

Tugas Lab 2 (TL 2)
Analisis Hasil Tokenisasi Byte-Pair Encoding (BPE)
Deadline: Jumat 6 September 2024 jam 22:00

Pada tugas ini, Anda diminta untuk menghitung akurasi hasil tokenisasi algoritma Byte-Pair Encoding (BPE) dan menganalisis hasilnya.

1. APA YANG PERLU DIPAHAMI?

Untuk menyelesaikan tugas ini, Anda perlu memahami cara kerja BPE dan cara menghitung akurasi hasil tokenisasi.

1.1. Cara Kerja BPE. Sebelum algoritma BPE dijalankan, sekumpulan teks akan dipecah terlebih dahulu melalui suatu proses yang disebut sebagai pratokenisasi. Ada banyak pendekatan yang bisa digunakan, tetapi pada tugas ini, asumsikan bahwa pemecahannya berdasarkan *whitespaces*. Setiap token hasil pratokenisasi ini akan kembali dipecah ke karakter penyusunnya. Karakter-karakter tersebut akan menjadi nilai awal himpunan kosakata *tokenizer*. Selanjutnya, kosakata akan ditambah dengan hasil *merge* dua *pair* yang paling banyak muncul pada suatu iterasi. Proses ini akan dilakukan hingga jumlah kosakata final ada sebanyak `vocab_size`. Perhatikan bahwa *hyperparameter* ini berbeda dari `number of merges (k)` yang dijelaskan di kelas, tetapi berkaitan erat.

1.2. Cara Menghitung Akurasi *Tokenizer*. Akurasi *tokenizer* didefinisikan sebagai berikut.

$$Akurasi = \frac{T}{N}$$

dengan:

- 1) T = Jumlah token di *gold standard* yang berhasil ditokenisasi dengan benar.
- 2) N = Jumlah token di *gold standard*.

Sebagai contoh, misalkan berikut adalah perbandingan antara hasil tokenisasi *gold standard* dan *tokenizer* dari dua kalimat.

TABLE 1. Perbandingan hasil tokenisasi *gold standard* dan *tokenizer*

| No. | Tokenisasi <i>gold standard</i> | Tokenisasi <i>tokenizer</i> |
|-----|--|--|
| 1 | 'Buku', 'nya', 'mahal', '' | 'Bukunya', 'mahal', '' |
| 2 | 'Seharusnya', 'kamu', 'tidak', 'terlambat', '' | 'Seharus', 'nya', 'kamu', 'tidak', 'terlambat', '' |

Pada contoh di atas, bisa dilihat bahwa jumlah token dari *tokenizer* yang benar pada kalimat pertama ada dua ('mahal' dan '') sedangkan pada kalimat kedua ada empat ('kamu', 'tidak', 'terlambat', dan ''). Lalu, diketahui bahwa jumlah token pada *gold standard* ada sembilan (jumlah semua token pada kolom pertama). Oleh karena itu, akurasinya adalah $6/9 \approx 0.67$. Perhatikan bahwa perhitungan ini tidak sesederhana menghitung jumlah token yang sama, melainkan harus mempertimbangkan juga urutan tokennya.

2. APA YANG HARUS DIKERJAKAN?

Setelah memahami informasi di atas, berikut adalah beberapa hal yang perlu dilakukan.

- 1) *Clone*/unduh repositori GitHub [ini](#) yang berisi:
 - `bpe.ipynb`: Kode untuk melatih dan menguji *tokenizer* yang perlu dilengkapi.
 - `akurasi.py`: Kode untuk menghitung akurasi yang perlu dilengkapi.
 - `train/test.jsonl`: Dataset untuk *training/testing* yang diadaptasi dari [UD Indonesian-GSD](#).

- 2) Pada `bpe.ipynb`, isi NPM Anda pada variabel `npm`. Perhatikan bahwa NPM Anda akan mempengaruhi sampel dari dataset *testing* yang Anda dapatkan.
- 3) Latih *tokenizer* pada dataset *training* dengan tiga nilai `vocab_size` berbeda: 500, 1000, dan 2000. Artinya, Anda memiliki tiga tokenizer yang berkorespondensi dengan nilai `vocab_size` berbeda.
- 4) Implementasikan algoritma perhitungan akurasi pada `akurasi.py` (perhatikan contoh pemanggilannya pada baris ke-40). Lalu, gunakan implementasi tersebut untuk menguji akurasi *tokenizer* pada `bpe.ipynb`.
- 5) Buat laporan dengan format `pdf` yang berisikan:
 - (a) Penjelasan singkat cara kerja algoritma Anda.
 - (b) Tiga nilai akurasi *tokenizer* (untuk masing-masing `vocab_size`).
 - (c) Analisis singkat terkait pengaruh nilai `vocab_size` yang berbeda terhadap nilai akurasi yang didapatkan.

3. PENGUMPULAN TUGAS

Kumpulkan *file* berikut dalam satu file *zip*:

- 1) *File* laporan dengan aturan nama `NLP-TL2-[NamaAnda].pdf`.
Contoh: `NLP-TL2-LuthfiBalaka.pdf`.
- 2) *File* `BPE.ipynb` **dan** `akurasi.py` yang sudah dilengkapi.

Catatan: Pastikan *file* laporan tidak bermasalah **dan** program bisa dijalankan tanpa *error*. Akan ada pengurangan poin jika ada masalah terkait kedua hal tersebut.