The making of Apache Lucene™ vector search

10/06/2022

Lucene vector search

- Overview
- What is KNN vector search?
- HNSW algorithm
- Open development
- How to use
- Future directions

Who am I?

- Mike Sokolov

  Lucene committer, Amazon principal engineer, long time searcher

  This talk represents contributions from many in the Apache Lucene community

Who are you?

- How many of you have written an application that uses full-text search?

Who are you?

- Have you written an application that uses full-text search?
- Have you trained a language model on a corpus of text?

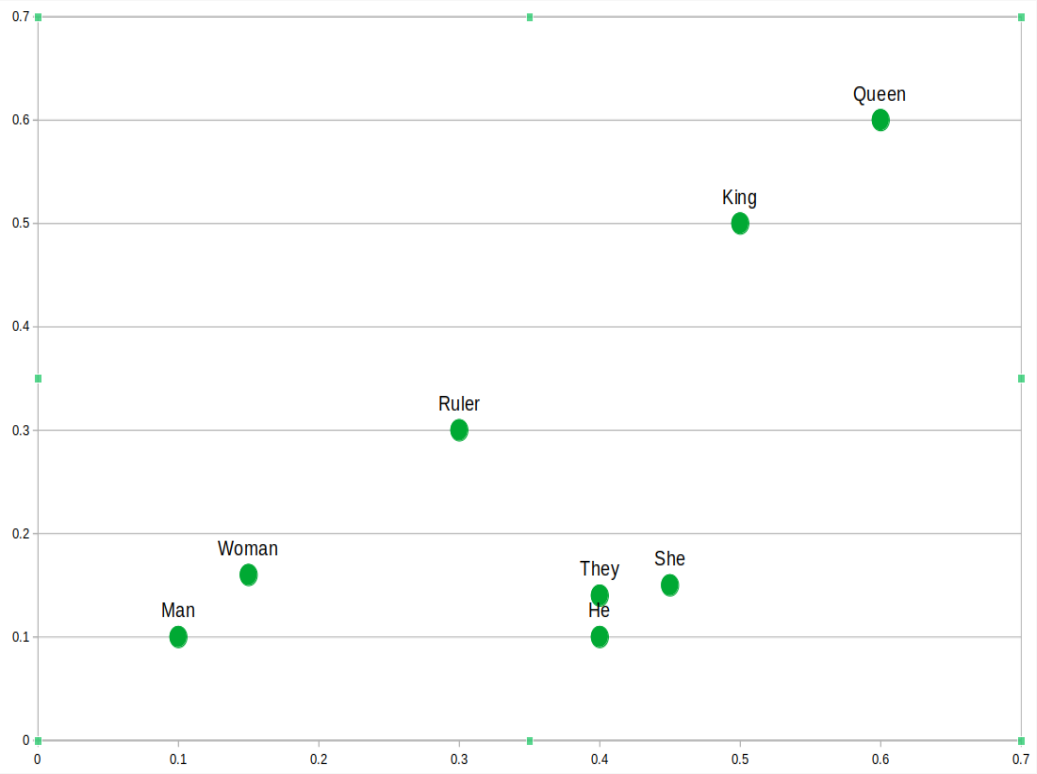
Who are you?

- Have you written an application that uses full-text search?
- Have you trained a language model on a corpus of text?
- Have you heard of huggingface?

Lucene vector search

- Overview
- What is KNN vector search?
- HNSW algorithm
- Open development
- How to use
- Future directions

Word Embeddings



Nearest-neighbor search

- Proliferation of linguistic datasets: GloVe, word2vec, BERT
- Distance representing linguistic similarity (“meaning”)
- Queries and documents as vectors

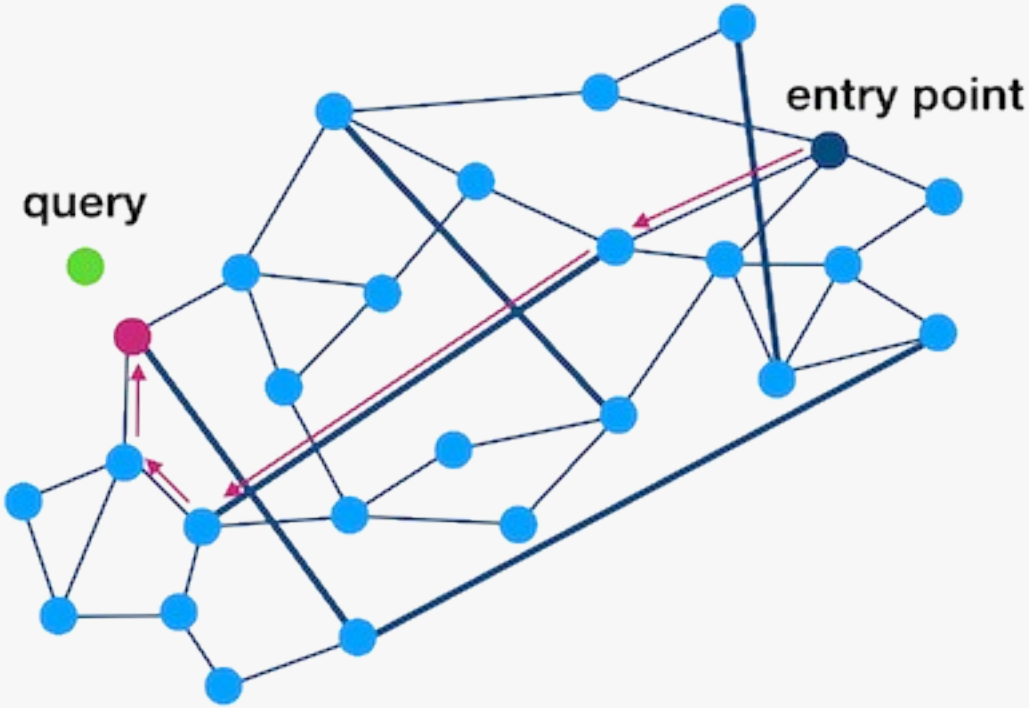
Dimensionality

- Classic TFIDF relevance model is called the “Vector Space Model.” What’s the difference?
- Sparse vectors in a very high-dimensional space (size of the corpus – millions – no hard limit)
- Geospatial searches vectors that have low dimension; Points/kd-tree limited to 8 dimensions.
- Neural search vectors are dense; dimensions in the 100’s (limited to 1024).

Lucene vector search

- Overview
- What is KNN vector search?
- HNSW algorithm
- Open development
- How to use
- Future directions

Graph search

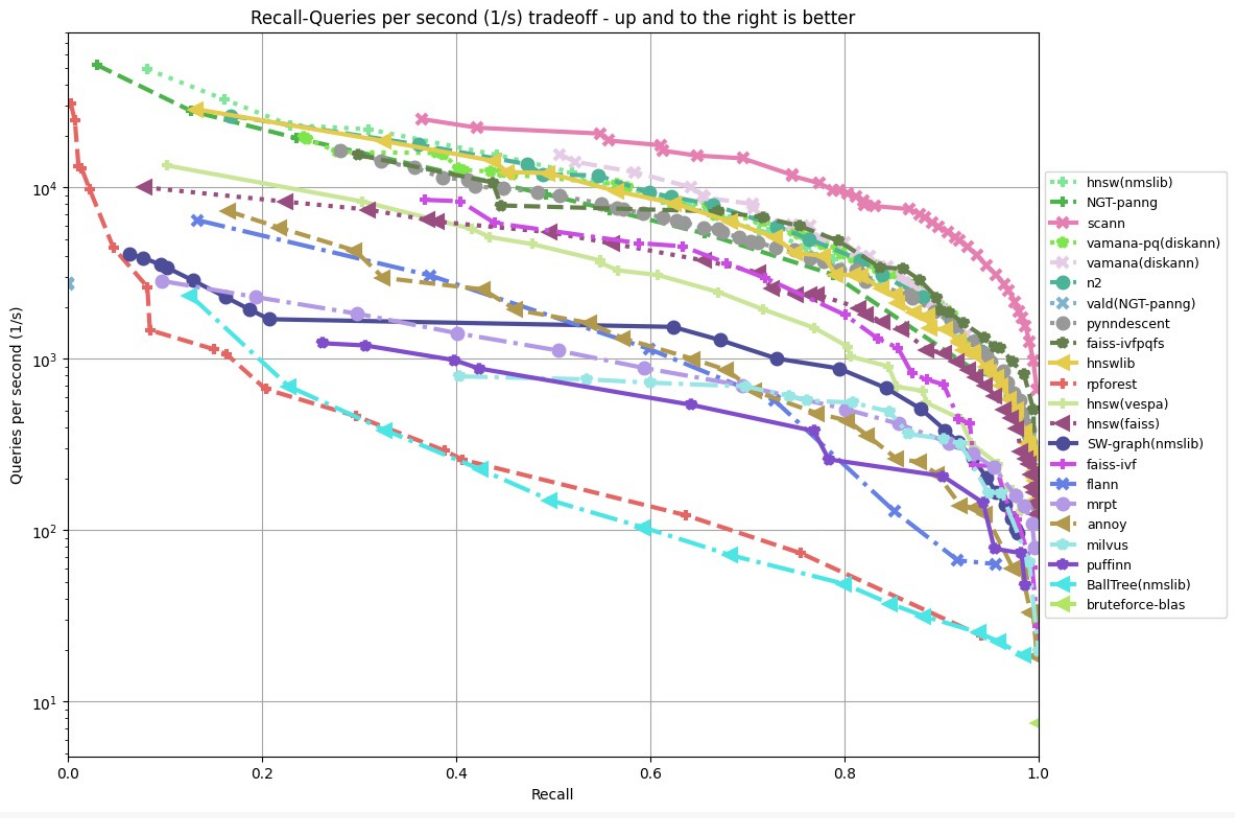


*opensearch.org

Graph indexing

- HNSW search links new nodes to M nearest neighbors
  - Reverse links and prune, preserving diversity
- Bigger M -> more precise; higher cost.
- Beam-width controls extent of search

Speed / Accuracy



* ann-benchmarks.com

Lucene vector search

- Overview
- What is KNN vector search?
- HNSW algorithm
- Open development
- How to use
- Future directions

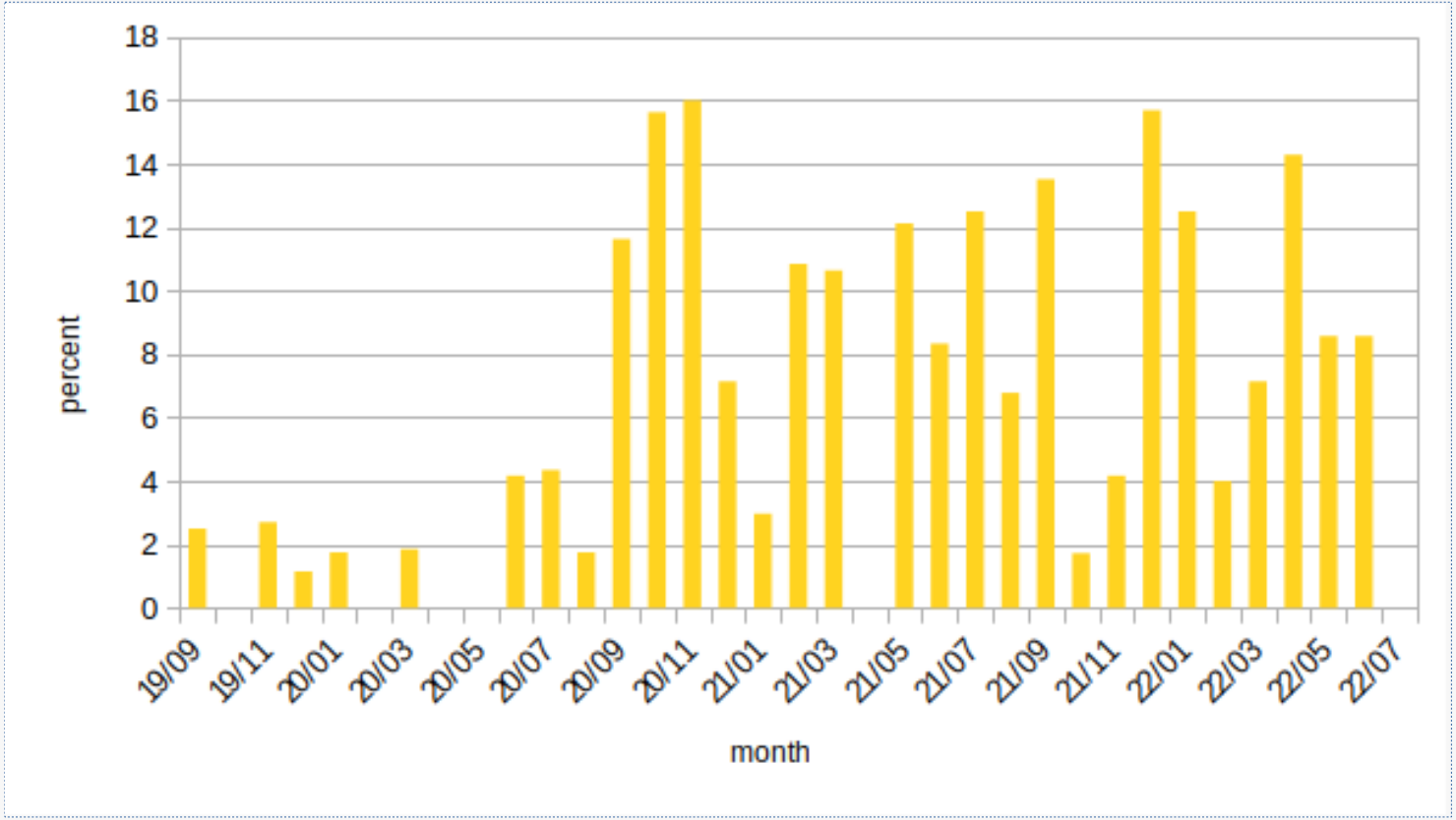
LUCENE-9004, Oct 2019

- Hacky prototype, many gaps
  - Single-layer graph (not hierarchical)
  - Built on top of existing index formats
  - Missing some algorithmic advances
- Lots of interest and comments and suggestions
  - And contributions!

iterations commence

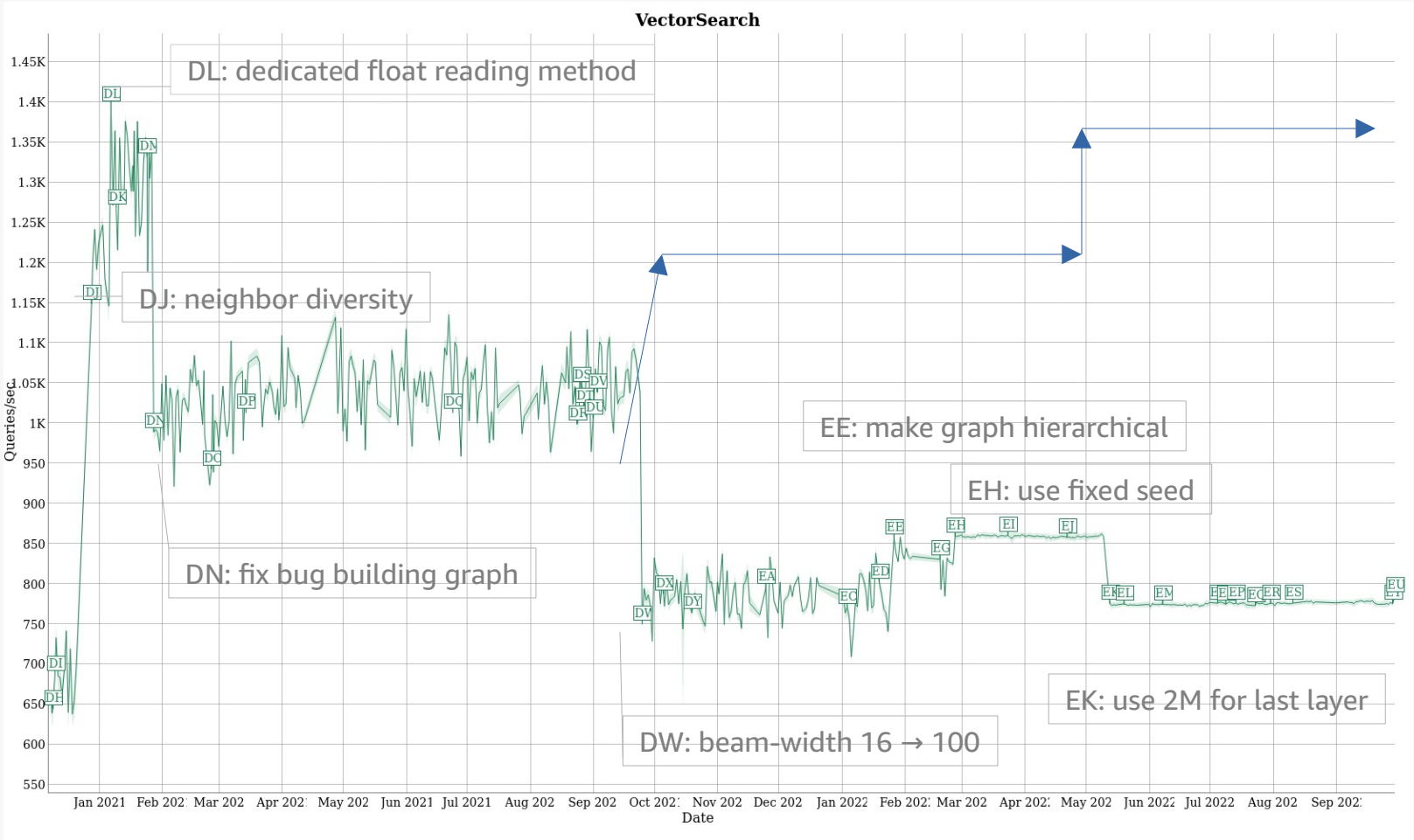
- Nov 2019 - HNSW index format (Tomoko Uchida)
  - Lucene's Codec abstraction separates implementation / file format from "user level" API.
  - API designed to support future ANN algorithms that may require completely different data structures
- Feb 2020 - ann-benchmarks (Julie Tibshirani)
- Sep 2020 - total rewrite, "beta" quality

Lucene vector issues %



Release timeline

- 9.0 – initial release (Dec 2021)
  - Vector Query handles deletions
  - Prune non-diverse neighbors when making graph
  - Careful benchmarking; roughly ½ speed of native-code HNSW
- 9.1 – hierarchical graph (Mayya Sharipova)
- 9.3 – prefiltering (Kaival Parikh)
- 9.4 – low-precision encoding (1 byte/sample)
  - no longer waits for flush() to build graphs


VectorSearch
DL: dedicated float reading method
DJ: neighbor diversity
DN: fix bug building graph
EE: make graph hierarchical
EH: use fixed seed
EK: use 2M for last layer
DW: beam-width 16 → 100
Queries/sec
Date
1.45K
1.4K
1.35K
1.3K
1.25K
1.2K
1.15K
1.1K
1.05K
1K
950
900
850
800
750
700
650
600
550
Jan 2021
Feb 202
Mar 202
Apr 202
May 202
Jun 2021
Jul 2021
Aug 202
Sep 202
Oct 202
Nov 202
Dec 202
Jan 2022
Feb 202
Mar 202
Apr 202
May 202
Jun 2022
Jul 2022
Aug 202
Sep 202

Lucene vector search

- Overview
- What is KNN vector search?
- HNSW algorithm
- Open development
- How to use
- Future Directions

Lucene knn vector API

- KnnVectorField
  - document.addField
    - (new KnnVectorField(“field”, float[] vector))
- KnnVectorQuery
  - indexSearcher.search
    - (new KnnVectorQuery(“field”, float[] vector,
      - int topK)

Examples

- Lucene's `demo` module
- `luceneutil`
- Both use GloVe *word: vector* dictionary
- YMMV; results highly dependent on embeddings

HNSW indexing parameters

- Lucene 9.4 default
  - M=16, beam-width=100
- `IndexWriterConfig.setCodec(new Lucene94Codec() {`
  - `KnnVectorsFormat knnVectorsFormat() {`
    - `return new Lucene94HnswVectorsFormat`
      - `(M, beamWidth);`
- KnnGraphTester (in Lucene's test jar)
  - for tuning indexing parameters

Luke Demo

- Embeddings `all-MiniLM-L6-v2`
  - `https://sbert.net`
- 1.2M products from Amazon dataset
  - https://github.com/amazon-research/esci-data
- Luke hacked to support KNN search

Lucene vector search

- Overview
- What is KNN vector search?
- HNSW algorithm
- Open development
- How to use
- Future Directions

Future ideas

- Compressed graph encoding
- JDK Vector API for dot-product computation
- Other ANN algorithms, eg combining HNSW and quantization
- How best to rank, with term matches?

Thank you

QUESTIONS?