Asdos PJ: Muhammad Faisal Adi Soesatyo
CSCE604231: PENGOLAHAN BAHASA MANUSIA (NLP)
September 25, 2024

**Tugas Lab 4 (TL 4)**
**Eksperimen Pengembangan N-gram *Language Model***
**Deadline Jumat, 27 September 2024 jam 22:00**

Pada Tugas Lab 4 ini, peserta diminta untuk melakukan eksperimen pengembangan N-gram *Language Model* serta mengukur kualitas model yang dibangun menggunakan metrik *perplexity*.

## 1. Apa yang Harus Dikerjakan?

Data yang digunakan pada tugas lab ini berasal dari https://dumps.wikimedia.org/idwiki/latest/idwiki-latest-pages-articles.xml.bz2. Konten dataset tersebut merupakan potongan kalimat dari artikel yang terdapat pada laman wikimedia.

Secara umum terdapat tiga hal yang perlu dikerjakan pada tugas ini,

1) Pembangunan model
2) Evaluasi model
3) Analisis model

### 1.1. Pembangunan Model.

Terkait task pembangunan model terdapat beberapa hal yang perlu diperhatikan

1) *Clone*/unduh repository GitHub ini yang berisi kode program dari eksperimen pengembangan *n-gram language model* ini.
2) Pada repository tersebut terdapat dua file utama yang perlu diperhatikan, yakni `preprocess_data.py` serta `ngram_lm.py`.
   (a) File `preprocess_data.py` merupakan file untuk melakukan *preprocess* pada dataset yang akan digunakan. Beberapa tahap *preprocessing* yang perlu dilakukan
   - Load dataset yang telah disediakan. Terdapat dua jenis dataset, yakni `train-set` serta `test-set` masing-masing berjumlah 80 dan 20 kalimat.
   - Memecah korpus tersebut menjadi beberapa kalimat.
   - Terdapat dua macam skenario yang perlu Anda lakukan,
     – Melakukan lowercasing (*uncased*)
     – Membiarkan dataset apa adanya (*cased*)
   - Melakukan *tokenization* untuk masing-masing kalimat tersebut. Agar sesuai dengan bahasa dari dataset, Anda diharapkan menggunakan library Aksara[1] untuk melakukan *tokenization*.

   Setelah preprocessing selesai dilakukan, Anda melakukan tahap awal pembangunan n-gram model. Beberapa tahap tersebut di antaranya,
   - Membuat dictionary yang berisi pasangan kata dan kemunculan kata tersebut di dalam corpus.
   - Dari dictionary tersebut, membangun koleksi kata (*vocabulary*) dengan *threshold* sebesar 3.
   - Dengan memanfaatkan *vocabulary* tersebut, Anda perlu melakukan penanganan OOV.
   - Simpan ke dalam file sebagai *pickle*.

   (b) File `ngram_lm.py` merupakan file untuk membangun model, *generate probabilities*, serta menghitung perplexity. Terdapat beberapa method yang perlu menjadi perhatian pada file ini, yakni
   - `generate_n_grams` merupakan method untuk menghasilkan seluruh kemungkinan n-gram yang terdapat di corpus.

[1] https://github.com/ir-nlp-csui/aksara

- `count_probability` merupakan method untuk menghitung probabilitas suatu n-gram. Pada method ini, Anda **diharuskan** menggunakan *add-one (laplace) smoothing*. **Dilarang** menggunakan *Maximum Likelihood Estimation (MLE)*.
- `count_perplexity` merupakan method untuk menghitung perplexity dengan memanfaatkan method `count_probability` dari model yang telah Anda bangun pada `generate_n_grams`. Terdapat beberapa hal yang perlu diperhatikan pada method ini, yakni
  - Ukuran dataset yang dibangun akan cukup besar, sehingga implementasi **diharapkan** menggunakan penjumlahan logaritma untuk mencegah terjadinya *underflow* maupun *overflow*.
  - Method perplexity diharapkan bekerja pada level *corpus*, bukan pada level kalimat.

  Kedua informasi di atas dapat Anda akses pada Slide N-gram LM - SCELE.

1.2. **Evaluasi Model.** Setelah program untuk membangun n-gram model selesai pada tahap 1.1, Anda diharapkan membangun dua jenis model, yakni unigram serta bigram.

Berikut adalah hal yang perlu Anda lakukan dalam task evaluasi model,

1) Membangun dua jenis gram model, yakni **unigram** dan **bigram**.
   Mengingat pada tahap pertama Anda telah membuat abstraksi untuk sebuah n-gram model maka untuk membangun unigram serta bigram bukanlah hal yang rumit. Anda dapat memanfaatkan method `generate_n_grams` yang terdapat pada file `ngram_lm.py` dengan mengatur nilai parameter $n$, di mana $n$ merupakan jenis gram LM yang ingin Anda bangun. $n = 1$ artinya unigram, $n = 3$ artinya trigram, dst.
2) Membuat lima kalimat untuk masing-masing model, sehingga terdapat total 10 kalimat untuk diuji pada kedua gram model tersebut. Kriteria untuk masing-masing kalimat sebagai berikut,
   - Menggunakan bahasa Indonesia.
   - Berjumlah sebanyak 7-10 token termasuk dengan start-stop token (`<s>` dan `</s>`).
   - Terkait *sentence generator* Anda dapat memanfaatkan method `probabilities_for_all_vocab` yang terdapat pada file `ngram_lm.py`.
   - Apabila *generated token* sama dengan *given word*, maka Anda dipersilakan untuk memilih kata dengan probabilitas tertinggi kedua. Apabila ternyata kata dengan probabilitas tertinggi kedua masih sama dengan *given word*, Anda dipersilakan memilih kata dengan probabilitas tertinggi ketiga, dst.
3) Uji perplexity pada masing-masing kalimat tersebut di masing-masing model yang telah dibangun.

1.3. **Analisis Model.** Pada tahap ini, Anda diharapkan melaporkan hasil pengembangan serta evaluasi model yang telah dilakukan sebelumnya. Terdapat beberapa poin yang perlu dilaporkan, di antaranya

1) Pada tahap 1.1 disebutkan bahwa terdapat dua macam skenario yang perlu dilakukan, yakni *lowercasing (uncased)* serta *no lowercasing (cased)*. Bagaimana performa kedua model, baik unigram maupun bigram dari segi perplexitynya? Model mana yang secara perplexity lebih baik?
2) Proses apa yang Anda lakukan saat implementasi *tokenization* menggunakan *library* Aksara[2]?
3) Laporkan kalimat yang telah Anda hasilkan pada tahap 1.2 untuk model unigram dan bigram. Bagaimana pendapat Anda terkait masing-masing kalimat tersebut? Anda dipersilakan meninjau dari sisi tata bahasa, keselarasan makna antar kata, dsb.
4) Hitunglah *probability* keempat kalimat berikut dengan model unigram dan bigram yang telah Anda bangun.
   (a) **Unigram**
   - `<s>` saya sedang menunggu di peron 5 stasiun tersebut `</s>`
   - `<s>` para pekerja terlihat lincah saat membersihkan lokomotif tersebut `</s>`

   (b) **Bigram**
   - `<s> <s>` pak ustad berceramah di atas mimbar masjid `</s>`
   - `<s> <s>` para murid diajarkan budi pekerti di sekolah `</s>`

[2]https://github.com/ir-nlp-csui/aksara

5) Laporkan ukuran masing-masing vocab pada model unigram dan bigram yang telah Anda bangun. Vocabulary tersebut termasuk dengan komponen `<UNK>`.
6) Lakukan analisis pada nilai perplexity untuk masing-masing kalimat di kedua model unigram dan bigram yang telah Anda lakukan pada tahap 1.2.

## 2. Objektif TL 4

Memahami konsep serta cara kerja *rule based n-gram language model*.

## 3. Luaran TL 4

Luaran dari TL 4 ini terdapat dua jenis, yakni

1) Kode program N-gram Language Model.
2) Laporan hasil analisis model pada tahap 1.3.

Kode program hasil pengembangan beserta laporan analisis tahap 1.3 harap dikumpulkan sebagai zip file. Di dalam zip file tersebut setidaknya harus mengandung file-file serta folder berikut,

- `NLP-TL4-SourceCode-[NPM]/`
  - `main.py`
  - `ngram_lm.py`
  - `preprocess_data.py`
  - `requirements.txt`
  - `data/`
    - `idwiki-corpus.txt`
    - `idwiki-train.txt`
    - `idwiki-test.txt`
- `NLP-TL4-LaporanHasilAnalisis-[NPM].pdf`

Mohon struktur serta penamaan masing-masing file serta folder diperhatikan agar tidak terjadi kesalahan tulis. Penalti sebesar **10 poin** apabila Anda mengumpulkan serta penamaan *file* tidak sesuai dengan ketentuan di atas.

## 4. Pengumpulan Tugas

*File* yang dikumpulkan:

1) Zip file dengan format penamaan `NLP-TL4-[NamaMahasiswa].zip`.
   Contoh: `NLP-TL4-CommanderKowalski.zip`.
2) Anda akan dikenakan penalti sebesar **5 poin** apabila format serta penamaan *file* yang Anda submit tidak sesuai ketentuan.