

THE UNIVERSITY of EDINBURGH

# Text Technologies for Data Science

**INFR11145**

# Term Representation

Instructor:
**Walid Magdy**

28-Oct-2026

1

## Lecture Objectives

- Learn about
  - Local representation
  - Distributed representation
  - Embeddings
  - Dense Bi Encoder
- Implement:
  - Whatever you can in CW3

## Query Expansion

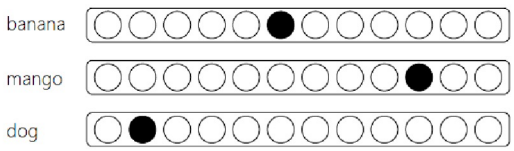
- QE can solve some issue in term matching
- Still can lead to query drift!
- Inconsistent performance!

- How to deal with terms of the same meaning to my query?

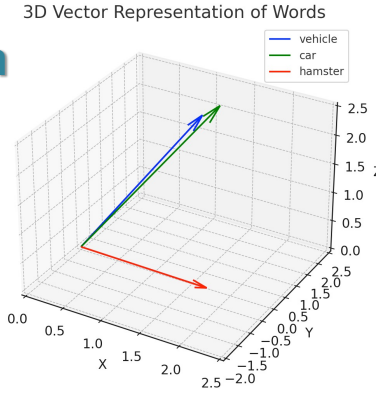


## Local Term Representation

- So far, we rely on "term" match
- Preprocessing solves good portion of the matching problem (e.g. *retrieving, retrieval, retrieved, retrieves*)
- Fails with similar meanings (*Car ≠ Vehicle*)
- Terms so far have "local" or "lexical" representation
- A term in a vector is 1 for itself only and 0 for others (sparse representation)



# Distributed Representation




- So far, a term is a definite term
  - Car ≠ Vehicle ≠ Hamester
  - Local Representation
- **Semantic representation**
  - A term, phrase, or a paragraph can be presented as a vector
  - Objective: terms/sentences with closer meaning get higher dot product that terms/sentences with different meanings
  - Ideally: multimodel → sentences and images of similar content shall get closer representation



# Sparse (Lexical) vs Dense (Semantic)

**Sparse (BM25 / TF-IDF)**

**Representation:** High-dimensional sparse vectors (vocabulary size)

**Matching Mechanism:** Exact string & token occurrence

**Index Structure:** Inverted index

**Primary Bottleneck:** Vocabulary mismatch & synonyms

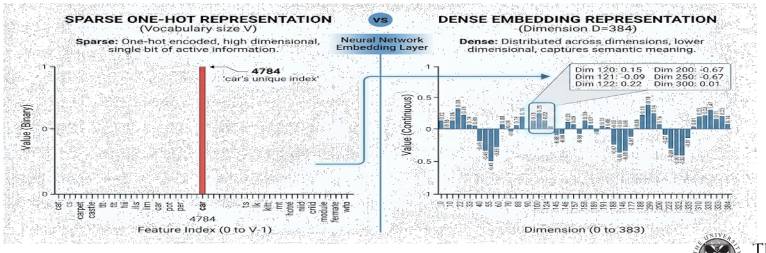
**Dense (Embedding Search)**

**Representation:** Continuous vectors (e.g., 384–1536 floats)

**Matching Mechanism:** Conceptual geometric similarity

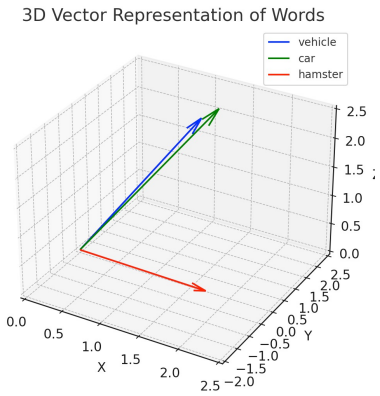
**Index Structure:** Vector DB (e.g. HNSW graph)

**Primary Benefit:** Semantic understanding & zero word overlap



# Distributed Representation

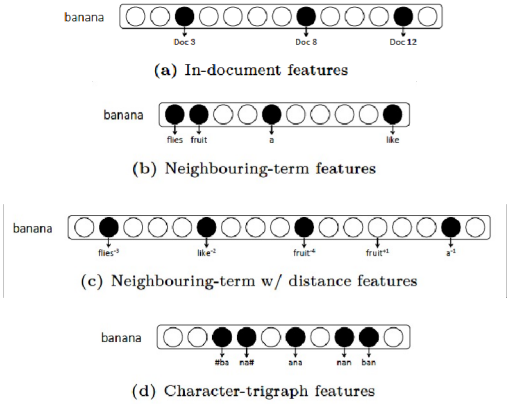
- Observed representation
- Latent representation
  - W2V
  - GLOVE
  - BERT
  - Bi-Encoder
  - Cross-Encoder






# Observed Distributed Representation

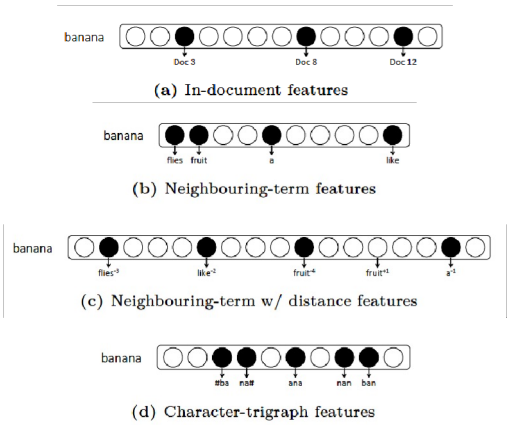
- Initial trials included "Observed Distributed Representation"
- Each term is represented by a vector of "features" of that term.
- Features can be:
  - In-document
  - Neighbour terms
  - Neighbour terms w/ distance
  - Character n-grams
  - A combination of all
- Issues?



# Observed Distributed Representation

**Performance?**

- Efficiency:
- Effectiveness:






# Latent Distributed Representation

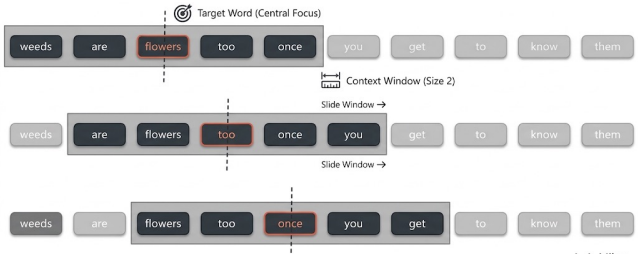
- Latent – Dense – Semantic

## Embeddings

- Compared to observed feature spaces:
  - usually learned from observed features.
  - typically have fewer dimensions.
  - more dense.
  - dimensions may be less interpretable.
  - may generalize better.

# 1st Generation Embeddings – Word2Vec

- Goal → A word is represented as a dense real-valued vector.
- How → Shallow neural model learning from billion words corpus.
- Predict middle word from neighbors within fixed size context window.

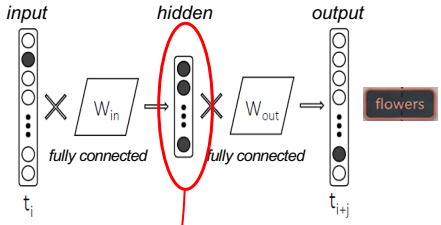



- Two different architectures:
  - **Skip-Gram**
  - **CBOW**



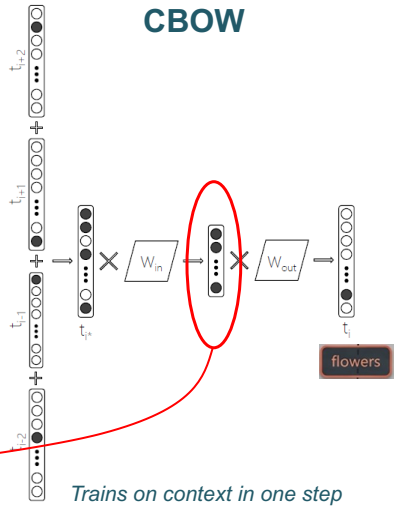
# Word2Vec

## Skip-Gram




*Trains on context word by word*

## CBOW




*Trains on context in one step*

**Embeddings**



# 1st Generation – Word2Vec

- Skip-Gram and CBOW (2013)
- Another method, GLOVE (2014)
  - Count-based model instead of W2V predictive model

**Performance?**

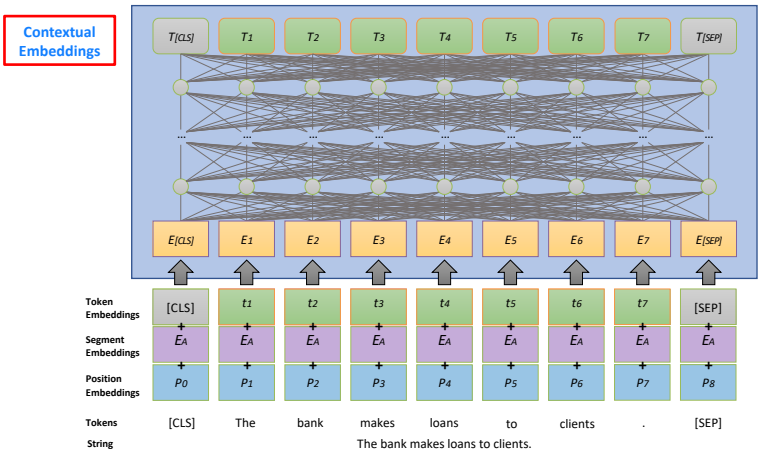
- Efficiency:
- Effectiveness:

- Issue?



# 2nd Generation – Transformers

- Dense Bi-Encoder (e.g. sentence-BERT)



# 2nd Generation – Transformers

## Performance?

| Rank | Model | Submission Date | MRR@10 On Eval | MRR@10 On Dev |
|---|---|---|---|---|
| 1 | **BERT + Small Training** Rodrigo Nogueira and Kyunghyun Cho - New York University | January 7th, 2019 | 35.87 | 36.53 |
| 2 | **IRNet (Deep CNN/IR Hybrid Network)** Dave DeBarr, Navendu Jain, Robert Sim, Justin Wang, Nirupama Chandrasekaran – Microsoft | January 2nd, 2019 | 28.061 | - |
| 3 | **Neural Kernel Match IR (Conv-KNRM)**(1)Yifan Qiao, (2)Chenyan Xiong, (3)Zhenghao Liu, (4)Zhiyuan Liu-Tsinghua University(1, 3, 4); Microsoft Research AI(2) [Dai et al. '18] | Novmeber 28th, 2018 | 27.12 | 29.02 |
| 4 | **Neural Kernel Match IR (KNRM)** ((1)Yifan Qiao, (2)Chenyan Xiong, (3)Zhenghao Liu, (4)Zhiyuan Liu-Tsinghua University(1, 3, 4); Microsoft Research AI(2) [ Xiong et al. '17] | December 10th, 2018 | 19.82 | 21.84 |
| 5 | **Feature-based LeToR: simple-feature based RankSVM**(1)Yifan Qiao, (2)Chenyan Xiong, (3)Zhenghao Liu, (4)Zhiyuan Liu-Tsinghua University(1, 3, 4); Microsoft Research AI(2) | December 10th, 2018 | 19.05 | 19.47 |
| 6 | **BM25** | Novmeber 1st, 2018 | 16.49 | 16.70 |



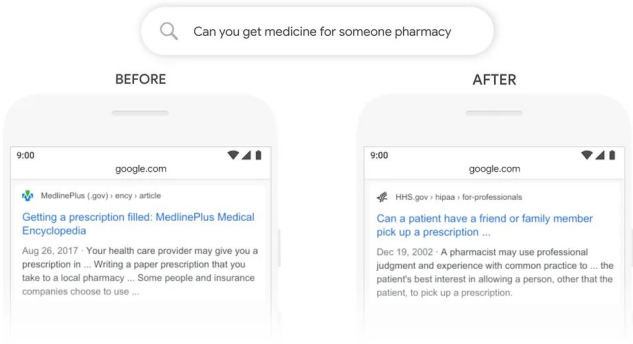
# Deployment

## Google Search - 2019

Understanding searches better than ever before

Oct 25, 2019
6 min read

*We're making a significant improvement to how we understand queries, representing the biggest leap forward in the past five years, and one of the biggest leaps forward in the history of Search.*

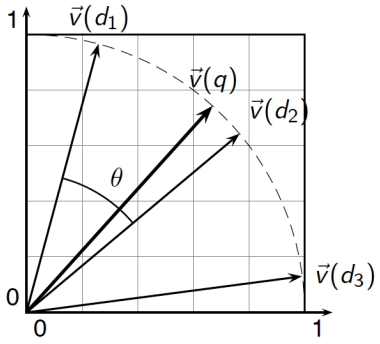



Source

# Matching (Q, D)?

- Documents → Dense embeddings (vector)
- Queries --> Dense embeddings (vector)

- Recall this?




- Cosine Similarity
- Q vs D in collections

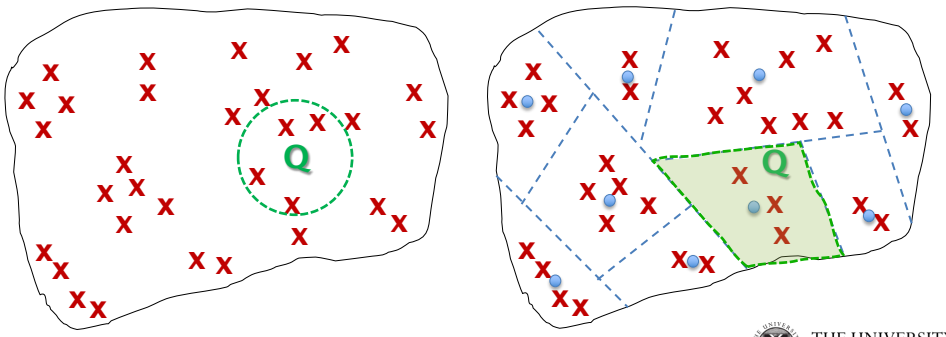
- Find closes Ds for Q
- KNN

- Issue?



# KNN vs ANN (in abstract)

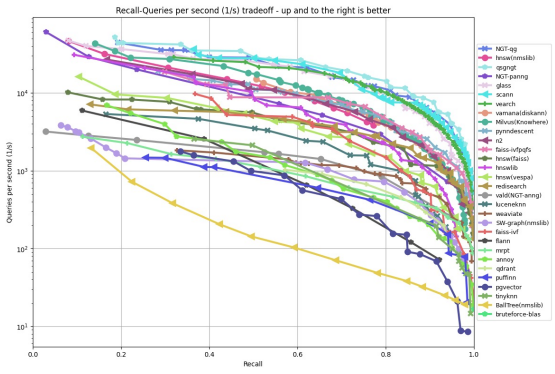
- Dot product of Q vector with D vectors
- Rank Ds by closest to Q → Nearest Neighbour
- Vector length is limited (100s), but for many Ds, process becomes inefficient.
- Solution: KNN → ANN (Approximate NN)



# • Approximate Nearest Neighbour (ANN)

- Multiple approaches
- Increase speed of dense search by 100x – 10000x
- Retains recall at ~80-99%
- Most popular:
  - ANN oh yeah (ANNOY)
  - Graph-based (HNSW)
  - Inverted File Clustering (IVF)
  - (self-reading ☺)

https://ann-benchmarks.com/





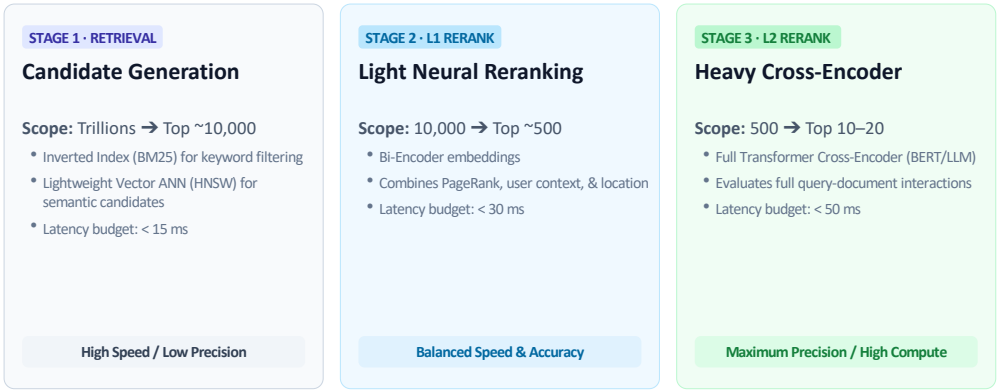

# What about what we studied so far?

- Inverted Index?
- BM25?
- PRF?

- Term-based search is **indispensable**
  - Very efficient
  - Highly scalable
  - Limitations: ranking might be suboptimal!

- Solution:
  - Used as essential first step
  - Advanced methods are used later for reranking top results.

# Typical Search on Huge Collections






# Term Representation (History)

- **2013–2015 — Word Embeddings Era**
  - **Breakthrough:** *Word2Vec* and *GloVe* → dense word representations
  - **Impact:** Limited — used for query expansion and semantic similarity.
  - **Limitation:** Static embeddings; no context awareness
- **2018–2019 — Transformer Revolution**
  - **Breakthrough:** *BERT* and the *Transformer* architecture.
  - **Impact:** Major leap — contextualized embeddings enable deep query–document understanding.
  - **Key Systems:** monoBERT, duoBERT → strong results
- **2020–2021 — Dense Retrieval and Dual Encoders**
  - **Breakthrough:** *Dense Passage Retrieval (DPR)*, *ColBERT*, *ANCE*.
  - **Impact:** Enables *end-to-end neural retrieval* using dense embeddings.
  - **Technology:** ANN (Approximate Nearest Neighbor) search with FAISS, ScaNN.
  - **Trend:** Emergence of *vector-based retrieval pipelines*.

# Term Representation (History)

- **2022–2023 — LLMs and Retrieval-Augmented Generation (RAG)**
  - **Breakthrough:** Integration of *Large Language Models* (GPT, T5, FLAN) in IR.
  - **Impact:** LLMs enhance *query rewriting*, *re-ranking*, and *document expansion*.
  - **Trend:** Retrieval becomes part of *RAG pipelines* — retrieval + generation for QA and reasoning.
- **2024–2025 — Multimodal and Graph-Augmented IR**
  - **Breakthroughs:**
  - *Multimodal retrieval* (text + image + audio).
  - *Graph-based RAG (GraphRAG)* for structured contextual reasoning.
  - *Domain-specialized IR models* (biomedical, legal, financial).
  - **Impact:** Focus on *interpretable*, *trustworthy*, and *multimodal* retrieval.
  - **Trend:** IR as a foundation for *knowledge-augmented AI*.



# Summary

- New approaches uses embeddings for term, query, and document representation
- More accurate, but slower
- Approximation speeds up the process
- Typical Search on large collections:
  - Classical methods → Dense search → Heaving reranking

# Resources

- *An Introduction to Neural Information Retrieval* (Bhaskar Mitra and Nick Craswell, FnTIR)
- *Approximate Nearest Neighbor Search on High-Dimensional Data: Experiments, Analyses, and Platform* (Li et al., *IEEE TKDE*)
- Pre-training Methods in Information Retrieval (Fan et al., FnTIR)

- YouTube Videos:
  - What Is Vector Search? Difference Between Vector & Semantic Search Explained, Elastic (2 mins)
  - Vector Search & Approximate Nearest Neighbors (ANN) | FAISS (HNSW & IVF), Mustafa Zaki (18 mins)