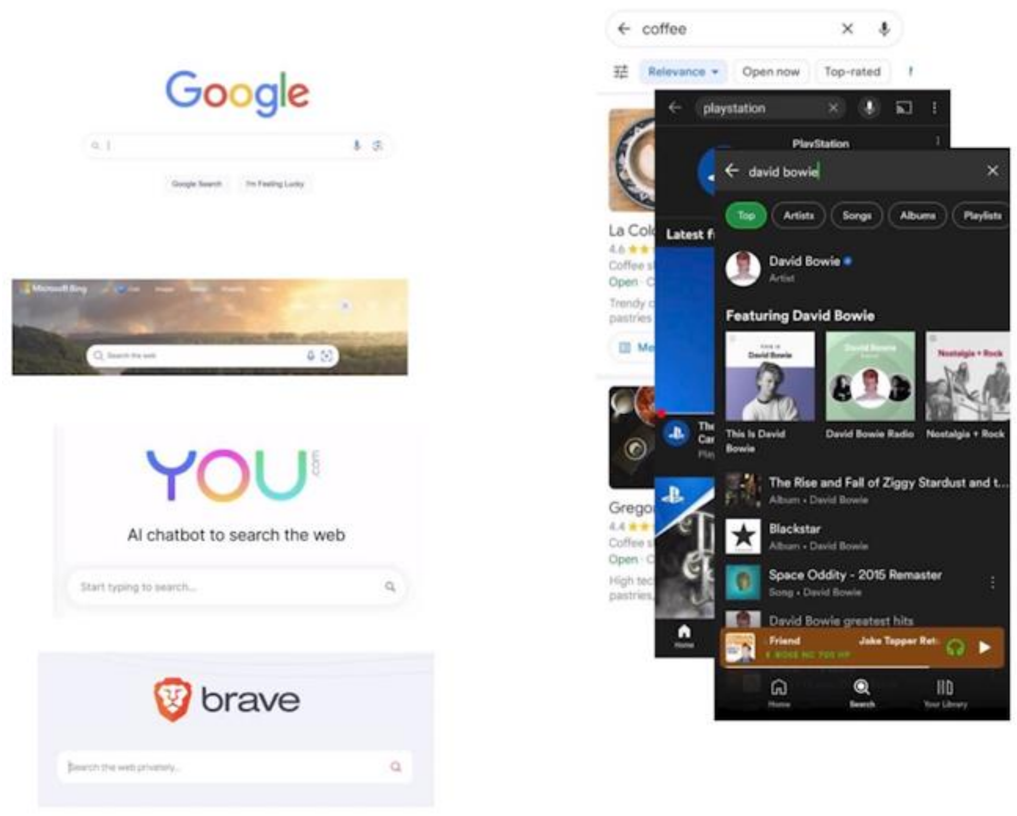
Contents

Motivation

Whether we want to sift through millions of social media posts, extract information from reports of medical trials and academic research, or simply retrieve relevant texts from thousands of documents, reports, and notes generated by an organization as a part of its daily operation, we would need an information retrieval system with the capability to match a query and its search intent to the relevant documents.



The traditional approach for information retrieval, such as BM25/TF-IDF, relies on word frequencies within indexed documents and on key terms within the query to estimate the relevance of said documents to the query. This approach has two key limitations that affect its accuracy. Documents that do not contain the keywords but include terms that are semantically similar to those keywords may be missed. For a pool of documents containing different languages, and especially languages with different scripts and alphabets, the keyword approach would fail.

With the language model approach, both queries and documents are embedded in a common latent space, and we can build a semantic matching function with the measure of relevance defined as the cosine similarity between the vector embeddings, regardless of the language, alphabet script, or presence of specific key terms. The aforementioned approach is called bi-encoding.

Cosine-similarity

Cosine similarity is a numerical measure of similarity between two vectors, widely used in various fields such as natural language processing and information retrieval. It calculates the cosine of the angle between the vectors, providing a value between -1 and 1, where higher values indicate greater similarity. Its scale-invariant and directional independence properties make it particularly valuable for comparing documents, text, and feature vectors in recommendation systems.

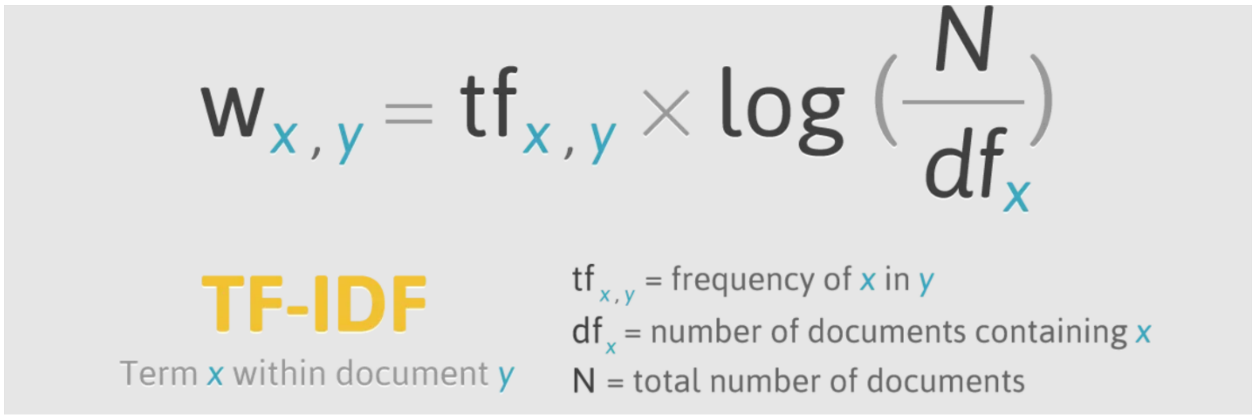
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

TF-IDF

TF-IDF, which stands for Term Frequency-Inverse Document Frequency, is a numerical statistic that reflects the importance of a word or term within a document relative to a collection of documents (corpus). TF-IDF is commonly used in information retrieval and text mining for ranking the relevance of documents to a query.

High TF-IDF: A high TF-IDF score indicates that the term is important in the given document and rare across the corpus.

Low TF-IDF: A low TF-IDF score suggests that the term is either common across all documents or not present often in the document of interest.



$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

BM25 algorithm

BM25 (Best Matching 25) is a ranking function used in information retrieval and text mining. BM25 is commonly used for scoring and ranking documents based on their relevance to a query. It is particularly effective for keyword search and is widely employed in search engines.

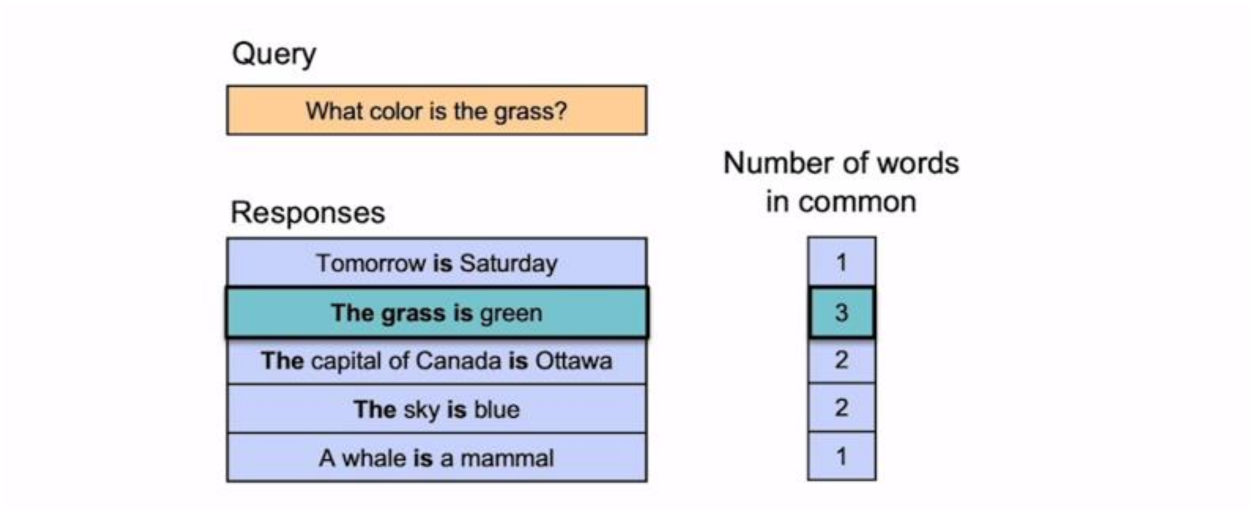
The BM25 score for a document D with respect to a query Q is calculated as the sum of the scores for individual query terms. The formula for calculating the BM25 score is as follows:

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

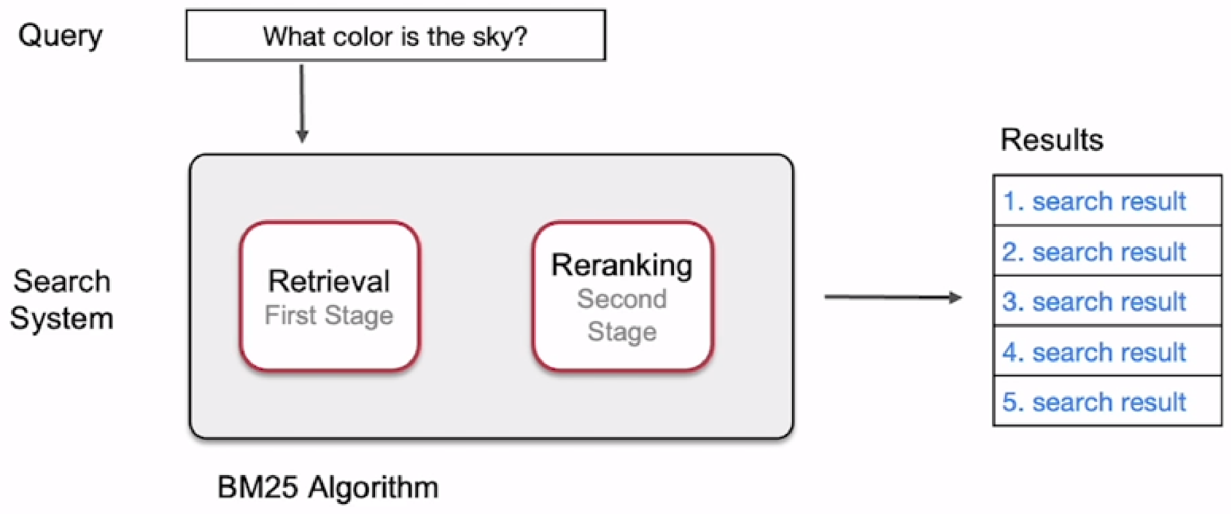
In this formula, IDF(q) represents the inverse document frequency of the query term q, TF(q, D) denotes the modified term frequency of term q in document D, |D| represents the length of document D, and avgdl is the average document length in the corpus. Parameters k1 and b are tunable constants that control the impact of term frequency saturation and document length normalization, respectively. BM25 takes into account both term frequency and document length normalization, which helps address the issue of document length bias.

Keyword Search

Keyword search refers to the process of searching for specific words or phrases within a set of documents, databases, or information sources to retrieve relevant information. The goal is to identify and retrieve documents that contain the specified keywords.



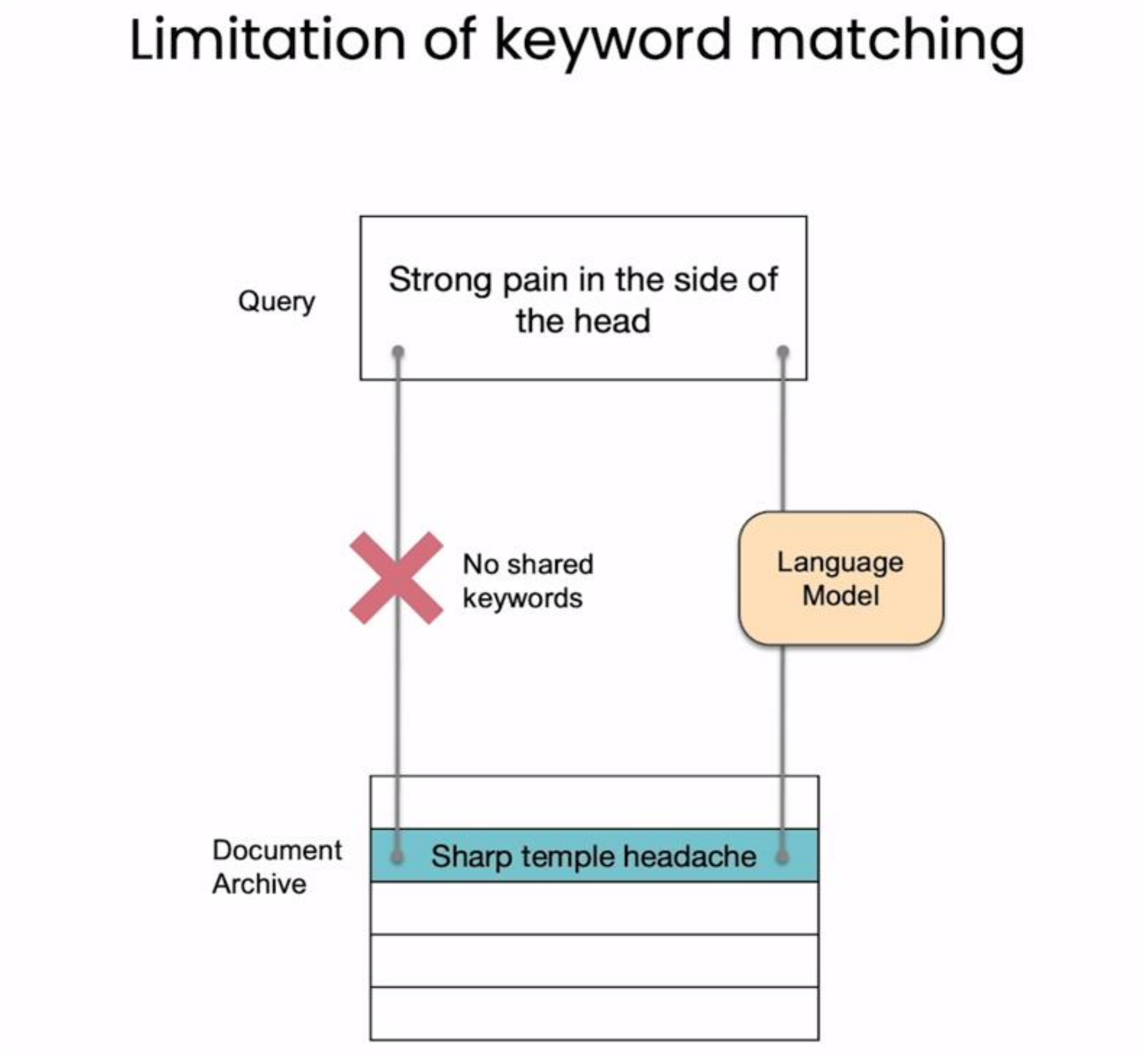
Keyword Search Internals



Inverted Index

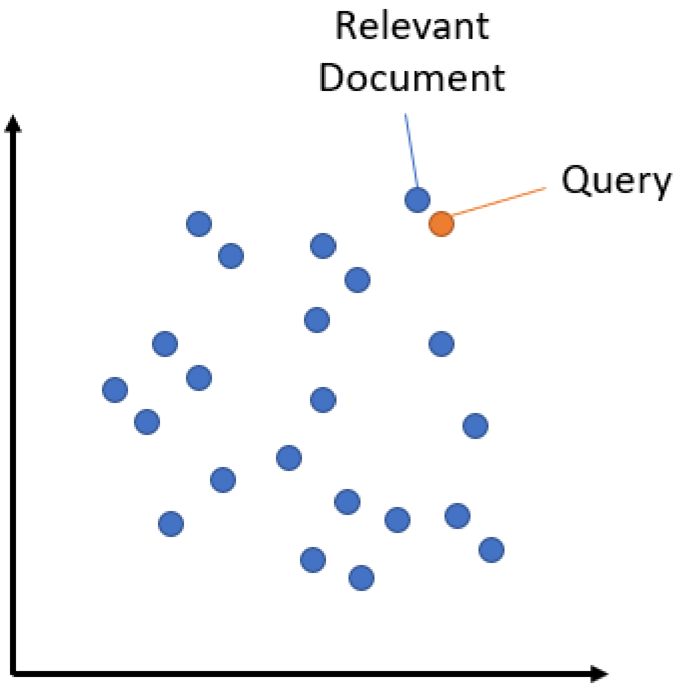
Keyword	Document IDs
abacus	1, 38, 18
...	
Color	23, 804
...	
Sky	804, 922

Keyword matching is sensitive to the precise wording of queries, potentially overlooking relevant content due to variations or synonyms. It lacks contextual awareness, leading to imprecise results as it may not distinguish between different meanings of a keyword in diverse contexts.The method struggles to capture semantic nuances, limiting its effectiveness in understanding deeper connections and meanings beyond literal keyword presence.



Semantic Search

Semantic search seeks to improve search accuracy by understanding the content of the search query. In contrast to traditional search engines which only find documents based on lexical matches, semantic search can also find synonyms. The idea behind semantic search is to embed all entries in your corpus, whether they be sentences, paragraphs, or documents, into a vector space.At search time, the query is embedded into the same vector space and the closest embeddings from your corpus are found. These entries should have a high semantic overlap with the query.



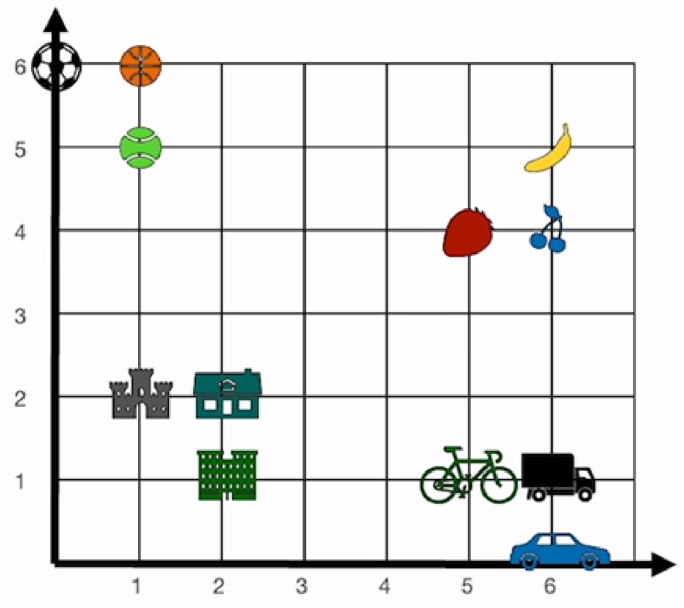
Embeddings

Embeddings are mathematical representations of objects, such as words, phrases, or documents, in a continuous vector space. The concept of embeddings is fundamental in natural language processing (NLP) and machine learning, particularly for tasks like semantic analysis, sentiment analysis, and information retrieval. Here are key points about embeddings:

Vector Representation: Embeddings represent objects as vectors, typically in a high-dimensional space. Each dimension in the vector captures a specific feature or aspect of the object, and the combination of these dimensions defines the position of the object in the vector space.

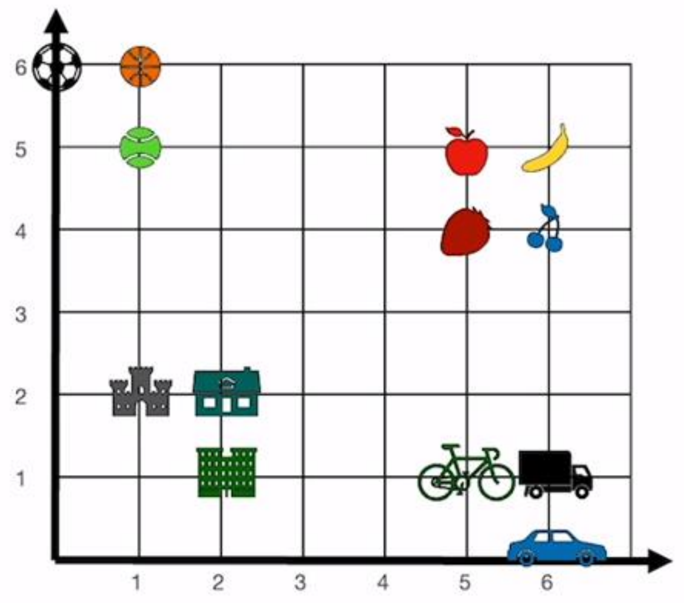
Semantic Similarity: Embeddings aim to capture semantic relationships between objects. In the context of words, similar words are represented by vectors that are close to each other in the vector space. This enables algorithms to understand the semantic meaning and context of words.

Quiz: Where would you put the word "apple"?



Word	Numbers	
Apple	?	?
Banana	6	5
Strawberry	5	4
Cherry	6	4
Soccer	0	6
Basketball	1	6
Tennis	1	5
Castle	1	2
House	2	2
Building	2	1
Bicycle	5	1
Truck	6	1
Car	6	0

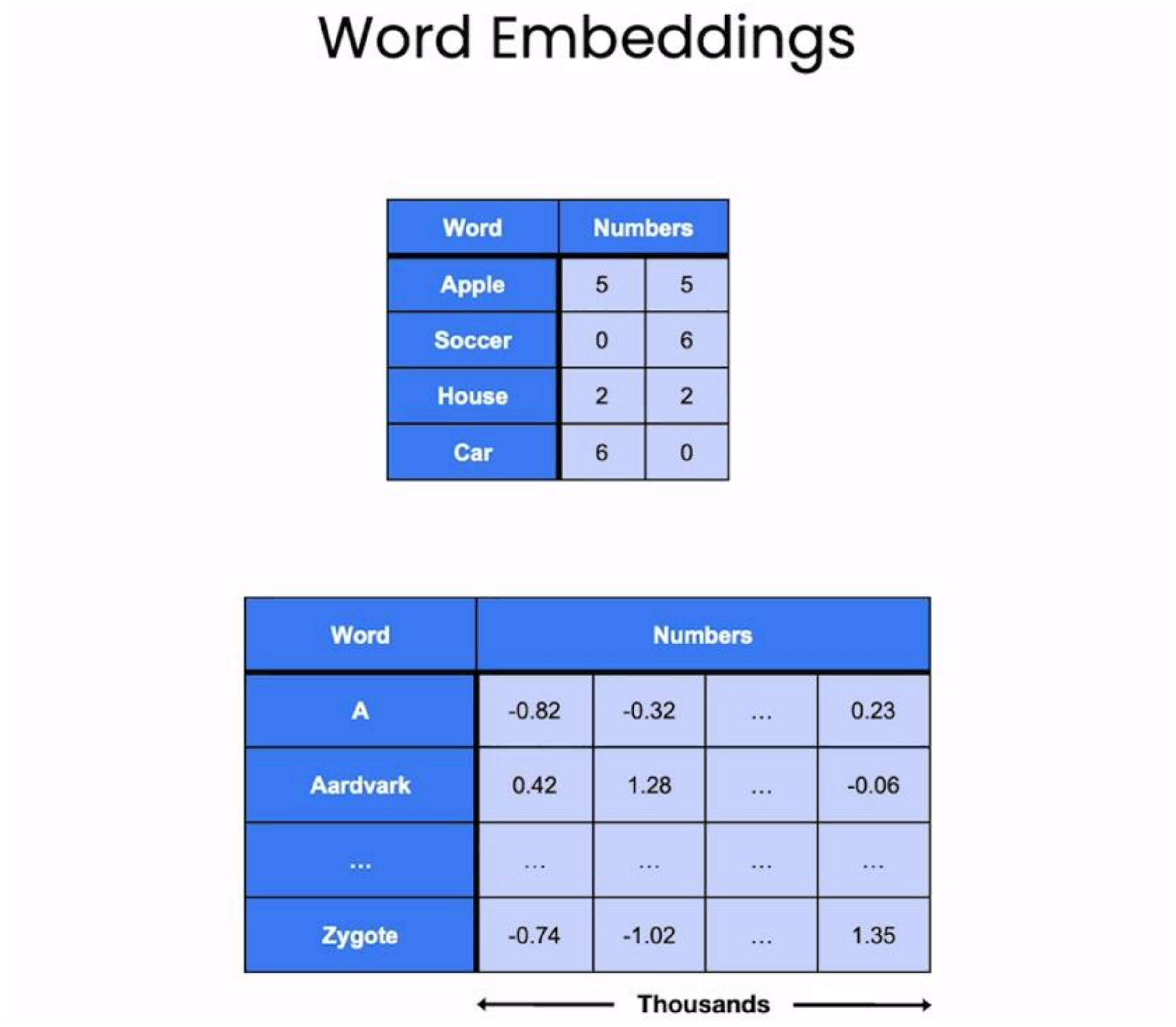
Quiz: Where would you put the word "apple"?



Word	Numbers	
Apple	5	5
Banana	6	5
Strawberry	5	4
Cherry	6	4
Soccer	0	6
Basketball	1	6
Tennis	1	5
Castle	1	2
House	2	2
Building	2	1
Bicycle	5	1
Truck	6	1
Car	6	0

Word embedding in NLP is an important term that is used for representing words for text analysis in the form of real-valued vectors. It is an advancement in NLP that has improved the ability of computers to understand text-based content in a better way. It is considered one of the most significant breakthroughs of deep learning for solving challenging natural language processing problems.

In this approach, words and documents are represented in the form of numeric vectors allowing similar words to have similar vector representations. The extracted features are fed into a machine learning model so as to work with text data and preserve the semantic and syntactic information. This information once received in its converted form is used by NLP algorithms that easily digest these learned representations and process textual information.

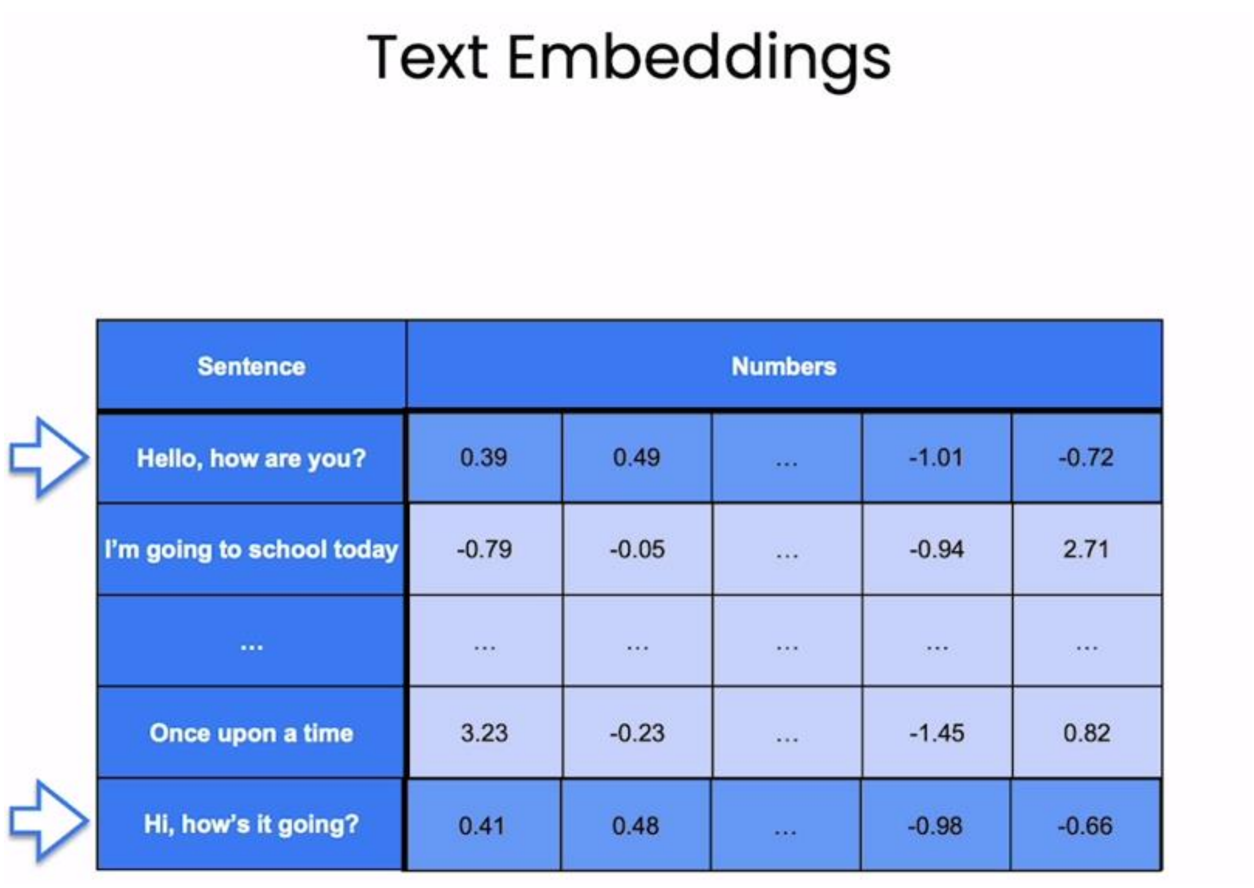


Word Embeddings

Word	Numbers	
Apple	5	5
Soccer	0	6
House	2	2
Car	6	0

Word	Numbers			
A	-0.82	-0.32	...	0.23
Aardvark	0.42	1.28	...	-0.06
...	...	...	...	...
Zygote	-0.74	-1.02	...	1.35

Text embeddings mitigate certain drawbacks inherent in word embeddings by considering the broader context of phrases, sentences, or documents. Unlike word embeddings, which lack context, text embeddings capture the semantic nuances within a given linguistic context, addressing issues of ambiguity and polysemy associated with individual words. They excel in resolving long-range dependencies and intricate sentence structures, offering a more comprehensive representation of the overall meaning. Moreover, text embeddings, especially those derived from transformer models like BERT, demonstrate superior performance in various natural language processing tasks, surpassing the limitations of word embeddings in tasks such as sentiment analysis, question answering, and document summarization. By leveraging contextual information, text embeddings enhance the nuanced understanding of language, contributing to more accurate and context-aware language processing models.

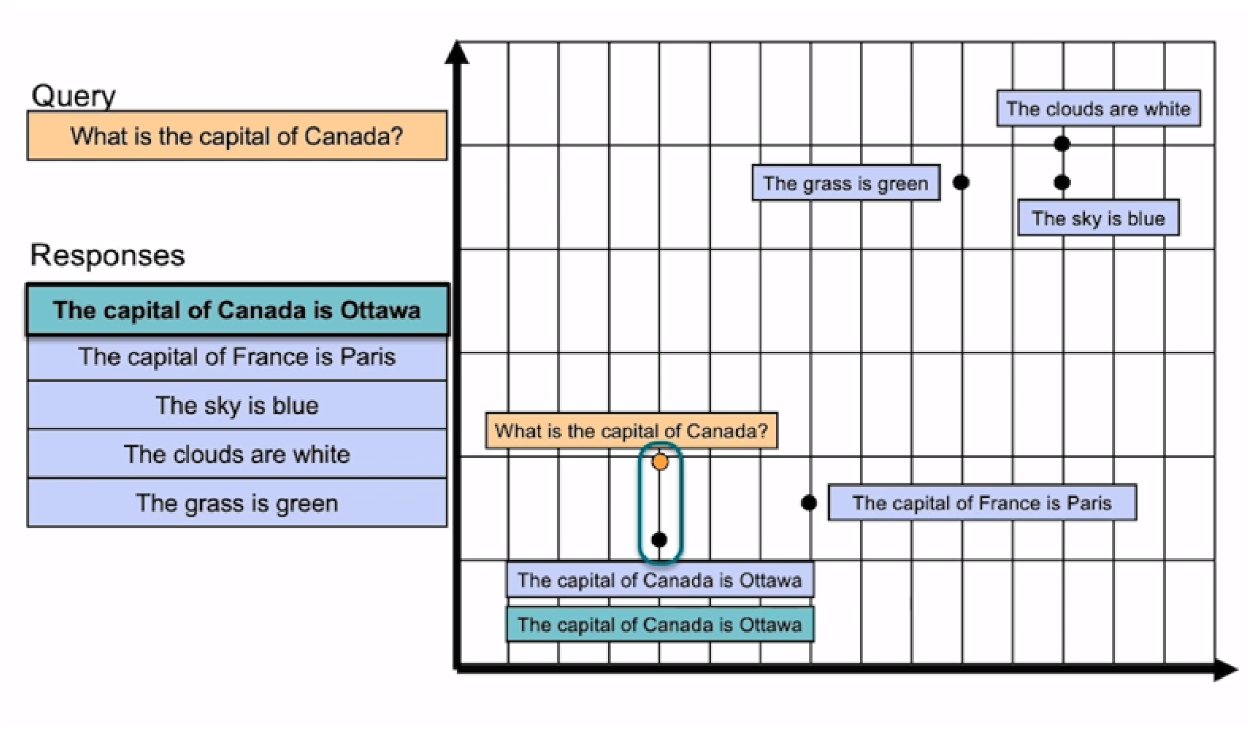


Text Embeddings

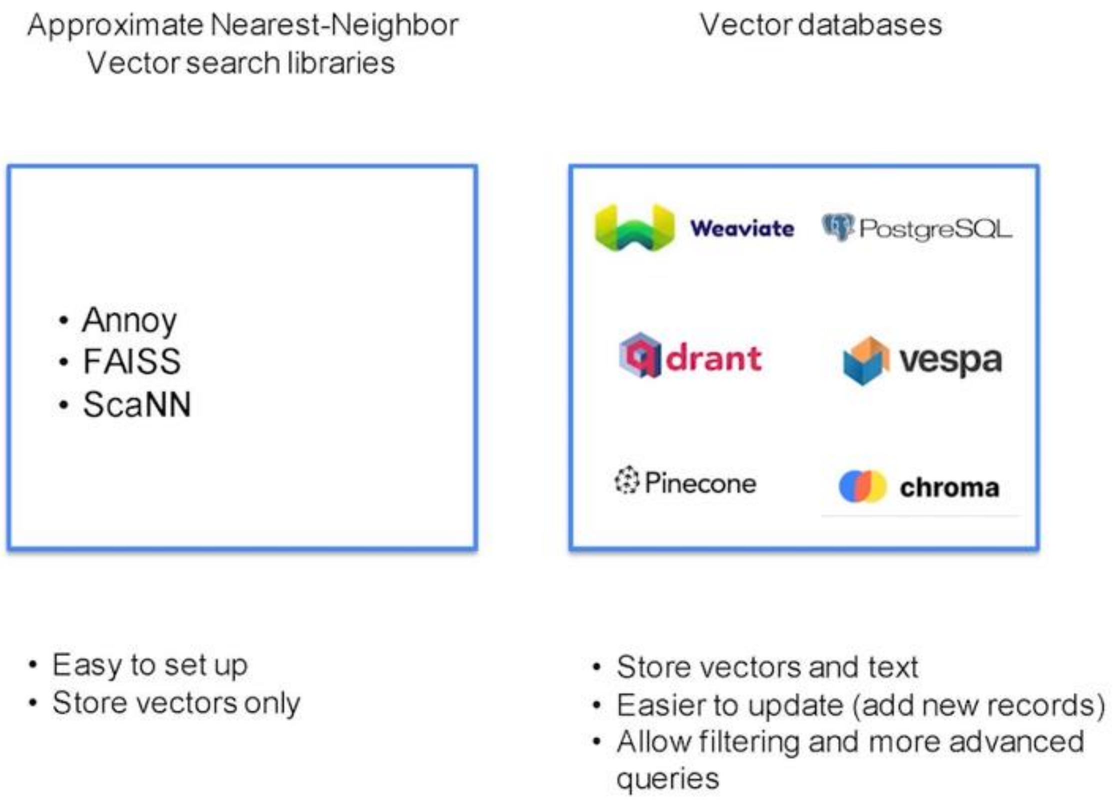
Sentence	Numbers				
Hello, how are you?	0.39	0.49	...	-1.01	-0.72
I'm going to school today	-0.79	-0.05	...	-0.94	2.71
...	...	...	...	...	...
Once upon a time	3.23	-0.23	...	-1.45	0.82
Hi, how's it going?	0.41	0.48	...	-0.98	-0.66

Dense Retrieval

Dense retrieval is a paradigm in information retrieval that focuses on encoding and retrieving information using dense vector representations. In this approach, both queries and documents are represented as dense vectors in a high-dimensional space, allowing for efficient and accurate similarity computations. Unlike traditional sparse representations, such as bag-of-words, dense retrieval captures semantic relationships and contextual information, leading to more nuanced and context-aware retrieval. Dense retrieval models, often based on neural networks, enable the modeling of complex interactions between words and phrases, contributing to improved relevance in search results. These models leverage pre-trained embeddings or are fine-tuned on large-scale datasets to enhance their understanding of language nuances. Dense retrieval has demonstrated success in various applications, including question answering, document ranking, and information retrieval tasks, offering a promising approach for handling the challenges of large-scale and diverse datasets. Its ability to capture intricate semantic relationships positions dense retrieval as a powerful tool in modern information retrieval systems.


Query
What is the capital of Canada?
Responses
The capital of Canada is Ottawa
The capital of France is Paris
The sky is blue
The clouds are white
The grass is green
The clouds are white
The grass is green
The sky is blue
What is the capital of Canada?
The capital of France is Paris
The capital of Canada is Ottawa
The capital of Canada is Ottawa

ANN vector search vs Vector databases



Approximate Nearest-Neighbor Vector search libraries

- Annoy
- FAISS
- ScaNN

- Easy to set up
- Store vectors only

Vector databases

Weaviate
PostgreSQL
qdrant
vespa
Pinecone
chroma

- Store vectors and text
- Easier to update (add new records)
- Allow filtering and more advanced queries

References

1. https://www.deeplearning.ai/short-courses/large-language-models-semantic-search/
2. https://medium.com/@abhishekranjandev/building-a-semantic-search-engine-with-machine-learning-and-jupyter-notebooks-fbcb15b538c5
3. https://www.kaggle.com/code/ajitrajput/semantic-search-engine-using-nlp/notebook?source=post_page-----cec19e8cfa7e--------------------------------