Uživatelská dokumentace

Tato dokumentace obsahuje návod jak spustit ukázku nové implementace indexu HNSW, která je součástí bakalářské práce na téma "Aproximace KNN problému".

Všechny cesty uvedené v tomto souboru jsou relativní k cestě složky, která obsahuje tuto dokumentaci.

Potřebné programy

- Docker
- Překladač C++17
- Python 3.9

Ukázka

1. Ujistěte se, že služba Docker je zapnutá.
2. Spusťte skript `RUNME.py` pomocí interpretu Python verze 3.9. Na Windows například takto:

```
py -3.9 RUNME.py
```

Tento skript provede následující.

- Vytvoří virtuální prostředí interpretu Python ve složce `.venv`.
- Vytvoří nativní C++ řešení ve složce `src/cmakeBuild`.
- Spustí porovnání původní a nové implementace indexu HNSW nad malými kolekcemi se 100 000 prvky.
- Otevře stránku s výsledky v jedné kartě internetového prohlížeče.
- Otevře stránku s dokumentací indexu ve druhé kartě.

Výsledky srovnání

Výsledky jsou zaznačeny do grafů, které zobrazuje webová stránka. Ta je generována ve složce `src/website`. Původní implementace je v grafech označována slovem `original`, nová implementace slovem `new`.

Virtuální prostředí

Pokud není uvedeno jinak, skripty uvnitř složky `src/scripts` vždy spouštějte pomocí vygenerovaného virtuální prostředí. Prostředí aktivujete pomocí aktivačního skriptu ve složce `.venv/Scripts`. Výběr skriptu závisí na použitém OS a interpretu.

OS	INTERPRET	CESTA K AKTIVAČNÍMU SKRIPTU
Linux		./.venv/Scripts/activate
Windows	Batch	.\.venv\Scripts\activate.bat
Windows	Powershell	.\.venv\Scripts\Activate.ps1

Seznam skriptů

Následuje seznam skriptů ve složce `src/scripts`, které umožňují uživateli provést více operací než úvodní skript.

NÁZEV SKRIPTU	STRUČNÝ POPIS SKRIPTU
buildProject	Vytvoří virtuální prostředí, nativní C++ řešení a jeho Python rozhraní.
clean	Odstraní vygenerované soubory a vrátí projekt do původního stavu.
datasetGenerator	Vygeneruje datové soubory pro debugování.
datasetToText	Převede datový soubor do textového formátu.
formatCMakeTemplates	Vygeneruje CMakeLists.txt.
generateTables	Vygeneruje LaTeX tabulky podobné těm, které jsou v bakalářské práci.
latexTable	Vygeneruje LaTeX tabulku na základě výsledků srovnání.
runBenchmarks	Spustí srovnání, vygeneruje a otevře webovou stránku s výsledky.
runRecallTable	Postaví nový index a zobrazí tabulku závislosti přesnosti na parametru vyhledávání ef_{search}.
SIMDCapability	Zobrazí SIMD rozšíření instrukční sady procesoru, která jsou k dispozici.

Podrobný popis skriptů

U každého skriptu je uveden jeho účel, parametry a příklad spuštění. Pokud skript obsahuje alespoň jeden parametr, pak použitím parametru `--help` nebo `-h` zobrazíte nápovědu v anglickém jazyce.

buildProject

Tento skript lze spustit bez virtuálního prostředí.

Vytvoří virtuální prostředí interpretu Python, stáhne potřebné softwarové balíčky, vygeneruje nativní C++ řešení pro knihovnu nového indexu, vytvoří rozhraní v jazyce Python pro nový index a otestuje funkčnost tohoto indexu spuštěním skriptu `runRecallTable`.

PARAMETR	ZKRATKA	VYŽADOVÁN	VÝZNAM
--clean	-c	Ne	Vrátí projekt do původního stavu před jeho opětovným sestavením.
--cleanResults	-r	Ne	Pokud je `--clean` nastaven, odstraní naměřené výsledky.
--ignorePythonVersion	-i	Ne	Umožňuje spustit skript s libovolnou verzí interpretu Python.

Pokud je specifikován parametr `ignorePythonVersion`, skript nemusí fungovat správně.

Příklad spuštění:

```
1  py -3.9 buildProject.py --clean --cleanResults
```

clean

Tento skript lze spustit bez virtuálního prostředí.

Odstraní datové soubory pro debugování, C++ nativní řešení a Python rozhraní. Pokud je spuštěn mimo virtuální prostředí, pak odstraní toto prostředí. Naměřené výsledky odstraněny nebudou, pokud o to uživatel nepožádá.

PARAMETR	ZKRATKA	VYŽADOVÁN	VÝZNAM
--results	-r	Ne	Odstraní naměřené výsledky srovnání, vygenerované grafy a tabulky.

Příklad spuštění:

```
py clean.py --results
```

datasetGenerator

Vygeneruje datové soubory pro debugování uvedené v konfiguračním souboru `src/config/debugDatasets.json`. O konfiguraci tohoto skriptu se více dočtete v kapitole `Datové soubory` níže v této dokumentaci.

Příklad spuštění:

```
py datasetGenerator.py
```

datasetToText

Převede vybraný datový soubor ze složky `src/data` do textového formátu.

PARAMETR	ZKRATKA	VYŽADOVÁN	VÝZNAM
--name	-n	Ne	Název datového souboru bez přípony. Pokud není uveden, výchozím souborem je `angular-small`.

Příklad spuštění:

```
py datasetToText --name euclidean-medium
```

formatCMakeTemplates

Vygeneruje soubor `src/index/CMakeLists.txt` a doplní do něj správnou definici maker tak, aby došlo pouze ke kompilaci těch funkcí, pro které je k dispozici vhodné SIMD rozšíření instrukční sady procesoru.

Příklad spuštění:

```
py formatCMakeTemplates.py
```

generateTables

Vygeneruje LaTeX tabulky podobné těm, které jsou v bakalářské práci, ale pouze v případě, že jsou pro ně dostupné naměřené výsledky. Tyto výsledky lze získat spuštěním následujících příkazů. Avšak tato měření mohou trvat více než 12 hodin.

```
py runBenchmarks.py -a ..\config\heuristic.yaml ..\config\naive.yaml -d lastfm-64-dot -r 5
py runBenchmarks.py -a ..\config\heuristic.yaml ..\config\prefetch.yaml -d glove-50-angular -r 5
py runBenchmarks.py -a ..\config\original.yaml ..\config\prefetch.yaml -d sift-128-euclidean -r 5
py generateTables.py
```

Vygenerované tabulky jsou dostupné ve složce `src/figures`.

latexTable

Vygeneruje jednu LaTeX tabulku na základě výsledků srovnání implementací.

PARAMETR	ZKRATKA	VYŽADOVÁN	VÝZNAM
--algorithms	-a	Ano	Seznam implementací oddělený mezerami.
--dataset	-d	Ano	Název datového souboru.
--label	-la	Ne	Identifikátor tabulky.
--legend	-le	Ne	Názvy implementací v tabulce. Pokud není uveden, budou použity původní názvy.
--output	-o	Ano	Cesta k výstupnímu souboru.
--percent	-p	Ne	Přidá do tabulky sloupec s procentuálním rozdílem časů stavby.
--recompute	-r	Ne	Znovu vypočítá výkonnostní metriky z naměřených výsledků. Tato operace může trvat více než 10 minut.

Příklad spuštění:

```
py latexTable.py -a new-prefetch original -d sift-128-euclidean -le "Nová impl." "Původní impl." -o ..\figures\table.tex -p
```

runBenchmarks

Před spuštěním se ujistěte, že je služba Docker zapnutá.

Spustí srovnání implementací v jednom nebo více Docker kontejnerech, vypočítá výkonnostní metrika, vygeneruje webovou stránku s výsledky a otevře ji v nové kartě internetového prohlížeče. Kód vygenerované stránky lze poté najít ve složce `src/website` a můžete ji opětovně zobrazit otevřením souboru `index.html`.

PARAMETR	ZKRATKA	VYŽADOVÁN	VÝZNAM
--algoDefPaths	-a	Ano	Seznam cest ke konfiguračním souborům oddělených mezerami.
--datasets	-d	Ano*	Seznam datových souborů oddělených mezerami.
--force	-f	Ne	Spustí již provedená měření znovu.
--datasetsPath	-p	Ano*	Cesta k textovému souboru se seznamem datových souborů.
--runs	-r	Ne	Počet opakování měření. Výchozí hodnota je 1.
--workers	-w	Ne	Počet paralelně spuštěných Docker kontejnerů. Výchozí hodnota je 1.

O konfiguraci srovnání se více dočtete v kapitole `Konfigurace srovnání`. Datové soubory využité ke srovnání nejsou ty samé, které jsou využívány k debugování. Jejich seznam najdete v kapitole `Testované datové soubory`.

* Pouze jeden z parametrů označených hvězdičkou by měl být uveden.

Příklad spuštění:

```
1  py runBenchmarks.py -a ..\config\noBit.yaml -f -p
   ..\config\datasets.txt -r 5 -w 2
```

runRecallTable

Postaví index nové implementace a vyhledá v něm nejbližší sousedy s různými hodnotami parametru vyhledávání ef_{search}. Poté vypíše tabulku závislosti přesnosti vyhledávání na tomto parametru. Konfigurace tohoto skriptu se nachází v souboru `src/config/recallTable.json` a více se o ní dočtete v kapitole `Konfigurace recallTable`.

Příklad spuštění:

```
py runRecallTable.py
```

SIMDCapability

Zobrazí SIMD rozšíření instrukční sady procesoru, která jsou k dispozici. Využívána ostatními skripty pro vygenerování správných maker v jazyce C++.

Příklad spuštění:

```
py SIMDCapability.py
```