



FAKULTAS
**ILMU
KOMPUTER**

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

Tugas Pemrograman 3 Learning-to-Rank

Deadline: 26 November 2022, 23:55 WIB

Ketentuan:

1. Tugas Pemrograman 3 ini terdiri dari 1 buah tutorial LETOR.
2. Buatlah program 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: Tugas3_1906262623.zip
4. Kumpulkan tugas pada submisi yang telah disediakan di SCeLe sebelum tanggal **26 November 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.

Learning To Rank: LambdaMART

Pengertian

Merupakan suatu teknik yang digunakan untuk menyelesaikan permasalahan Ranking. Perbedaan utama dari Learning to rank dan tradisional supervised learning lainnya adalah:

- Traditional Supervised Learning
Biasanya digunakan untuk melakukan prediksi berupa class atau angka numerikal skor dari suatu instance/data, lalu diurutkan.
- Learning to Rank
Menyelesaikan permasalahan *ranking* untuk melakukan urutan secara optimal. Learning to rank umumnya tidak terlalu memperdulikan skor yang didapat, lebih mementingkan hasil *order* (urutan).

LambdaMART

LambdaMART bersama dengan RankNet dan LambdaRank merupakan algoritma Learning to Rank yang ditemukan oleh Chris Burges dan rekannya di Microsoft Research. Ketiganya mentransformasikan ranking ke bentuk *pairwise classification* atau *regression problem*. LambdaMART merupakan perpaduan dari LambdaRank dan MART (Multiple Additive Regression Trees).

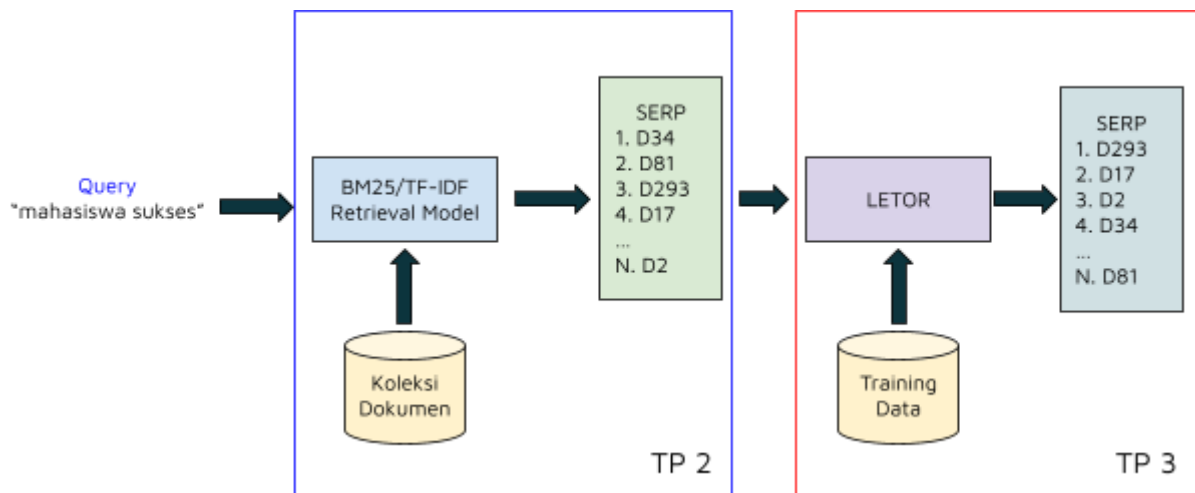
Terdapat dua implementasi populer LambdaMART:

- RankLib yang merupakan bagian dari Lemur Project.
- LightGBM yang disediakan oleh Microsoft.

Pada tutorial yang diberikan, akan menggunakan LightGBM sebagai contoh implementasi.

Cara Kerja LambdaMART menggunakan LightGB dapat dilihat pada link [ini](#)

Petunjuk Pengerjaan Tugas



Sebelumnya di Tugas Pemrograman 2, Anda diminta mengimplementasikan *ranked retrieval model* dengan pembobotan menggunakan sparse retrieval model, yaitu TF-IDF dan BM25. Output yang dihasilkan dari Tugas Pemrograman 2 adalah top-100 dokumen terbaik dari suatu query.

Pada Tugas Pemrograman 3 ini, anda akan menggunakan kembali program dari Tugas Pemrograman 2 yang nantinya akan ditambahkan dengan program LETOR (*learning-to-rank*). Anda diminta untuk me-ranking ulang pada hasil top-100 dari hasil ranking TP2 dengan menggunakan LambdaMART. Anda diminta untuk mempelajari tutorial Learning-to-rank yang telah disediakan dan mengimplementasikannya ke kode milik anda sendiri.

Berikut cara mengimplementasikan LETOR:

1. Persiapkan data yang akan dilakukan *re-ranking*
2. Membuat LSI/LSA Model
3. Train LightGBM LambdaMART Model
4. Melakukan Prediksi

Untuk memudahkan Anda, berikut *walkthrough* singkat terkait pengerjaan TP3:

1. Implementasi LETOR toolkit dengan mengadaptasi salah satu toolkit LETOR yang dapat diakses melalui tutorial berikut: ristek.link/TutorialLETOR. **Kemas code tersebut dalam suatu class atau fungsi dengan format .py.** Kode di notebook biasanya kurang rapih dan tidak modular. Anda perlu merapihkan kode di notebook tersebut dengan menggunakan class atau fungsi.
2. Bandingkan hasil ranking dengan menggunakan LETOR dan tidak, tuliskan hasil evaluasi dalam file .txt

Bonus

Sebelumnya Anda telah melakukan implementasi re-ranking menggunakan LETOR. Terkait bonus, silahkan lakukan eksperimen yang dapat meningkatkan hasil menjadi lebih baik, seperti menambahkan fitur-fitur baru, melakukan hyperparameter tuning pada library LambdaMART, atau coba cari dan implementasikan library LETOR lain sebagai perbandingan. Tuangkan hasil evaluasi mu dalam file berformat .txt. Kemudian, lakukan perbandingan terkait hasil dari sebelum dan setelah eksperimen dan tuliskan hasil evaluasi perbandingan tersebut.

Poin penilaian:

- letor.py 90 poin
- evaluasi.txt 10 poin
- Bonus 10 poin

Catatan:

- Untuk metrics, scorenya cukup 2 angka di belakang koma.
- Tugas akan dinilai bukan berbasis output, melainkan berbasis ide yang dikerjakan.
- Mohon untuk menuliskan kode dengan rapi agar penilai dapat mengerti ide dan kode yang dituliskan.

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>
- Data pada tugas ini merupakan koleksi Medline dari University of Glasgow: http://ir.dcs.gla.ac.uk/resources/test_collections/
- [Intuitive explanation of Learning to Rank \(and RankNet, LambdaRank and LambdaMART\)](#)
- [A Practical Guide to LambdaMART in LightGbm](#)

Selamat mengerjakan!