Transforming your search with cutting-edge techniques and LLMs

Atita Arora

07 October 2023
About me

- Started 2008
- Computer Applications
- Strategic Business Management
- Vector / Semantic Search
- Language analysis
- Information retrieval

Mom of 2 boys
Opensource
Loves to travel, eat, cook

Atita Arora
Search Architect / Engineer / Enthusiast for NLP, Vectors, ML & AI.

QR Code
Searching for keywords like it's 1999...

Search apps powered by neural search and LLMs! 🧠🔍
Building Vector Search in Chorus: A Technical Deep Dive

In my last blog we established how vectors function internally and how they could be a game-changer in the world of e-commerce. We hope the first blog helped set the tone for a more focussed discussion of the technical details of our implementation of vector search in Chorus, our open source reference implementation of e-commerce search.

So how does vector search in Chorus really work?

LLM-Powered Search Applications: A New Era of Semantic Search

There’s no denying that the digital age has made information more accessible than ever before. However, with the information explosion, finding the precise piece of data you need has become increasingly difficult. Enter the world of semantic search, where search engines understand user intent and the contextual meaning of terms, thus offering a new level of precision.

Vector Similarity beyond Search

When making use of unstructured data, there are traditional go-to solutions that are well-known for developers:

- **Full-text search** when you need to find documents that contain a particular word or phrase.
- **Vector search** when you need to find documents that are semantically similar to a given query.

Sometimes people mix those two approaches, so it might look like the vector similarity is just an extension of full-text search. However, in this article, we will explore some promising new techniques that can be used to expand the use-case of unstructured data and demonstrate that vector similarity creates its own stack of data exploration tools.

Vector search in Elasticsearch: The rationale behind the design

There are different ways to implement a vector database, which have different trade-offs. In this blog, you’ll learn more about how vector search has been integrated into Elasticsearch and the trade-offs that we made.
The Evolution of Language Models - From Word Embeddings to Multimodal Transformers

Key Time periods

Pre 2010 - Focus on statistical models and predefined rules

2010 - Introduction of Word Embeddings

2018 - Introduction of Transformers

And rest is just the history !!
The Evolution of Search Applications

1. Pattern / Exact Search (Early Years)
2. Synonyms + Stemming (Mid-2000s)
3. Natural Language Processing (2010)
4. Multi-word Synonyms
5. Personalisation
6. Entity Recognition
7. Semantic search
8. Multimodal / Multilingual search
Shirt Jacket 'with' Chest Pockets

Shirt Jacket 'without' Chest Pockets

https://www.peek-cloppenburg.de/de/search/context-damen?q=Hemdjacke%20ohne%20Brusttaschen%20&searchType=userInput
Challenges of the Keyword Search

1. Actual user query intent is still a mystery! (language, long tail)
2. Vocabulary mismatch
3. Missing standardization
4. Variety of attributes
5. Trends and Popularity
6. Multimodal Content (Text, Image, Audio, Video)

...........................And there are certainly many more domain level challenges!!
Vector / Semantic Search comes to the Rescue!!

- Involves a way of representing "an object" using a list of characteristics or features.

- Vector search involves converting a "source object" into a vector embedding using a desired transformer, then index and search through the vector embeddings.

Encoding Example:

[-0.06275664, -0.038876858, -0.031730175,......-0.027633812]

- The goal of semantic search is to provide search results that align with the user's intent, even if the query and indexed content do not match precisely in terms of keywords.

Convertible laptop with touchscreen

==

Touchscreen laptop with 360-degree hinge

iphone with transparent backcover ≠

transparent backcover for iphone
How can it help?

- Fix recall issues caused by:
  - long-tail queries
  - vocabulary mismatch
  - multilingual search

- Multimodal (Use text, images, audio, video)

- Also exposes a lot of potential in:
  - Marketing and Promotions - similar users
  - Recommendations and Suggestions - similar products
  - Serchandising - filter, facets, synonyms

- Boosting precision by leveraging meaning that goes beyond the individual words themselves and focuses on the **context and intent** behind the language used.
Query: black shirt to go with blue jeans men

Results with default algo

Results with match by image vector ->
Query: date night dress for a girl

Results with default algo

Results with match by text vector ->
Query: smartwatch by sumsung

Results with default algo

Results with match by image vector

[Images of search results with various smartwatches and options]
Query: Men shorts in size m

Results with default algo

Results with match by text vector ->
Query: Fashionable formal footwear for women

Results with default algo

Results with match by image vector ->
Query: Bag for College

Results with default algo

Results with match by image vector ->
Query: Blakdress with something shimmery

No Results with 'Default Algo'

Results with text vector ->
From Keywords to Semantic Search in 3 steps

1. **Vector Extraction**
   - Data gather
   - Config changes & fields selection
   - Model Selection
   - Adding Vectors to the data

2. **Conversion and Transformation**
   - Encoding Service - Conversion of text query -> Text / Image Vector
   - Transformation of Search API to accommodate Vectors queries

3. **Optimization for Relevancy Improvements and Efficiency**
   - Offline Experimentation for relevancy improvements
   - Fine-tuning and performance optimizations
Implementation Architecture

Vector Extraction

Conversion & Transformation

Encoding Service

Encoding

(minlim/text
clip/text)

Query

(Main Query /
Boost Query)

Rewritten Query with
embeddings

Rewriter selected
i.e. embed/embig

Search Engine

Vectorised Dataset

+ Query Rewriter Config
+ Query Embeddings Rewriter

Results

docs json

Data Encoding Process

Text Vector Model -sentence-transformers-MiniLM-L6-v2

Image Vector Model - openai/clip-vit-large-patch14

Welcome to Chorus online store!

Multi-project open-source collaboration moving towards an open source stack for e-commerce search.

Here are some queries to start with:

- A query for Woman Jeans
- Any search the same as boxsearch?
  - Look at the product in the catalog.

Implementation Architecture
Keep it Simple : From Theory to Practice

- Identify a designated business use-case to try Vector / Semantic Search
- Exemplify the value with a POC:
  - Using existing Lucene-based search or leverage dedicated vector search engines / DBs
  - Leverage pre-existing models (from Hugging Face)
- Experiment to prove improvements
- Focus on building change process understanding
Identifying what to Encode/Use

- Correlated with user queries / types of searches
- Product Attributes
- Images / Multimedia
- Domain specific fields (allergens / occasion )
- Categorical data
Caveats

- Language model suitable for the business case
- Limitations of the models (Token length, language etc)
- Dimensions are not free - Infrastructural changes
- Fast and scalable vector encoding service
- Persistent 'K' retrievals
Step 3 - How to Evaluate?

Categorise your user queries and evaluate individual capability of your search / IR

Exact Queries
- phillips shb1100
- Krups f08801
- Fossil Watch AM4183

Product Type Queries
- Bluetooth headset
- Backpack / rucksack
- Perfumes

Symtomatic Queries
- wintercap for women
- craft kits for kids
- cleaning stuff for equipments

Feature:
- glow in dark toy
- organising stuff for office
- Longsleeve tshirt

Thematic:
- Datanight dress for girls
- LOL dolls
- Organising stuff for office

Compatibility:
- power adapter for lenovo
- headsets with noise cancellation
- Silver Toned Analogue Watch
- Gym wear for men

Slang/Acronym:
- Kicks
- notebook
- Beanie

Typos
- smartwatch by sumsung
- Toner cartridge from leximark brand
- Blakdress with shimmery material

Could be evaluated on:
- # zero results
- nDCG@10 , MAP , MRR etc
- Conversions / ATB
- # Search refinements
- Ctr , abandonment etc
## Evaluating the vector / semantic retrieval

<table>
<thead>
<tr>
<th>Query Type / Algo</th>
<th>Exact</th>
<th>Product Type</th>
<th>Symptoms</th>
<th>Feature</th>
<th>Thematic</th>
<th>Compatibility</th>
<th>Slang/ Acronym</th>
<th>Typos</th>
<th>Avg nDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25</td>
<td>0.76</td>
<td>0.47</td>
<td>0.44</td>
<td>0.44</td>
<td>0.04</td>
<td>0.78</td>
<td>0.34</td>
<td>0.15</td>
<td>0.44</td>
</tr>
<tr>
<td>Match by Text Vector</td>
<td>0.26</td>
<td>0.87</td>
<td>0.59</td>
<td>0.67</td>
<td>0.63</td>
<td>1</td>
<td>0.88</td>
<td>0.90</td>
<td>0.70</td>
</tr>
<tr>
<td>Match by Image Vector</td>
<td>0.50</td>
<td>0.73</td>
<td>0.50</td>
<td>0.72</td>
<td>0.66</td>
<td>0.80</td>
<td>0.84</td>
<td>0.64</td>
<td>0.66</td>
</tr>
<tr>
<td>Boost by Text Vector</td>
<td>0.55</td>
<td>0.65</td>
<td>0.62</td>
<td>0.66</td>
<td>0.10</td>
<td>1</td>
<td>0.85</td>
<td>0.92</td>
<td>0.66</td>
</tr>
<tr>
<td>Boost by Image Vector</td>
<td>0.58</td>
<td>0.67</td>
<td>0.50</td>
<td>0.65</td>
<td>0</td>
<td>0.80</td>
<td>0.84</td>
<td>0.78</td>
<td>0.60</td>
</tr>
</tbody>
</table>
Other things to experiment with

- Fine-tuned / self trained models
- Combining Vectors with managed searches*
- Vector based Re-ranking model
- Combining scores from text + image search results

*Promoted searches, legal binding, brand conflicts etc
Query: organising stuff for office

Getting Better through Fine Tuning..

Results with default algo

Results with match by image model

Results with fine-tuned image model

Fine-tuned Model courtesy: Jina.ai
What are other implementations?

- Semantic Retrieval
- Auto Data Labelling
- Anomaly Detection
- Retrieval Augmented Generation
- Summarisation
- Results Diversity
References

- Available at:

- Query Types https://baymard.com/blog/ecommerce-search-query-types

- Evaluation Tool https://quepid.com/

- Model Courtesy https://huggingface.co/models
Thank you!!
Querqy Core

Parser

“user input”

<<parses>>

Query

<<creates>>

Rewriter

<<rewrites>>

Rewrite Chain

<<uses>>

Lucene

<<creates>>

Lucene Query

<<supplies to ES/Solr>>

QueryBuilder (ES/OS)

QParser (Solr)

<<creates>>

Controller

<<creates>>

Search engine / request adapter

<<supplies>>

Lucene / request adapter

<<supplies>>

ES / OS / Solr Dismax request adapter
Women of Search
About Women of Search

● Group started in Mar 2021

● A vibrant community dedicated to empowering & celebrating women in the world of search technologies (& related tech)

● Provide a platform for networking, mentorship, and knowledge-sharing among women professionals.

● Showcase, celebrate, brag about the achievements of women

● Practice public speaking and groom your soft skills :}
Updates

● 'Happy Hour' continues … every 1st Wednesday of each month at 9 am PST.

● Lots of interesting topics and discussions

● Share useful resources and experiences

Slack : relevancy.slack.com / #women-of-search

LinkedIn : https://www.linkedin.com/groups/12659464/

Calendar: https://bit.ly/2SywlDh

Google group: https://groups.google.com/g/women-of-search
Thank you !!!

In the realm of generative AI, women's voices echo softly, but their impact resounds loudly!