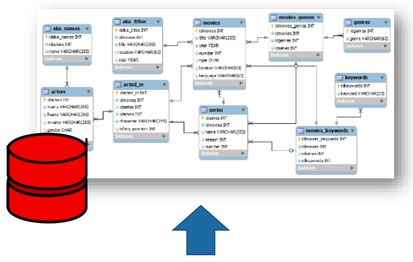
Fast Approximated Nearest Neighbor Joins For Relational Database Systems

Dresden Database Systems Group

Vector Data in Database Systems



Word2Vec / Glove — Word Embeddings

SIFT / GIST — Image Descriptors

Deepwalk / Node2Vec — Graph Embeddings

Vector Representation

Unstructured Text

Multimedia Data

Graph Data

Enables
- Similarity Queries
- Inductive Reasoning (Analogies)

Requires
- Integration of Fast Vector Search Operations

Integration of kNN-Join Operation

queries

q_2:
```
SELECT m1.title, m2.title
FROM movies m1 kNN-Join(3) movies m2
ON m1.title ~ m2.title
WHERE m1.year= … AND m2.genre IN (…)
```

database

m_id	title	…
1	Inception	…
2	Godfather	…
3	Forest Gump	…
…	…	…
10^6	Scarface	…

+

term	vector
1972	[0,21; 0,58; …; -0,77]
Brando	[-0,46; 0,25; …; 0,44]
Copolla	[0,76; 0,48; …; -0,51]
…	[-0,46; 0,53; …; 0,85]
Untouchables	[0,86; -0,22; …; 0,12]

results

title	VARCHAR
Inception	Shutter Island
Inception	The Dark Knight
…	…
Untouchables	Scarface

Legend: query set R, target set T

$$kNN(R \bowtie T) = \{(\boldsymbol{r}, \boldsymbol{t}) | \, \boldsymbol{t} \in kNN(\boldsymbol{r}, T), \boldsymbol{r} \in R\}$$

Challenges

- **Batch-wise kNN Search** for large query sets
  → Reduce interface and retrieval times
- **High Dimensional Data**
  → Previous Work mainly focus on low dimensional data (e.g. geographical data)
- **Adaptive kNN-Join Algorithm**
  → Different cardinalities of join operands
  → Only one index for all vectors
- **Different Demands on Precision and Response Time**

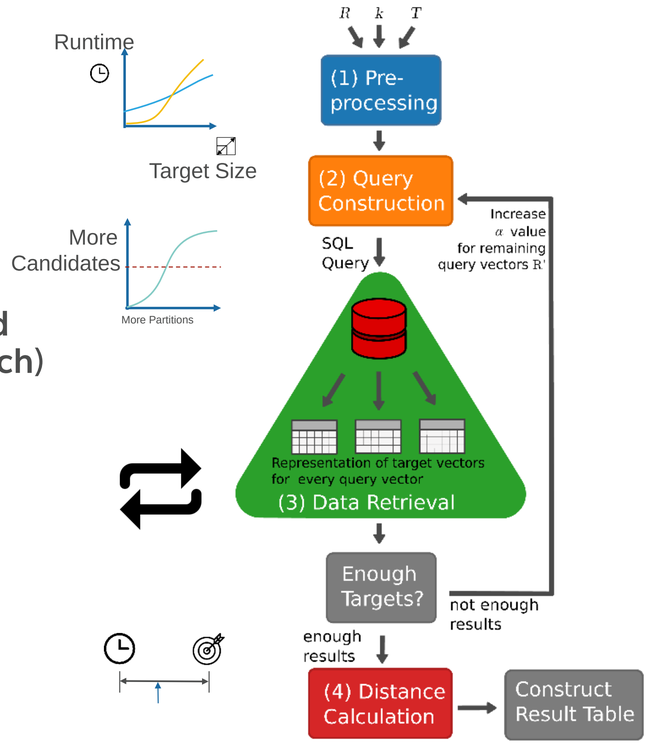
Fast Inverted Search Algorithm

(1) Preprocessing
- Precompute distance values
- **Adaptive to target set size**

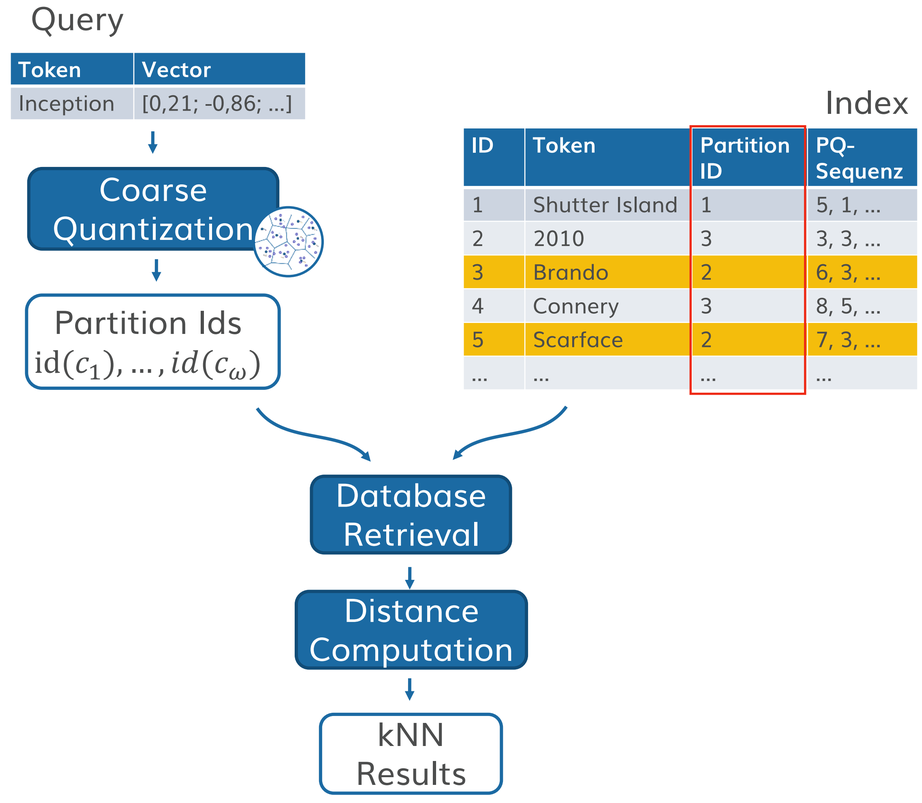
(2) Query Construction
- Determine ω (Number of Part.)
- **Estimate number of retrieved entries (Probabilistic Approach)**

(3) Data Retrieval
- Retrieve kNN candidates

(4) Distance Calculation
- **User can adjust trade-off between precision and response time**



Inverted Indexing



Estimation of Number of Partitions ω :

$$P(m \geq n) = \sum_{n \leq m \leq d} \frac{\binom{K}{m}\binom{N-K}{d-m}}{\binom{N}{d}}$$

Parameter:
Population Size $\boldsymbol{N}$: Number of all index vectors
Number of success states in population $\boldsymbol{K}$: Number of targets
Number of draws $\boldsymbol{d}$: Number of vectors in selected partitions
Number of Successes $\boldsymbol{m}$:Number of targets in selected partitions

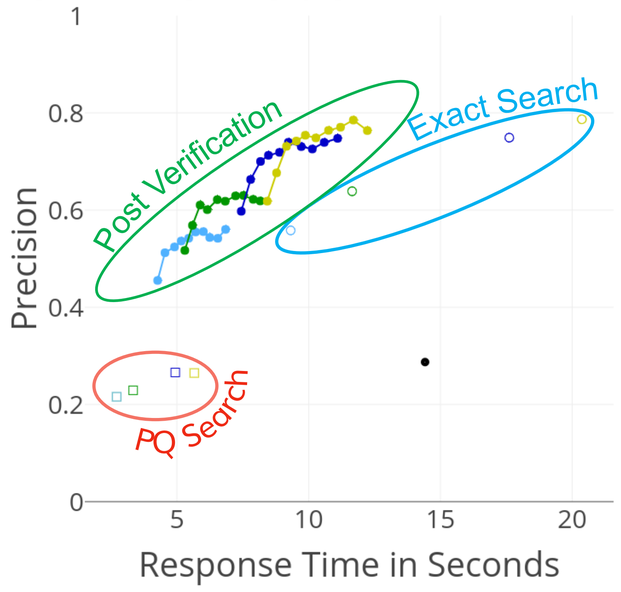
Evaluation of Algorithm with Different Distance Calculation Methods

[1]

Product Quantization
Fast Approximated Search Method

Exact Distance Calculation
Fast Approximated Search Method

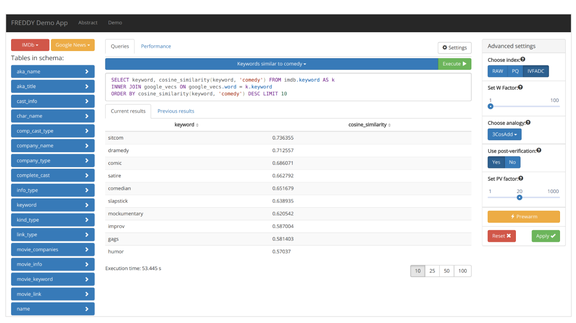
Post Verification
Post verify candidate solutions obtained by product quantization



Setup

5,000 query vectors
50,000 target vectors

Web Demo



Word Embedding operations that use the kNN-Join Index can be explored in our web demo

TECHNISCHE UNIVERSITÄT DRESDEN

Contact: Michael.Guenther@tu-dresden.de https://wwwdb.inf.tu-dresden.de/research-projects/freddy/