Deep Learning Vector Search Service

Jeffrey Zhu, Program Manager

OpML ’19
Evolution of Search

Classic information retrieval is based on keyword matches and user behavior signals
  • Query rewrite and other alteration techniques cannot enumerate all keyword expansions
  • Insufficient user signals for tail queries

Novel search scenarios have emerged
  • Natural language/Conversation, Question and Answer, image/multimedia

Vector search is a critical technique to improve search and enabling new scenarios
Content Vectorization

Use deep learning model to encode content as a vector

- Distance between vectors represents semantic similarity
- Better semantic representation, tolerant to out of vocabulary, spelling errors, connective words.

Raw Query: coffee and teasouth melbourne

Score: Max Cosine similarity


Title: “coffee tea suppliers in south melbourne melbourne city victoria”

Anchor: “coffee beans suppliers south melbourne”
Vector Recall by Nearest Neighbor Search

- Hash query to this bucket
- Search among points in bucket

Semantic word 1
Semantic word 2
Semantic word 3

(c) Codewords Assignment
quantizer 1
quantizer 2
quantizer 3

NNG

KD-tree

TP-tree

HNSW
From Keyword to Semantic Vector Search

Q: {is it legal for 17 year old to buy a gun}

Bag of Words

- legal
- own
- buy

AND

OR

gun

Inverted Index Matching

<table>
<thead>
<tr>
<th>Word</th>
<th>Posting</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>Posting 1</td>
</tr>
<tr>
<td>buy</td>
<td>Posting 2</td>
</tr>
<tr>
<td>legal</td>
<td>Posting 3</td>
</tr>
<tr>
<td>...</td>
<td>Posting 4</td>
</tr>
</tbody>
</table>

Vector Representation

(0.78, 0.8, 0.4, 0.3, 0.9, ...)
(0.75, 0.6, 0.1, 0.7, 0.2, ...)

Nearest Neighbor Search in Semantic Space

Ranking

- BM25F
- Semantic similarity

Re-ranking
SpaceV: Semantic Vector Search at Scale

- Better fidelity (NCG@infinity) than keyword search + BM25F ranker with the same document sets
- Additional fidelity gain after combining with keyword search

<table>
<thead>
<tr>
<th>L1 Fidelity on full index</th>
<th>Overall</th>
<th>Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword + Vector Search</td>
<td>+3.24</td>
<td>+5.14</td>
</tr>
</tbody>
</table>
Deep Learning Vector Search Service

- **Platform Capabilities**
  - Performance: <10ms search latency
  - Scale: 100B+ vector index size
  - Agile: Fast experimentation + deploy
  - Flexible: Pluggable ANN algorithms

- **Distributed serving**
  - Randomly partition vectors into smaller vector indexes
  - Serving queries is distributed and aggregated before returning

**Query: coffee in Melbourne**

1. Encode semantic representation
2. Local ANN search over each partition
3. Merge results
SpaceV: Semantic Vector Search at Scale

- High scale and Low latency
  - 40B+ vectors
  - Served with N (N=3) replica in 500+ servers
  - High capacity: 240M vectors per machines * 1,800 QPS at most
  - Low latency: 5ms in average and 8ms in 95%ile

<table>
<thead>
<tr>
<th></th>
<th>QPS per replica</th>
<th>Avg latency (ms)</th>
<th>50% latency (ms)</th>
<th>95% latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Traffic</td>
<td>1,200</td>
<td>5.341</td>
<td>4.764</td>
<td>8.004</td>
</tr>
<tr>
<td>Peak Traffic</td>
<td>1,800</td>
<td>6.177</td>
<td>5.159</td>
<td>9.293</td>
</tr>
</tbody>
</table>
Key Innovations

- SPTAG - Approximate Nearest Neighbor Algorithm
  - Balanced k-means tree over relative neighbor graph

- Distributed Vector Index Serving
  - K-means clustering for distributed serving

- Lower Cost Serving Hardware
  - Offload index from memory to Solid State Disk (SSD)
SPTAG – Space Partition Tree and Graph

- Hybrid approach to achieve high recall for both low and high dimension vectors
  - BKT: Balanced K-means Tree
  - RNG: Relative Neighbor Graph

- Designed for efficiency, scale, and agility
  - Better trade-off between recall and latency
  - User customized distance
  - Incremental update

Object function:
\[
\min_{H,C} \|X - HC\|_F^2 + \lambda \|1^T H\|_2^2
\]

Cluster chosen:
\[
k = \arg\min_i f(x_t, c_i) + \lambda s_i
\]
SPTAG – Space Partition Tree and Graph

- Evaluation
  - Three datasets: 1M 2048 dim, 4M 100 dim, 36M 100 dim
  - Two algorithms: HNSW, BKT & RNG

- Open source available at https://github.com/Microsoft/SPTAG
**Distributed Vector Index Serving**

- Challenges with Distributed Serving
  - Poor scalability
  - Too much resource usage for each query
  - Poor latency – long tail

Diagram:
- **Query**
- **100 dims vector**
- **Expensive Broadcast**
- **Top K** → **Aggregation cost** → **Results**
Distributed Vector Index Serving

- Data partitioning with balanced k-means clustering
  - Each data partition maps to specific cluster
  - Each query is only sent to closest clusters (instead of global broadcast)

- Evaluation
  - Selecting top 5 out of 22 clusters can get the same recall as baseline, and only use 23% capacity
SSD Serving

- Challenges
  - Memory is bottleneck to lower cost serving
  - Memory cache hit rate is low due to ANN random access pattern

- ANN algorithm for SSD
  - Build head index from partial vector and serve in memory
  - Cluster tail vectors with head vectors as a center and serve in SSD
SSD Serving

- Evaluation
  - Dataset: 13 million 100 dim vectors
  - 67% memory saving

<table>
<thead>
<tr>
<th></th>
<th>Index Size</th>
<th>Metadata Size</th>
<th>In Memory</th>
<th>In SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Serving</td>
<td>32.3G</td>
<td>6.6G</td>
<td>32.3G</td>
<td>-</td>
</tr>
<tr>
<td>SSD Serving</td>
<td>47.5G</td>
<td>6.6G</td>
<td>6.6G</td>
<td>40.9G</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>99%</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Serving</td>
<td>1.05ms</td>
<td>1.32ms</td>
<td>0.962</td>
</tr>
<tr>
<td>SSD Serving</td>
<td>3.07ms</td>
<td>5.90ms</td>
<td>0.929</td>
</tr>
</tbody>
</table>
Takeaways

• Vector search is a critical technique to improve web search and power new capabilities such as question and answering, image search, etc.

• Key innovations in ANN algorithm and distributed vector index serving allows DLVS platform to serve high scale vector search scenarios (100B+ vectors)

• Core ANN algorithm (SPTAG) is open source and available for developers to use
  • https://github.com/Microsoft/SPTAG