ElasticSearch 101

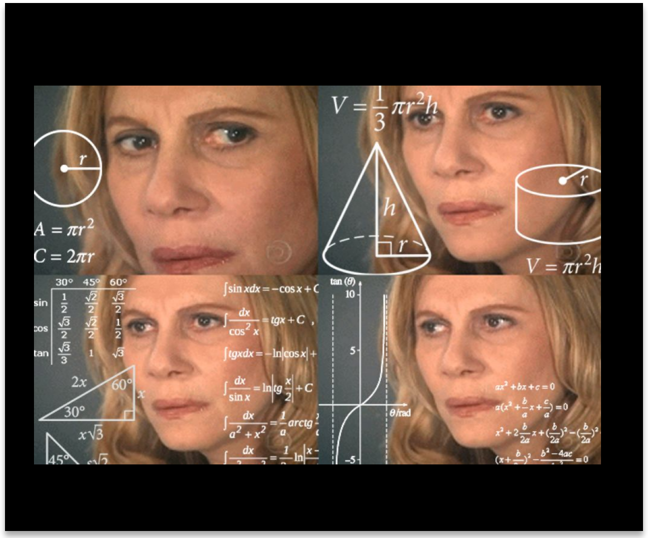
Dan Salo

November 17th, 2021

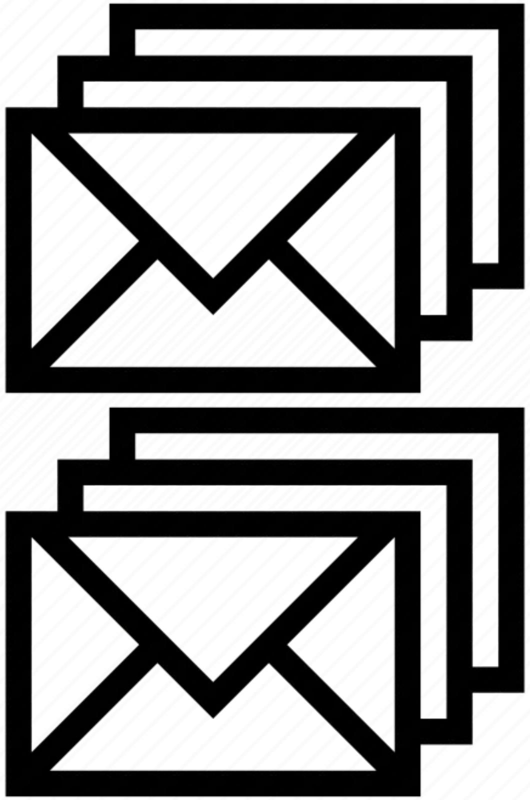
So … we want to build a search engine!

Query:

"Show me all emails related to PFPT trades in the last 3 days between Bank A and Bank B"



Documents:



Relevant Documents:

Core Components

Query Representation
- Numeric
- Words, synonyms, stop words ...

Document Representation
- Numeric
- Words, topics, clusters, features ...

Retrieval function
- How to map query to documents

Relevance
(Binary or Continuous)

https://sites.cs.ucsb.edu/~tyang/class/293S17/slides/Topic2IRModels.pdf

Outline

Boolean Retrieval

Ranked Retrieval

- Vector Space Model
- Semantic Search
- Learning to Rank

Search Evaluation

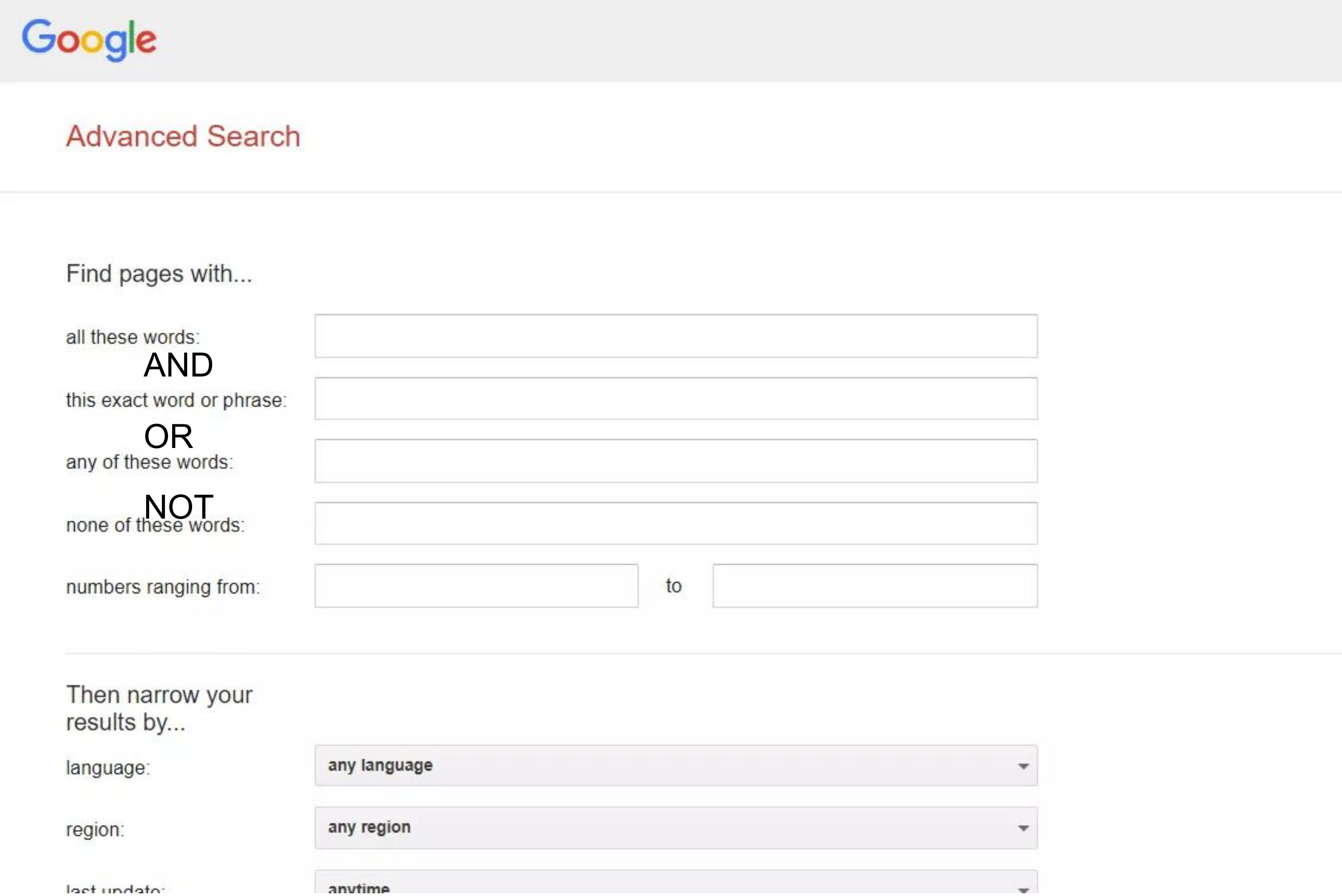
What are the concepts?

How does ES implement?

Why would we want to use?

Boolean Retrieval

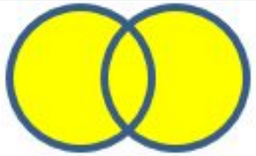
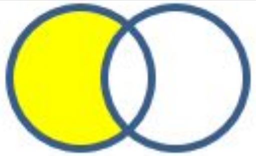
Boolean Retrieval Example



https://www.google.com/advanced_search

Boolean Retrieval Basics

How to Search Using Boolean Operators:

Concept	Search Examples	Results
AND	politics **AND** media children **AND** poverty "civil war" **AND** Virginia	 Results will include both terms
OR	"law enforcement" **OR** police labor **OR** labour 60s **OR** sixties	 Results will include one or both terms
NOT	"civil war" **NOT** American Caribbean **NOT** Cuba therapy **NOT** physical	 Excludes results with the term following NOT

Query representation:

- Set of keywords
- Boolean operators

Document representation:

- Set of keywords

Retrieval function:

- Set operation
- *Binary* relevance

https://mc.libguides.com/c.php?g=39000&p=247825

Boolean Retrieval Extras

Query Representation

- Stems/Lemmas/Tokens
- Stopword removal
- Query expansion
  - Thesaurus Synonyms
  - Knowledge Base
  - Relevance Feedback

Document Representation

- Stems/Lemmas/Tokens
- Stopword removal
- Document Expansion
  - Topic Modeling
  - Named Entities

Closing the Semantic/Vocab Gap

Retrieval Function

- Pre-match filtering
- Proximity Match
- Fuzzy Match
- Wildcard/Regex Match

20 years of Lucene

Written in 1999
Apache in **2001**
Top-level in 2005

Named after author's wife's middle name.

Written in 2004
Apache in **2006**
Top-level in 2007

Merged with Lucene in 2010

Separated from Lucene in February 2021.

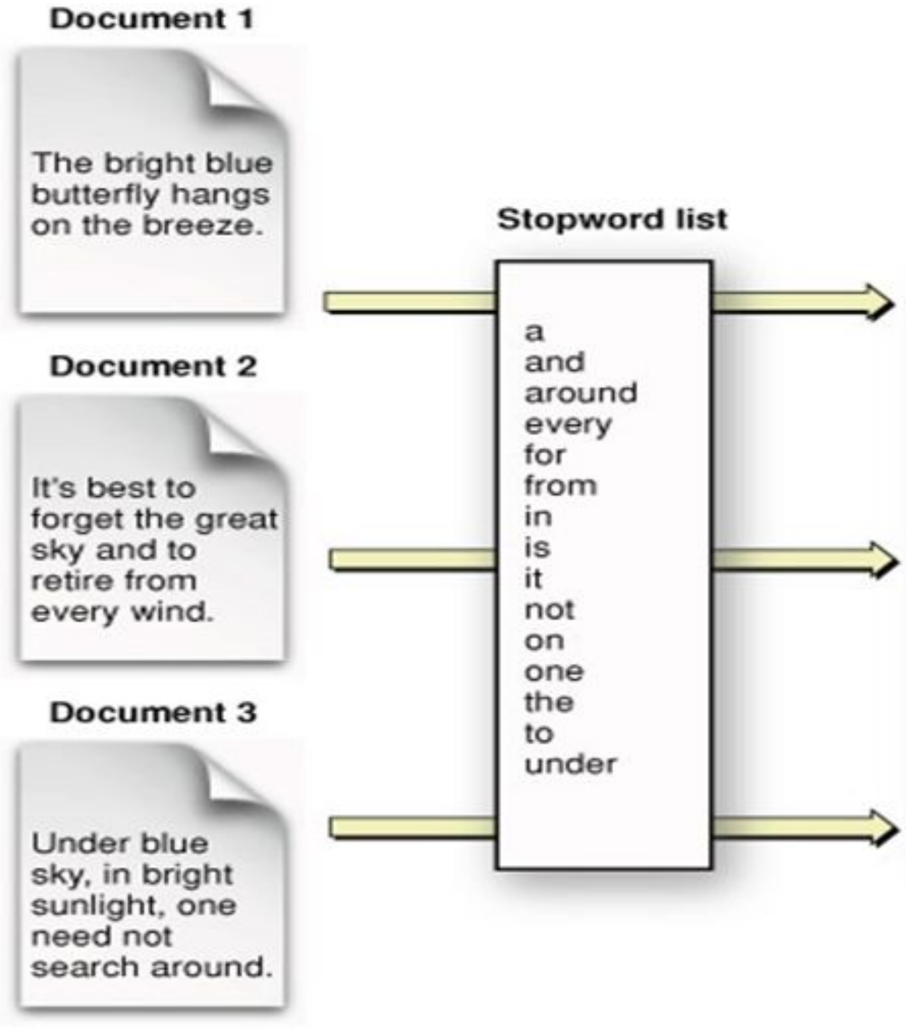
Released in **2010**
Couples with Logstash, Kibana, etc.

Managed by publicly traded Elastic

First stable release July **2021**

AWS forked ES 7.10 and continues development under Apache license

Inverted Index



Inverted index

ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

https://tutorialseye.com/elasticsearch-storage-architecture-using-inverted-indexes.html

Inverted Index

- Opposite of forward index
- Optimized for query:
  "Which documents contain term *X*?"

Boolean Query: "blue" AND "sky"
Relevant Document: Doc 3

Inverted index

ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

elasticsearch

Mapping

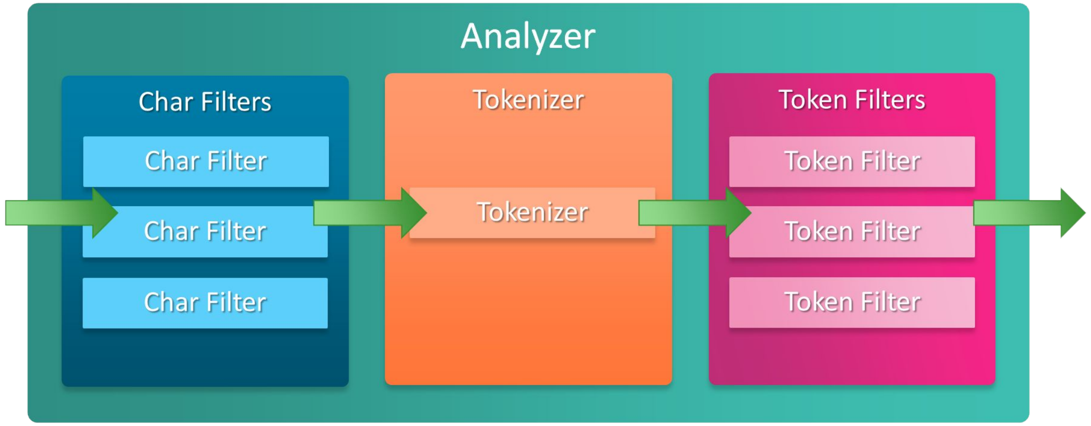
- Can be explicit or dynamic
- Similar to NoSQL
- Categories of data types:
  - Binary / Boolean
  - Keyword
  - Numbers
  - Dates
  - Vectors
  - Geo
  - **Text (Inverted Index)**
  - Relational (object, nested, flattened)

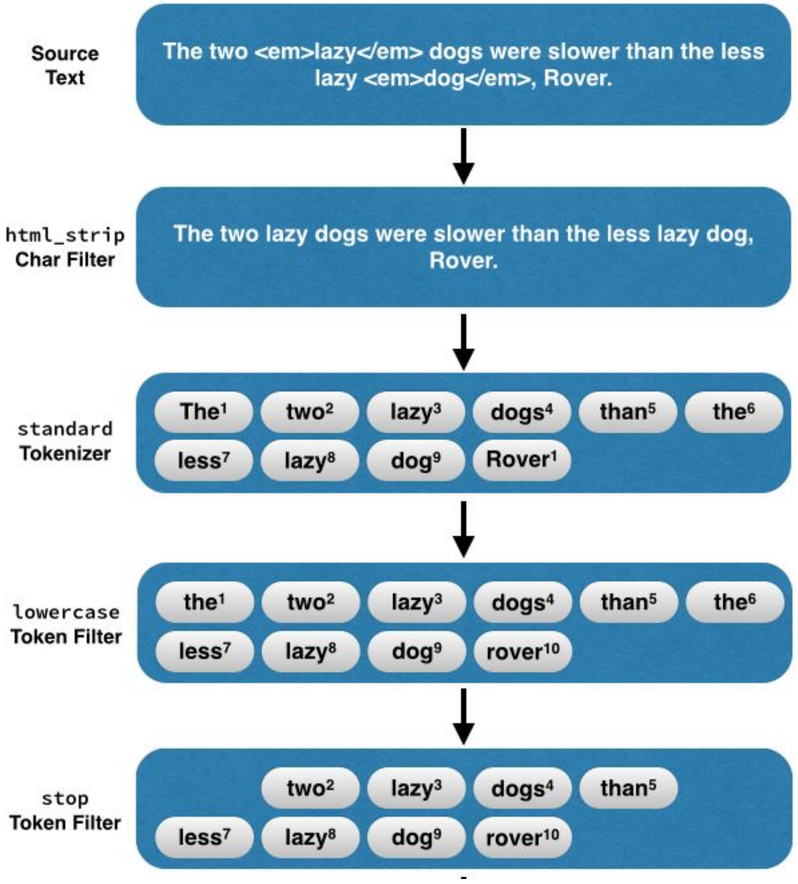
```
"id": {
    "type": "text"
},
"body": {
    "type": "text"
},
"subject": {
    "type": "text"
},
"date": {
    "type": "date"
},
"to": {
    "type": "text"
},
"from": {
    "type": "text"
},
"entities": {
    "type": "nested",
    "properties": {
        "score": {
            "type": "float"
        },
        "text": {
            "type": "text",
        },
        "type": {
            "type": "text",
        }
    }
```

https://wiki.proofpoint.com/wiki/display/RESERO/Data+Stores+Schema

elasticsearch

Analysis





https://www.elastic.co/guide/en/elasticsearch/client/net-api/current/writing-analyzers.html

elasticsearch

Filtering vs. Querying

Filtering	Querying
• Happens before querying • Cached • Does not calculate relevance score • Binary / Exact searches	• Happens after filtering • Not cached • Calculates relevance score • Full text search

Querying

elasticsearch

Term Query

- Not Analyzed

Match Query

- Analyzed
- “fuzziness”: fuzzy match search

Match Phrase Query

- Analyzed
- “slop”: proximity search

Analysis

- Analyzer: “synonyms”
- Token filter: “stemmer”

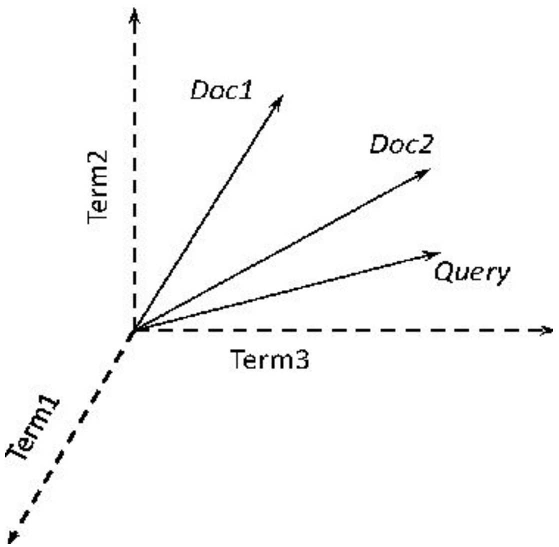
Observations of Boolean Search

- User input must be formulated into a Boolean query
- Deterministic
- Provides many parameters for tuning
- Number of results returned are either “feast” or “famine”
- Typically paired with ranked retrieval in the event of a “feast”

Ranked Retrieval

Vector Space Model

Vector Space Model Basics



$$\mathit{similarity}(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Query representation:

- *Sparse* vector based on (weighted) terms

Document representation:

- *Sparse* vector based on (weighted) terms

Retrieval function:

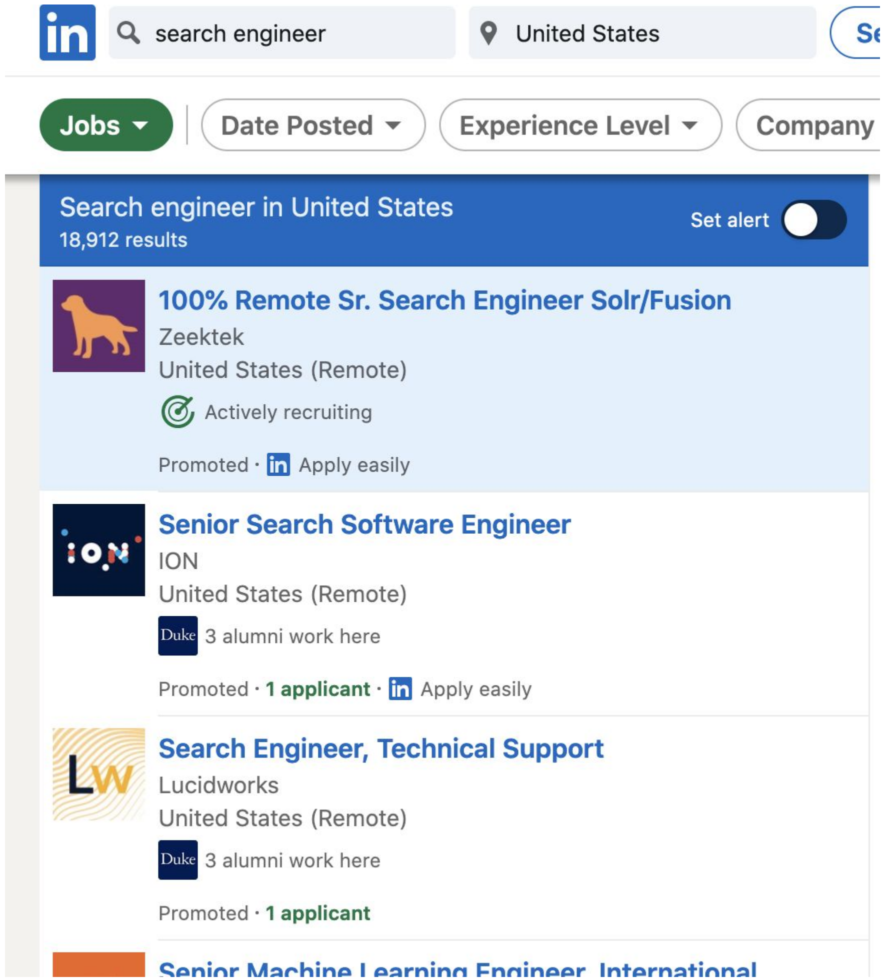
- Cosine similarity between Query and all Docs
- *Continuous* value for relevance

http://bitsearch.blogspot.com/2011/01/vector-space-model-for-scoring.html

Vector Space Model Extras

Search Engineers can …

- Choose their term weighting scheme:
  - Okapi BM25 (ES default)
  - TF-IDF
- Boost on certain fields (constant)
- Add a function to modify field score (variable)
- Decay weight for numeric values
- …



https://www.researchgate.net/figure/Topic-based-vector-space-model-visualization_fig2_298215705

elasticsearch

Lucene Practical Scoring Function

```
score(q,d)  =  ①
            queryNorm(q)  ②
          · coord(q,d)    ③
          · ∑ (           ④
                tf(t in d)     ⑤
              · idf(t)²        ⑥
              · t.getBoost()   ⑦
              · norm(t,d)      ⑧
            ) (t in q)    ④
```

queryNorm: normalization coefficient to compare between queries

coord: number of query terms in document

For each term t in query q:

tf: term frequency in document

idf: inverse document frequency in corpus

getBoost: query-time or index-time boost for a term

norm: inverse square root of number of terms in field

https://www.compose.com/articles/how-scoring-works-in-elasticsearch/
http://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/package-summary.html#scoring

elasticsearch

Elasticsearch Altogether

Boolean Retrieval

Boolean Filtering

Boolean Querying

Ranked Retrieval

Practical Scoring Function

Optimize for Recall

Optimize for Precision

Observations of Vector Space Models

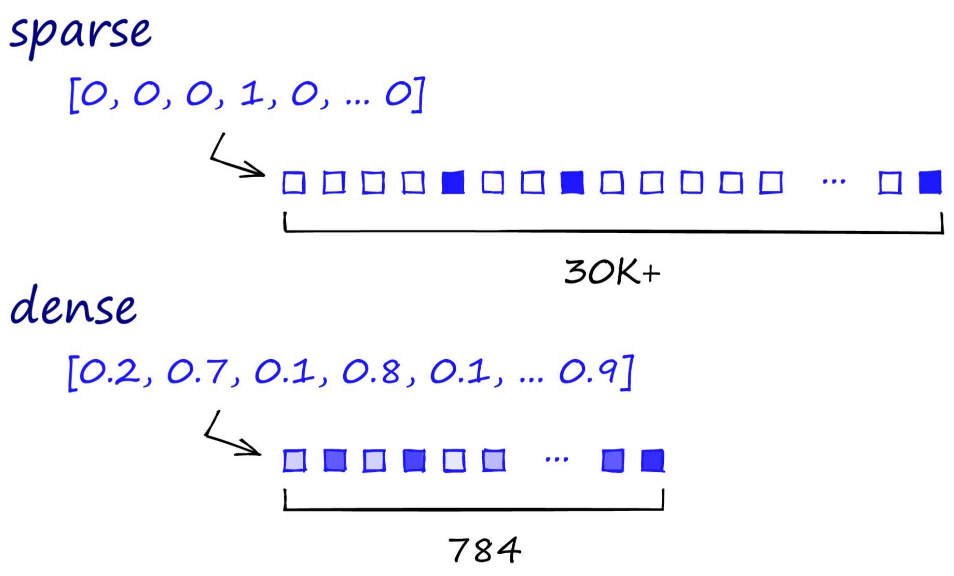
- Simple, mathematically based approach.
- Considers both local (tf) and global (idf) word occurrence frequencies.
- Allows efficient implementation for large document collections.
- Missing semantic information (e.g. word sense).
- Missing syntactic information (e.g. phrase structure, word order, proximity information)
- Assumption of term independence (e.g. ignores synonymy).

https://www.microsoft.com/en-us/research/uploads/prod/2017/01/cao-nie-gao-robertson.sigir08.pdf

Ranked Retrieval

Semantic Search

Dense Vectors



Embedding Algorithms?
Deep Dive Talk in October!

e.g. Word2Vec, LSI, ELMo, BERT,

Query Representation:

- *Dense* vector based on embedding algorithms

Document representation:

- *Dense* vector based on embedding algorithms

Retrieval function:

- (Approximate) Nearest Neighbors
- *Continuous* value for relevance

Approximate Nearest Neighbors Search

Techniques

- Product Quantization
- Hierarchical Navigable Small World (HNSW)
- …

Libraries

- FAISS (Facebook)
- ANNOY (Spotify)
- LOPQ (Yahoo)
- Nmslib
- …
- https://github.com/erikbern/ann-benchmarks

Dense Vectors in Elasticsearch

elasticsearch

In 2019, ES implemented “dense_vector” and “script_score”

Similarity is calculated over **all documents**.

Good for scoring documents, but not in the initial retrieval step.

So what about ANN?

ANN in Elasticsearch

elasticsearch

Investigate various implementations of ann search for vector fields #42326

Closed **mayya-sharipova** opened this issue on May 21, 2019 · 54 comments

mayya-sharipova commented on May 21, 2019 • edited Contributor

Assignees

Integrate ANN search #78473

Open 15 of 18 tasks **jtibshirani** opened this issue on Sep 29 · 3 comments

jtibshirani commented on Sep 29 • edited by mayya-sharipova Member

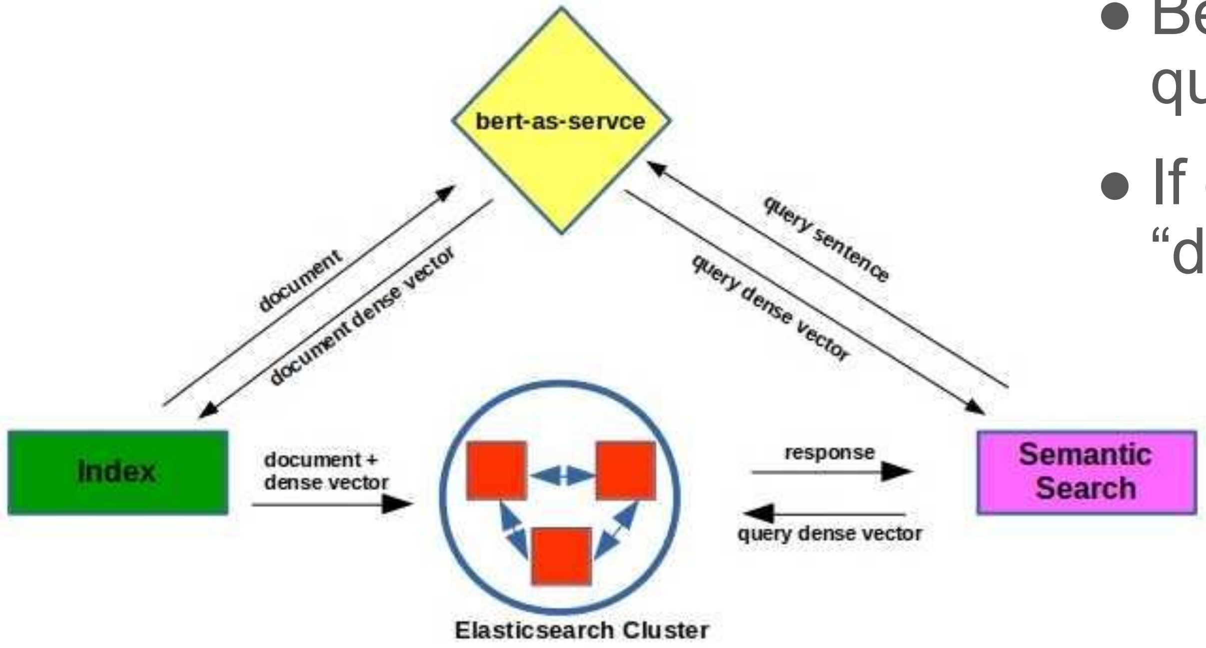
Background

Currently Elasticsearch supports storing vectors through the `dense_vector` field type and using them

Plugins?

If you don’t need ANN …

elasticsearch

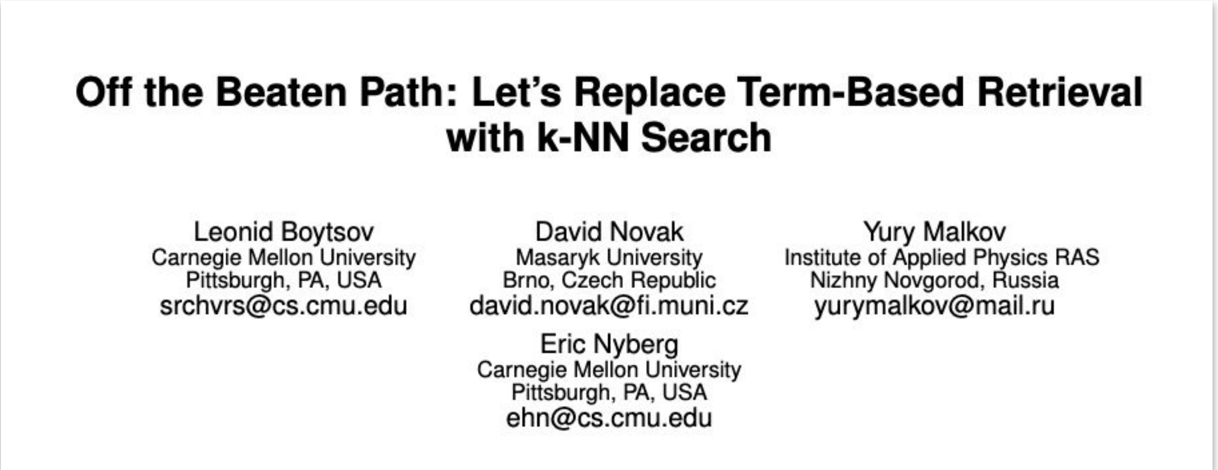


- Bert as service run at index time over all documents.
- Bert as service run at query time over query.
- If corpus is “manageable” size, then ES “dense_vector” is fine.

https://xplordat.com/2019/10/28/semantics-at-scale-bert-elasticsearch/

Semantic Search Observations

- BERT *et al* leverage contextual information in query text
- **Query formulation** is a key determinant
- Replacement for term-based retrieval? I don't think so for trade alert to comms use case …



Off the Beaten Path: Let's Replace Term-Based Retrieval with k-NN Search

Leonid Boytsov
Carnegie Mellon University
Pittsburgh, PA, USA
srchvrs@cs.cmu.edu

David Novak
Masaryk University
Brno, Czech Republic
david.novak@fi.muni.cz

Yury Malkov
Institute of Applied Physics RAS
Nizhny Novgorod, Russia
yurymalkov@mail.ru

Eric Nyberg
Carnegie Mellon University
Pittsburgh, PA, USA
ehn@cs.cmu.edu

http://boytsov.info/pubs/cikm2016.pdf

Ranked Retrieval

Learning to Rank

LTR vs. Document Classification

- Both can output a binary decision: 1 or 0
- Difference: ranking is dependent on documents **and the query**
- Therefore, features for LTR ought to contain query and document information.
- Data:
  - Query & documents & relevance judgements

"president in 2010" Obama ✅

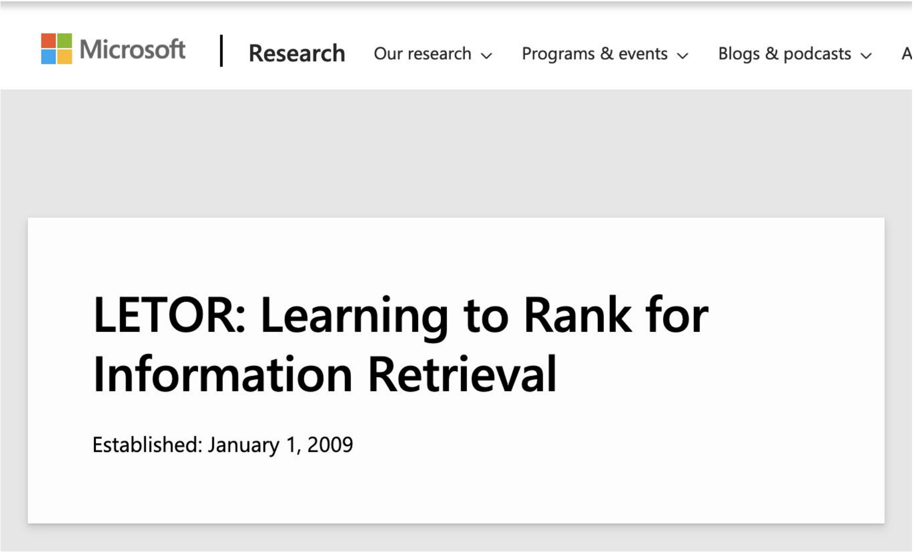
"current president" Obama ❌

https://www.youtube.com/watch?v=2UpLin5T_E4

Possible Features

Feature	Descriptions
1	BM25
2	document length (dl) of body
3	dl of anchor
4	dl of title
5	dl of URL
6	HITS authority
7	HITS hub
8	HostRank (SIGIR feature)
9	Inverse document frequency (idf) of body
10	idf of anchor
11	idf of title
12	idf of URL
13	Sitemap based feature propagation (SIGIR feature)
14	PageRank
15	LMIR.ABS of anchor
16	BM25 of anchor
17	LMIR.DIR of anchor
18	LMIR.JM of anchor
19	LMIR.ABS of extracted title (SIGIR feature)
20	BM25 of extracted title (SIGIR feature)
21	LMIR.DIR of extracted title (SIGIR feature)
22	LMIR.JM of extracted title (SIGIR feature)
23	LMIR.ABS of title
24	BM25 of title
25	LMIR.DIR of title
26	LMIR.JM of title
27	Sitemap based feature propagation (SIGIR feature)
28	tf of body
29	tf of anchor
30	tf of title
31	tf of URL
32	tf*idf of body
33	tf*idf of anchor
34	tf*idf of title
35	tf*idf of URL
36	Topical PageRank (SIGIR feature)
37	Topical HITS authority (SIGIR feature)
38	Topical HITS hub (SIGIR feature)
39	Hyperlink base score propagation: weighted in-link (SIGIR feature)
40	Hyperlink base score propagation: weighted out-link (SIGIR feature)
41	Hyperlink base score propagation: uniform out-link (SIGIR feature)
42	Hyperlink base feature propagation: weighted in-link (SIGIR feature)
43	Hyperlink base feature propagation: weighted out-link (SIGIR feature)
44	Hyperlink base feature propagation: uniform out-link (SIGIR feature)

LETOR Features



https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/letor-4-0/

Algorithms

- Methods:
  - Tree-based methods (LambdaMART, MART)
  - SVMRank and Propensity SVM Rank
  - Linear Models
  - Deep Nets
- Objectives:
  - Supervised
  - Semi-supervised
  - Listwise
  - Pairwise

Two-Stage Learning to Rank for Information Retrieval

Van Dang, Michael Bendersky, and W. Bruce Croft

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
{vdang, bemike, croft}@cs.umass.edu

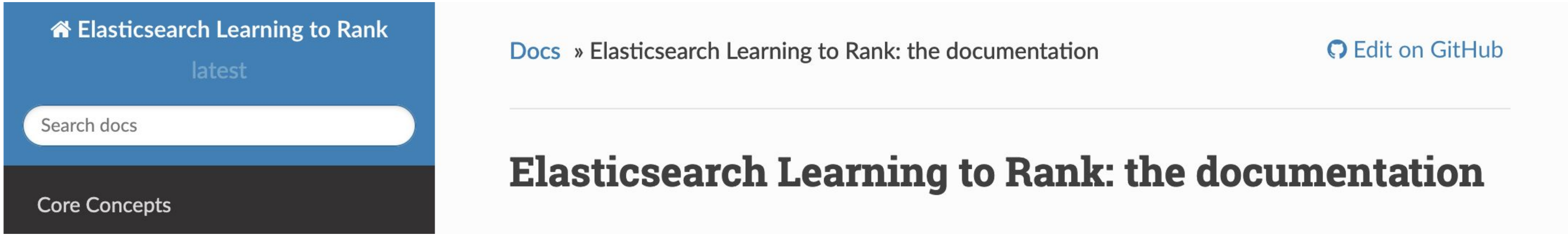
From RankNet to LambdaRank to LambdaMART: An Overview

Christopher J.C. Burges
Microsoft Research Technical Report MSR-TR-2010-82

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.366.7926&rep=rep1&type=pdf
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MSR-TR-2010-82.pdf
https://arxiv.org/pdf/1608.04468.pdf

Plugins

elasticsearch



- Allows you to store features (Elasticsearch query templates) in Elasticsearch
- Logs features scores (relevance scores) to create a training set for offline model development
- Stores linear, xgboost, or ranklib ranking models in Elasticsearch that use features you've stored
- Ranks search results using a stored model

https://github.com/o19s/elasticsearch-learning-to-rank

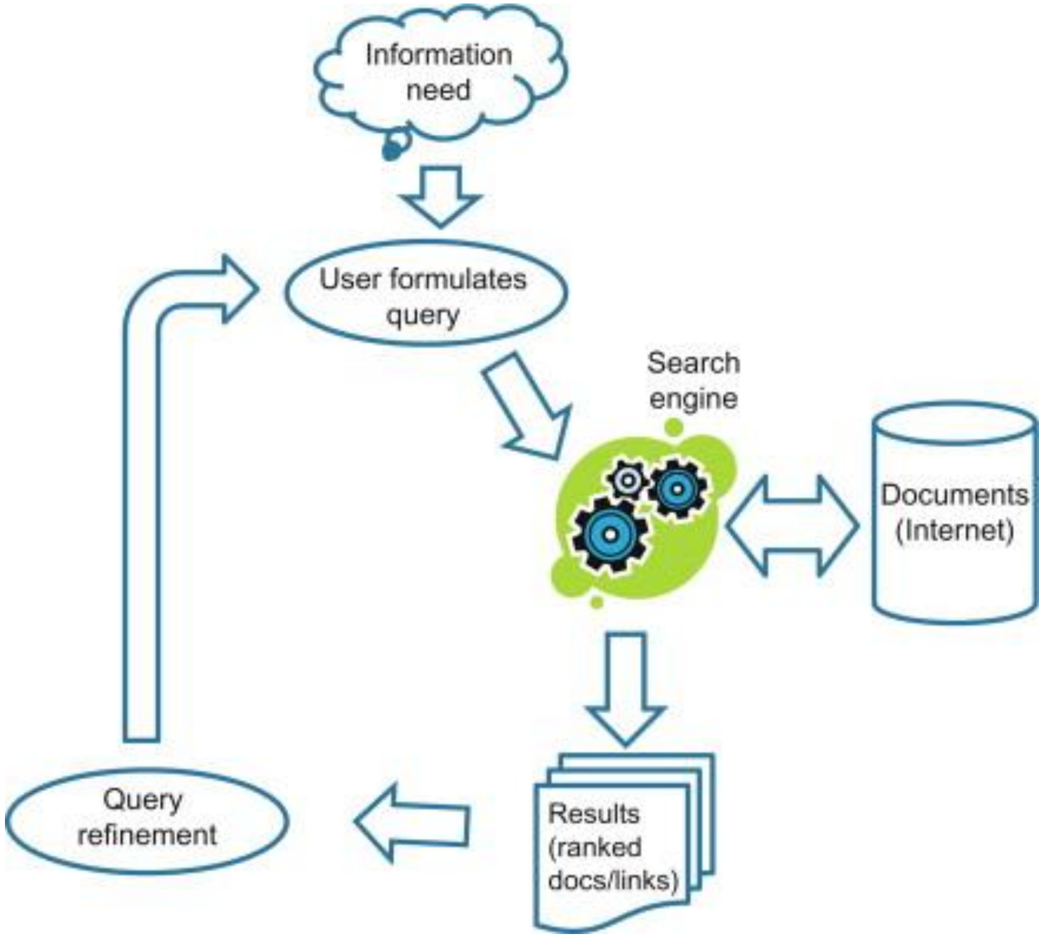
LTR Observations

- User clicks (surrogate for relevance judgments) are necessary
- User clicks can be noisy
- Either bootstrap search engine first without LTR or warm start LTR

Search Evaluation

Relevancy in Search

- Search relevance is the measure of accuracy of the relationship between the search **query** and the search **results**.
- How to assess relevancy of results?
  - User click[1]
  - Next paginated group (signifies none were relevant)
  - Query reformulation[2] (signifies none were relevant & a time-dependent signal)



[1]https://dl.acm.org/doi/10.1145/3404835.3462894
[2]https://www.sciencedirect.com/topics/computer-science/query-reformulation
https://ccc.inaoep.mx/~villasen/bib/AN%20OVERVIEW%20OF%20EVALUATION%20METHODS%20IN%20TREC%20AD%20HOC%20IR%20AND%20T

Binary Relevance

Single Document

Relevance as a binary label *per query*

	Relevant	Non-relevant	Total
Retrieved	A	B	A+B
Not retrieved	C	D	C+D
Total	A+C	B+D	A+B+C+D

Figure 6. Categories for precision, recall, and accuracy.

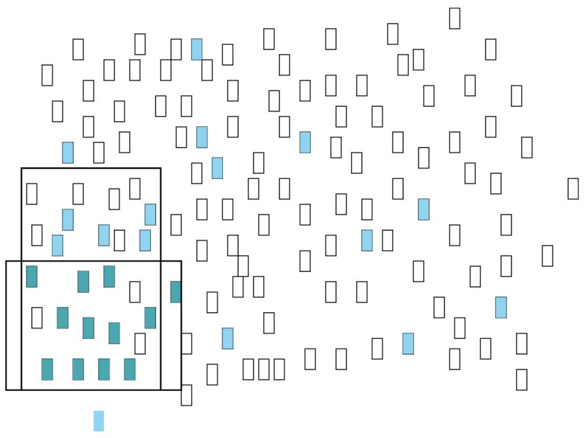


Figure 7. Example of the use of precision and recall.

Corpus

How to combine P/R for multiple documents in a corpus?

$$P_{11} = \frac{1}{11} \sum_{j=0}^{10} \frac{1}{N} \sum_{i=1}^{N} \tilde{P}_i(r_j) \tag{6.9}$$

with $\tilde{P}_i(r_j)$ being the precision (interpolated or measured) at the jth recall point for the ith query (out of N queries). $r_0, r_1, \ldots r_{10}$ are the 11 standard recall

$$MAP = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{Q_j} \sum_{i=1}^{Q_j} P(rel = i) \tag{6.11}$$

with Q_j being the number of relevant documents for query j; N the number of queries, and $P(rel = i)$ the precision at ith relevant document.

https://ccc.inaoep.mx/~villasen/bib/AN%20OVERVIEW%20OF%20EVALUATION%20METHODS%20IN%20TREC%20AD%20HOC%20IR%20AND%20T

Data Sets

- Contains documents and queries with associated relevance judgements
- Most datasets seem to use natural language queries or questions
- Different than LETOR dataset

MS MARCO: A Human Generated MAchine Reading COmprehension Dataset

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang
Microsoft AI & Research

Cranfield collection

The Cranfield collection comes in two forms: the 1400 collection; and the 200 collection

The bits

- **cran.all** - The documents
- **cran.qry** - The queries
- **cranqrel** - The relevance assesments
- **readme** - Some attempt at explanation especially about the relevance judgements

- cran.tar.gz - All the bits put together

https://github.com/oaqa/FlexNeuART/tree/master/scripts/data_convert

Next Steps

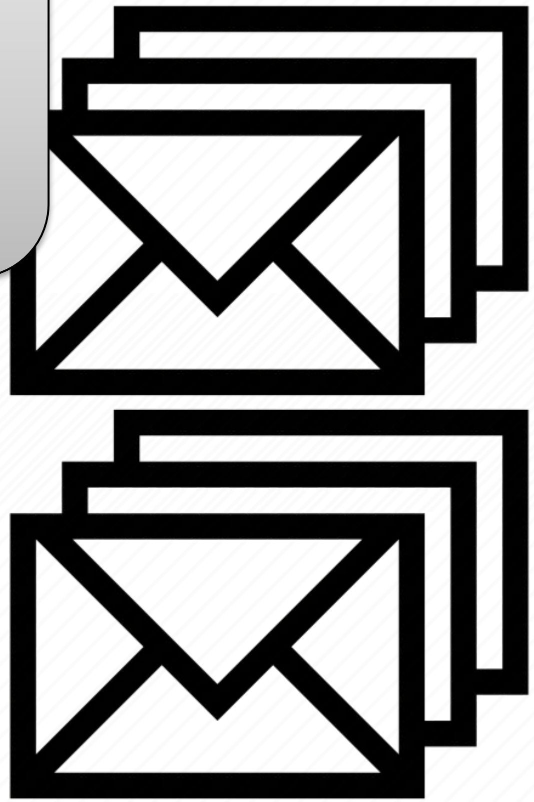
How will user formulate query?
Write test code that takes user input and generates ES queries

Query:

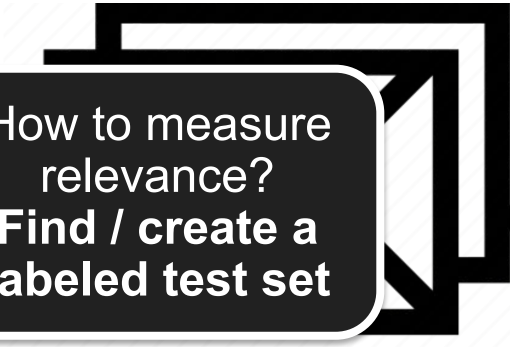
"Show me all emails related to PFPT trades in the last 3 days between Bank A and Bank B"

Semantic search?
Experiment with embeddings on email / chat data

ocuments:



Relevant Documents:



How to measure relevance?
Find / create a labeled test set

Learn to rank?
Experiment with different algorithms on labeled test set

Thank You

Information Retrieval Courses and Books

- 2021, Ray Mooney, CS 371: https://www.cs.utexas.edu/users/mooney/ir-course/
- 2009, C. Manning, Intro to IR: https://nlp.stanford.edu/IR-book/information-retrieval-book.html
  - 2021 class: https://web.stanford.edu/class/cs276/

Basic Techniques

- Traditional Search Evaluation in TREC: https://ccc.inaoep.mx/~villasen/bib/AN%20OVERVIEW%20OF%20EVALUATION%20METHODS%20IN%20TREC%20AD%20HOC%20IR%20AND%20TREC%20QA.pdf
- Vector Space model Basics: https://github.com/socrateszhang/InfoRetrivalModels
- Psuedo-Relevance Feedback: https://www.microsoft.com/en-us/research/uploads/prod/2017/01/cao-nie-gao-robertson.sigir08.pdf
- Lucene Scoring: http://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/package-summary.html#scoring