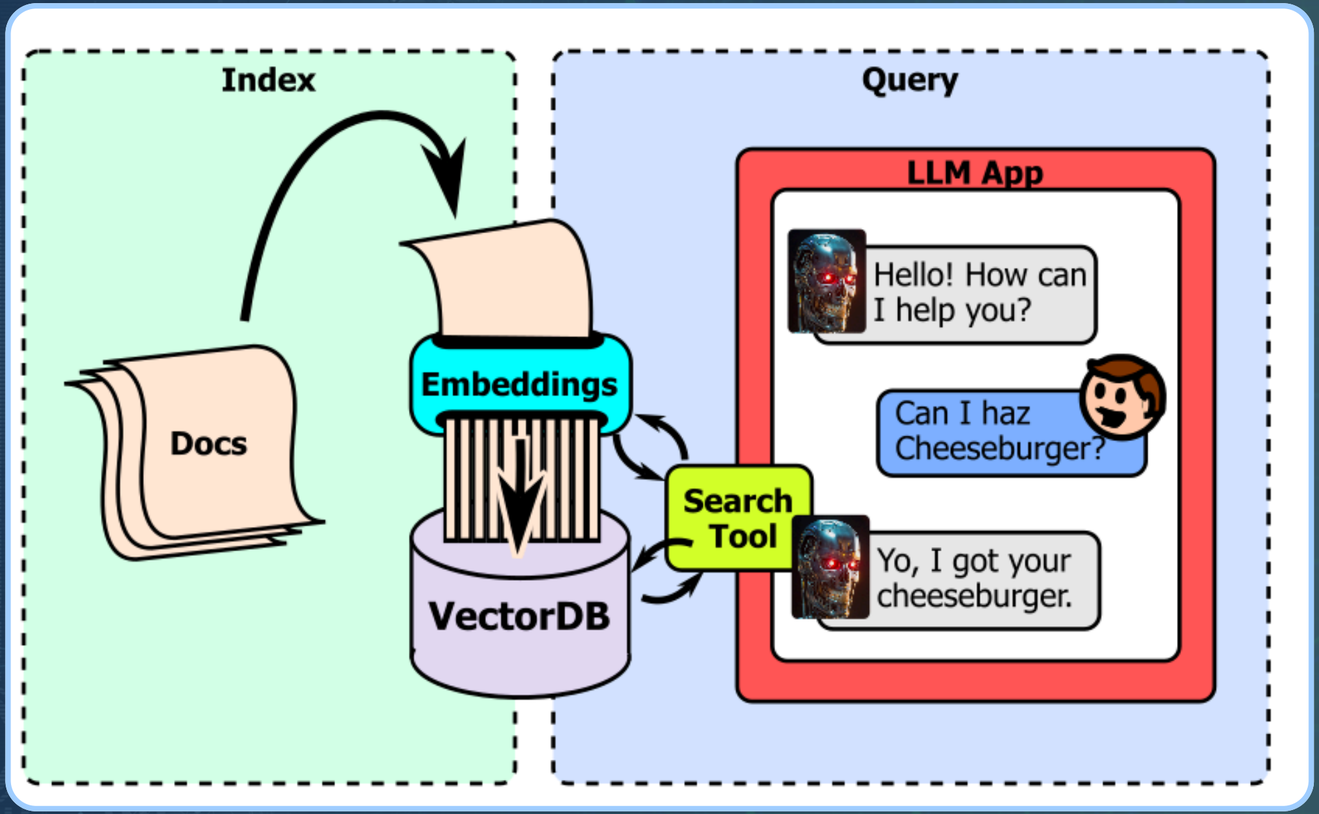
Semantic Search – “It’s how you do RAG”

“It’s easy”

- Chunk the documents
- Use BERT-like model to embed as vectors
- Store in a vector store
- At query time you embed the query and retrieve the nearest docs

“It’s cool”

Unlike “old fashioned” lexical search which uses exact token matching, Semantic Search matches based on *meaning!*

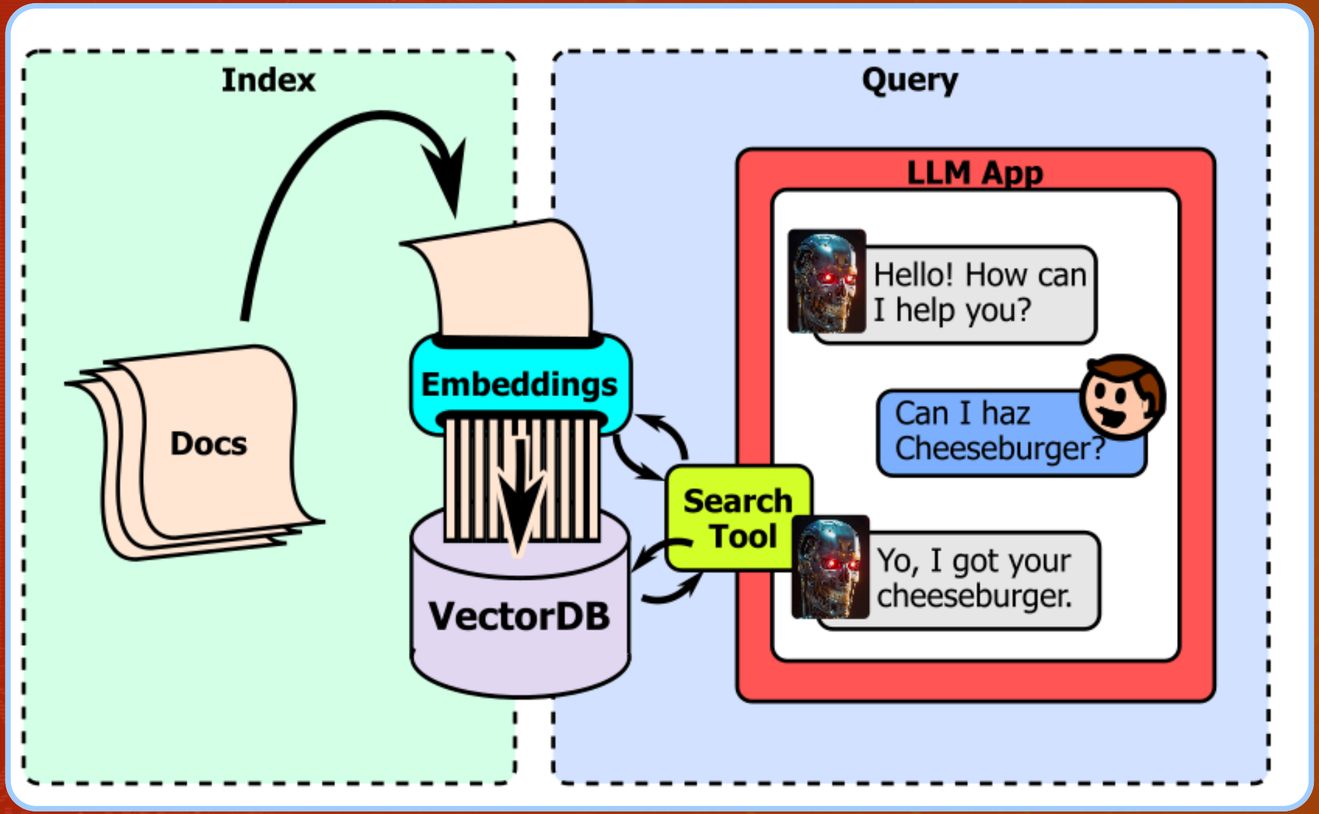


Arcturus Labs

Semantic Search – “It’s how you do RAG”

BUT *there are challenges*

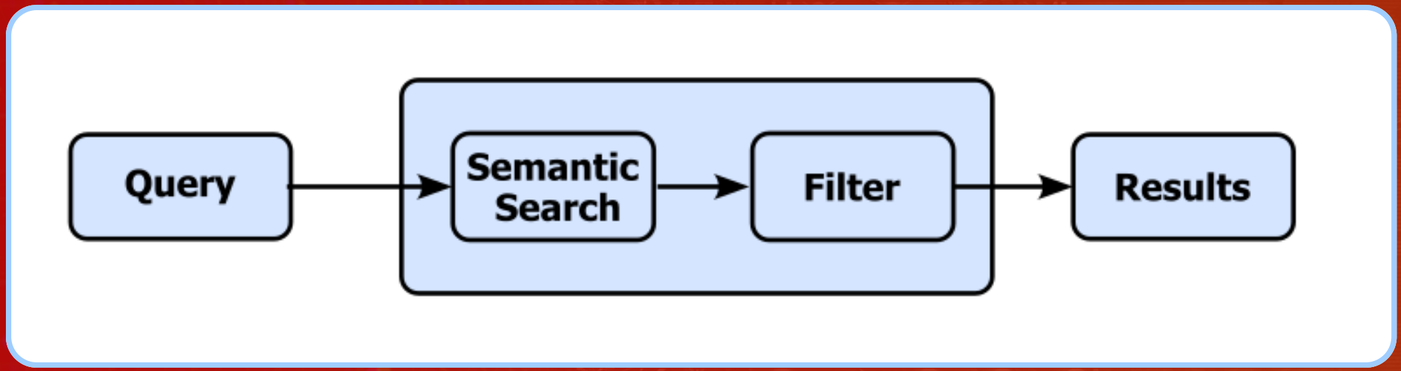
- Impossible to find
  - Exact term matches (e.g. ids, people names)
  - Phrase matches
  - New jargon introduced since training
- Fixing relevance problems is VERY involved
- Filtering is clunky - hard to “slice and dice” data set



Arcturus Labs

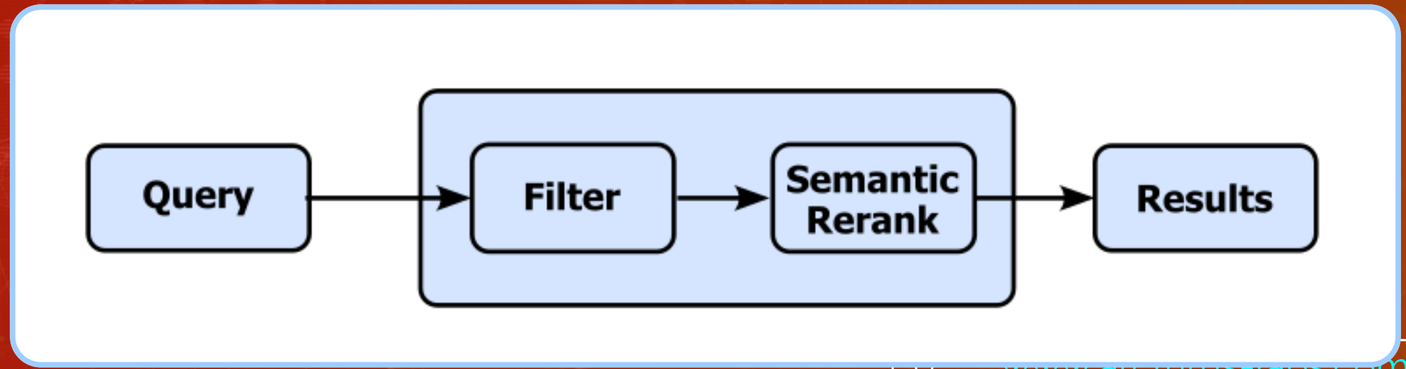
Semantic Search – “It’s how you do RAG”

How *do* you filter?



your filter might remove all the matches

reranking is brute force



IN THIS TALK:

- “Body slam” introduction to lexical search
- Example with RAG underscoring benefits
- Discuss the problems
- Point toward a hybrid search

Arcturus Labs

(really friggin' fast)

Let's Learn Lexical Search!

- Indexing
- Searching
- Results

www.arcturus-labs.com

Arcturus Labs

Lexical Indexing

Data Types

Numbers:

- bool
- int
- float
- date/time

Strings:

- keywords - indivisible strings
- text - will be processed into array of tokens

the star of the show

Etc:

- Geo points
- Geo shapes
- Intervals of time
- compound types

Arcturus Labs

Lexical Indexing

Analysis – converts a string into an array of normalized tokens

The quick brown fox jumps over the lazy dog.

character filtering

the quick brown fox jumps over the lazy dog.

tokenization

[the] [quick] [brown] [fox] [jumps] [over] [the] [lazy] [dog]

stopwording

[_] [quick] [brown] [fox] [jumps] [over] [_] [lazy] [dog]

stemming

[quick] [brown] [fox] [jump] [over] [lazi] [dog]

Arcturus Labs

Lexical Indexing

Inverted Index

Document 12:	[he] [was] [quick]
Document 43:	[quick] [brown] [fox] [jump] [over] [lazi] [dog]
Document 88:	[slow] [red] [fox] [ran] [under] [porch]

Arcturus Labs

Lexical Indexing

Inverted Index

Document 12: [he] [was] [quick]

Document 43: [quick] [brown] [fox] [jump] [over] [lazi] [dog]

Document 88: [slow] [red] [fox] [ran] [under] [porch]

the Inverted Index

[quick] → [3, 12, **43**, 67, 81, 92]
[brown] → [7, 15, **43**, 56, 78, 84, 97]
[fox] → [2, 18, 29, **43**, 88]
[jump] → [5, 21, 34, **43**, 59, 90]
[over] → [9, 22, 35, **43**, 51, 66, 73, 95]
[lazi] → [1, 19, 27, **43**, 99]
[dog] → [4, 20, 31, **43**, 50, 61, 72, 87, 100]

Arcturus Labs

Lexical Indexing

Differences

Semantic	Lexical
docs are pre-chunked	no chunking
index is often larger than text	index is often smaller than text
data is opaque (vectors)	data is transparent (tokens)

Arcturus Labs

Lexical Search

Fast Searching

the Inverted Index

quick	→	[3, 12, 43, 67, 81, 92]
brown	→	[7, 15, 43, 56, 78, 84, 97]
fox	→	[2, 18, 29, 43, 88]
jump	→	[5, 21, 34, 43, 59, 90]
over	→	[9, 22, 35, 43, 51, 66, 73, 95]
lazi	→	[1, 19, 27, 43, 99]
dog	→	[4, 20, 31, 43, 50, 61, 72, 87, 100]

Find all "fox" documents

"brown" and "fox" documents

"fox" or "dog" documents

Arcturus Labs

Lexical Search

Relevant Searching

the Inverted Index

quick → [3, 12, 43, 67, 81, 92]
brown → [7, 15, 43, 56, 78, 84, 97]
fox → [2, 18, 29, 43, 88]
jump → [5, 21, 34, 43, 59, 90]
over → [9, 22, 35, 43, 51, 66, 73, 95]
lazi → [1, 19, 27, 43, 99]
dog → [4, 20, 31, 43, 50, 61, 72, 87, 100]

"brown" and "fox" documents

- Matching docs are scored based on ***TF*IDF****
- It means:

$$\frac{\text{Term Frequency}}{\text{Doc Frequency}}$$

- Which *really* means

$$\frac{\text{num times the word appears in the doc}}{\text{num docs where the word appears}}$$

- With multi-field search, you can apply boosts and filtering

*more accurately it's BM25

Arcturus Labs

Lexical Searching

Differences

Semantic	Lexical
Searches by approx. nearest neighbor	Searches by matching tokens
Search and filter in different steps	Search by all fields simultaneously
Relevancy is distance (and that's it)	Can debug and apply boosts w/o reindexing

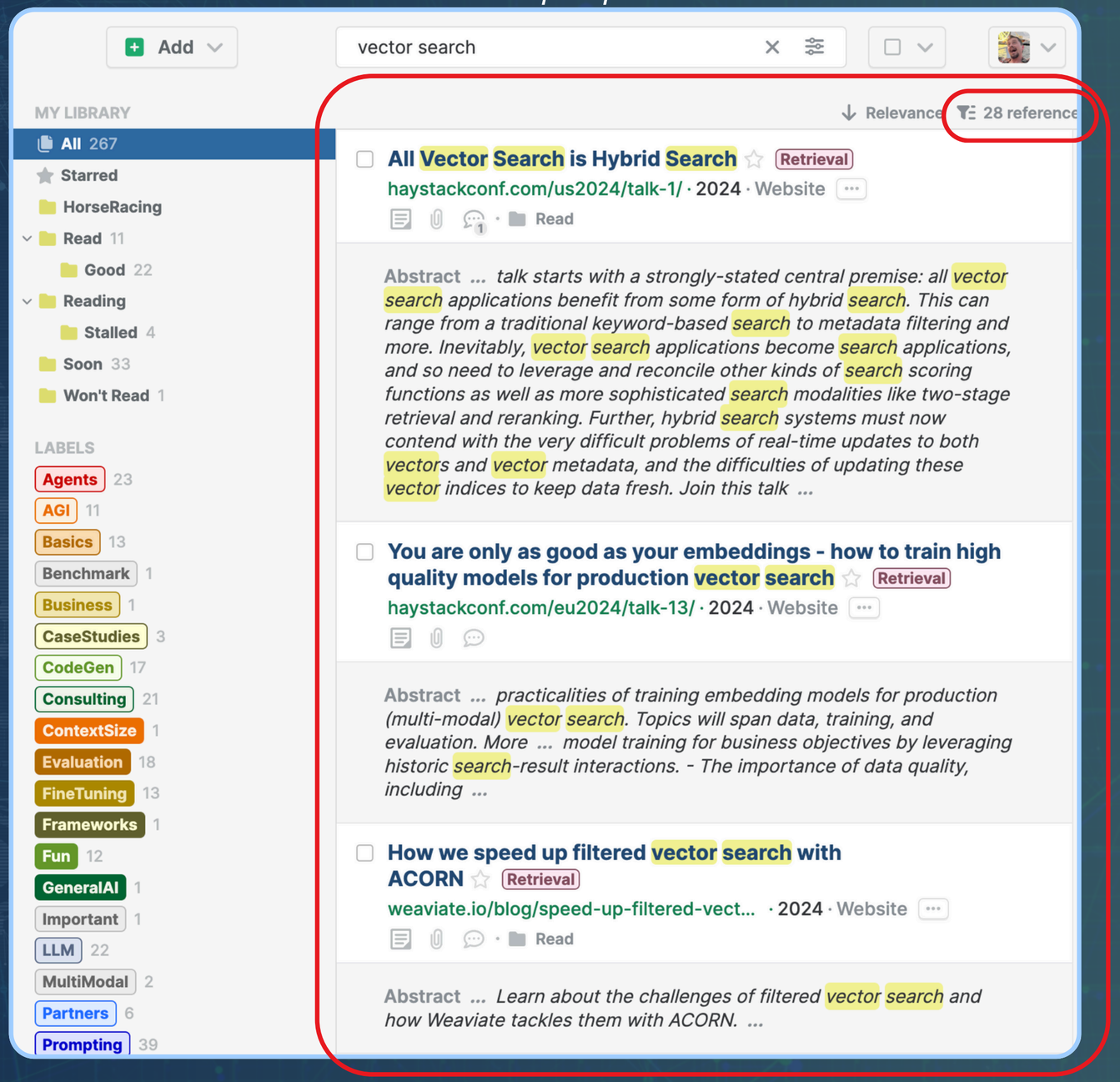
Arcturus Labs

Lexical Results

Result Set

- How many matching documents are in the set
- A list of the top N documents

Paperpile



Arcturus Labs

Lexical Results

Result Set

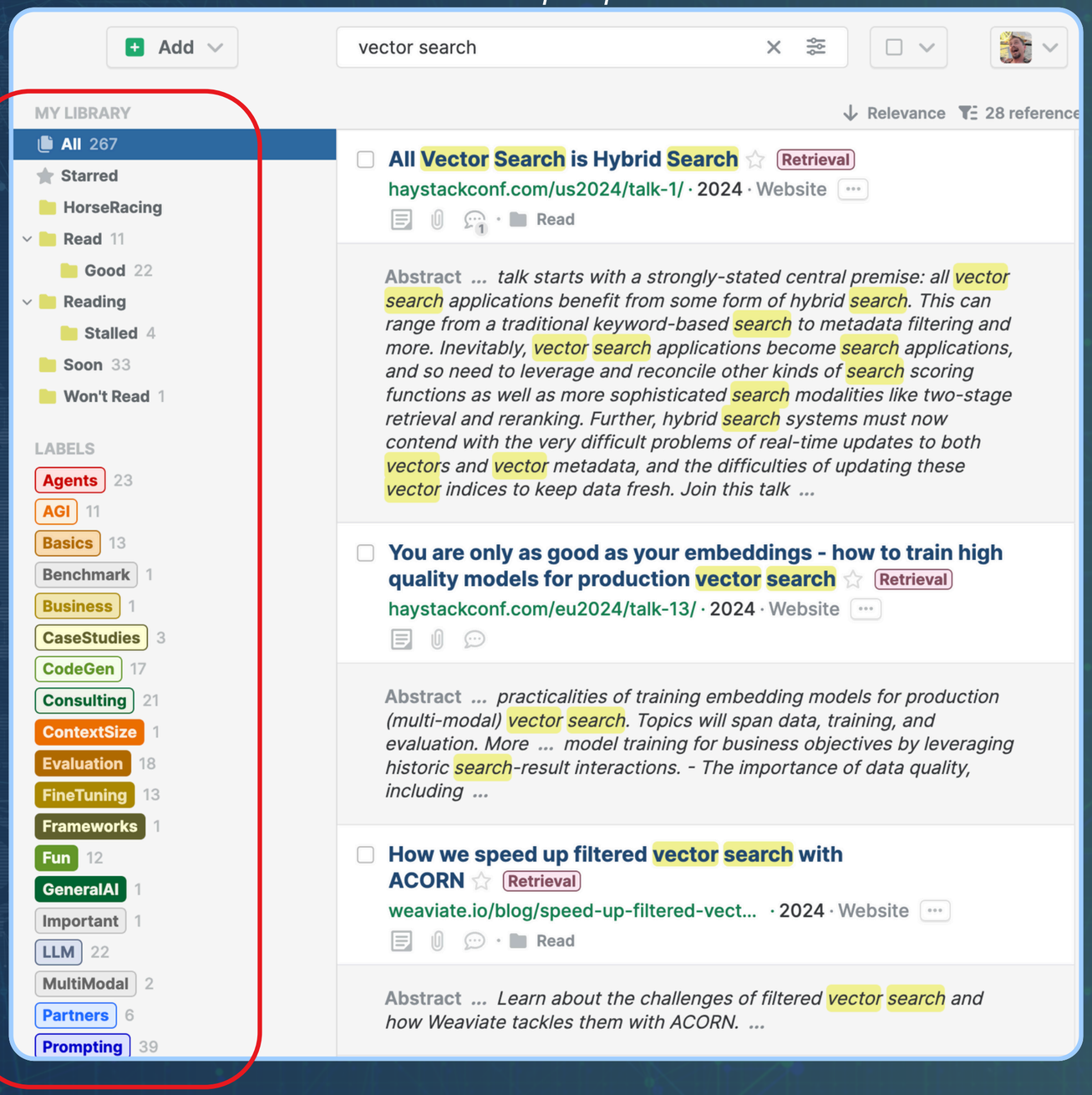
- How many matching documents are in the set
- A list of the top N documents

Aggregate Analytics

Can summarize fields of all matching docs (not just the top N)

- average, max, min
- histograms, cardinality
- facet counts

Paperpile



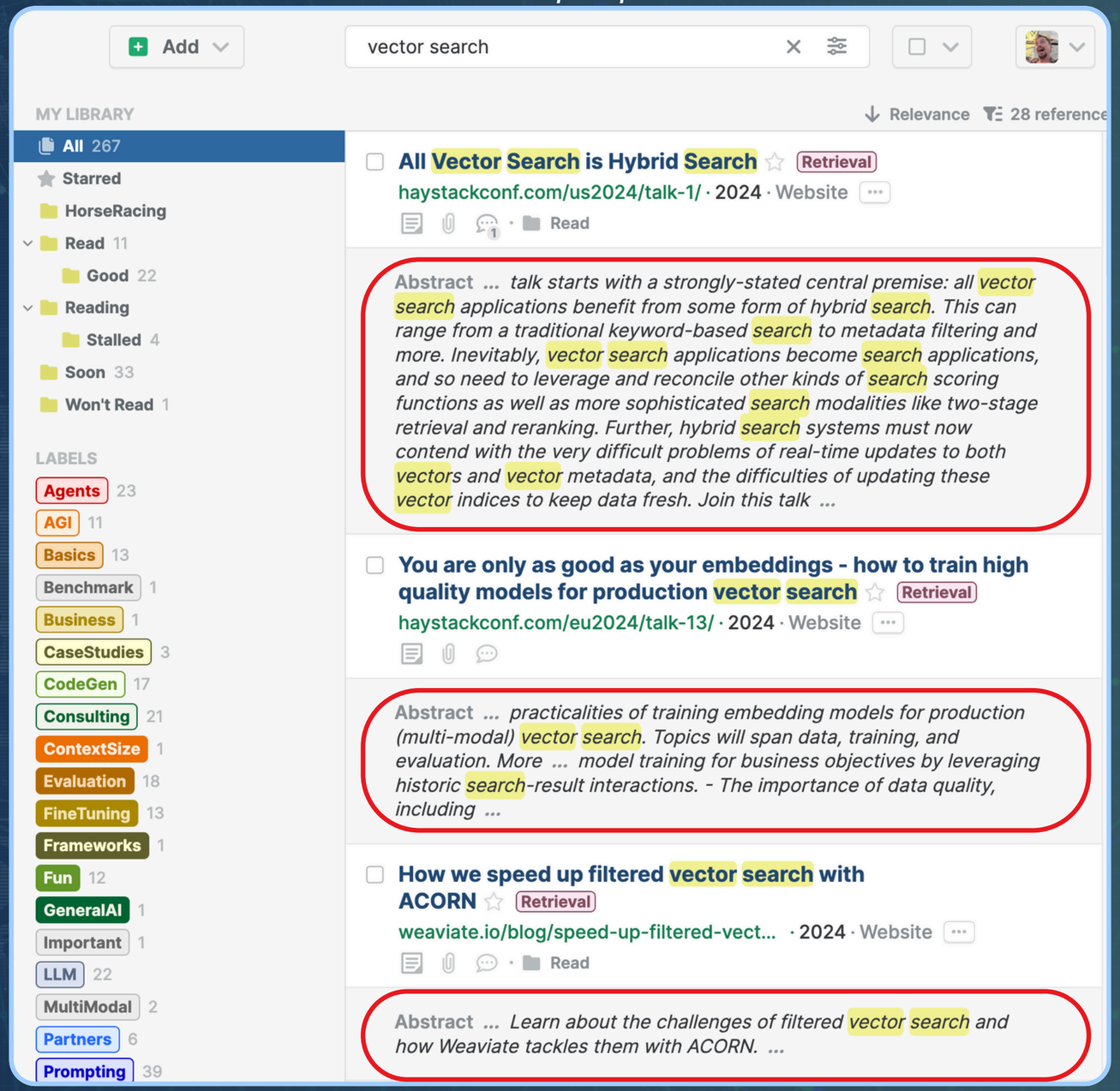
Arcturus Labs

Lexical Results

Snippets

- Segments of text that contain the interesting phrases in context.
- It's like search-time chunking.

Paperpile



Arcturus Labs

Lexical Results

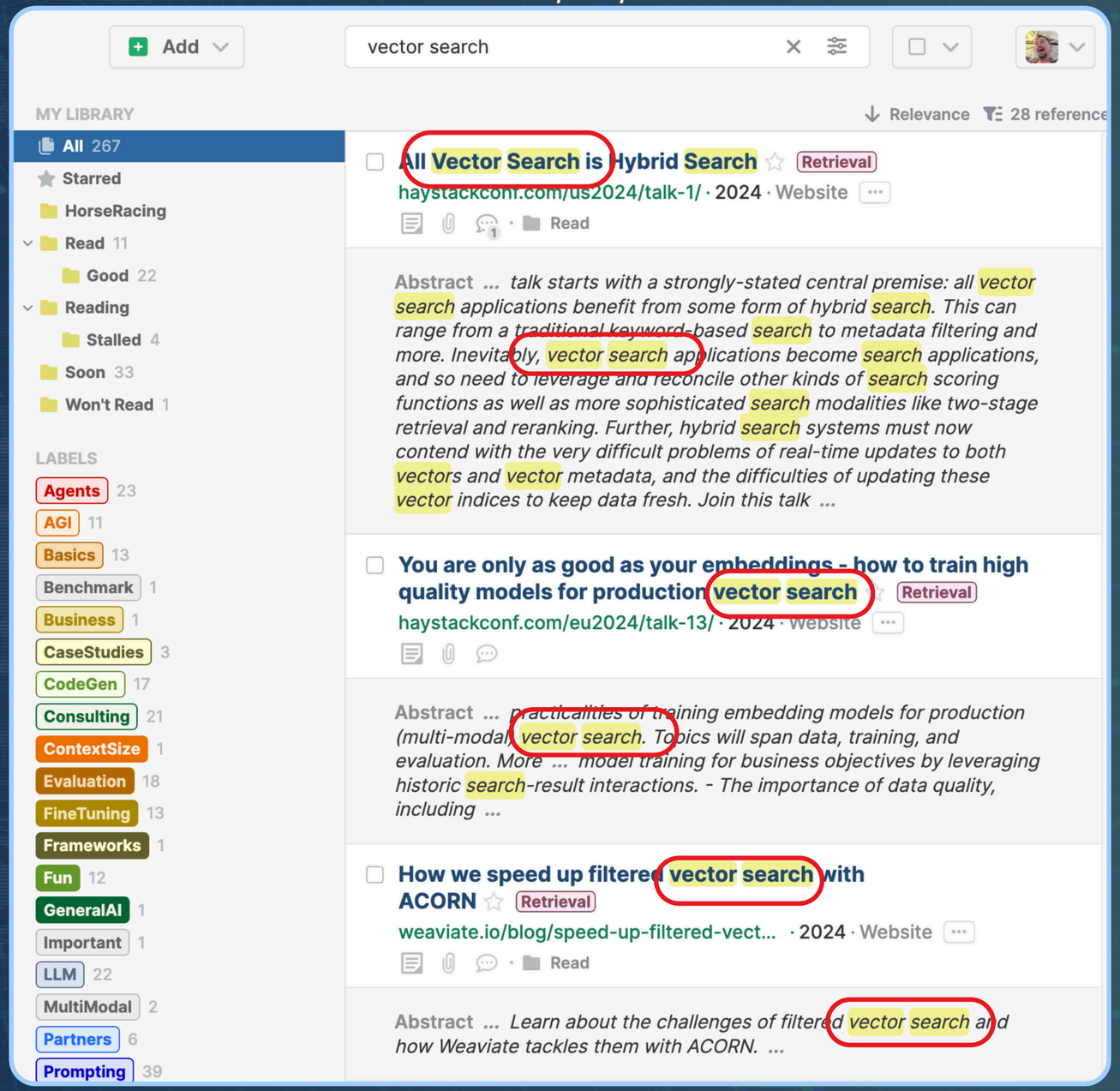
Snippets

- Segments of text that contain the interesting phrases in context.
- It's like search-time chunking.

Highlights

- Search text highlighted in results
- Mostly for human consumption.

Paperpile



Arcturus Labs

Lexical Results

Differences

Semantic	Lexical
The chunks and their metadata	The full doc (all fields)
	Snippets
No Facets	Facets Aggregate Data

Maybe good for RAG

Watch us apply this to RAG!

Arcturus Labs

Supercharging RAG with Lexical Search

www.arcturus-labs.com

Arcturus Labs

Indexing Docs

WANDS (Wayfair ANnotation Dataset)

- E-commerce product dataset
  - Home items
  - Furniture
  - Appliances
- Used to benchmark search relevance algorithms.
- 42,994 items.
- Fields:
  - product_name
  - product_class
  - product_description
  - rating_count

I added "availability" which lists the states where the product is available

Elasticsearch Mapping

```
mapping = {
    "mappings": {
        "properties": {
            "product_id": {"type": "keyword"},
            "product_name": {
                "type": "text",
                "analyzer": "english",
                "fields": {
                    "exact": {
                        "type": "text",
                        "analyzer": "standard"
                    }
                }
            },
            "product_class": {"type": "keyword"},
            "product_description": {
                "type": "text",
                "analyzer": "english",
                "fields": {
                    "exact": {
                        "type": "text",
                        "analyzer": "standard"
                    }
                }
            },
            "rating_count": {"type": "integer"},
            "average_rating": {"type": "float"},
            "availability": {"type": "keyword"},
        }
    }
}
```

Arcturus Labs

Searching

Search query

```
def high
"query": {
    "bool": {
        "shou
            {
            }
            {
            }
        ],
        "must
            {
            }
        ],
```

```
if availability:
    search_query["query"]["bool"]["filter"].append(
        {
            "term": {
                "availability": availability
            }
        }
    )
if product_class:
    search_query["query"]["bool"]["filter"].append(
        {
            "term": {
                "product_class": product_class
            }
        }
    )
if min_average_rating:
    search_query["query"]["bool"]["filter"].append(
        {
            "range": {
                "average_rating": {"gte": min_average_rating}
            }
        }
    )

search_query["size"] = num_results
```

Search results as text

```
for laptops or tablets . the space-saving design lets you put
Average Rating: 5.0
---
Product ID: 37239
Product Name: anti-fatigue comfort floor mat kitchen mat
Product Class: ['Kitchen Mats']
Product Description: prop a foot up , take a wide stance , ma
your feet ! this contoured , not flat anti fatigue mat provid
standing desk mat is engineered from the ground up to be the
anti fatigue mat ? the patented active standing mat has a con
excellent cushioning comfort that you can feel when standing
Average Rating: 4.0
---
Product ID: 9458
Product Name: standing desk converter 100 % natural bamboo ad
```

www.arcturus-labs.com

Arcturus Labs

Formatted Response

Facets definition

```
"aggs": {
    "product_class": {
        "terms": {
            "field": "product_class",
            "size": 10
        }
    }
}
```

Facet counts as text

```
Facet Counts:

product_class:
  Desks: 553
  TV Stands & Entertainment Centers: 273
  Kitchen Mats: 128
  Plant & Telephone Tables: 120
  Office Chairs: 108
  Area Rugs: 90
  Bathroom Storage: 79
  Toilet Paper Holders: 59
  Patio Umbrella Stands & Bases: 54
```

Arcturus Labs

LLM Search Tool

Definition

```
tools = [{
    "type": "function",
    "function": {
        "name": "search_catalog",
        "description": "Search for products in the catalog using various filters.
        Sometimes the results will be an imperfect match for the query. If you feel that
        the results can be improved, you should refine the query by adding a
        product_class filter or by modifying the query string to use different search
        terms.",
        "parameters": {
            "type": "object",
            "properties": {
                "query_string": {
                    "type": "string",
                    "description": "The search query to match against product names and
                    descriptions"
                },
                "product_class": {
                    "type": "string",
                    "description": "Filter results by product class. It is important to
                    use exact string matches from the product_class list, so only use
                    this after making a preliminary query_string-only search and
                    reviewing the product_class facet.",
                    "optional": True
                },
                "min_average_rating": {
                    "type": "number",
                    "description": "Filter results by minimum average rating - this
                    should be a number between 0 and 5",
                    "optional": True
                },
            },
```

query_string

product_class

min_average_rating

Implementation

```
from functools import partial

tool_lookup = {
    "search_catalog": partial(
        high_level_search,
        availability=user.state,
    )
}
```

www.arcturus-labs.com

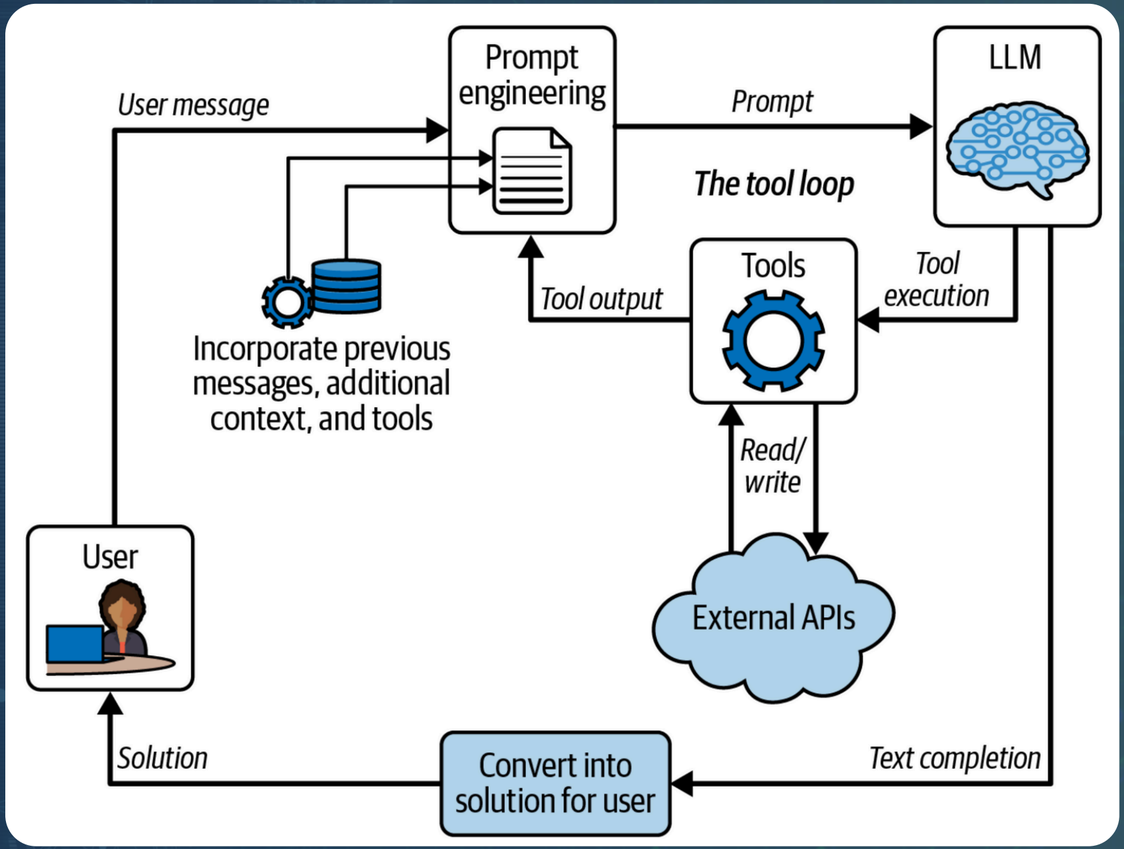
Arcturus Labs

Application Assembled

```
system = """
You are a helpful assistant that can the user find products from the catalog.

The user will discuss what they are looking for and it is your job to research
the catalog and find the best matches. Research follows these steps:
1. Make a preliminary search based on whatever the user says they want.
2. Review the results in order to get a sense of what is available. Pay special
attention to the product_classes and counts that are available.
3. Prior to answering the user, make additional refined searches based on what
you learned from the results of the preliminary search. If the results contain
irrelevant items mixed in, then consider adding a product_class filter to narrow
the scope.
4. Finally, report back to the user about all that you've discovered.

When reporting the results follow these steps:
1. Start with a quick summary of the relevant results (across all searches) that
is addresses how they will help the user based upon the context of the
conversation.
2. If it makes sense, describe the natural grouping of the results. Then you
should present the top most relevant results sorted by relevance. Make sure to
manually filter out results that you deem irrelevant.
3. At the end, make recommendations for further research that you can do to help
the user find what they are looking for.
"""
```

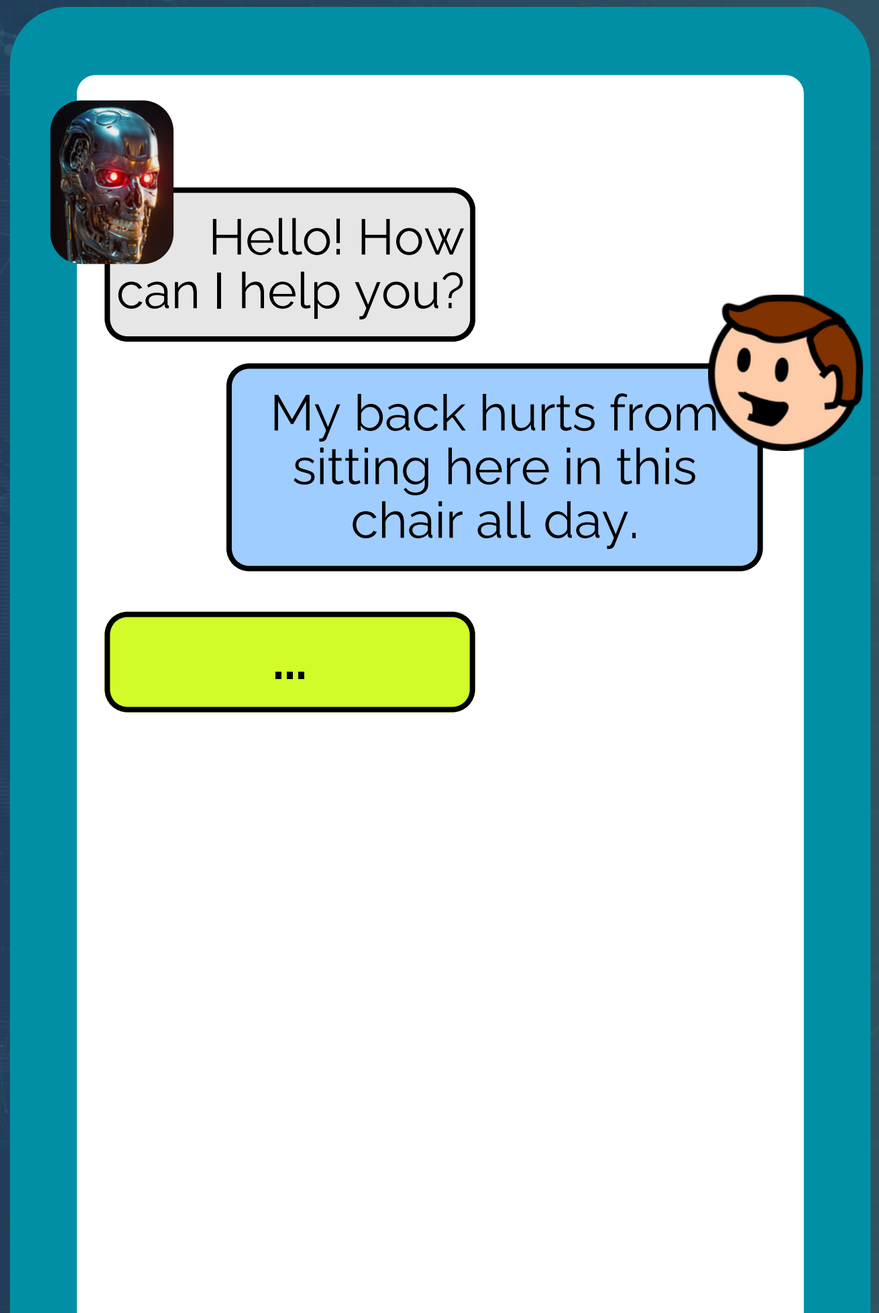


Arcturus Labs

Lexical RAG in Action

Cool things:

- Search is automatically pre-filtered to items available in this state.



Arcturus Labs

Lexical RAG in Action

Cool things:

- Search is automatically pre-filtered to items available in this state.

Hello! How can I help you?

My back hurts from sitting here in this chair all day.

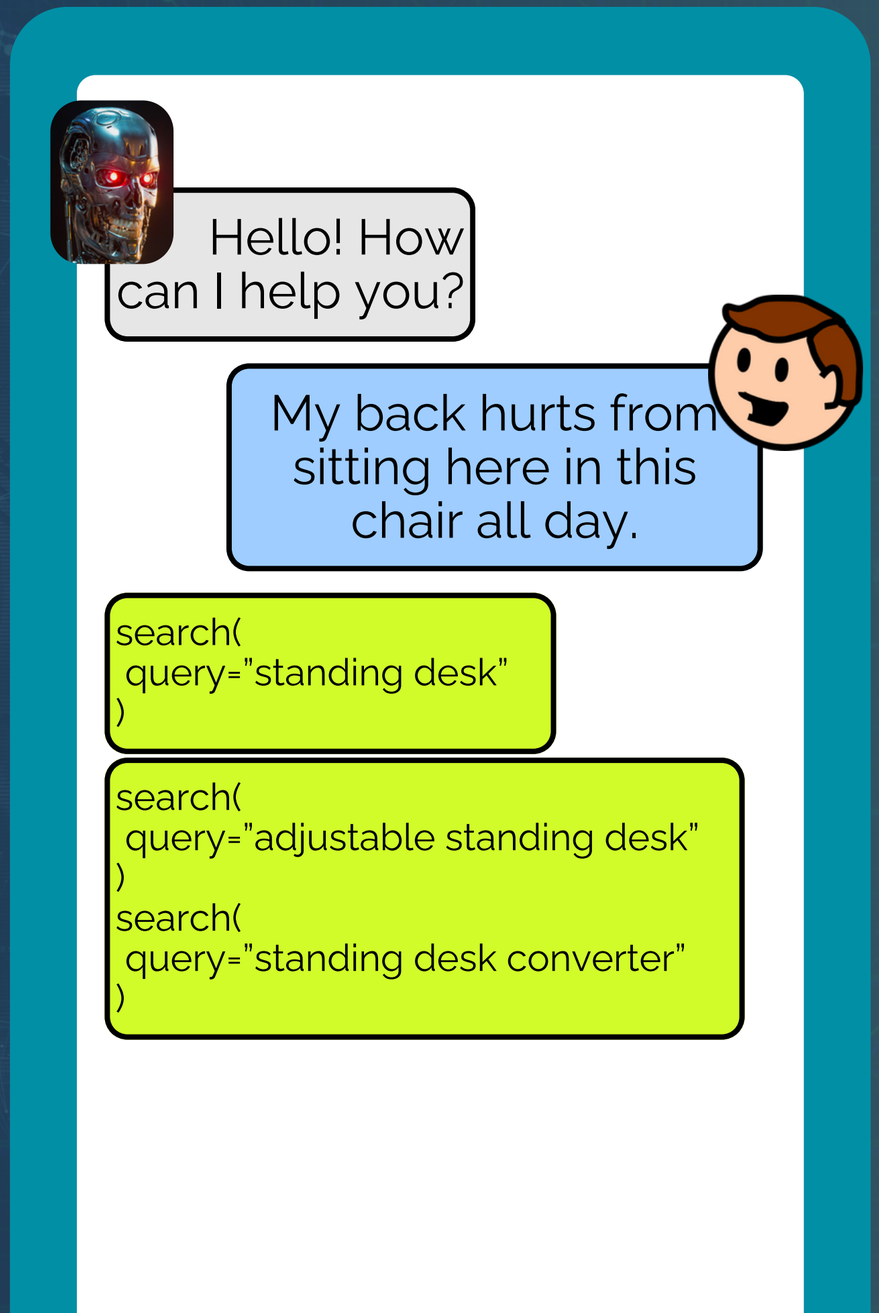
```
search(
 query="ergonomic chair"
)
search(
 query="standing desk"
)
```

Arcturus Labs

Lexical RAG in Action

Cool things:

- Search is automatically pre-filtered to items available in this state.



Arcturus Labs

Lexical RAG in Action

Cool things:

- Search is automatically pre-filtered to items available in this state.
- The LLM can see the facets.

```
product_class:
  Desks: 553
  TV Stands & Entertainment Cente
  Kitchen Mats: 128
  Plant & Telephone Tables: 120
  Office Chairs: 108
  Area Rugs: 90
  Bathroom Storage: 79
  Toilet Paper Holders: 59
  Patio Umbrella Stands & Bases:
```

Hello! How can I help you?

My back hurts from sitting here in this chair all day.

```
search(
 query="standing desk"
)
```

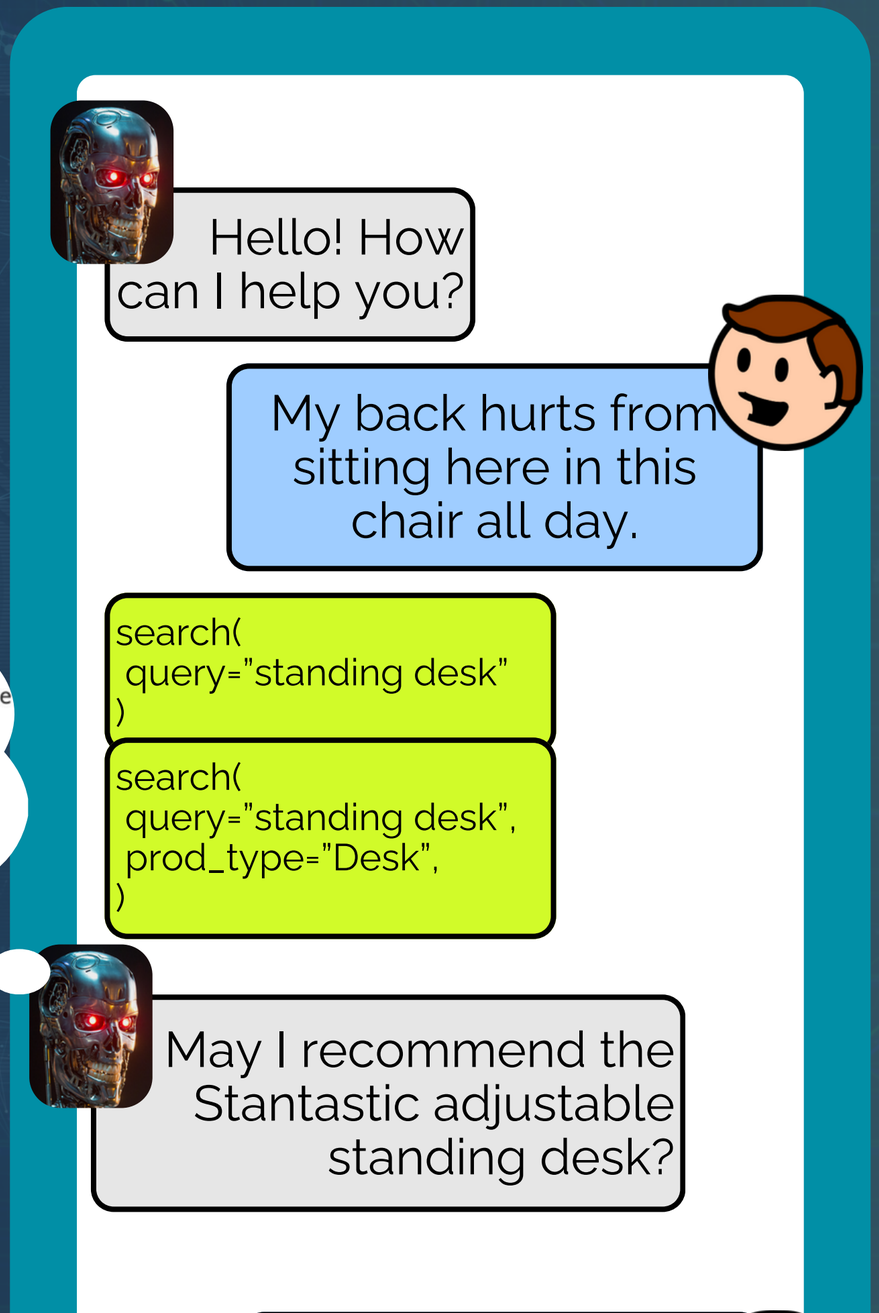
Arcturus Labs

Lexical RAG in Action

Cool things:

- Search is automatically pre-filtered to items available in this state.
- The LLM can see the facets.
- The LLM can use what it learned to filter search.

```
product_class:
  Desks: 553
  TV Stands & Entertainment Cente
  Kitchen Mats: 128
  Plant & Telephone Tables: 120
  Office Chairs: 108
  Area Rugs: 90
  Bathroom Storage: 79
  Toilet Paper Holders: 59
  Patio Umbrella Stands & Bases:
```



Arcturus Labs

Lexical RAG in Action

Cool things:

- Search is automatically pre-filtered to items available in this state.
- The LLM can see the facets.
- The LLM can use what it learned to filter search.
- The LLM can also add filters based on the user's interactions. Filters can be anything!

In principle you can even modify the relevance score.

I *prefer* lower-priced desks.

can I help you?

My back hurts from sitting here in this chair all day.

```
search(
 query="standing desk"
)
```

```
search(
 query="standing desk",
 prod_type="Desk",
)
```

May I recommend the Stantastic adjustable standing desk?

I only want the best rated desks.

```
search(
 query="standing desk",
 prod_type="Desk",
 min_rating=4.5,
)
```

Arcturus Labs

Limitations of Lexical Search and Hybrid Search to the Rescue?

www.arcturus-labs.com

Arcturus Labs

Limitations of Lexical Search

I know

You know

We know

Arcturus Labs

Limitations of Lexical Search

This is what Lexical Search thinks

Lexical Search

- Uses Bag of Words + Phrase Matching
- Doesn't get synonyms
- Doesn't get homonyms
- Is clueless about negation
- Misses context clues that modify intent

... the very things that Semantic Search excels at.

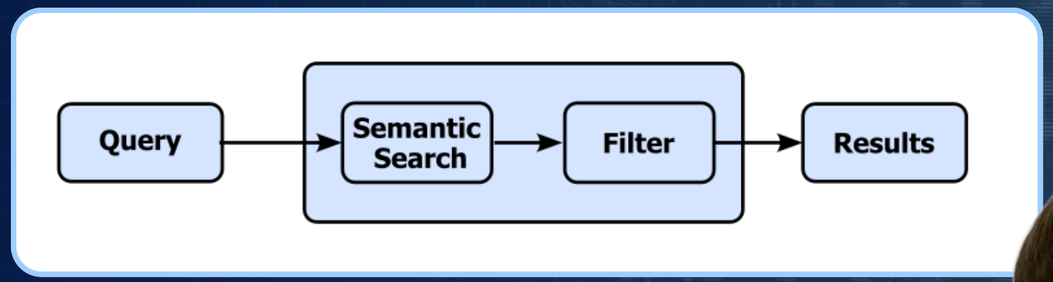
www.arcturus-labs.com

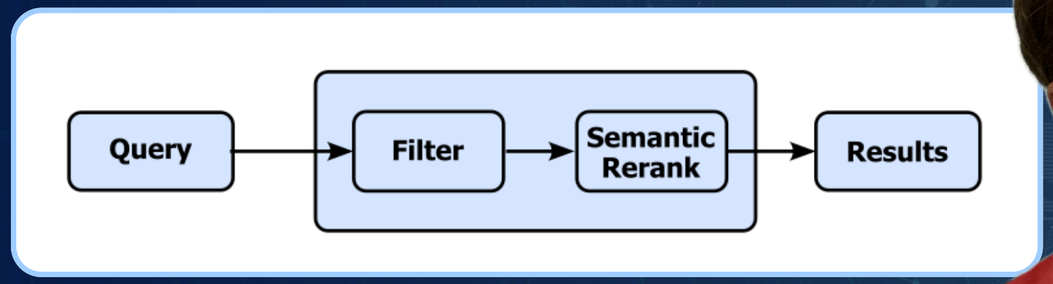
Arcturus Labs

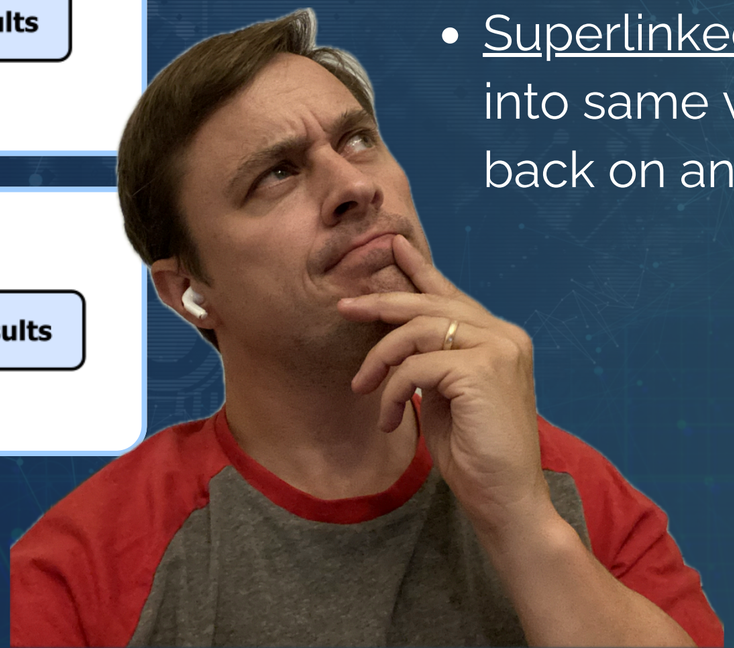
Hybrid Search?

I want filters, facets and boosting from Lexical Search.
I want search-by-meaning from Semantic Search.

Are we really stuck here?







Maybe. But we're working on it.

- Lexical search + reranking
- SPLADE - use embedding model to generate synthetic synonyms - shove that into lexical search.
- ACORN (now in Weaviate) - filters items *while* traversing HNSW datastructure
- Superlinked - embeds multiple datatypes into same vector(?) and can then piggy-back on any vector store

www.arcturus-labs.com

Arcturus Labs

Thank you!

www.arcturus-labs.com

www.arcturus-labs.com