1 Month Work Progress

Demo: KNN Search with Incremental Query Answering

Jaouhara Chanchaf

September 6th, 2022

Summary

Done:

- Developed the first version of the user interface.
- Updated DStree incremental query answering code to run Kashif.
- Assessed Kashif performance for different k values.

In prgress:

- Build a good use case for the demo paper.
- Issues: good results are retrieved even with small k values, increasing k tends to improve ranking of results rather than introducing better results (in terms of intersection size).

Issues:

- Query compilation time is a bottleneck.
- Kashif recall is very low. (best recall≈ 0.16 for k $= 100$)
- Cannot explain why mean recall drops for a higher k value (avg. recall for $100nn = 0.16$, avg. recall for $1000nn \approx 0.14$)

Summary

Next:

- UI:
  - Figure out how to avoid loading the index for every query (load the index when the application starts).
  - Implement pagination of query results.
  - Add the option of ranking results by overlap size or by vector distances
  - Merge results of the same table.

On hold (data discovery):

- Compare QCR index and Josie and Compare Kashif with PEXESO.
- Summarize LSH paper.
- Read Hercules paper and experiment with code.

Performance evaluation

Experiment on 100k tables (1GB of data) holding 5 million vectors.

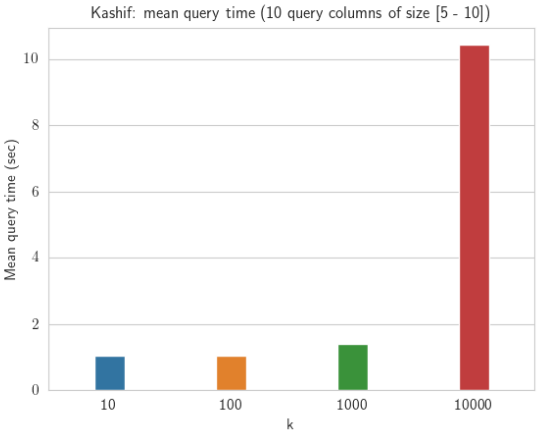

Figure: Kashif Mean Query Time

Performance evaluation

Experiment on 100k tables (1GB of data) holding 5 million vectors.

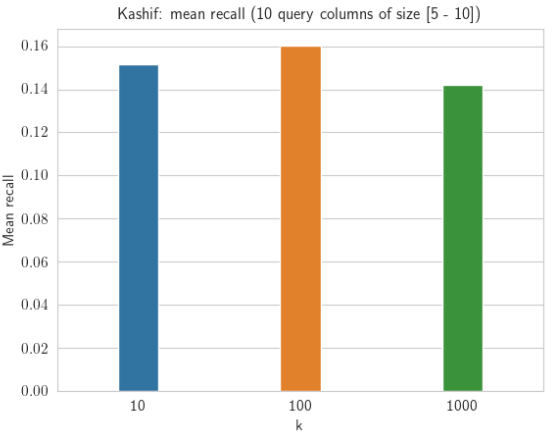

Figure: Kashif Average Recall

Use case

Context:
A user wants to retrieve tables about sport teams per country/ continent. He/she chooses a query column with continent names (very vage query).

After several search attempts with different small k values (k = {10, 100, 500, 550, 599}, he/she decides to take a large k value (k = 2000) in hope of finding the desired table at the expense of query-time.
query: continent.csv Desired table: "Eliteprospects.com - Nittorps IK"

k	found table	query-time (in sec)
10	no	5.30
100	no	6.13
500	no	6.28
600	yes	6.37
800	yes	6.40
2000	yes	7.16

Work update (Thursday, Sept 8th 2022)

Done:

- AI 1: ICDE requirements for the demo paper
  `https://icde2023.ics.uci.edu/demonstration-track/`
- AI 3: Why Kashif has low recall? That was due to a bug in the brute force implementation.
- AI 5

Next:

- AI 2:
  What makes Incremental query answering useful?
  Can early results be useful to the user? we must prove that through an example.

Recall evaluation

Experiment on 200 tables, 1k columns, 10k vectors.

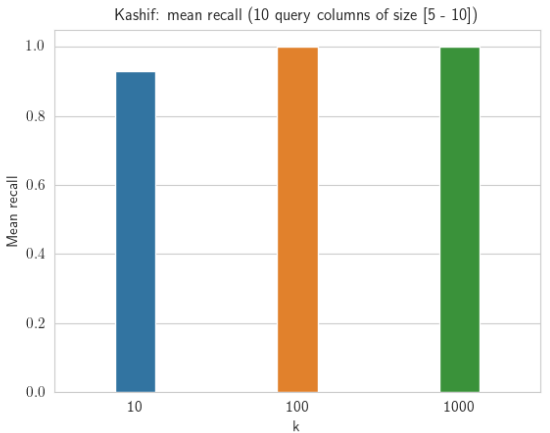

Figure: Kashif Average Recall

Performance evaluation

Experiment on 100k tables, 494k columns, 5M vectors.

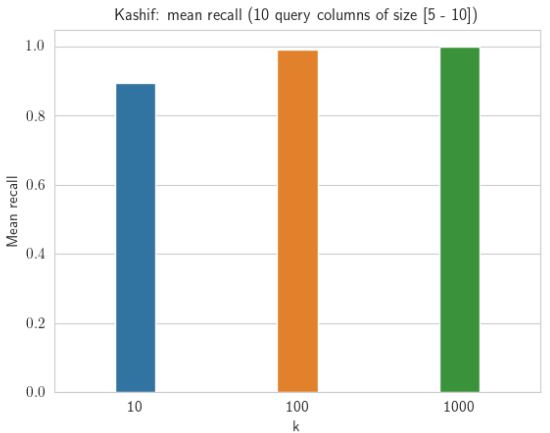

Figure: Kashif Average Recall