

[과제 3] 대규모 말뭉치(KCC 원시말뭉치)에서 가장 유사한 문장 상위 n개 추출

<과제 목적> 대규모 말뭉치 (빅 텍스트 데이터) 분석의 문제점, 해결방안 실습

- 입력: 한글 문장 1개
- 출력: 입력문장과 가장 유사한 n개의 문장 추출 및 유사도, 소요시간 출력
- 방법: 위 2번의 형태소분석 (또는 WPM) 방식의 유사도 계산 모듈을 대규모 말뭉치에 적용, 말뭉치의 각 문장들과 순서대로 비교하여 가장 유사도가 높은 상위 n개 문장 및 유사도, 소요시간을 출력

n값은 실행할 때 command line 인자 (또는 사용자 입력) 으로 받음

<참고1>

말뭉치의 각 문장에서 토큰 추출 방식은 입력문장에서 토큰 추출과 동일한 방식 (형태소 분석기 또는 WPM) 을 사용해야 함.

<참고2>

말뭉치의 각 문장에 대한 토큰 추출은 유사도 비교 전에 한꺼번에 배치처리 방식으로 처리하여 저장하는 방법, 또는 유사도 계산 직전에 토큰을 추출해도 됨.

이 때 문장에서 토큰 추출하는 시간을 반드시 소요시간 (실행시간) 에 포함시켜야 함

<참고3>

말뭉치의 각 문장/라인을 미리 소팅한 후에 사용해도 되며, 소팅 시간은 소요시간 (실행시간) 에서 제외함.

<참고4>

각 문장의 어절들을 문장 내에서 소팅하여 저장한 후에 사용해도 되며, 이 처리시간은 소요시간 (실행시간) 에서 제외함.

- 제출하는 모든 실습 및 과제의 소스 코드는 아래 GitHub 경로에 있습니다.

<https://github.com/joshua-dev/bigdata>

실행 결과

tokenizer (SPM model) 생성

```
(venv) ~ /KMU/2020-1/bigdata/bigdata/week03 > make argv=10
vocabulary_size >> 10000
sentencpiece_trainer.cc(116) LOG(INFO) Running command: --input=../src/spm/KCC940_Korean_sentences_UTF8.txt --model_prefix=bpe
ix=BPE --vocab_size=10000
sentencpiece_trainer.cc(49) LOG(INFO) Starts training with :
TrainSpec {
    input_file: ../src/spm/KCC940_Korean_sentences_UTF8.txt
    input_format: BPE
    model_prefix: BPE
    model_type: UNIGRAM
    vocab_size: 10000
    self_test_sample_size: 0
    character_coverage: 0.9995
    input_sentence_size: 0
    shuffle_input_sentence: 1
    seed_sentencepiece_size: 1000000
    shrinking_factor: 0.75
    max_sentencepiece_length: 4192
    num_threads: 6
    num_sub_iterations: 2
    max_sentencepiece_length: 16
    split_by_unicode_script: 1
    split_by_number: 1
    split_by whitespace: 1
    treat_whitespace_as_suffix: 0
    hard_vocab_limit: 1
    use_all_vocab: 0
    unk_id: 0
    bos_id: 1
    eos_id: 2
    pad_id: -1
    unk_piece: <unk>
    bos_piece: <>
    eos_piece: </>
    pad_piece: <pad>
    unk_surface: ?
}
NormalizerSpec {
    name: mmt_nfkc
    add_dummy_prefix: 1
    remove_extra_whitespaces: 1
    escape_whitespaces: 1
    normalization_rule_tsv:
}
trainer_interface.cc(267) LOG(INFO) Loading corpus: ../src/spm/KCC940_Korean_sentences_UTF8.txt
trainer_interface.cc(315) LOG(INFO) Loaded all 50000 sentences
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <unk>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: </>
```

```
Python

eos_id: 2
pad_id: -1
unk_piece: <unk>
bos_piece: <>>
eos_piece: </><
pad_piece: <pad>
unk_surface: ???

}
NormalizerSpec {
    name: mmf_nfkc
    add_dummy_prefix: 1
    remove_extra_whitespaces: 1
    escape_whitespaces: 1
    normalization_rule_tsv:
}

trainer_interface.cc(267) LOG(INFO) Loading corpus: ./src/spm/KCC940_Korean_sentences_UTF8.txt
trainer_interface.cc(315) LOG(INFO) Loaded all 50000 sentences
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <unk>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <>>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: </><
trainer_interface.cc(335) LOG(INFO) Normalizing sentences..
trainer_interface.cc(384) LOG(INFO) all chars count=383555
trainer_interface.cc(392) LOG(INFO) Done! 99.952% characters are covered.
trainer_interface.cc(402) LOG(INFO) Alphabet size=1124
trainer_interface.cc(403) LOG(INFO) Final character coverage=0.999502
trainer_interface.cc(403) LOG(INFO) Done! preprocessed 50000 sentences.
unigram_model_trainer.cc(129) LOG(INFO) Making suffix array...
unigram_model_trainer.cc(133) LOG(INFO) Extracting frequent sub strings...
unigram_model_trainer.cc(184) LOG(INFO) Initialized 121066 seed sentencieces
trainer_interface.cc(441) LOG(INFO) Tokenizing input sentences with whitespace: 50000
trainer_interface.cc(441) LOG(INFO) Done! 179546
unigram_model_trainer.cc(470) LOG(INFO) Found 179546 sentences for EM training
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=66907 obj=14.2945 num_tokens=408743 num_tokens/piece=6.10912
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=60041 obj=13.0399 num_tokens=410596 num_tokens/piece=6.83859
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=45025 obj=13.0625 num_tokens=426458 num_tokens/piece=9.47184
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=44990 obj=13.0224 num_tokens=426771 num_tokens/piece=9.48591
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=33742 obj=13.1862 num_tokens=447472 num_tokens/piece=13.2611
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=33740 obj=13.1487 num_tokens=447499 num_tokens/piece=13.2632
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=25300 obj=13.361 num_tokens=449615 num_tokens/piece=18.5582
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=25300 obj=13.317 num_tokens=449612 num_tokens/piece=18.5581
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=18978 obj=13.5784 num_tokens=45336 num_tokens/piece=25.9899
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=18978 obj=13.5254 num_tokens=45333 num_tokens/piece=25.9898
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=14233 obj=13.0351 num_tokens=51793 num_tokens/piece=36.3528
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=14233 obj=13.7754 num_tokens=518018 num_tokens/piece=36.3556
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter0 size=11000 obj=14.0986 num_tokens=541137 num_tokens/piece=49.1943
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter1 size=11000 obj=14.0302 num_tokens=541137 num_tokens/piece=49.1943
trainer_interface.cc(507) LOG(INFO) Saving model: BPE.model
trainer_interface.cc(531) LOG(INFO) Saving vocabs: BPE.vocab
Enter a Hangul sentence ... -
```

상위 n개 유사 문장 추출 및 소요 시간 출력

- 10000개의 단어로 학습했을 때

```
bash
pad_piece: <pad>
unk_surface: ???
}
NormalizerSpec {
    name: nmt_nfkc
    add_dummy_prefix: 1
    remove_extra_whitespaces: 1
    escape_whitespaces: 1
    normalization_rule_tsv:
}

trainer_interface.cc(267) LOG(INFO) Loading corpus: ./src/spm/KCC940_Korean_sentences_UTF8.txt
trainer_interface.cc(315) LOG(INFO) Loaded all 50000 sentences
trainer_interface.cc(330) LOG(INFO) Adding meta_piece<>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <>-
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: </>-
trainer_interface.cc(335) LOG(INFO) Normalizing sentences...
trainer_interface.cc(384) LOG(INFO) all chars count=383555
trainer_interface.cc(392) LOG(INFO) Done: 99.9502% characters are covered.
trainer_interface.cc(402) LOG(INFO) Alphabet size=1124
trainer_interface.cc(403) LOG(INFO) Final character coverage=0.999502
trainer_interface.cc(435) LOG(INFO) Done! preprocessed 50000 sentences.
unigram_model_trainer.cc(129) LOG(INFO) Making suffix array...
unigram_model_trainer.cc(184) LOG(INFO) Extracting frequent seed strings...
unigram_model_trainer.cc(441) LOG(INFO) Tokenizing input sentences with whitespace: 50000
trainer_interface.cc(451) LOG(INFO) Done! 179546
unigram_model_trainer.cc(470) LOG(INFO) Using 179546 sentences for EM training
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=6907 obj=14.2945 num_tokens=408743 num_tokens/piece=6.10912
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=60041 obj=13.0399 num_tokens=410596 num_tokens/piece=6.83859
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=54987 obj=12.9966 num_tokens=414685 num_tokens/piece=7.54151
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=54816 obj=12.9776 num_tokens=415084 num_tokens/piece=7.57231
trainer_interface.cc(507) LOG(INFO) Saving model: BPE.model
trainer_interface.cc(531) LOG(INFO) Saving vocabcs: BPE.vocab
Enter a Hangul sentence: 어느날 머리에서 뿐이 자랐다
1. 4%: 50.000000%
2. 10%: 50.000000%
3. 5%: 50.000000%
4. 7%: 50.000000%
5. 16%: 50.000000%
6. 17%: 50.000000%
7. LAC-저스 투수 류현진이 지난달 27일 샌프란시스코 자이언츠와의 원정경기에서 투구하고 있다: 44.444444%
8. 흥 전 지사는 23일 오전 자신의 페이스북을 통해 자는 어릴 때부터 낙동강변에서 자랐고 국회 한경노동위원회에 5년을 있었기 때문에 이를 잘 알고 있다며 이같이 밝혔다: 44.444444%
9. 그러면서 중요한 선거를 앞둔 이 시점에 어처구니없는 짓을 저질러 자유한국당에 공격의 발미를 제공한 자가 그 드루킹이라는 것을 알게 되니 미리에서 갑자기 스팀이 올라오면서 뚜껑이 확 열린다고 꼬집었다: 44.444444%
10. 중앙선 기관리워먼트는 6일 보도자료를 배포하고 전국 선거·보궐 선거·국회 의원 재·보궐 선거 전투표율 89%부터 9일 까지 실시한다며 별도의 선고는 필요 없지만 전 가권이 있는 사람으면 전국 어느 사전투표소에서 투표할 수 있다고 밝혔다: 33.333333%
Elapsed time: 6.718964s
(venv) ~/KMU/2026-1/bigdata/bigdata/week03 > _
```

- 20000개의 단어로 학습했을 때

```
bash
name: nmt_nfkc
add_dummy_prefix: 1
remove_extra_whitespaces: 1
escape_whitespaces: 1
normalization_rule_tsv:
}

trainer_interface.cc(267) LOG(INFO) Loading corpus: ./src/spm/KCC940_Korean_sentences_UTF8.txt
trainer_interface.cc(315) LOG(INFO) Loaded all 50000 sentences
trainer_interface.cc(330) LOG(INFO) Adding meta_piece<>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <>-
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: </>-
trainer_interface.cc(335) LOG(INFO) Normalizing sentences...
trainer_interface.cc(384) LOG(INFO) all chars count=383555
trainer_interface.cc(392) LOG(INFO) Done: 99.9502% characters are covered.
trainer_interface.cc(402) LOG(INFO) Alphabet size=1124
trainer_interface.cc(403) LOG(INFO) Final character coverage=0.999502
trainer_interface.cc(435) LOG(INFO) Done! preprocessed 50000 sentences.
unigram_model_trainer.cc(129) LOG(INFO) Making suffix array...
unigram_model_trainer.cc(184) LOG(INFO) Extracting frequent seed strings...
unigram_model_trainer.cc(441) LOG(INFO) Tokenizing input sentences with whitespace: 50000
trainer_interface.cc(451) LOG(INFO) Done! 179546
unigram_model_trainer.cc(470) LOG(INFO) Using 179546 sentences for EM training
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=6907 obj=14.2945 num_tokens=408743 num_tokens/piece=6.10912
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=60041 obj=13.0399 num_tokens=410596 num_tokens/piece=6.83859
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=54987 obj=12.0625 num_tokens=426458 num_tokens/piece=9.4714
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=44990 obj=13.0224 num_tokens=426771 num_tokens/piece=9.48591
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=3742 obj=13.1862 num_tokens=447472 num_tokens/piece=13.2616
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=3740 obj=13.1487 num_tokens=447499 num_tokens/piece=13.2632
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=25305 obj=13.364 num_tokens=469515 num_tokens/piece=18.5582
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=25305 obj=13.317 num_tokens=469512 num_tokens/piece=18.5581
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=0 size=22000 obj=13.432 num_tokens=480227 num_tokens/piece=21.8285
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iter=1 size=22000 obj=13.41 num_tokens=480225 num_tokens/piece=21.8284
trainer_interface.cc(507) LOG(INFO) Saving model: BPE.model
trainer_interface.cc(531) LOG(INFO) Saving vocabcs: BPE.vocab
Enter a Hangul sentence: >> 어느날 머리에서 뿐이 자랐다
1. 그러면서 중요한 선거를 앞둔 이 시점에 어처구니없는 짓을 저질러 자유한국당에 공격의 발미를 제공한 자가 그 드루킹이라는 것을 알게 되니 미리에서 갑자기 스팀이 올라오면서 뚜껑이 확 열린다고 꼬집었다: 60.000000%
2. LAC-저스 투수 류현진이 지난달 27일 샌프란시스코 자이언츠와의 원정경기에서 투구하고 있다: 50.000000%
3. 0: 50.000000%
4. go: 50.000000%
5. 4%: 50.000000%
6. 0: 50.000000%
7. co: 50.000000%
8. 9%에서 1: 50.000000%
9. 1%: 50.000000%
10. 0%에서 3: 50.000000%
Elapsed time: 7.679332s
(venv) ~/KMU/2026-1/bigdata/bigdata/week03 > _
```

- 50000개의 단어로 학습했을 때

```

bash
}
trainer_interface.cc(267) LOG(INFO) Loading corpus: ./src/spm/KCC940_Korean_sentences_UTF8.txt
trainer_interface.cc(315) LOG(INFO) Loaded all 50000 sentences
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <unk>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: <>
trainer_interface.cc(330) LOG(INFO) Adding meta_piece: </>
trainer_interface.cc(335) LOG(INFO) Normalizing sentences...
trainer_interface.cc(384) LOG(INFO) all chars count=3835555
trainer_interface.cc(392) LOG(INFO) Done: 99.9502% characters are covered.
trainer_interface.cc(402) LOG(INFO) Alphabet size=1124
trainer_interface.cc(406) LOG(INFO) Final character coverage=0.999502
trainer_interface.cc(452) LOG(INFO) Extracting frequent sub strings...
unigram_model_trainer.cc(129) LOG(INFO) Making suffix array...
unigram_model_trainer.cc(133) LOG(INFO) Extracting frequent sub strings...
unigram_model_trainer.cc(184) LOG(INFO) Initialized 121866 seed sentencieces
trainer_interface.cc(441) LOG(INFO) Tokenizing input sentences with whitespace: 50000
trainer_interface.cc(451) LOG(INFO) Done: 179546
unigram_model_trainer.cc(470) LOG(INFO) Using 179546 sentences for EM training
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=0 size=66907 obj=14.2945 num_tokens=408743 num_tokens/piece=6.10912
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=60041 obj=13.0399 num_tokens=410596 num_tokens/piece=6.83859
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=0 size=4525 obj=13.0625 num_tokens=426459 num_tokens/piece=9.4714
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=44998 obj=13.0224 num_tokens=426771 num_tokens/piece=9.48591
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=44992 obj=13.0224 num_tokens=426771 num_tokens/piece=9.48591
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=44996 obj=13.0224 num_tokens=426771 num_tokens/piece=9.48591
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=44997 obj=13.0224 num_tokens=426771 num_tokens/piece=9.48591
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=0 size=5305 obj=13.364 num_tokens=469615 num_tokens/piece=18.5582
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=5305 obj=13.317 num_tokens=469612 num_tokens/piece=18.5581
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=0 size=14233 obj=13.7754 num_tokens=518018 num_tokens/piece=36.3956
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=0 size=11000 obj=14.0906 num_tokens=541137 num_tokens/piece=49.1943
unigram_model_trainer.cc(486) LOG(INFO) EM sub_iters=1 size=11000 obj=14.0302 num_tokens=541137 num_tokens/piece=49.1943
trainer_interface.cc(507) LOG(INFO) Saving model: BPE.model
trainer_interface.cc(531) LOG(INFO) Saving vocabs: BPE.vocab
Enter a Hangul sentence-> 어느 날 머리에서 뿔이 자랐다
1: 4941 : 66.666667%
2: 1841 : 66.666667%
3: 7801 : 66.666667%
4: 중앙선거관리위원회는 6일 보도자료를 배포하고 지방선거 및 국회의원 재·보선 사전투표를 8일부터 9일 까지 실시한다며 별도의 선고는 필요 없으며 선거권이 있는 사람이라면 전국 어느 사전투표소에서 투표할 수 있다고 밝혔다 : 50.000000%
5: LACI저스 투수 류현진이 지난달 27일 샌프란시스코 자이언츠와의 원정경기에서 투구하고 있다: 50.000000%
6: 0: 50.000000%
7: 정세균 국회 의장이 12일 오전 국회 의사당 잔디광장에서 열린 2018 평창 동계올림픽 마스코트 제작식에서 수호랑과 반다비를 배경으로 자원봉사자 및 서동초등학교 어린이들과 파이팅을 외치고 있다: 50.000000%
8: 1%p다 : 50.000000%
9: 0: 50.000000%
10. 1%포인트나: 50.000000%
Elapsed time: 8.875745s
(venv) ~/KMU/2028-1/bigdata/bigdata/week03 >
```

Installation

```
pip install -r requirements.txt
```

Run

```
// n means the number of high similarity sentences to be printed.  
// ex) make argv=10  
make argv=n
```

구현 방법 및 결과 분석

Google에서 제공하는 SPM model의 tokenizer API (sentencepiece) 를 사용했다.

주어진 문장과 가장 유사한 상위 n개의 문장을 추출하는 알고리즘은 다음과 같이 구현했다.

1. KCC 원시 말뭉치를 이용하여 spm model (bpe model) 을 생성한다.
2. processor를 생성하고 1을 통해 얻은 model을 로딩하여 tokenizing 준비를 마친다.
3. 문장이 입력되면 processor를 통해 tokenizing을 수행한다.
4. KCC 원시 말뭉치를 읽어들여 문장 단위로 자르고 문장 부호를 제거하여 processor에게 전달한다.
5. processor는 4로부터 전달받은 각 문장들을 tokenizing한다.
6. 3과 5로부터 얻은 입력 문장 토큰과 KKC 원시 말뭉치의 각 문장들에 대한 토큰을 비교하여 유사도를 계산하고 연관 배열에 저장 한다.
7. 6에서 얻은 연관 배열을 유사도 순으로 내림차순 정렬하고 유사도가 가장 높은 상위 n개 문장과 수행 시간을 출력한다.

이 때 model 생성 (학습) 및 토큰 추출 시간은 중간에 I/O가 발생하기 때문에 각각 측정하여 마지막에 더한 값을 최종 소요 시간으로 사용했으며 sorting하는 데에 걸린 시간은 포함시키지 않았다.

또한 model을 생성할 때 몇개의 문장으로 학습시킬지 입력하는데, 이를 vocabulary size라 한다.

이번 과제에선 vocabulary size를 10000개, 20000개, 50000개로 늘려가면서 유사한 문장 추출의 변화를 관찰했다.

그 결과 같은 말뭉치로 학습한 모델에 같은 문장을 입력해도 vocabulary size에 따라 tokenizing 방법이 달라지기 때문에 유사한 문장들의 유사도 순위가 바뀔 수 있고

순위가 같더라도 유사도가 다를 수 있는 것을 확인할 수 있다.

- 1 을 구현한 함수 train

```
def train(vocab_size: int):  
    ...  
    train creates spm model using a given text data.  
    @param vocab_size the number of sentences the model will use to train  
    ...  
    spm.SentencePieceTrainer.Train(f'--input=./src/spm/KCC940_Korean_sentences_UTF8.txt --model_prefix=BPE --vocab_size={vocab_si
```

- 2 ~ 6 을 구현한 함수 get

```
def get(inputSentence: str) -> dict:  
    ...  
    get returns similarites of sentences in KCC sentences to a given Hangul sentence.  
    @param inputSentence a Hangul sentence  
    @return a map containing similarities between sentence of KCC sentences and input sentence  
    ...
```

이 때, 6 에서 입력한 문장과 KCC 원시 말뭉치의 문장 간 유사도를 계산하기 위해 compare 함수를 구현했다.

```
def compare(tokens_first: list, tokens_second: list) -> float:  
    ...  
    compare compares two token lists using BPE model.  
    @param tokens_first a Hangul sentence tokens.  
    @param tokens_second a Hangul sentence tokens.  
    @return similarity between two sentence tokens using BPE model.  
    ...
```

- 7 을 구현한 함수 main

```
def main(argv):  
    ...  
    main runs training model, sorting and printing results and timing.  
    @param argv command-line parameter  
    ...
```