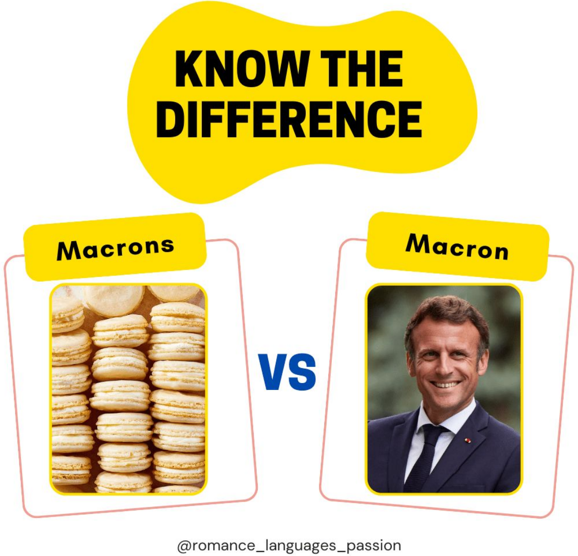
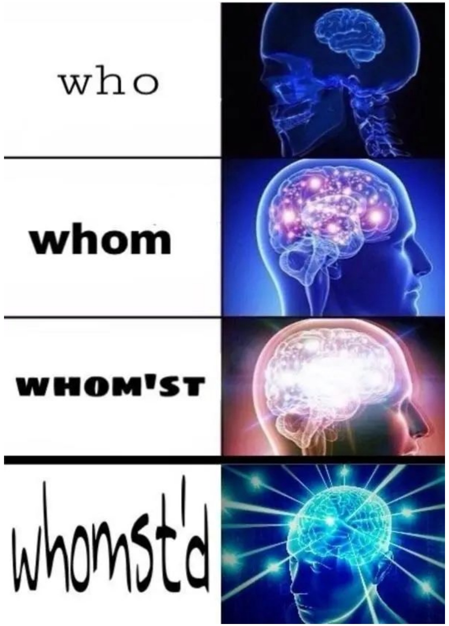
Natural Language Processing
The Finale





Hello.

Miguel Cardoso

Head of AI @ Twistag

Work on Product, AI and Software Engineering daily.

Recap

- Regex
- Tokenization
- Stopwords
- Stemming
- Lemmatization
- Bag of Words
- TF-IDF

Recap

- Regex → Useful to manipulate strings
- Tokenization
- Stopwords
- Stemming
- Lemmatization
- Bag of Words
- TF-IDF

Recap

- Regex → Useful to manipulate strings
- Tokenization → Pre-processing step applied on text
- Stopwords
- Stemming
- Lemmatization
- Bag of Words
- TF-IDF

Recap

- Regex → Useful to manipulate strings
- Tokenization → Pre-processing step applied on text
- Stopwords → Words that are considered irrelevant for NLP purposes, thus filtered out
- Stemming
- Lemmatization
- Bag of Words
- TF-IDF

Recap

- Regex
- Tokenization
- Stopwords
- Stemming
- Lemmatization
- Bag of Words
- TF-IDF

Useful to manipulate strings

Pre-processing step applied on text

Words that are considered irrelevant for NLP purposes, thus filtered out

Text normalization techniques

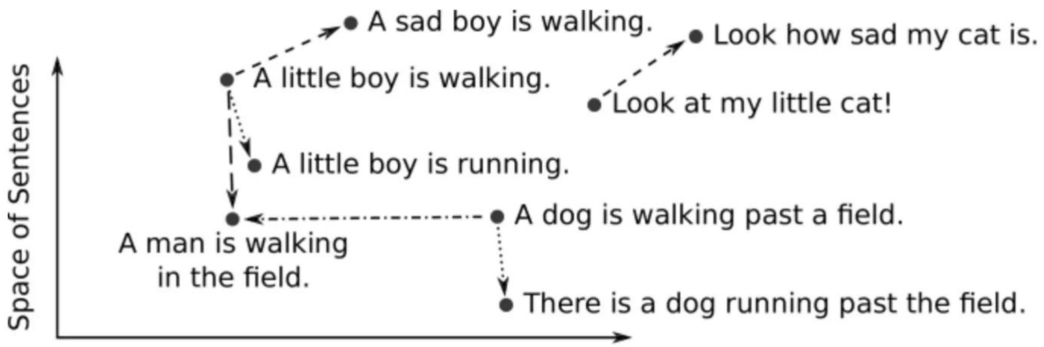
Recap

- Regex → Useful to manipulate strings
- Tokenization → Pre-processing step applied on text
- Stopwords → Words that are considered irrelevant for NLP purposes, thus filtered out
- Stemming → Text normalization techniques
- Lemmatization → Text normalization techniques
- Bag of Words → NLP techniques to represent text
- TF-IDF → NLP techniques to represent text

Natural Language Processing

This class will be about

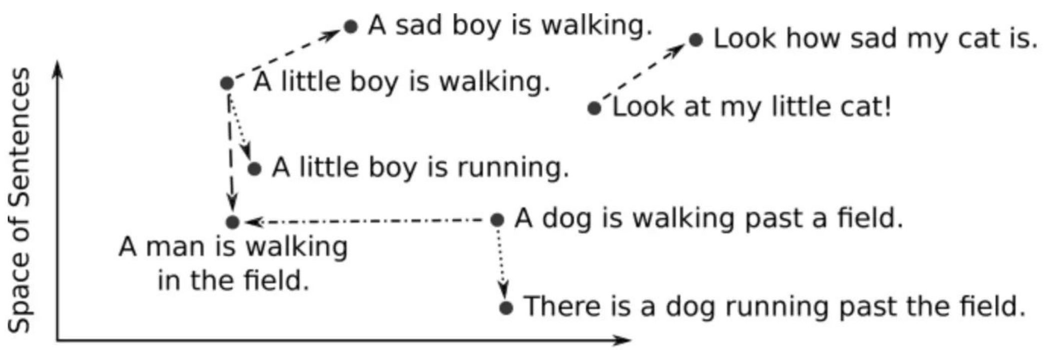
- Fetching information



Natural Language Processing

This class will be about

- Embeddings
- Similarity
- (Simplified) Search Engines



Natural Language Processing

References and useful resources

- https://www.pinecone.io/learn/vector-embeddings/
- https://www.elastic.co/pt/what-is/vector-embedding
- https://www.ibm.com/think/topics/vector-embedding
- https://www.timescale.com/blog/a-beginners-guide-to-vector-embeddings/
- https://www.cloudflare.com/learning/ai/what-are-embeddings/
- https://www.pinecone.io/learn/series/faiss/hnsw/
- https://medium.com/@gallaghersam95/visualizing-embedding-vectors-99cac1d164c4

Natural Language Processing

What are embeddings ?
What can we use them for ?
How do we get them ?

Natural Language Processing

What are embeddings ?
What can we use them for ?
How do we get them ?

Natural Language Processing

What are embeddings ?

Embeddings are vectors that represent real-world objects, like **words**, images, or videos, in a form that **computer algorithms** like machine learning models can easily process.

	Feature #1	Feature #2	Feature #3	Feature #4	...
Document #1	1	0	1	0	0
Document #2	0	0	2	0	0
Document #3	0	1	0	0	1
Document #4	0	0	0	1	0
⋮	0	0	1	0	0

DTM

Natural Language Processing

What are embeddings ?

Embeddings are vectors that represent real-world objects, like **words**, images, or videos, in a form that **computer algorithms** like machine learning models can easily process.

	Feature #1	Feature #2	Feature #3	Feature #4	...
Document #1	1	0	1	0	0
Document #2	0	0	2	0	0
Document #3	0	1	0	0	1
Document #4	0	0	0	1	0
⋮	0	0	1	0	0

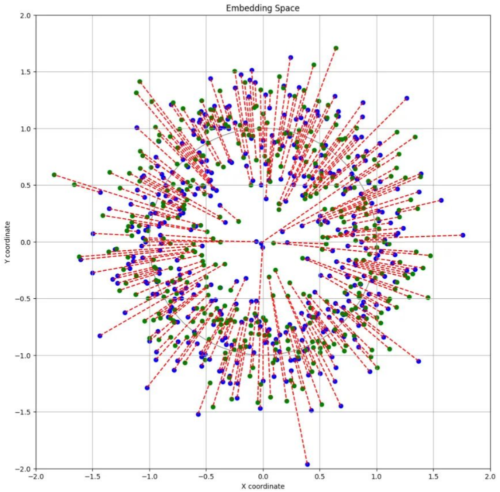
DTM

TL;DR it's just a bunch of numbers with meaning for computers

Natural Language Processing

What can we use them for ?

For information retrieval and as input for machine learning models.



Natural Language Processing

Embeddings are **vectors** that represent real-world objects, like **words**, images, or videos, in a form that **computer algorithms** like machine learning models can easily process.

They are used for **information retrieval** and as **input** for machine learning models.

Computed by specialized algorithms that **process text into vectors** like bag of words, TF-IDF, BM25 or deep learning embedding models.

Natural Language Processing

Computed by specialized algorithms that **process text into vectors** like bag of words, TF-IDF, BM25 or deep learning embedding models.

Syntactic

Semantic

Then what ?

Search.
Information Retrieval.
Embeddings.
Similarity.

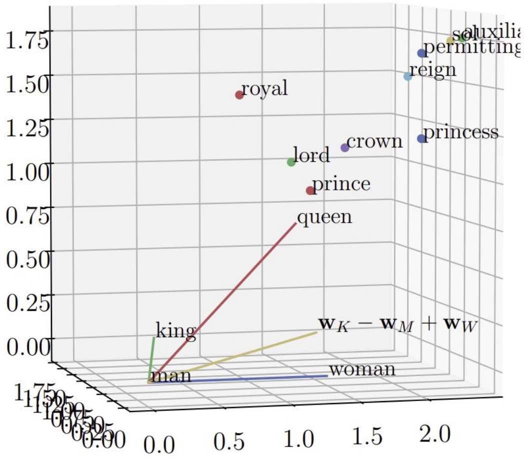
Then what ?

How do we search ?

Then what ?

How do we search ?

- Words exist in space
- Sentences are made of words
- Sentences exist in space
- Use distances to search



Then what ?

Demo.

Then what ?

Demo.

Use case:
Find the most similar and dissimilar colleagues based on your CVs.

Natural Language Processing - Search

There was a lot going on.

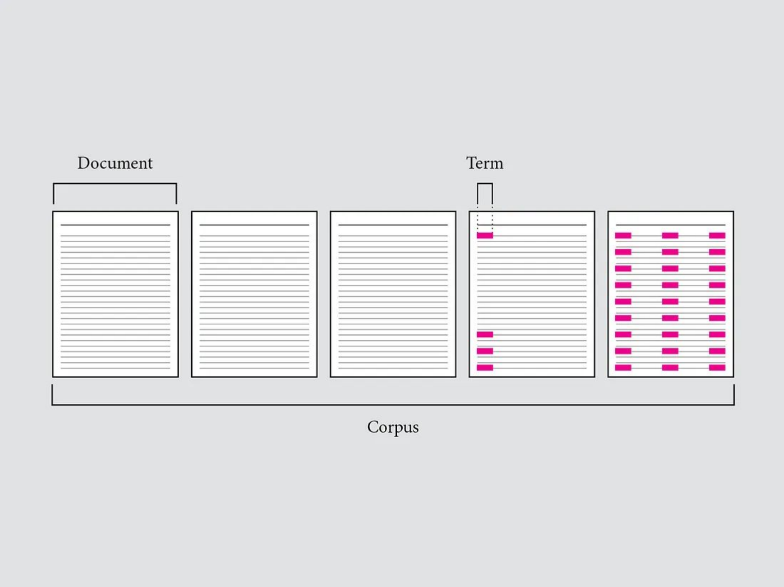
Natural Language Processing - Search

Relevant concepts

Natural Language Processing - Search

Relevant concepts

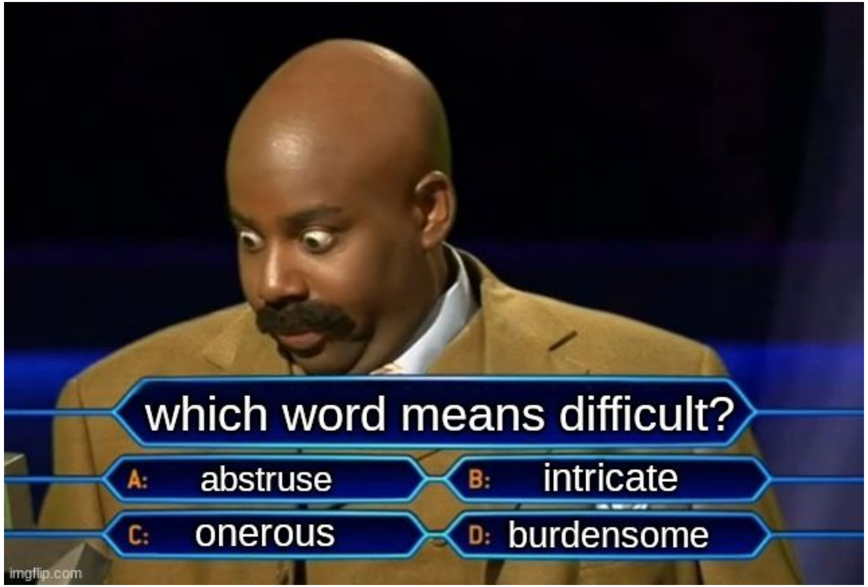
- Corpus



Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary

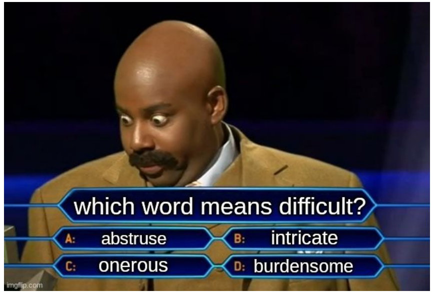


Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary

Vocabulary is hard.

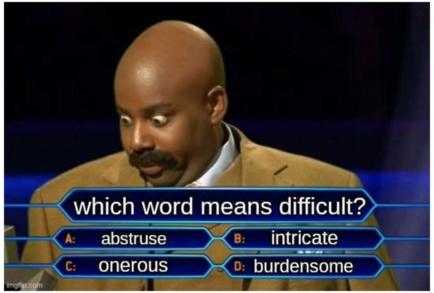


Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary

We need to have an explicit defined vocabulary.
A vocabulary is a set of terms (or **tokens**) known to the system.



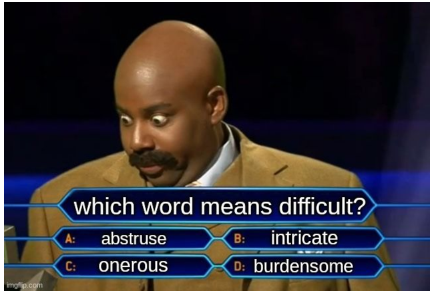
Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary

We need to have an explicit defined vocabulary.
A vocabulary is a set of terms (or **tokens**) known to the system.

Normally more important in syntactic systems.
However, proper tokenization is a big driver of modern LLMs.



Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words

Why don't scientists trust atoms?

Because they make up everything!

Word: Frequency

Why: 1
don't: 1
scientists: 1
trust: 1
atoms: 1
Because: 1
they: 1
make: 1
up: 1
everything!: 1

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures

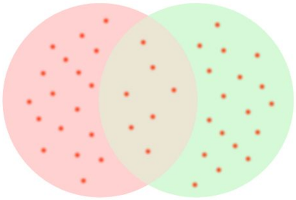
- Cosine Similarity
- Euclidean Distance
- Jaccard Similarity

https://myscale.com/blog/power-cosine-similarity-vs-euclidean-distance-explained/

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures



total elements in intersection

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

total elements in union i.e. Universal Set

$J(A,B)$ is thus probability of picking a random element from the universal set and finding that it is present in both the participating sets

similar to chances that you throw a dart and it hits the intersection

Jaccard Similarity Coefficient as Probability

Natural Language Processing - Search

Relevant concepts

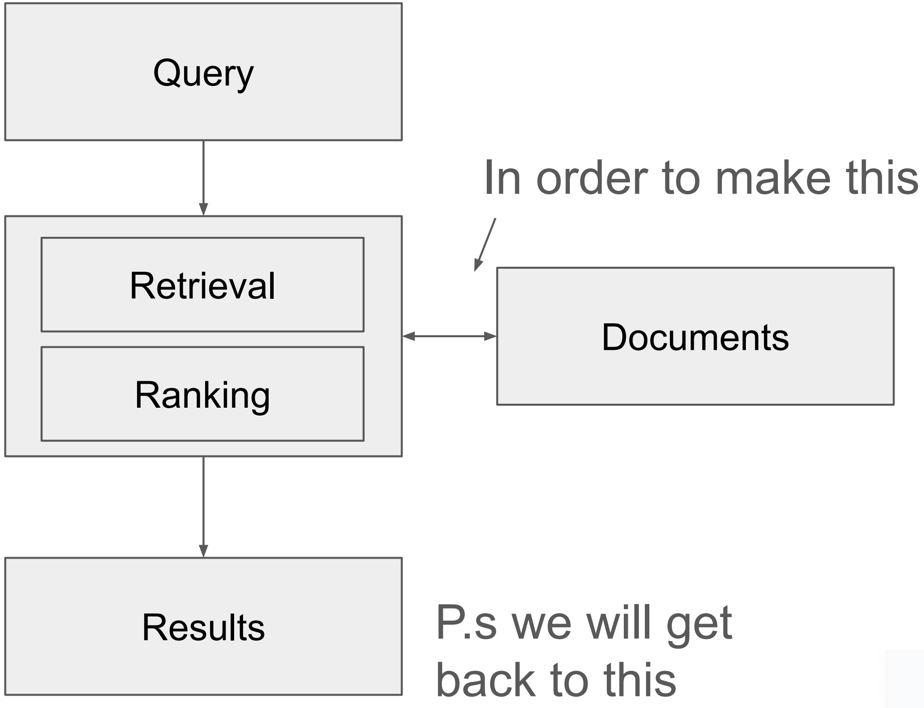
- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing

"... refers to the process of creating a searchable index or catalog of data"

Natural Language Processing - Search

Relevant concepts

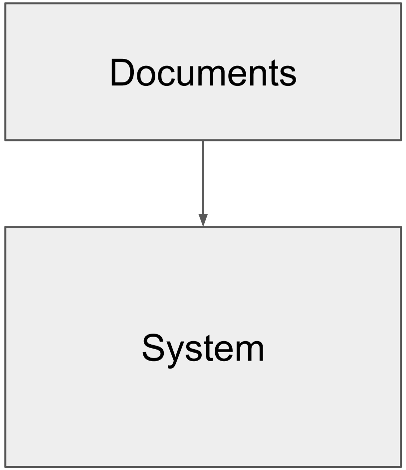
- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing



Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing



Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

Frequency of term t in document d

Total number of terms

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

$$idf\,(t, D) = log\left(\frac{N}{count\,(d \in D : t \in d)} \right)$$

Total number of documents

Number of documents d with term t.

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

$$idf\,(t, D) = log\,\left(\frac{N}{count\,(d \in D : t \in d)} \right)$$

$$tf\,idf\,(t, d, D) = tf\,(t, d)\ .\ idf\,(t, D)$$

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF

Evaluates the importance of a term (word) within a document relative to a collection of documents (a corpus)
I.e relative bag of words

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF
- Okapi BM25

Best Matching 25

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF
- Okapi BM25

Best Matching 25

… is a ranking function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity).

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF
- Okapi BM25

Best Matching 25

… is a ranking function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity).

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D) \cdot (k_1 + 1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

k1 and b are free parameters

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF
- Okapi BM25

Documents

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

Query

average document length

$$\text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

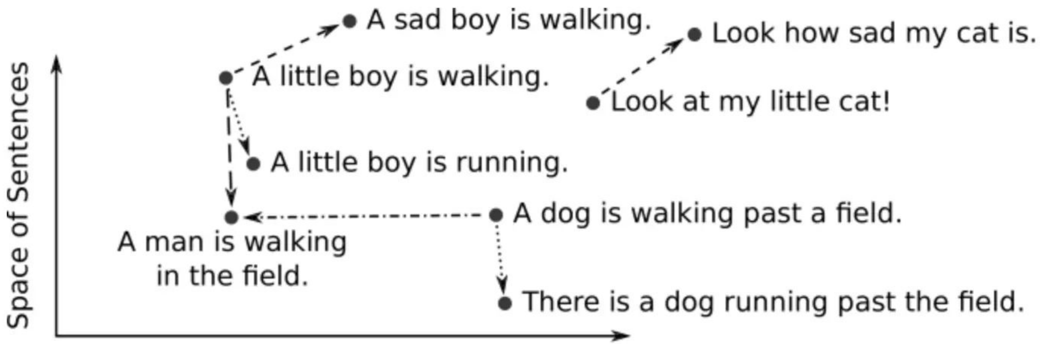
is the number of times that q_i occurs in the document D

Natural Language Processing - Search

Relevant concepts

- Corpus
- Vocabulary
- Bag of words
- Similarity Measures
- Indexing
- TF-IDF
- Okapi BM25

Lets focus on similarity



I want look for documents about sad cats.

How do I do that ?

Assume these can be documents like books, papers and so on.

Your turn

Your turn

Embeddings
Visualize

Information Retrieval

Ok, but how ?

Query

↓

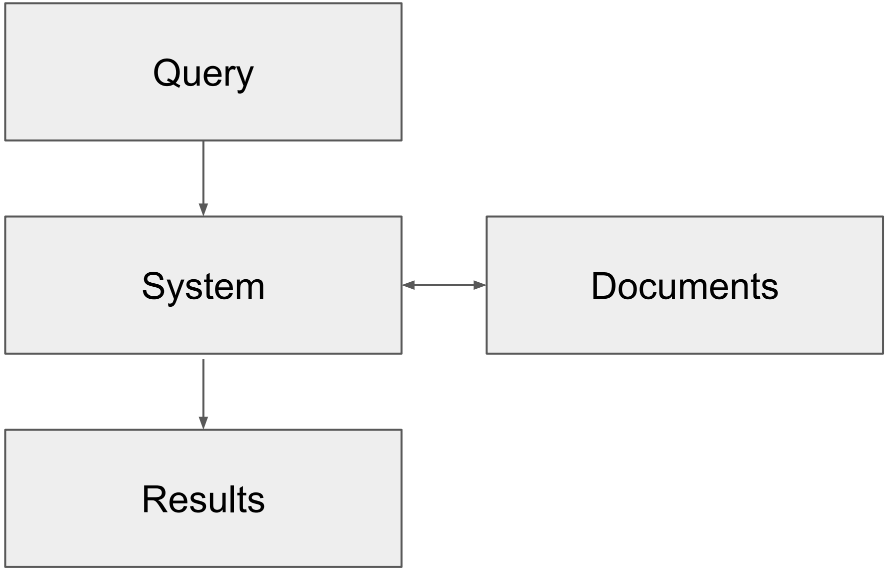
System

↓

Results

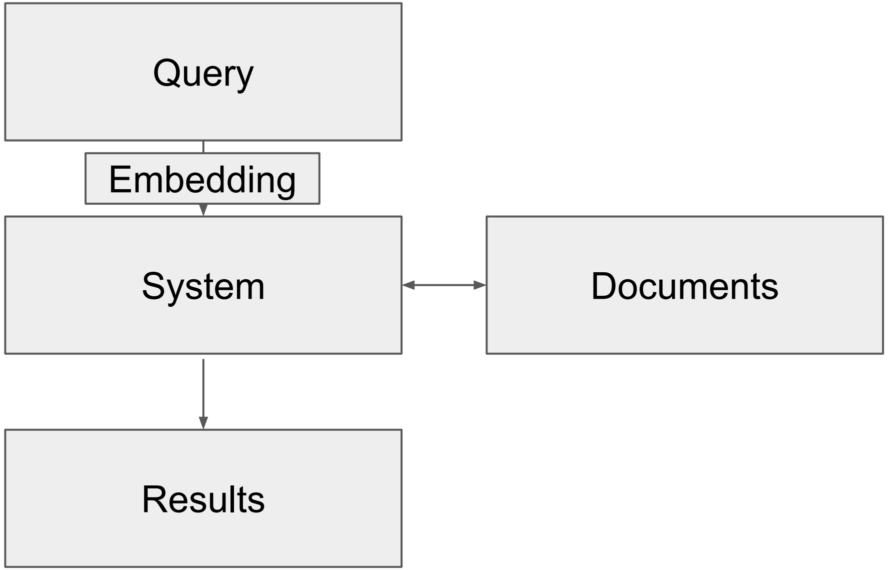
Information Retrieval

How do we search ?



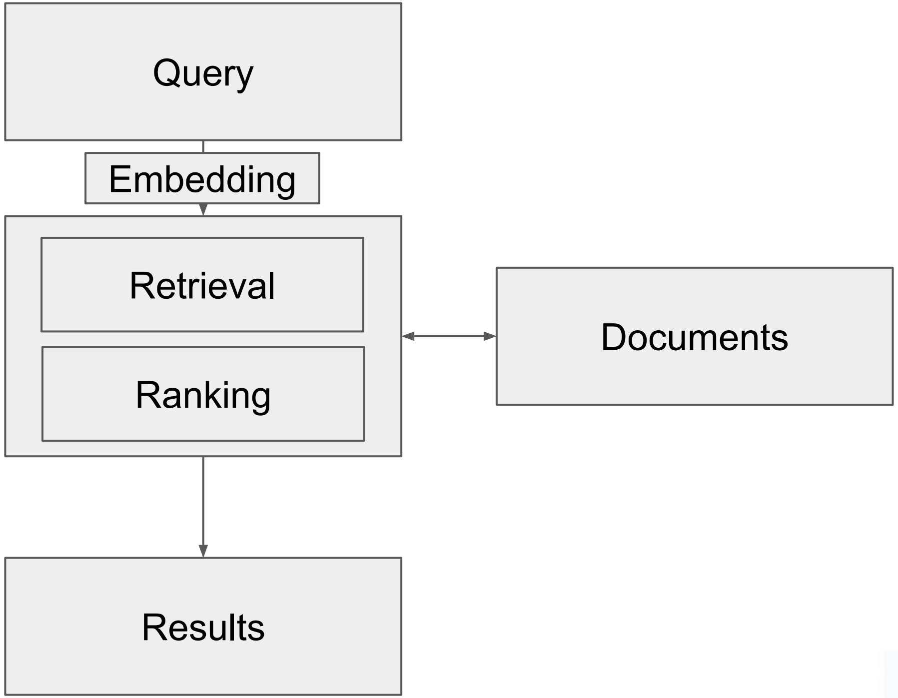
Information Retrieval

How do we search ?



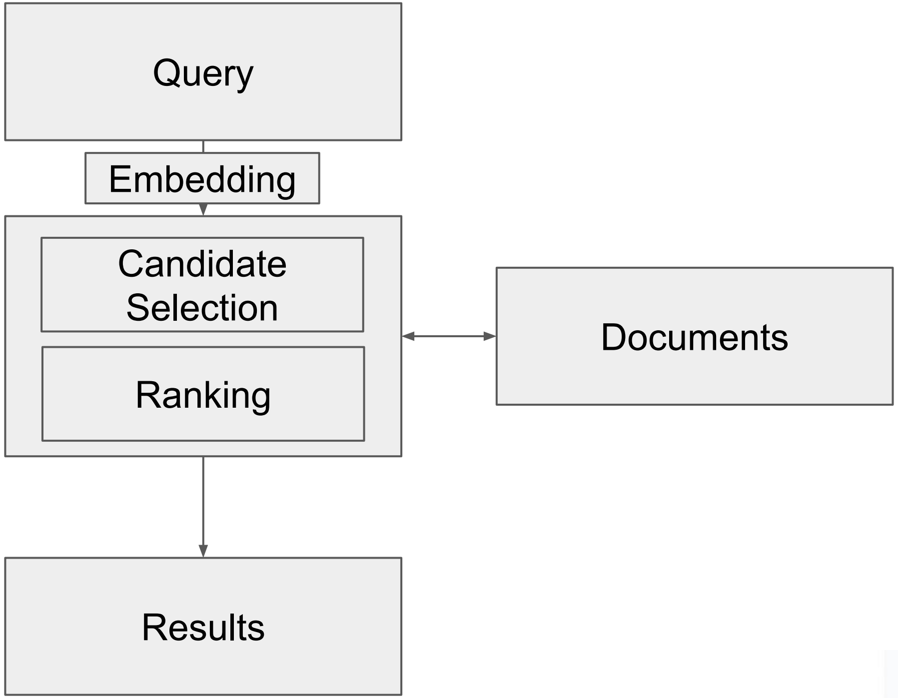
Information Retrieval

How do we search ?



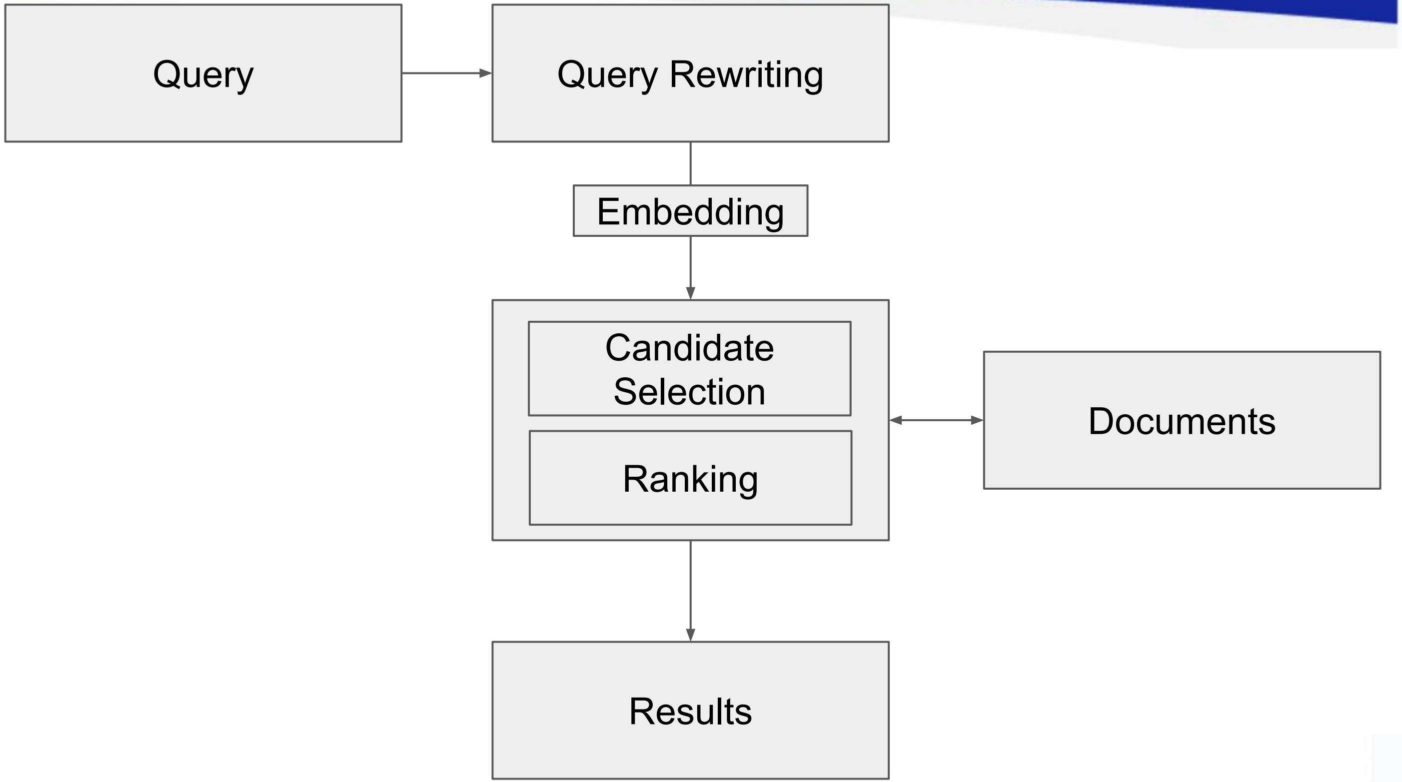
Information Retrieval

How do we search ?

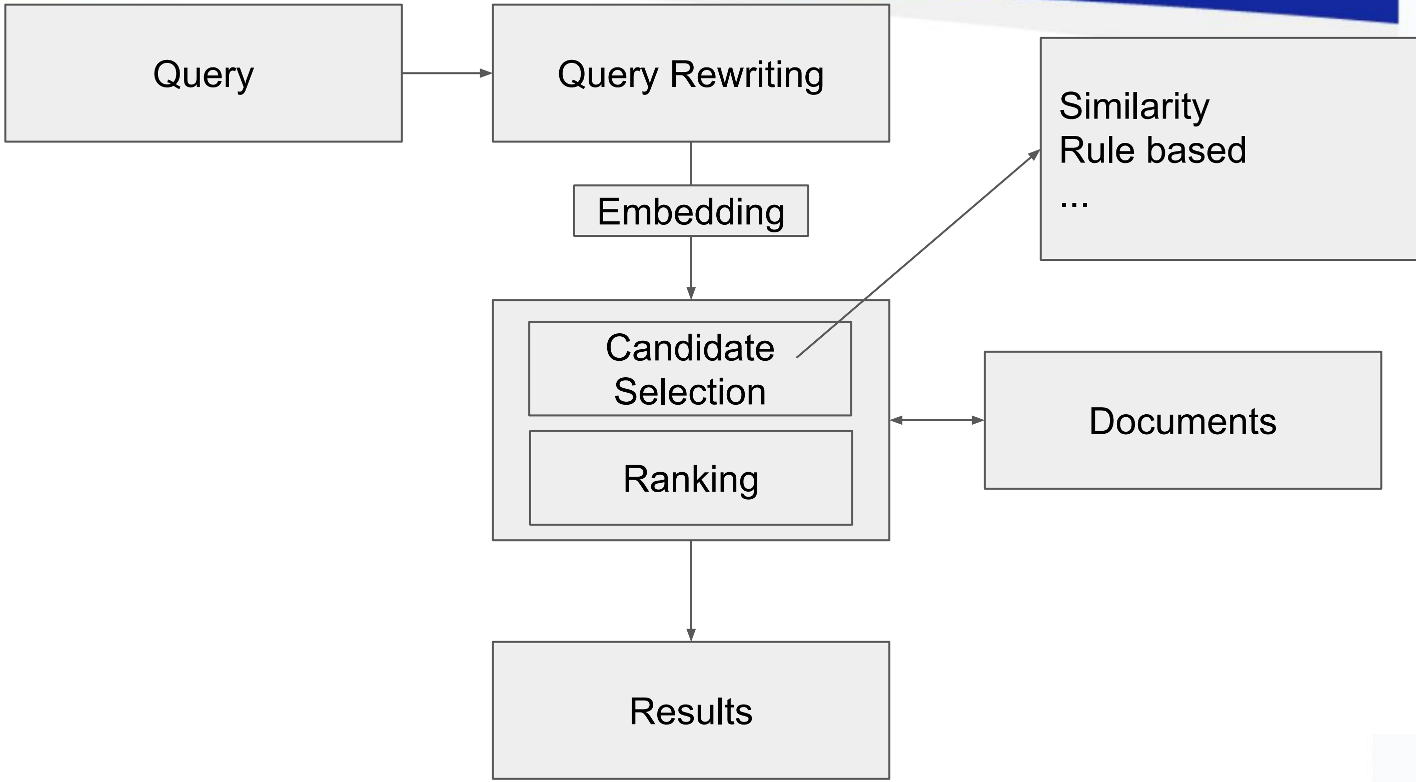


Information Retrieval

How do we search ?



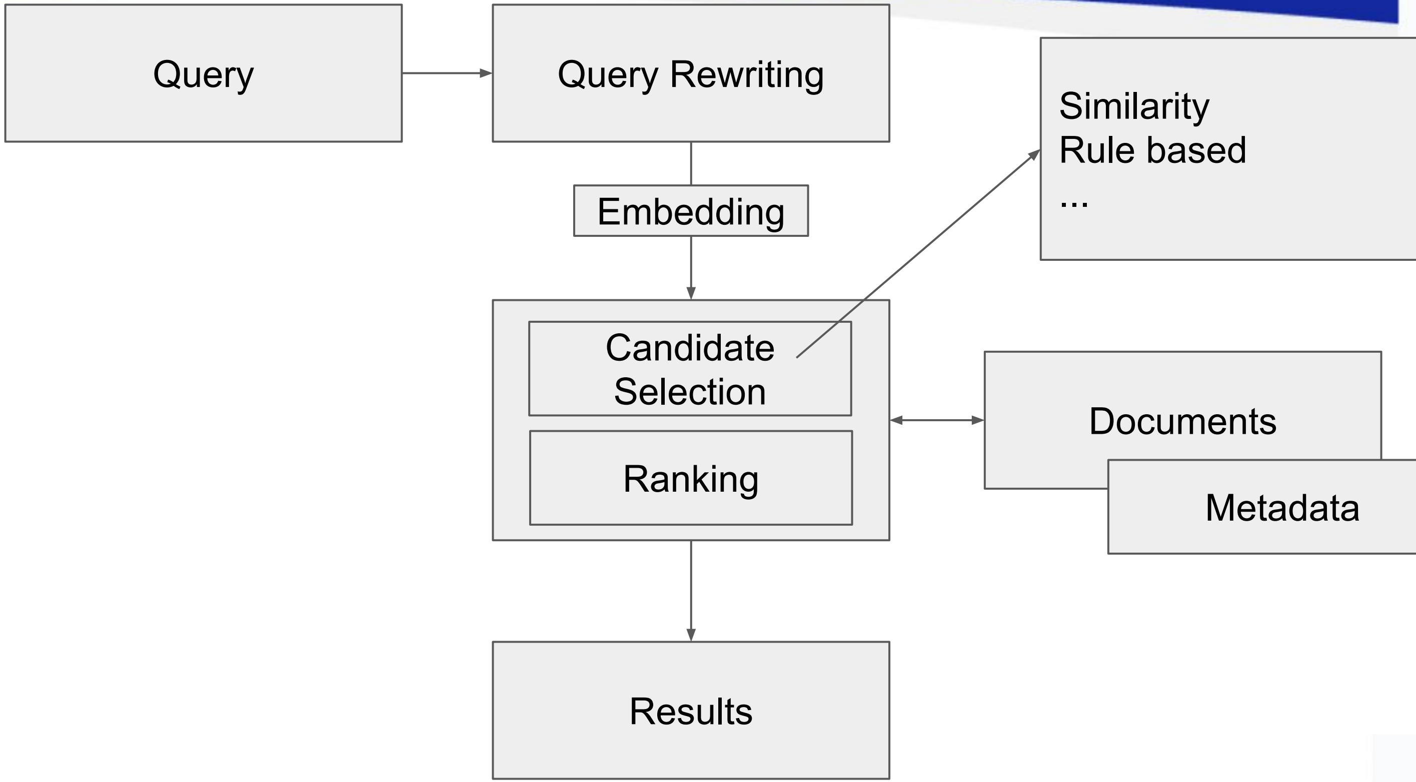
Information Retrieval



How do we search ?

Information Retrieval

How do we search ?



Your turn

Search engine

What is search really ?

What is search really ?

P.s recommending the right thing

Evaluation Metrics

- Precision (at K)
- Recall (at K)
- F1 Score
- NDCG

Evaluation Metrics

- Precision (at K)

Natural Language Processing - Search

Evaluation Metrics

- Precision (at K)

Precision = TP / (TP + FP)

True Positives
i.e The system classifies something as true and they are true

False Positives
i.e The system classifies something as true and they are false

Natural Language Processing - Search

Evaluation Metrics

- Precision (at K)

Precision = TP / (TP + FP)

All items

True Positives
i.e The system classifies them as true and they are true

False Positives
i.e The system classifies them as true and they are false

Evaluation Metrics

- Precision (at K)

Precision = TP / (TP + FP)

% of times the system got the true label right

Evaluation Metrics

- Precision (at K) is the proportion of recommended items in the top-k set that are relevant

Precision at K = TP at K / (TP at K + FP at K)

Natural Language Processing - Search

Evaluation Metrics

- Precision (at K) is the proportion of recommended items in the top-k set that are relevant

Precision at K = TP at K / (TP at K + FP at K)

Result A	Relevant
Result B	Relevant
Result C	Not Relevant

Natural Language Processing - Search

Evaluation Metrics

- Precision (at K) is the proportion of recommended items in the top-k set that are relevant

Precision at K = TP at K / (TP at K + FP at K)   Precision at 3 = ⅔ = 66%

Result A	Relevant
Result B	Relevant
Result C	Not Relevant

Evaluation Metrics

- Precision (at K) is the proportion of recommended items in the top-k set that are relevant

Precision at K = TP at K / (TP at K + FP at K)   Precision at 3 = ⅔ = 66%

Result A	Relevant
Result B	Not Relevant
Result C	Relevant

Natural Language Processing - Search

Evaluation Metrics

- Recall (at K)

Recall = TP/(TP+FN)

False Negatives
i.e The system classifies something as false and they are true

Evaluation Metrics

- Recall (at K) (also known as sensitivity) is the fraction of relevant instances that were retrieved.

Recall at K= TP at K / (TP at K + FN at K)
Relevant items = 5

Recall at 3 = 2/2 = 100%

Result A	Relevant
Result B	Not Relevant
Result C	Relevant

Evaluation Metrics

- Recall (at K) (also known as sensitivity) is the fraction of relevant instances that were retrieved.

Recall at K= TP at K / (All relevant items)
Relevant items = 5

Recall at 3 = ⅖ = 40%

Result A	Relevant
Result B	Not Relevant
Result C	Relevant

Evaluation Metrics

- Recall (at K) (also known as sensitivity) is the **fraction of relevant instances that were retrieved.**

Recall at K= TP at K / (All relevant items)
Relevant items = 5

Recall at 3 = $\frac{2}{5}$ = 40%

Result A	Relevant
Result B	Not Relevant
Result C	Relevant

Evaluation Metrics

- F1 Score is a metric that takes into account both precision and recall to provide a balanced evaluation of a system's performance

Evaluation Metrics

- F1 Score is a metric that takes into account both precision and recall to provide a balanced evaluation of a system's performance

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Evaluation Metrics

- F1 Score is a metric that takes into account both precision and recall to provide a balanced evaluation of a system's performance

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

… gives more weight to the lower of the two values. This means that if **either precision** or **recall is low** (i.e., the weaker of the two metrics), **the harmonic mean will also be low**, *reflecting the fact that the system is not performing well in at least one of these aspects*.
It penalizes systems that have an extreme imbalance between precision and recall.

Evaluation Metrics

- F1 Score at K is a metric that takes into account both precision and recall to provide a balanced evaluation of a system's performance

F1 Score at K = 2 * (Precision at K * Recall at K) / (Precision at K + Recall at K)

… gives more weight to the lower of the two values. This means that if **either precision** or **recall is low** (i.e., the weaker of the two metrics), **the harmonic mean will also be low**, *reflecting the fact that the system is not performing well in at least one of these aspects*.
It penalizes systems that have an extreme imbalance between precision and recall.

Natural Language Processing - Search

Evaluation Metrics

- Normalized Discounted Cumulative Gain (NDCG)

Evaluation Metrics

- Normalized Discounted Cumulative Gain (NDCG) assesses how well the top-ranked items in a list align with the preferences or relevance judgments of users, i.e order matters.

Evaluation Metrics

- Normalized **Discounted Cumulative Gain** (N**DCG**).

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} =$$

Evaluation Metrics

- Normalized **Discounted Cumulative Gain** (N**DCG**).

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} =$$

DCGp is the DCG at position p.

Evaluation Metrics

- Normalized **Discounted Cumulative Gain** (N**DCG**).

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

- DCG_p is the DCG at position p.
- rel_i is the relevance score of the item at position i in the ranking list (typically a non-negative number, where higher values represent higher relevance).

Evaluation Metrics

- Normalized **Discounted Cumulative Gain** (N**DCG**).

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

  - DCG_p is the DCG at position p.
  - rel_i is the relevance score of the item at position i in the ranking list (typically a non-negative number, where higher values represent higher relevance).
  - Log is used to produce a smooth reduction

Natural Language Processing - Search

Evaluation Metrics

- Normalized **Discounted Cumulative Gain** (N**DCG**).

  The premise of DCG is that **highly relevant documents appearing lower in a search result list should be penalized** as the graded relevance value is reduced logarithmically proportional to the position of the result.

Evaluation Metrics

- **Normalized** Discounted Cumulative Gain (**N**DCG).

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

$$\mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)}$$

represents the list of relevant documents (ordered by their relevance) in the corpus up to position p.

ideal discounted cumulative gain

Your turn

More search
Evaluation

Your turn

More search Evaluation

```
random.seed(42)
idx = random.sample(rel_set.keys(),1)[0]

print('Query ID %s ==>' % idx, qry_set[idx])
rel_docs = rel_set[idx]
print('Documents relevants to Query ID %s' % idx, rel_docs)
sample_document_idx = random.sample(rel_docs,1)[0]
print('Document ID %s ==>' % sample_document_idx, doc_set[sample_document_idx])
```

```
Query ID 14 ==> How much do information retrieval and dissemination systems, as well as automa
ted libraries, cost? Are they worth it to the researcher and to industry?
Documents relevants to Query ID 14 [17, 26, 35, 48, 55, 58, 66, 73, 82, 125, 157, 163, 166, 19
1, 213, 221, 222, 249, 280, 291, 294, 298, 306, 330, 335, 337, 347, 364, 365, 366, 367, 371, 3
80, 445, 457, 464, 465, 481, 489, 490, 494, 496, 506, 519, 527, 590, 593, 622, 628, 638, 689,
719, 722, 723, 726, 727, 730, 778, 821, 833, 838, 847, 848, 864, 871, 896, 1099, 1160, 1247, 1
304, 1352, 1357, 1362, 1365, 1367, 1370, 1371, 1373, 1374, 1375, 1376, 1409]
Document ID 48 ==> Adaptive Information Dissemination Sage, C.R. Anderson, R.R. Fitzwater, D.
R. Computer dissemination of information offers significant advantages over manual disseminati
on because the computer can use strategies that are impractical and in some cases impossible f
or a human.. This paper describes the Ames Laboratory Selective Dissemination of Information s
ystem with emphasis on the effectiveness of user feedback.. The system will accept any documen
t, abstract, keyword, etc., in a KWIC or Science Citation Index Source format.. User profiles
consist of words or word clusters each with an initially assigned significance value.. These v
alues are used in making the decision to notify a user that he may be interested in a particul
ar document.. According to responses, the significance values are increased or decreased and q
uickly attain an equilibrium which accurately describes the user's interests.. The system is e
conomical compared to other existing SDI systems and human intervention is negligible except f
or adding and deleting profile entries..
```

Your turn

More search
Evaluation

For another time

Wrap up

Embeddings exist in most systems nowadays.
Retrieval, recommendations and all advanced machine learning models use embeddings.
Word embeddings can be syntactic or semantic, there are trade-offs.
Embeddings are as or even more important than the algorithm that might use them.
Vector databases are very powerful.

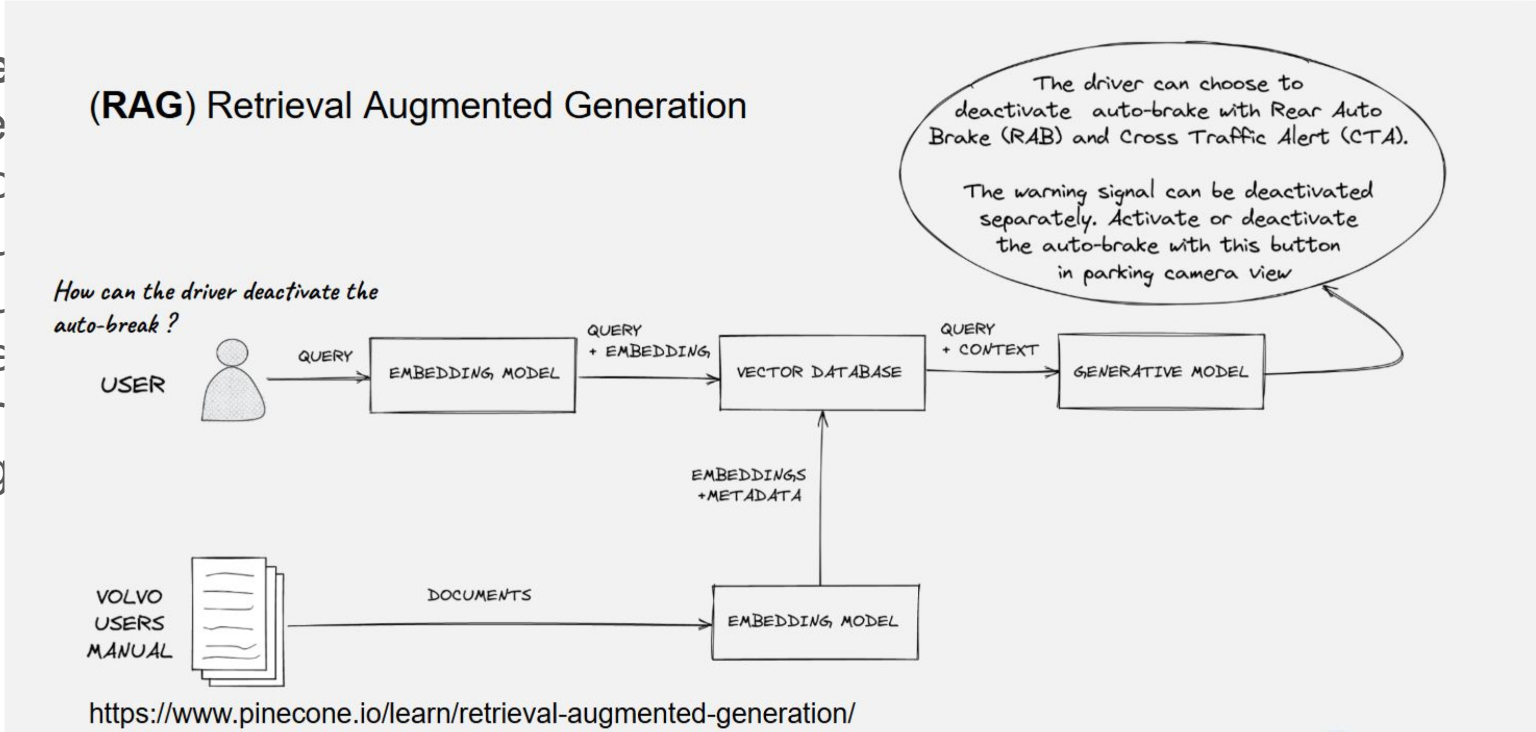
Today.

Relevant concepts
Vector Databases
Complex Indexing (Chunk, Summarization)
Query Expansion
Query Rewriting
Retrieval Augmented Generation
Graph Retrieval
Agentic Retrieval i.e ‘Reasoning and routing’

Buzzwords
Chatbots
Agents
Functional Calling
Enterprise Knowledge Search

Today.

Re
Ve
Co
Qu
Qu
Re
Gr
Ag

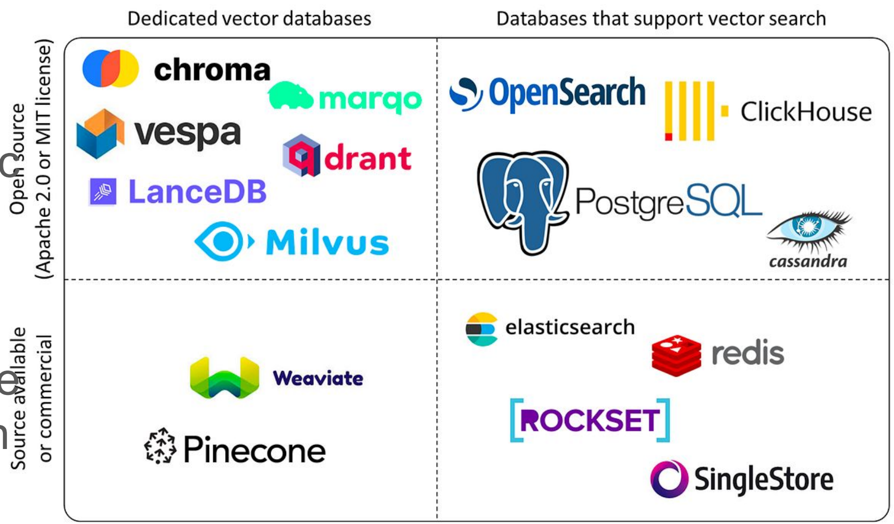


rch

Information Retrieval

Challenges

1. Go through the resources.
2. Create a plan on how you would build an actual retrieval system for a specific use-case, understand possible risks and how to evaluate it
3. Use any of these tools and platforms to build your specialized search engine for …. your study notes, your research papers, …
4. Build a **RAG** system on top of it.
5. Formally evaluate it



Natural Language Processing

Next up: