1. Task Overview

**Task:** Build an approximate K-NN Graph for a set of vectors. i.e., for each vector, find its approximate k nearest neighbors in a limited time. For this task, k is set to be 100.

**Dataset:** The final evaluation dataset is sampled from a billion-scale vector dataset, which consists of Bing queries encoded by Turing AGI v5 that trains Transformers to capture similarity of intent in web search queries.

<table>
<thead>
<tr>
<th>#</th>
<th>Num of vectors</th>
<th>Num of dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final evaluation dataset</td>
<td>10 million</td>
<td>100</td>
</tr>
</tbody>
</table>

**Measurement:** Compute the resulting average recall score on >=10,000 sample groundtruth vectors. The recall of one vector will be computed as follows:

$$\text{Recall} = \frac{\text{number of true top 100 nearest neighbors}}{100}$$

**Evaluation Environment:** Azure Standard F32s_v2 (32 CPU x 2.7 Ghz Processors, 64 GB Main Memory, 32GB Storage)

2. Solution Overview

```
Dataset
  ↓
Preprocessing
  ↓
KNNG Construction
  ↓
Output
```

3. Preprocessing

**Data Format:** Quantize the floating point numbers in the dataset.

**Data Loading:** Align the dataset by a certain byte.

4. KNNG Construction

Our method is based on KGraph [1]. The method is based on the following simple principle: a neighbor of a neighbor is also likely to be a neighbor [2]. The initial KNNG is continuously improved through iterations. In addition, in this algorithm, there are mainly two operations of update and join.

We optimize the algorithm using the following strategies:
- Reduce memory overhead.
- Use grid search strategy to choose better parameters to balance time and recall.

5. Acceleration

- **Parallelization**
  - Use openmp for distance calculation and other operations.
- **SIMD**
  - Use the AVX-512 instruction set to maximize CPU computing speed.

6. Result

<table>
<thead>
<tr>
<th>#</th>
<th>Recall</th>
<th>Runtime(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final evaluation dataset</td>
<td>0.974</td>
<td>1833</td>
</tr>
</tbody>
</table>

7. References

[1]. https://github.com/aaalgo/kgraph