CSCE604135 • Temu-Balik Informasi
Semester Gasal 2024/2025
Fakultas Ilmu Komputer, Universitas Indonesia

Tugas Pemrograman 1: *Boolean Retrieval*

Tenggat Waktu: Rabu, 25 September 2024, 23.55 WIB

Ketentuan:

1. Anda diberikan beberapa *file* kode Python yang berisi *template* kode.
2. Kumpulkan **semua program** (*.py*) dan **laporan** (*.pdf*) jika ada, yang sudah dibuat dalam format *.zip* dengan penamaan **TP1_NPM.zip** melalui submisi SCeLe.
   Contoh penamaan file: TP1_2006524290.zip
3. Kumpulkan *file zip* tersebut pada submisi yang telah disediakan di SCeLe sebelum **Rabu, 25 September 2024, 23.55 WIB**. Keterlambatan pengumpulan akan dikenakan penalti.
4. Tugas ini dirancang sebagai tugas mandiri. **Plagiarisme tidak diperkenankan dalam bentuk apapun**. Adapun kolaborasi berupa diskusi (tanpa menyalin maupun mengambil jawaban orang lain) dan memanfaatkan informasi dari literatur manapun masih diperbolehkan. **Pastikan** untuk mencantumkan nama kolaborator dan referensi literatur.
5. Anda boleh berkonsultasi terkait tugas ini asisten dosen berikut. Asisten dosen diperbolehkan membantu Anda dengan memberikan petunjuk.
   a. Muhammad Ilham Ghozali
      Email: walangsigit@gmail.com
      Discord: myticalcat
   b. Jaycent Gunawan Ongris
      Email: jaycent.gunawan@ui.ac.id
      Discord: jaycentg

Petunjuk Pengerjaan Tugas

Pada tugas ini, Anda akan mengimplementasikan *inverted index* pada dataset yang diberikan dengan skema *blocked sort-based indexing* (BSBI). Sebelum melakukan *indexing*, ingat bahwa Anda wajib untuk melakukan *pre-processing*, seperti *tokenization*, *stemming*, dan *stopwords removal*. Dalam melakukan *indexing*, Anda wajib melakukan *compression* dengan menggunakan metode **VB Encoding**. Setelah melakukan *indexing*, program Anda diharapkan dapat dijalankan untuk melakukan *boolean retrieval* berdasarkan *query* yang diberikan.

Boolean Retrieval

Sistem *boolean retrieval* yang Anda buat diharapkan dapat mendukung pencarian dengan operasi AND, OR, dan DIFF (silakan merujuk pada *slides* untuk implementasi algoritma ini pada *iterables* yang sudah *sorted*). Penggunaan tanda kurung juga harus diimplementasikan untuk menunjukkan *precedence* dari operasi—yang mana yang perlu diselesaikan terlebih dahulu. Akan tetapi, ingat bahwa *precedence* antara AND, OR, dan DIFF setara karena ketiganya merupakan *set operators*.

Sebagai referensi, Anda dapat menggunakan ekspresi *postfix* untuk mengevaluasi *query* yang diberikan. Secara algoritma, evaluasi dalam ekspresi *postfix* akan jauh lebih mudah dibandingkan dengan menggunakan ekspresi *infix*. Anda dapat melakukan eksplorasi terkait algoritma Shunting-Yard untuk mengonversi ekspresi *infix* ke *postfix*. Sebagai contoh, ekspresi *postfix* dari operasi `2 + 3` adalah `2 3 +`. Contoh implementasi yang lebih jelas untuk tugas pemrograman ini dapat Anda lihat di *template* kode yang diberikan.

Contoh *query* **valid** yang dapat diberikan oleh pengguna adalah sebagai berikut.

- `term1 AND term2`
- `term1 AND (term2 OR term3)`
- `term1 AND (term2 OR (term3 DIFF term4))`
- `term1 AND term2 DIFF (term3 OR (term4 AND term5))`
- `term1 OR ((term2 AND term3) DIFF (term4 OR term5))`

Contoh *query* **tidak valid** adalah sebagai berikut. Program Anda tidak akan diuji berdasarkan ini.

- `term1 AND`
- `(term1 AND) term2`
- `term1 AND ((term2 OR term3)`

Jangan lupa untuk melakukan *pre-processing* pada *query* juga. Akan tetapi, Anda tidak perlu melakukan *stopwords removal* pada tahap ini, dengan asumsi tidak ada *stopwords* yang akan dimasukkan pada *query*. Terdapat suatu *method* di class `QueryParser` untuk melakukan validasi apakah suatu *query* mengandung *stopwords* atau tidak. Panggil *method* ini untuk melakukan validasi *query*, lalu kembalikan *list* kosong dan pesan bahwa terdapat *stopwords* pada *query*, jika memang terdapat *stopwords* pada *query*. Alasan mengapa *stopwords removal* pada *query* tidak diperlukan dikarenakan untuk mencegah suatu ekspresi hanya mempunyai satu *operand*, seperti jika `term1` dianggap sebagai *stopword*, maka setelah *stopwords removal*, ekspresi `term1 AND term2` hanya akan menjadi `AND term2`. Ini tentu akan menimbulkan isu pada evaluasi.

Urutan Pengerjaan Tugas

1. `util.py`, *file* ini berisi berbagai *classes* dan *methods* pembantu untuk *files* lainnya.
2. `compression.py`, *file* ini berisi berbagai algoritma kompresi yang akan digunakan untuk *indexing*. *File* ini dapat dikerjakan secara paralel dengan `util.py`, karena tidak ada *dependencies* antara keduanya.
3. `index.py`, *file* ini berisi abstraksi *class inverted index*. Pastikan Anda telah menyelesaikan `compression.py` sebelum mengerjakan *file* ini.
4. `bsbi.py`, *file* ini berfungsi untuk melakukan *indexing*. Pastikan semua *files* yang telah disebutkan sebelumnya sudah dikerjakan sebelum mengerjakan *file* ini.
5. `search.py`, *file* ini berfungsi untuk melakukan *boolean retrieval*.

Dataset

Anda akan bekerja dengan sampel dataset dari kumpulan abstrak dokumen di arXiv dalam bahasa Inggris. Total dokumen yang ada sebanyak kurang lebih 250.000 dokumen yang tersebar di 25 *folder*, di mana masing-masing *folder* berisi kurang lebih 10.000 dokumen. Tiap *folder* akan merepresentasikan satu *intermediate index*.

Bonus

Anda akan diberikan bonus 10 poin jika Anda mengimplementasikan satu algoritma kompresi tambahan selain VB Encoding setelah selesai mengerjakan tugas wajib. Setelah itu, buatlah suatu laporan yang berisi waktu *indexing* dan ukuran *file indices* yang terbentuk. Bandingkan kedua aspek tersebut untuk algoritma-algoritma kompresi yang Anda telah buat dan berikan justifikasi mengapa hal tersebut bisa terjadi. Lampirkan laporan dalam bentuk `pdf`.

Template Kode dan Dataset

- Template kode: tp1_template.zip
- Dataset: arxiv_collections.zip

Deliverables

Kompres semua *files* dan *folders* berikut ini ke dalam suatu zip *file* dengan format penamaan seperti yang disebutkan pada bagian sampul dokumen soal ini. Anda tidak perlu mengumpulkan *folder* `arxiv_collections`, cukup satu (atau dua, jika mengerjakan bonus) *folder index* yang dibuat saja.

- `bsbi.py`
- `compression.py`
- `index.py`
- `search.py`
- `util.py`
- `/index_vb`

Jika *folder index* yang dihasilkan terlalu besar sehingga tidak bisa di-*upload* di SCeLE, silakan meng-*upload folder* tersebut di *cloud storage* seperti Google Drive terlebih dahulu, lalu sertakan URL-nya di dalam zip *file deliverables*. Pastikan URL tersebut dapat diakses.

Catatan Revisi:

-

Rubrik Penilaian

Komponen	Proporsi
Implementasi `bsbi.py`	50%
Implementasi `index.py`	20%
Implementasi `util.py`	20%
Implementasi `compression.py`	10%
Implementasi algoritma kompresi tambahan (Bonus)	5%
Laporan (Bonus)	5%

Selamat mengerjakan!