CSCE604135 • Perolehan Informasi
Semester Ganjil 2022/2023
Fakultas Ilmu Komputer, Universitas Indonesia

Tugas Pemrograman 1: Inverted Index & Boolean Retrieval

Deadline: 22 September 2022, 23:55 WIB

Ketentuan:

1. Tugas Pemrograman 1 ini terdiri dari 1 buah file .zip berisi template program dan dataset dokumen forum kesehatan dalam Bahasa Indonesia.
2. Lengkapi program template yang diberikan sesuai dengan petunjuk pengerjaan tugas yang disediakan.
3. Seluruh program (file .py) yang telah dibuat dikumpulkan dalam satu folder dan dikonversi ke dalam format .zip dengan format penamaan **TugasX_NPM.zip**
   Contoh: Tugas1_1906262623.zip
4. Kumpulkan tugas pada submisi yang telah disediakan di SCeLe sebelum tanggal **22 September 2022, 23.55 WIB**. Keterlambatan pengumpulan akan dikenakan penalti sebesar 30% untuk 3 hari setelah deadline. Setelahnya submisi tidak akan diterima.
5. Tugas ini dirancang sebagai **tugas mandiri**. Plagiarisme tidak diperkenankan dalam bentuk apapun. Adapun kolaborasi berupa diskusi (tanpa menyalin maupun mengambil jawaban orang lain) dan literasi masih diperbolehkan dengan mencantumkan kolaborator dan sumber.
6. **Anda boleh konsultasi dengan asisten dosen (LINK). Asisten dosen diperbolehkan membantu Anda dengan memberikan petunjuk.**

Petunjuk Pengerjaan Tugas

Pada Tugas Pemrograman 1 ini, Anda diminta untuk membuat *indexer* yang menghasilkan *inverted index* dari *scratch* dan mengimplementasikan model *boolean retrieval* sederhana. Pada tugas ini, sudah disediakan template program yang harus diisi oleh mahasiswa untuk bisa membuat sistem yang bisa berjalan dengan baik dan efisien.

Terdapat 5 buah program yaitu **bsbi.py, compression.py, index.py, search.py, dan util.py** yang sudah dilengkapi dokumentasi pada template yang diberikan. Setiap function dalam program juga sudah diberikan dokumentasi lengkap untuk memudahkan Anda dalam menyusun program. Bagian yang perlu anda lengkapi sudah ditandai dengan comment **#TODO.** Selain berisi program, beberapa file juga telah diisi sample test case pada bagian akhir (pada block if __name__ == '__main__') untuk memeriksa kebenaran dari program yang anda buat. Anda juga dibebaskan untuk mengubah bagian ini untuk testing lebih lanjut. Namun perlu diingat bahwa penilaian akan tetap dilihat pada kualitas program yang dibuat.

Untuk memudahkan Anda dalam mengerjakan, disediakan *walkthrough* pengerjaan sebagai berikut:

1. Buat implementasi pada file **util.py**. File ini berisi implementasi *mapping* sederhana untuk menyimpan pemetaan bagi sebuah term ke sebuah integer (term ID) dan juga sebuah dokumen ke sebuah integer (doc ID); serta sebaliknya.
2. Buat implementasi pada file **compression.py**. File ini berisi implementasi untuk mengubah representasi *postings* menjadi *sequence of bytes* yang akan disimpan pada memori. Pada kedua *class* tersebut terdapat method *encode* dan *decode* yang akan dipanggil dari class lain, sementara method lainnya berperan sebagai *helper method* untuk kedua method sebelumnya. Ketika melakukan encoding, pertama, *list of postings* perlu diubah ke dalam bentuk *list of gaps*. Kemudian, *list of gaps* akan di-*compress* dengan Variable-Bytes Compression. Proses decoding perlu disesuaikan agar hasil kompresi bisa kembali semula.
3. Buat implementasi pada file **index.py**. File ini berisi beberapa *class* yang merupakan abstraksi dari sebuah Inverted Index, termasuk implementasi untuk melakukan operasi membaca dan menulis index yang berada pada sebuah storage (dalam bentuk file di harddisk).
4. Buat implementasi pada file **bsbi.py**. File ini berisi sebuah class yang merupakan abstraksi dari proses indexing dengan metode **Blocked Sort Based Indexing** (BSBI). Dalam melakukan proses indexing, Anda perlu mengimplementasikan method *parse_block* untuk mengolah dokumen menjadi bentuk *list of pairs <termID, docID>* dan method *merge* untuk menggabungkan seluruh *inverted indices* yang sudah dibuat sebelumnya. Kedua method tersebut akan dipanggil oleh method utama *index* yang melakukan proses indexing secara keseluruhan.
   Setelahnya terdapat method *retrieve* dalam proses searching yang melakukan pengambilan dokumen-dokumen berdasarkan *query* yang diberikan. Menyesuaikan

flow dari *indexer* ini, Anda sebaiknya mengerjakan bagian terkait proses indexing terlebih dahulu baru setelahnya bagian mengenai proses searching.

NB: Langkah 1 dan 2 bisa dilakukan secara paralel karena kedua program tersebut bersifat independen dengan program lainnya.

Jika seluruh program sudah diimplementasikan, Anda bisa mengujinya dengan langkah sebagai berikut:

1. Jalankan file **bsbi.py** untuk membangun index dari dataset yang tersedia. Jika proses indexing berhasil maka akan muncul file *index* dan *posting-dictionary* pada direktori index. Proses ini bisa memakan waktu yang cukup lama.
2. Jalankan file **search.py** untuk melakukan searching pada index yang dibuat. Contoh untuk melakukan searching melalui query sudah tersedia pada file tersebut.

Bonus:
Implementasikan satu lagi algoritma untuk kompresi postings yang berbeda dari Variable-Byte Coding, misal **Elias-Gamma Coding** (silakan cari referensi dengan search engine favorit Anda). Kemudian, lakukan perbandingan empiris terkait besarnya ukuran index yang dihasilkan jika menggunakan VB Coding dan jika menggunakan algoritma kompresi yang lain tersebut. Juga lakukan perbandingan empiris terkait lamanya waktu saat indexing (*actual time*).

Poin penilaian**:**

- util.py 20 poin
- compression.py 20 poin
- index.py 20 poin
- bsbi.py 40 poin
- BONUS 10 poin

Referensi & Kredit:

- Soal tugas pemrograman ini merupakan hasil modifikasi dari tugas pemrograman kuliah serupa di Stanford University: https://web.stanford.edu/class/cs276/pa/pa1.zip
- Hakim, A. N. (2016). Pemrosesan Pertanyaan pada Sistem Tanya Jawab Bidang Kesehatan dengan Pendekatan Pembelajaran Mesin. Bachelor's Thesis, Universitas Indonesia, Kampus UI Depok.

Selamat mengerjakan!