FAKULTAS
ILMU
KOMPUTER

CSCE604135 • Perolehan Informasi
Semester Gasal 2023/2024
Fakultas Ilmu Komputer, Universitas Indonesia

Tugas Pemrograman 1: Inverted Index & Boolean Retrieval
Tenggat Waktu: Rabu, 27 September 2023, 23.55 WIB

Ketentuan:

1. Tugas Pemrograman 1 ini terdiri dari 1 buah *file* .zip berisi ***template*** **program** dan ***dataset*** dokumen QnA dalam bahasa Indonesia yang dapat diunduh pada LINK.
2. Lengkapi *template* program yang diberikan sesuai dengan petunjuk pengerjaan tugas yang disediakan.
3. Seluruh program (*file* .py) dan *folder* index yang telah dibuat dikumpulkan dalam satu *folder* dan dikonversi ke dalam format .zip dengan format penamaan **TugasX_NPM.zip**
   Contoh: Tugas1_2006123456.zip
4. Kumpulkan tugas pada submisi yang telah disediakan di SCeLe sebelum tanggal **27 September 2023, 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. Asisten dosen diperbolehkan membantu Anda dengan memberikan petunjuk.

Petunjuk Pengerjaan Tugas

Pada Tugas Pemrograman 1 ini, Anda diminta untuk membuat sendiri ***indexer*** **yang menghasilkan** ***inverted index*** dan mengimplementasikan model ***boolean retrieval*** sederhana. Pada tugas ini, sudah disediakan *template* program untuk memudahkan Anda dalam menyusun program.

Terdapat 5 buah program yaitu **bsbi.py**, **index.py**, **compression.py**, **util.py**, dan **search.py** yang sudah dilengkapi dokumentasi untuk membantu Anda. Bagian yang perlu Anda lengkapi sudah ditandai dengan comment #TODO. Selain berisi program, beberapa *file* juga telah diisi *sample test case* 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.

Berikut langkah pengerjaan untuk memudahkan Anda.

1. Unduh dan ekstrak *dataset*
   Dataset berasal dari multilingual MS Marco yang dapat diunduh pada LINK. Hasil zip dapat langsung diekstrak pada *folder* pengerjaan TP1. Keseluruhan *dataset* mencapai sekitar 190 ribu *file* dan dibutuhkan waktu yang cukup lama untuk ekstrak keseluruhan *file*. Oleh karena **waktu ekstraksi dan *indexing* yang cukup lama**, Anda disarankan untuk mulai mengerjakan tugas ini sebelum mendekati *deadline*.
2. Buat implementasi pada *file* **util.py**
   File ini berisi implementasi *mapping* sederhana untuk menyimpan pemetaan bagi sebuah *term* ke sebuah *integer* (term ID) dan sebuah dokumen ke sebuah *integer* (doc ID); serta sebaliknya.
3. 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, sedangkan *method* lainnya berperan sebagai *helper method*.
   Ketika melakukan *encoding*, *list of postings* perlu diubah ke dalam bentuk *list of gaps*. Kemudian, *list of gaps* dikompresi dengan Variable-Bytes Compression. Proses *decoding* perlu disesuaikan agar hasil kompresi bisa kembali semula.
4. 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*).
5. 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*, diperlukan *parsing block* untuk mengolah dokumen menjadi bentuk *list of pairs* <termID, docID>, *merge index* untuk menggabungkan seluruh *inverted indices* yang

sudah dibuat sebelumnya, serta *retrieve* dalam proses *searching* untuk mengambil dokumen-dokumen berdasarkan *query* yang diberikan.

Menyesuaikan alur dari *indexer* ini, Anda sebaiknya mengerjakan bagian terkait proses *indexing* terlebih dahulu, baru setelahnya bagian mengenai proses *searching.* Adapun langkah 1 dan 2 bisa dilakukan secara paralel karena kedua program tersebut bersifat independen dengan program lainnya.

Berikut langkah pengujian yang dapat Anda lakukan setelah seluruh program sudah diimplementasikan.

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, yakni sekitar 40-60 menit. Oleh karena itu, Anda dapat terlebih dahulu mencoba *indexing* pada sebagian blok terlebih dahulu untuk keperluan *debugging* bila diperlukan. Anda juga dapat mengimplementasikan struktur data yang dianggap lebih efisien daripada yang tersedia di *template*.
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. Lakukan juga perbandingan empiris terkait lamanya waktu saat *indexing* (*actual time*).

Submisi:

File .zip berisi util.py, compression.py, index.py, bsbi.py, dan *folder* index (serta 1 *file* .txt atau .pdf berisi hasil perbandingan algoritma kompresi *postings* lain sebagai bonus bila mengerjakan). Tidak perlu mengumpulkan folder collections.

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
- Bonifacio, L. H., Jeronymo, V., Abonizio, H. Q., Campiotti, I., Fadaee, M., Lotufo, R., & Nogueira, R. (2021). mMARCO: A Multilingual Version of MS MARCO Passage Ranking Dataset. arXiv [Cs.CL]. Retrieved from http://arxiv.org/abs/2108.13897

Selamat mengerjakan!