우리 같이 함께

Written by. GPT-2

제3회
AI
BOOKATHON

INDEX

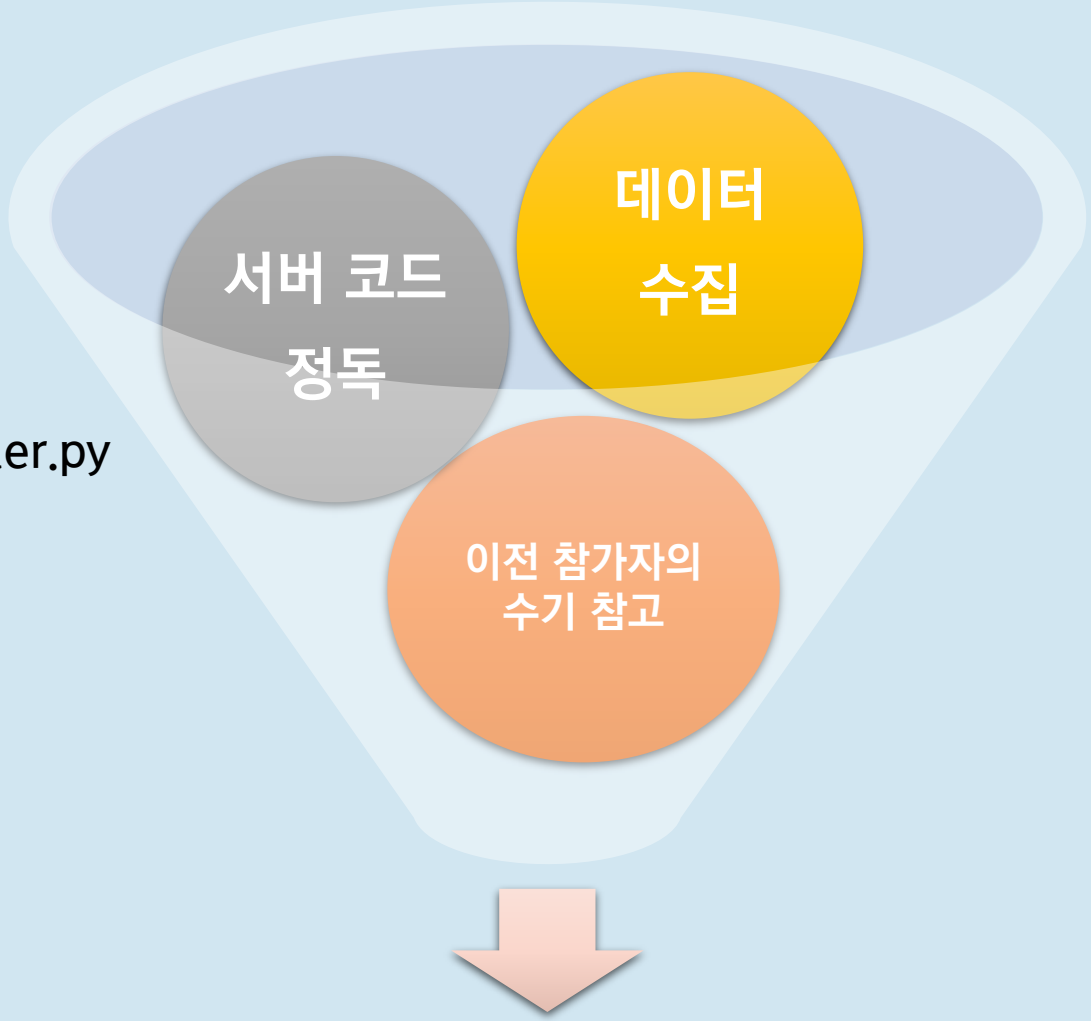
TEAM

데컬코마니

김유진
송재현
양지인

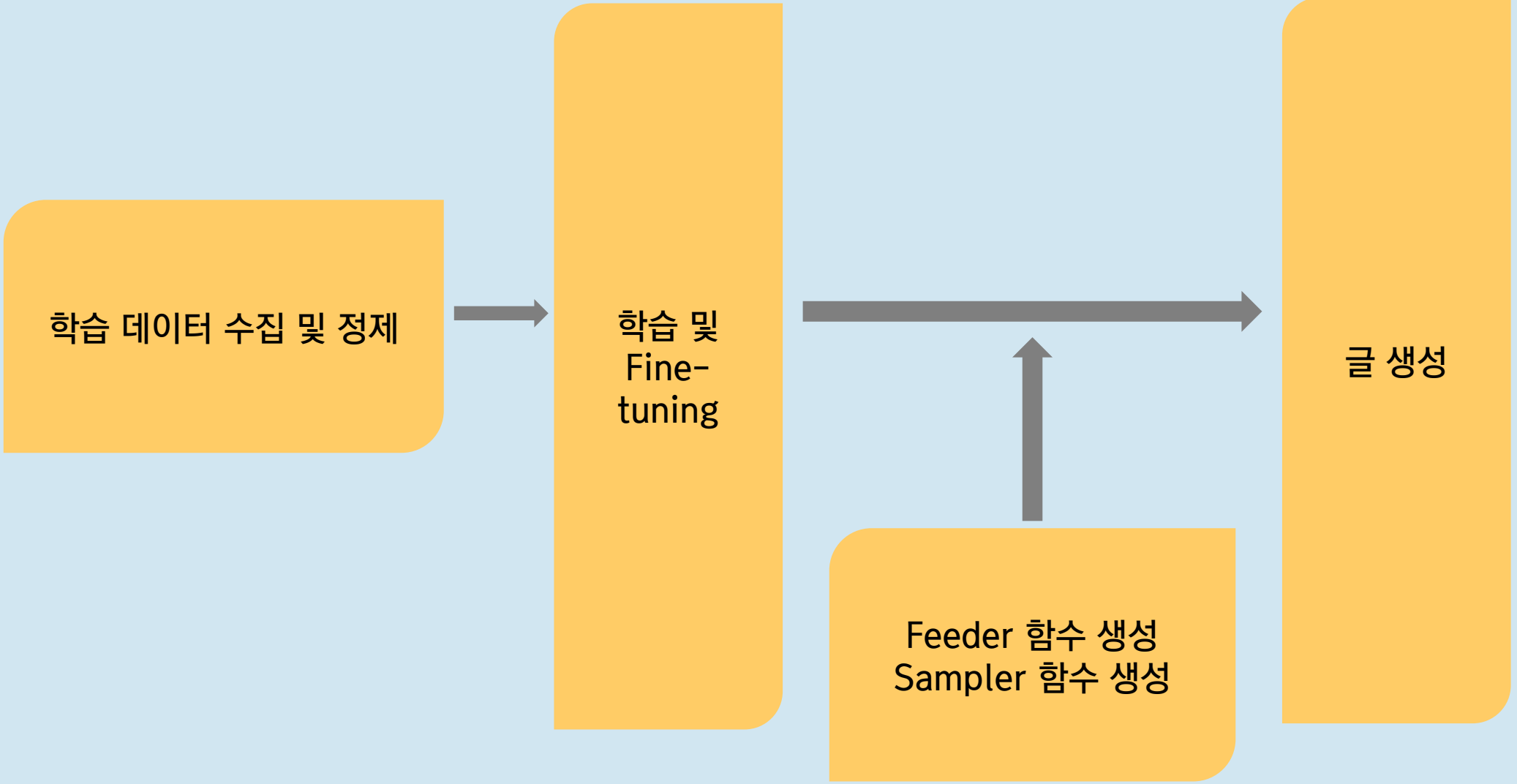
1. 준비과정

- feeder.py
- sampler.py
- engine_controller.py



무박2일 준비

2. Flow Chart



제3회 대학생 AI BOOKATHON 대회

2-1. 학습 데이터 수집

- 데이터 수집
  - 〈브런치〉 키워드 별 인기유저들의 글
  - 이전 참가자들이 공유한 〈브런치〉 여행 수집 데이터
  - 한겨레 사설 “삶의 창“
  - 이슬아 작가 수필
  - 아이유, 악동뮤지션, 김광석 노래의 가사
  - 〈광화문〉 연재소설
  - 〈재미수필〉 작가협회
  - 〈글틴-문학광장〉

2-2. 데이터 전처리

질 좋은 데이터를 학습 시키기 위해…

지친 꿈을 이끌고 계속 걷다 보니
첫발을 함께 떼어 달려왔던
친구들이 곁에 없다는 걸
어느 순간 깨닫게 되지
함께 이뤄갈까 성공해 다시 보자
지금쯤 현실의 처음을 겪고 있다면
그때완 다른 웃음 짓고 있으려나
그땐 함께 영원할 것만 같았지
어렸던 세상을 걷어내면
비탈지던 저 좁은 길가로 흩어져
화려하고 순수했던 꿈
너의 두 손에 넘쳐 흘렀던 그 한 움큼은
꼭 쥐고 살아가길
나로 시작될 거야 하늘을 날아보자
지금쯤 턱 막힌 장벽에 날개를 숨긴
그때 그 아이들과 우리의 꿈이
그땐 함께 영원할 것만 같았지
어렸던 세상을 걷어내면
비탈지던 저 좁은 길가로 흩어져
화려하고 순수했던 꿈
너의 두 손에 넘쳐 흘렀던 그 한 움큼과
그 두 손 모아 기도했던 시간이
이루어질 수 있도록
그땐 함께 이룰 거라고 믿었지
작은 손과 발로 서로를 잡고
뛰던 세상이 다였던 우리 어린 시절의
간절하고 행복했던 꿈

지친 꿈을 이끌고 계속 걷다 보니 첫발을 함께 떼어 달려왔던 친구들이 곁에 없다는 걸 어느 순간 깨닫게 되지 함께 이뤄갈까 성공해 다시 보자. 지금쯤 현실의 처음을 겪고 있다면 그때완 다른 웃음 짓고 있으려나. 그땐 함께 영원할 것만 같았지 어렸던 세상을 걷어내면 비탈지던 저 좁은 길가로 흩어져 화려하고 순수했던 꿈 너의 두 손에 넘쳐 흘렀던 그 한 움큼은 꼭 쥐고 살아가길 나로 시작될 거야 하늘을 날아보자. 지금쯤 턱 막힌 장벽에 날개를 숨긴 그때 그 아이들과 우리의 꿈이 그땐 함께 영원할 것만 같았지 어렸던 세상을 걷어내면 비탈지던 저 좁은 길가로 흩어져 화려하고 순수했던 꿈 너의 두 손에 넘쳐 흘렀던 그 한 움큼과 그 두 손 모아 기도했던 시간이 이루어질 수 있도록 그땐 함께 이룰 거라고 믿었지 작은 손과 발로 서로를 잡고 뛰던 세상이 다였던 우리 어린 시절의 간절하고 행복했던 꿈 너의 두 손에 넘쳐 흘렀던 그 한 움큼은 꼭 쥐고 살아가길 서투른 삶 걸음으로 상처를 입고 새로운 만남에 세상이 낯설어도 훗날 모두 이뤄 보일거야. 내가 알던 그때 그 아이들은

2-2. 데이터 전처리

광고성 글, 설명형 글, 너무 짧은 글👉 직접 보면서 정제

론-알프스, 부르고뉴, 알자르를 거쳐 이번에는 파리 근교 지역으로 떠나보려 한다. 우리가 알고 있는 파리는 보통 1구에서 20구까지다. 파리 인트라 뮤로스라고도 부른다. 그리고 파리를 포함해 20구 밖을 벗어난 파리 근교 지역을 통틀어 지칭하는 행정 구역이 일-드-프랑스로 이번에 떠나볼 곳은 일-드-프랑스에 속해 있는 곳이다. 일-드-프랑스에서 가장 유명한 곳은 아마도 베르사유. 그러나 만나볼 곳이 베르사유는 아니다. 베르사유만큼 알려지진 않았지만, 더 많은 이들이 알게 되었으면 하는 마음으로 선정한 곳으로 베르사유 못지않은 드넓은 정원 그리고 광활한 숲을 만나 볼 수 있다. 목적지는 퐁텐블로와 바르비종. 퐁텐블로와 바르비종은 차로 15분이면 오갈 수 있을 만큼 가깝다. 두 마을을 오가는 버스가 있지만 배차 간격이 길고, 운영 여부가 불투명할 때가 있어 나는 항상 택시를 이용했다. 파리에서 퐁텐블로까지는 기차로 이동할 수 있기 때문에

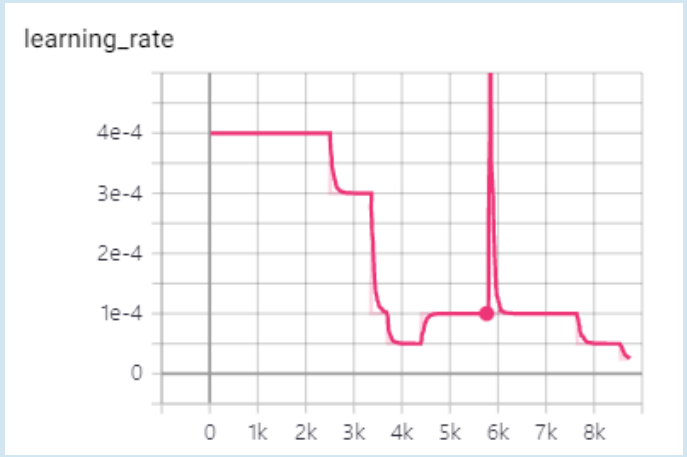
유행처럼 번지고 있는 'OO에서 한 달 살아보기' 까지는 못하더라도 파리까지 갔으니 아침에 눈 뜰 때부터 잠자리에 들 때까지 온전히 파리를 느끼고 싶어요. 반짝이는 불빛으로 수놓인 에펠탑의 아름다운 한 장면을 편안한 소파에 몸을 묻고 샴페인 한 잔으로 하루를 마무리하고 싶어요. 여기는 니까요! 1. Hotel Plaza Athenee 파리 호텔의 전설처럼 여겨지는 Hotel Plaza henee는 오뜨 쿠튀르(Haute Couture)의 중심부에 있는 클래식한 호텔입니다. '섹스 앤 시티'에서 캐리가 머물렀던 호텔로 더욱 유명해진 이 호텔은 돈 쓸 준비가 되어 있는 여행객을 호텔이라 해도 과언이 아닐 만큼 궁극의 럭셔리함을 휘감고 있답니다. 에펠탑 스위트룸에 묵면 바닥부터 천장까지 이어지는 온전한 에펠탑의 풍경을 감상할 수 있습니다. Star Rating ★★★★★ Price : €380 ~ €900 1박 Address : 25 Avenue Montaigne, 008 Paris, France 예약하러 가기 2. Shangri-La Hotel Paris 센 강 건너의 에펠 손에 닿을 듯 가까이 느낄 수 있는 최고급 호텔인 Shangri-La Hotel Paris는 19세기 로랜드 보나파르트(Roland Bonaparte) 왕자를 위해 지어진 저택을 보수하여 운영하고 있다. 프랑스의 고전적인 디테일을 살린 101개의 방과 스위트룸은 호화로운 별장 같은 느낌을 합니다. 객실에서 에펠탑을 볼 수 없다면 옥상 테라스에 올라가 보세요. 에펠탑의 낭만을 온 느낄 수 있습니다. Star Rating : ★★★★★ Price : €449 ~ €1,087 1박 dress : 10 Avenue d'Iéna, 75116 Paris, France 예약하러 가기 3. The Peninsula Paris 이름만 들어도 그 명성을 익히 알 수 있는 Peninsula Paris는 파노라마 같은 파리의 전경을 감상할 수 있음은 물론 파리에서 가장 큰 Spa를 이용할 수도 있습니다. 호텔에 자리한 고급 Bar와 레스토랑은 호텔 밖을 나가지 않고도 만족스러운 파리에서의 시간을 갖기에 충분합니다. 각 객실의 다양한 인테리어와 우아한 가구, 세련된 예술품은 물론이고 태블릿으로 작동되는 최고급 가전들까지 우아함과 첨단 기술의 디테일함이 당신의 눈을 어지럽게 만들 수도 있습니다. Star Rating : ★★★★★ Price : €490 ~ €971 1박 Address : 19 Avenue Kléber, 75116 Paris, France 예약하러 가기 4. Sofitel Paris Baltmore Tour Eiffel 개선문과 샹젤리제 명품거리를 쉽게 도보로 이동할 수 있는 거리에 위치한 인터내셔널 체인 소피텔 호텔은 완벽한 파리 여행을 실현할 수 있는 몇 안 되는 호텔이라고 할 수 있습니다. 1920년대 역사적인 건물을 사용하고 있는 이 호텔은 감동적인 에펠탑의 전망과 함께 고급 Mybed 침구가 만족스러운 호텔 Stay를 경험할 수 있게 해줄 것입니다. 이 호텔에 위치한 Carge Blanche 레스토랑은 미식가를 위해 언제든지 음식을 서빙할 준비를 하고 있답니다.

인도로 진짜 다시 떠나고 싶다. 코로나 때문에 요즘은 당최 C
거 인도여행 후 만들어 놓았던 이 영상으로 인도여행을 떠나본
영상이다. 인도사람들의 얼굴을 5초이상 쳐다볼 수 없다. 그
눈을 본적이 있는가?

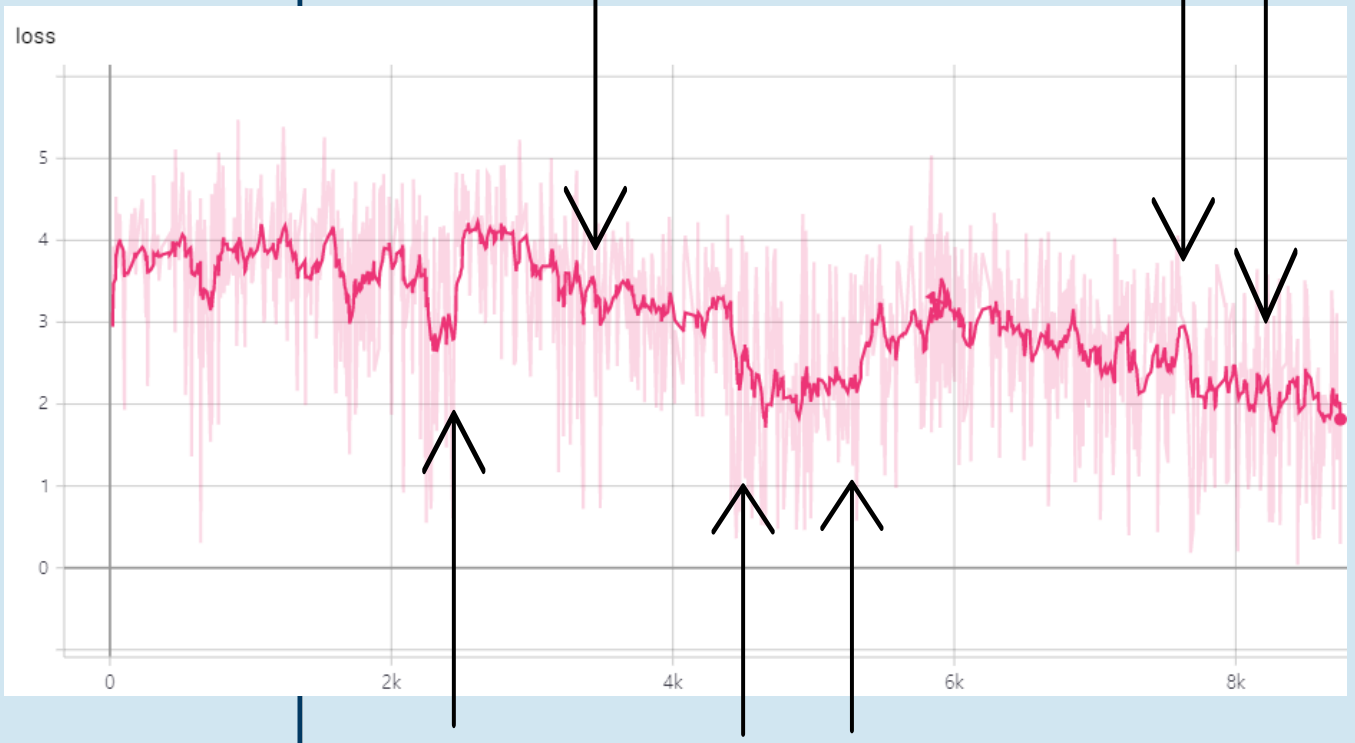
2-3. 학습 시작

모델 이름	학습 데이터	learning rate	input_token_length	batch size	accumulate gradients	결과
decal_test1 (input token length 확인용)	brunch(2222개 파일)	4e-4	750	1		
decal_test2	brunch(2222개 파일)	4e-4	600	1	32	5개 sample 전부 원문 통째로 내뱉음 / inferring 개소리함
decal_test3	brunch(2222개 파일)	2e-4	256	1	32	
decal_test4	brunch / book(2450부터) / book+광화소설+가사(2550부터) / 글틴, 재미수필, 컬럼(4400부터) / 컬럼, 광고형 글 등 데이터 삭제(5360부터)	4e-4 / 3e-4(2500부터) /1e-4(3350부터) / 5e-5(3700부터) / 1e-4 (4400부터) / 5e-5(7650부터)	256 / 450(2500부터)	1 /2(3350부터)	32 / 16 (3350부터) / 10 (3500부터)	Tensorboard 모양 이상, loss값이 줄지 않음 → input_token_length 450으로 수정

2-3. Fine-tuning



Learning Rate 조정



학습 데이터 추가/제거

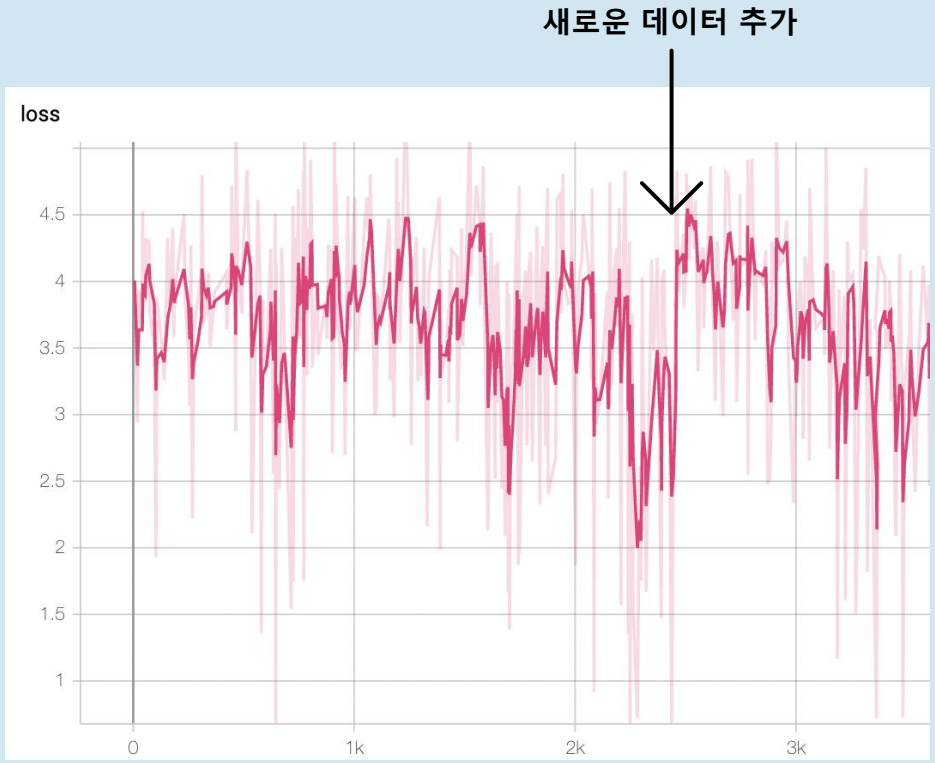
제3회 대학생 AI BOOKATHON 대회

2-3. Fine-tuning

최종 모델명: decal_test4

1) 0~2500 steps
   Learning_rate: 4e-4
   Input_token_length: 256
   Batch size: 1
   Accumulate gradients: 32

2) 2500~3350 steps
   데이터 추가: book+광화소설+가사
   Learning_rate: 3e-4
   Input_token_length: 450
   Batch size: 1
   Accumulate gradients: 32



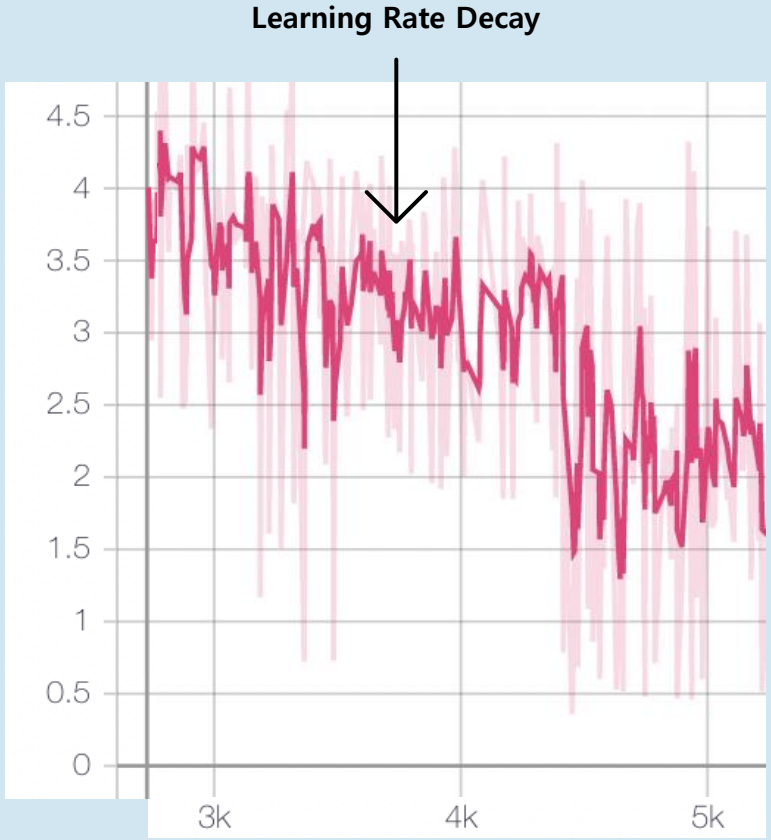
제3회 대학생 AI BOOKATHON 대회

2-3. Fine-tuning

최종 모델명: decal_test4

3) 3350~3700 steps
   Learning_rate: 1e-4
   Input_token_length: 450
   Batch size: 2
   Accumulate gradients: 10
   (3500 steps부터)

4) 3700~4400 steps
   Learning_rate: 5e-5
   Input_token_length: 450
   Batch size: 2
   Accumulate gradients: 10



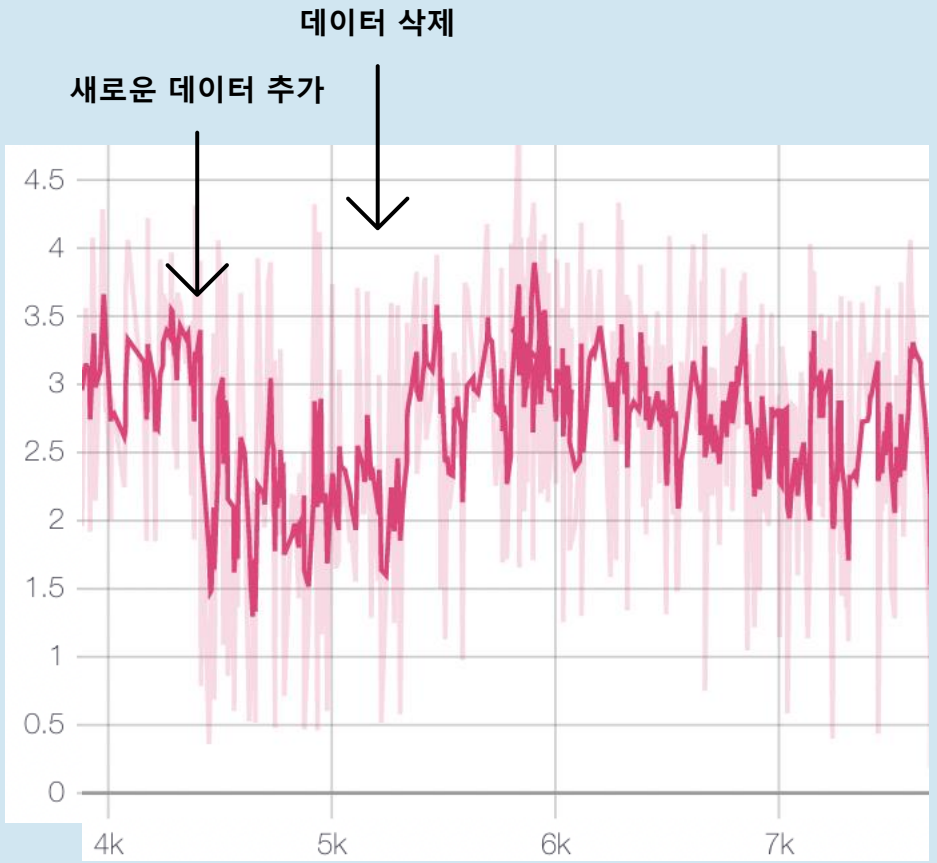
제3회 대학생 AI BOOKATHON 대회

2-3. Fine-tuning

최종 모델명: decal_test4

5) 4400~5360 steps
데이터 추가: 글틴, 재미수필, 컬럼
('함께', '가족', '환경' 키워드)
Learning_rate: 1e-4
Input_token_length: 450
Batch size: 2
Accumulate gradients: 10

6) 5360~7650 steps
데이터 삭제: 컬럼, 광고형 글 등
Learning_rate: 1e-4
Input_token_length: 450
Batch size: 2
Accumulate gradients: 10



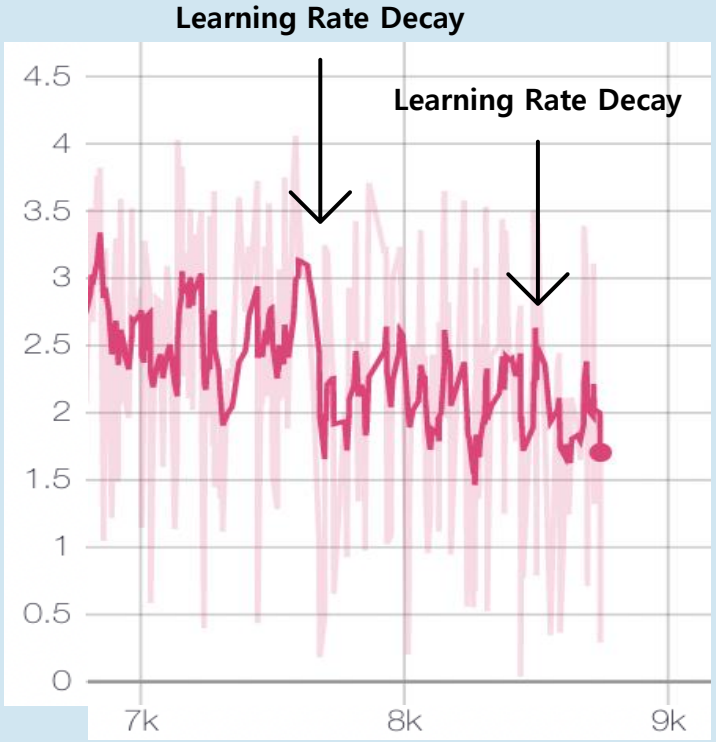
제3회 대학생 AI BOOKATHON 대회

2-3. Fine-tuning

최종 모델명: decal_test4

7) 7650~8550 steps
   Learning_rate: 5e-5
   Input_token_length: 450
   Batch size: 2
   Accumulate gradients: 10

8) 8550~8750 steps
   Learning_rate: 2.5e-5
   Input_token_length: 450
   Batch size: 2
   Accumulate gradients: 10



제3회 대학생 AI BOOKATHON 대회

2-3. 학습 과정 테스트

👉 약 4,000 Step : 무슨 말이지…?

(Infer_text = 아침밥을 먹었다.) 쿠바 사람이니까 알아보기로 했다. 다녀오는 내내 버스 지나간 후 맞춰 저녁식사하라는 조언을 얻었다. 그 뒤로는 내가 밥을 먹으러 밖으로 지나가기로 했다. 그렇게 설레던 시간이 끝나고 자주 돌아가야 했다. 사실 이 여행이 비교적 무서운게 여행지였다. 어쩔 수 없이할지, 새로펌을 해보고 싶기도 했다. 쿠바는 홀로개의 마켓을 볼 수 있었던 곳이라는 게 가장 좋았다. 후회로 종주 가도 한번 해보자, 많이 알다간병도 가네.

👉 약 7,600 Step : 첫 문장의 연결은 나아졌으나 아직 알 수 없는 단어와 문장

(Infer_text = 아침밥을 먹었다.) 긴장을 이기지 못하고 수프를. 무역 홍염 맞춰 검색창기를 하는 것처럼 너무 배가 고파져 꾸역꾸역 누웠다. 6시간의 비행 동안 땀 어찌 배되게 잘해서 도시민을게 편해 네 에펠탑 앞에 머물렀다. 숙소도 다르게 하고 배낭과 캐리어를 얻어혔다. 7월 말이 되자마자 공항에서 나를 기다리던 수많은 순례자들이 공항 안쪽으로 나와 가장 가까운 데서 공항까지 100km가 넘게 이동한 끝에 우리를 태운 비행기에 올랐다.

2-3. 학습 과정 테스트

👉 약 8,000 Step : 단어 문제는 해결됐으나 문장이 여전히 말이 안됨…

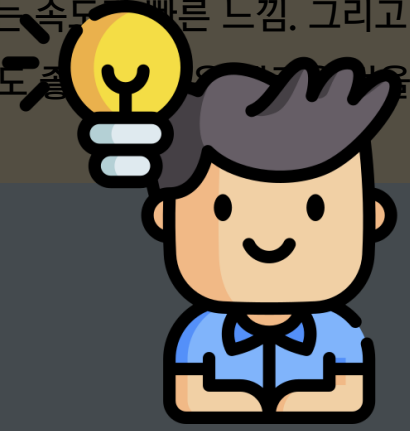
(Infer_text = 아침밥을 먹었다.) 열심히 걷고, 발을 담갔다. 그래도 여전히 숨은 체력이 남아있었다. 걷는 속도만큼 몸을 움직이는 느낌이다. 시작이 끝나는 느낌. 숨이 한편에 보이는 느낌. 빠르게 걸으면서 다리가 가는 느낌. 그리고 다음날부터는 꽤 괜찮아질까 싶다. 아- 걷는 속도가 빠른 느낌. 그리고 속도계에 의하면, 이런 느낌이구나... 그냥 걷는 것이다. 그냥 걷는 것도 좋지만 걸음을 느리게 하는 것도 좋다. 오늘은 새로운 것을 배우고 운동도 하는 날. 아- 걷는 것을 너무 좋아한다.

제3회 대학생 AI BOOKATHON 대회

2-3. 학습 과정 테스트

👉 약 8,000 Step : 단어 문제는 해결됐으나 문장이 여전히 말이 안됨…

(Infer_text = 아침밥을 먹었다.) 열심히 걷고, 발을 담갔다. 그래도 여전히 숨은 체력이 남아있었다. 걷는 속도만큼 몸을 움직이는 느낌이다. 시작이 끝나는 느낌. 숨이 한편에 보이는 느낌. 빠르게 걸으면서 다리가 가는 느낌. 그리고 다음날부터는 꽤 괜찮아질까 싶다. 아- 걷는 속도가 빠른 느낌. 그리고 속도계에 의하면, 이런 느낌이구나... 그냥 걷는 것이다. 그냥 걷는 것도 좋지만 걸음을 느리게 하는 것도 좋[illegible]을 배우고 운동도 하는 날. 아- 걷는 것을 너무 좋아한다.



Customizing 해보자!

더 나은 학습 성능을 위해, Feeder를

더 나은 문장 생성을 위해, Sampler를

2-4. Customization
-feeder.py에 _decal_topic_sent_sample_append() 생성

LDA 토픽 모델링을 적용하기 위한 함수

```python
# Customizing Part
def _decal_topic_sent_sample_append(self, text, length, text_file):
    sentences = split_text_into_sentences(text)
    sentences_tokenized = [self.tokenizer.morphs(sent) for sent in sentences if sent is not '']

    begin_index = random.randrange(0, len(sentences_tokenized)-2) if len(sentences_tokenized) > 2 else 0
    selected_sents = sentences_tokenized[begin_index:]

    folders = [folder for folder in  Path(self.data_dir_path).glob('*')]
    for folder in folders:
        try:
            topic_folder = [files for files in Path(folder).rglob('*.txt')]
            file_path = [file for file in Path(folder).glob(text_file)]
            if file_path[0] in topic_folder:
                topic_path = topic_folder
                topic_path.remove(file_path[0])
                break
        except:
            continue

    token_ids = self.tokenizer.convert_tokens_to_ids([token for sent in selected_sents for token in sent])
    while len(token_ids) < length:
        appending_text = random.choice(topic_path).read_text(encoding='utf-8')
        token_ids += self.tokenizer.convert_tokens_to_ids(self.tokenizer.morphs(appending_text))

    return token_ids[:length]
```

2-4. Customization
-feeder.py에 _decal_topic_sent_sample_append() 생성

```
# 전체 데이터 POS tagging
# NNG 일반명사, NNP 고유명사 -> nouns 이것만 남기기
def pos_tagging(text):
    pos_tagged=[]
    for t in text:
        tagged=mecab.pos(t)
        pos_tagged.append(tagged)

    return pos_tagged
```

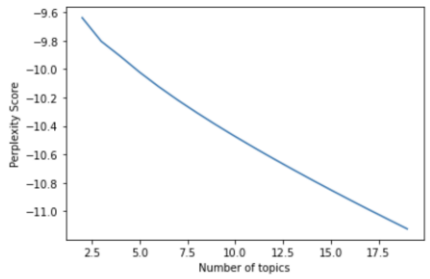
```
def get_nouns(pos_list):

    result=[]
    for i in pos_list:
        nouns = [n for n, t in i if t in ['NNG', 'NNP']]
        for w in nouns:
            if w not in stop_words:
                result.append(w)
    return result
```

```
get_nouns(pos_tagged)
```

```
['아침',
 '새',
 '노랫소리',
 '잠',
 '창문',
 '햇살',
 '냉기',
 '재채기',
 '눈',
 '창밖',
 '학교',
 '산책',
 '아버지',
 '양손',
 '효과',
 '약수',
 '아침',
 '어머니',
 '분주',
 '냉수',
 '아들',
 '게으름',
 '아침',
 '향기',
 '구수',
 '밥',
```

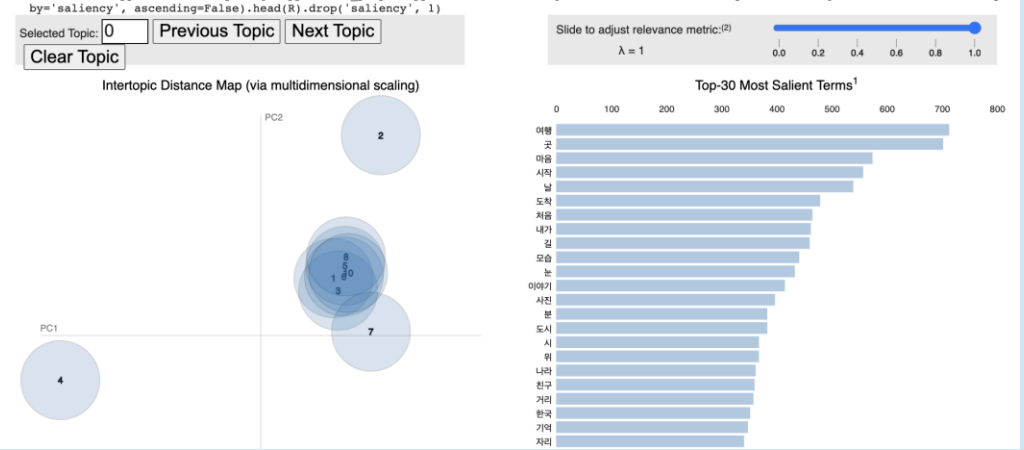
```
## 언어 모델 평가 방법
### 확률 모델이 결과를 얼마나 정확하게 예측하는지 판단. 낮을수록 정확하게 예측
import matplotlib.pyplot as plt
perplexity_values = []
for i in range(2,20):
    ldamodel=gensim.models.ldamodel.LdaModel(corpus, num_topics=i, id2word=dictionary
    perplexity_values.append(ldamodel.log_perplexity(corpus))
```

```
x=range(2,20)
plt.plot(x, perplexity_values)
plt.xlabel('Number of topics')
plt.ylabel('Perplexity Score')
plt.show()
```




```
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis

pyLDAvis.enable_notebook()
vis = gensimvis.prepare(ldamodel, corpus, dictionary)
vis
```

```
/usr/local/lib/python3.7/dist-packages/pyLDAvis/_prepare.py:247: FutureWarning: In a future version of pandas all arguments of DataFrame.drop ex
  by='saliency', ascending=False).head(R).drop('saliency', 1)
```



2-4. Customization
-feeder.py에 _decal_topic_sent_sample_append() 생성

```python
# 전체 데이터 POS tagging
# NNG 일반명사, NNP 고유명사 -> nouns 이것만 남기기
def pos_tagging(text):
    pos_tagged=[]
    for t in text:
        tagged=mecab.pos(t)
        pos_tagged.append(tagged)

    return pos_tagged
```

```python
def get_nouns(pos_list):

    result=[]
    for i in pos_list:
        nouns=[i for i, t in i if t in ['NNG', 'NNP']]
        for w in nouns:
            if w not in stop_words:
                result.append(w)
    return result
```

LDA 토픽 모델링으로 문서들의 토픽을 분류해 토픽별 디렉토리에 해당 문서들을 저장

↓

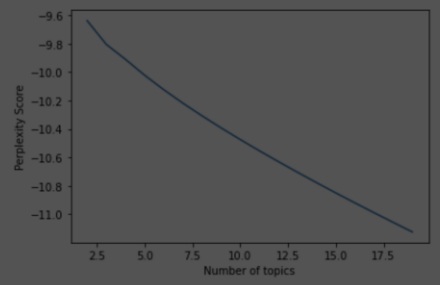
feeder.py에 생성한 함수를 통해, 주제가 유사한 문서로 append하기 위함!!

```python
get_nouns(pos_tagged)
```

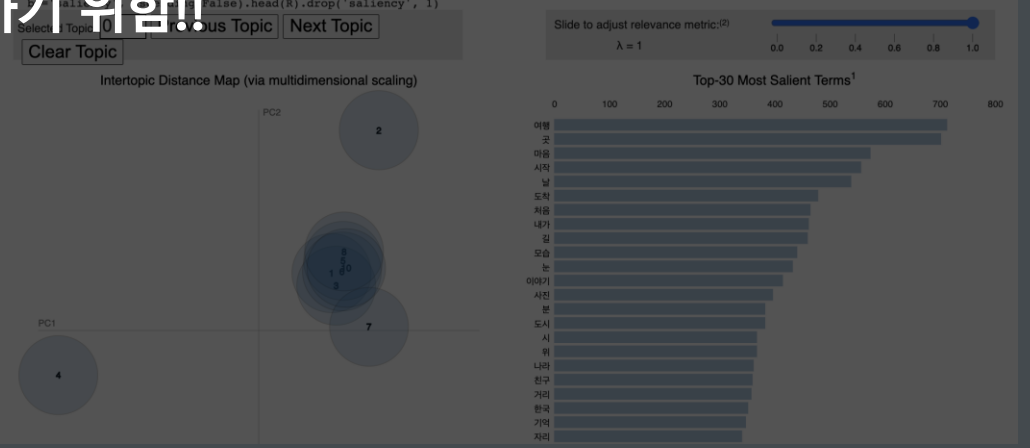
```
['아침',
 '새',
 '노랫소리',
 '잠',
 '창문',
 '햇살',
 '냉기',
 '재채기',
 '눈',
 '창밖',
 '학교',
 '산책',
 '아버지',
 '양손',
 '효과',
 '약수',
 '아침',
 '어머니',
 '분주',
 '냉수',
 '아들',
 '게으름',
 '아침',
 '향기',
 '구수',
 '밥',
```

```python
## 언어 모델 평가 방법
### 확률 모델이 결과를 얼마나 정확하게 예측하는지 판단. 낮을수록 정확하게 예측
import matplotlib.pyplot as plt
perplexity_values = []
for i in range(2,20):
    ldamodel=gensim.models.ldamodel.LdaModel(corpus, num_topics=i, id2word=dictionary
    perplexity_values.append(ldamodel.log_perplexity(corpus))
```

```python
x=range(2,20)
plt.plot(x, perplexity_values)
plt.xlabel('Number of topics')
plt.ylabel('Perplexity Score')
plt.show()
```




```python
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
```



2-4. Customization
-feeder.py에 _decal_topic_sent_sample_append() 생성

```python
# 전체 데이터 POS tagging
# NNG 일반명사, NNP 고유명사 -> nouns 이것만 남기기
def pos_tagging(text):
    pos_tagged=[]
    for t in text:
        tagged=mecab.pos(t)
        pos_tagged.append(tagged)

    return pos_tagged
```

```python
def get_nouns(pos_list):

    result=[]
    for i in pos_list:
        nouns=[i for i,t in i if t in ['NNG', 'NNP']]
        for w in nouns:
            if w not in stop_words:
                result.append(w)
    return result
```

LDA 토픽 모델링으로 문서들의 토픽을 분류해 토픽별 디렉토리에 해당 문서들을 저장

↓

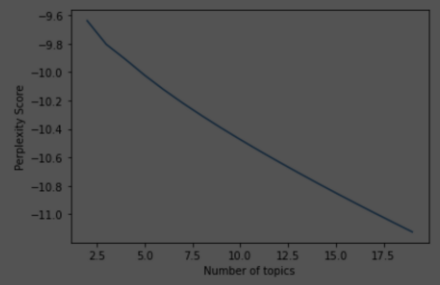
feeder.py에 생성한 함수를 통해, 주제가 유사한 문서로 append하기 위함!!

```python
get_nouns(pos_tagged)
```

```
['아침',
 '새',
 '노랫소리',
 '잠',
 '창문',
 '햇살',
 '냉기',
 '재채기',
 '눈',
 '창밖',
 '학교',
 '산책',
 '아버지',
 '양손',
 '효과',
 '약수',
 '아침',
 '어머니',
 '분주',
 '냉수',
 '아들',
 '게으름',
 '아침',
 '향기',
 '구수',
 '밥',
```

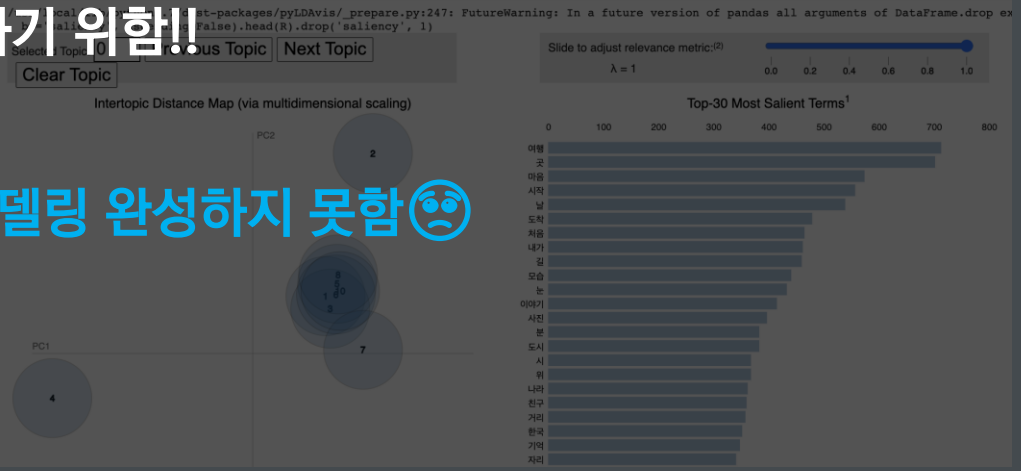
```python
## 언어 모델 평가 방법
### 확률 모델이 결과를 얼마나 정확하게 예측하는지 판단. 낮을수록 정확하게 예측
import matplotlib.pyplot as plt
perplexity_values = []
for i in range(2,20):
    ldamodel=gensim.models.ldamodel.LdaModel(corpus, num_topics=i, id2word=dictionary
    perplexity_values.append(ldamodel.log_perplexity(corpus))
```

```python
x=range(2,20)
plt.plot(x, perplexity_values)
plt.xlabel('Number of topics')
plt.ylabel('Perplexity Score')
plt.show()
```



시간 부족으로 인해 토픽모델링 완성하지 못함😥

```python
import pyLDAvis
import pyLDAvis.gensim_models as gensimvis
```



2-4. Customization
-sampler.py에 top_p_and_k() 생성

```python
class NsgDecodingSampler(object):
    def __init__(self, config):
        self.method = config.sampling_method
        self.top_k = config.top_k
        self.top_p = config.top_p
        self.temperature = config.temperature

    def sample(self, logits):
        if self.method == 'top_p':
            return top_p_logits(logits, self.top_p, self.temperature)
        elif self.method == 'top_k':
            return top_k_logits(logits, self.top_k)
        elif self.method == 'top_p_and_k':
            return top_p_and_k_logits(logits, self.top_p, self.top_k, self.temperature)
        else:
            raise NotImplemented
```

top-p & top-k
각각의 단점을 보완하기 위한 전략

```python
def top_p_and_k_logits(logits, p, k, temperature):
    if k == 0:
        # no truncation
        return logits

    def _top_k(log):
        values, _ = tf.nn.top_k(log, k=k)
        min_values = values[:, -1, tf.newaxis]

        return tf.where(
            log < min_values,
            tf.ones_like(log, dtype=logits.dtype) * -1e10,
            log,
        )

    with tf.variable_scope('top_p_logits'):
        logits_sort = tf.sort(logits, direction='DESCENDING')


        probs_sort = tf.nn.softmax(logits_sort/temperature)
        probs_sums = tf.cumsum(probs_sort, axis=1, exclusive=True)
        logits_masked = tf.where(probs_sums < p, logits_sort, tf.ones_like(logits_sort)*1000)
        # 누적확률까지 맞으면 logits_sort 유지, 아니면 1000으로 변경.
        # 그다음 이걸 reduce_min -> 최소 확률 찾음
        min_logits = tf.reduce_min(logits_masked, axis=1, keepdims=True) # [batchsize, 1]

        # min_logit보다 크거나 같아야 남겨줌.
        top_p_result = tf.where(
            logits < min_logits,
            tf.ones_like(logits, dtype=logits.dtype) * -1e10,
            logits,
        )

        # min_logit보다 크거나 같아야 남겨줌.
        top_p_result = tf.where(
            logits < min_logits,
            tf.ones_like(logits, dtype=logits.dtype) * -1e10,
            logits,
        )

    #갯수 카운트용
    for_count = tf.where(
        top_p_result==tf.constant([-1e10], dtype=top_p_result.dtype),
        tf.zeros_like(top_p_result, dtype=top_p_result.dtype),
        top_p_result
    )

    cnt = tf.count_nonzero(for_count)

    return tf.cond(
        tf.less_equal(cnt,k),
        lambda: top_p_result,
        lambda: tf.cond(
            tf.equal(k, 0),
            lambda: logits,
            lambda: _top_k(top_p_result),
            )
    )
```

제3회 대학생 AI BOOKATHON 대회

2-4. Customization
-sampler.py에 top_p_and_k() 생성

```
    # min_logit보다 크거나 같아야 남겨줌.
    top_p_result = tf.where(
        logits < min_logits,
        tf.ones_like(logits, dtype=logits.dtype) * -1e10,
        logits,
    )

#갯수 카운트용
for_count = tf.where(
    top_p_result==tf.constant([-1e10], dtype=top_p_result.dtype),
    tf.zeros_like(top_p_result, dtype=top_p_result.dtype),
    top_p_result
)

cnt = tf.count_nonzero(for_count)

return tf.cond(
    tf.less_equal(cnt,k),
    lambda: top_p_result,
    lambda: tf.cond(
        tf.equal(k, 0),
        lambda: logits,
        lambda: _top_k(top_p_result),
        )
)
```

1. top_p inferring 결과 변수 생성
2. top_p에서 선택된 개수 확인
3. 그 개수가 top_k의 k개보다 적거나 같으면 top_p 결과 리턴
4. 그 개수가 k보다 크면, 결과값에 top_k inferring 알고리즘 적용

제3회 대학생 AI BOOKATHON 대회

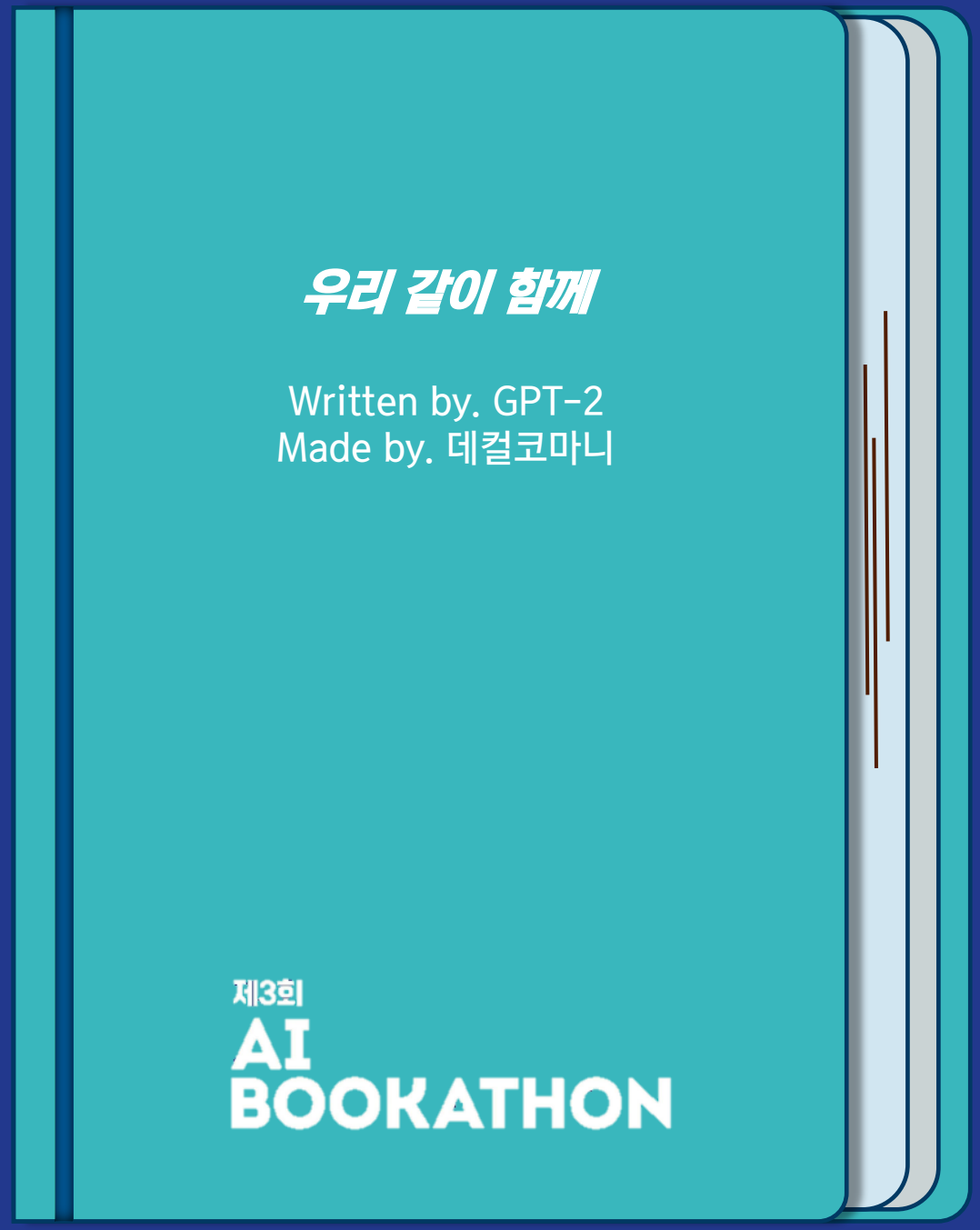
2-4. 학습 과정 테스트

Top_p(0.99 ~ 0.995),
Top_k(5 ~ 20),
Temperature(0.5~1) 변수
모두 조정해가며
다양한 inferring 알고리즘을 사용해
데이터 생성

2. 날씨가 점점 더워진다. 날씨가 점점 더워진다. 하지만, 내가 어디에 있는 지도를 보고, 내가 어떤 곳에 있는 지도를 찾아본다. 오늘은 그 와중에 내가 발견한 나의 모습을 조금 더 가까이 마주할 날이다. 그때 내가 있는 곳은 바로 '아는' 곳이었고, 아는 친구를 통해서 '아는 친구'가 되었다. 그렇게 나는 매일 아침 6시 30분부터 늦은 시간까지 집 앞의 공원이나 길거리에 나와 산책을 하고 있는 것이다. 하루에도 수만 명이 이 공원을 보러 와공원 주변의 산책로에는 사람들로 북적이고 그들을 따라 가면 좋은 점을 쉽게 찾아볼 수 있다. 공원 날씨가 점점 더워진다. 이제 곧 사막이 시작된다.
3. 날씨가 점점 더워진다. 날씨가 점점 더워진다. 오늘도 역시나. 오늘도. 그래도.....,. 날씨가 점점 더워진다. 내일 아침 일찍 공항에 도착했다. 아직 어둠이 가시지 않은 날씨 때문인지, 햇살에 비친 출국심사 모습이 여전히 낯설게 느껴지고 있다. 하지만 맑은 날씨 영향인지, 입국심사는 아침 일찍부터 시작되었음을 알려주고 있었다. 출국 줄에는 해외여행 중이던 가족들이 설레는 눈빛으로 설레는 얼굴을 양 목에 건 채 출국장으로 향하고 있다. 출국 심사대를 지나쳐야 하는 시간이 너무나도 불안정해서 나는 이제 좀 익숙해져서 또다시 새로운 모습으로 입국장으로 걸어 들어갔다.
4. 날씨가 점점 더워진다. 날씨가 점점 더워진다. 오늘은 따뜻한 커피 한잔을 마시러 카페에 갈 예정이다. 날씨가 매우 청명하고 공기도 맑다. 밖에 나간다면 햇빛에 비쳐지는 거리를 거닐며 시간을 보내기에 좋다. 카페에서 나와 비슷하게 생긴 테이블 마운틴 앞에 앉아 잠시 쉬어가기로 했다. 오늘 하루는 이 거리를 걷는 데 크게 문제 되지 않을 것 같다. 조금 일찍 나왔더라면 그 거리를 잘 보낼 수 있을 거 같다. 오늘은 이 근처에 앉아 커피 한잔을 마시며 하루를 마감하는 시간이다. 조금 걸어가다 보니 다시 카페가 나왔다.
5. 날씨가 점점 더워진다. 날씨가 점점 더워진다. 오늘은 어제와 다른 하루를 맞이하려고 한다. 오늘도 새로운 곳으로의 방문을 갈망한다. 새로운 만남의 기분이다. 새로운 곳을 기대하며 설레는 아침이다. 새로운 만남의 경험을 만들어주신 이민자 아주머니에게 감사를 느낀다. 오랜만에 연락이 왔다. 그녀는 날씨를 물었다. "왜 이렇게 춥던지. 추위를 안 타는 게 제일 이쁘기도 하고. 너무 덥고 춥다.

Generated: 1. 날씨가 점점 더워진다. 날씨가 점점 더워진다.. 긴장을 풀어준 다음, 따뜻한 햇빛을 잠시 쐬었다. 얼른 상쾌한 바다 냄새에 취해 한잔 해가 저문 뒤,지도 않았던 몸도 소복히 짐을 차리고 기분을 상하도록 만든다. 파도소리가 귀를 간지럽혔다. 이내 요시코 해변에 다시 와야겠다는 생각이 든다. 슬로베니아의 제 1 맥주집에서 저녁을 먹고 근처 마트에서 맥주를 사 왔다. 무엇보다도 현지 물가에 대도시인산책의 맑은 날씨가 한몫했다. 한 접시에 56유로 (약 6만6천원) 꼴이니, 우리나라로 치면 소복이 받 현지 화폐에 가깝지만, 현지 물가에 비하면 꽤 합리적인 가격일 날씨가 점점 더워진다.
2. 날씨가 점점 더워진다. 날씨가 점점 더워진다. 이런 날은 다른 나라로 여행을 떠나도 될 정도의 실컷 예뻐서 당신을 미소 짓게 하는 것 같다. 스페인의 말라가서 000점을 걷는다. 다른 나라들과 비교해서 분위기가 괜찮은 편인데, 스페인은 늦은 시간인 데다가 스페인에 가기 전에 해가 떠있는 풍경이 좋다. 저녁이 되어가고 있다. 화창한 날씨 속에 우리를 비롯해 많은 관광객들이 즐비하고, 쇼핑몰에서 조금 떨어진 곳에 위치해있는 쇼핑몰을 구경한다. 일몰의 아름다움과 노을빛이 더해져 더 낭만적이고 아름다운 분위기를 띄고 있다. 스페인의 12월 날씨는 무척이나 따뜻하다. 바람도 불
3. 날씨가 점점 더워진다. 날씨가 점점 더워진다. 이런 날씨에도 불구하고 이 노래가 위로가 되는 것 같다. 춥고 외로웠던 맘을 녹여주는 것 같다. 그래서 오늘도 이 노래를 들으러 집으로 간다. 그래서 우리 딸은 이 노래를 들으면서 눈물을 흘리는 걸지도 모른다. 울기도 힘들고 외로웠던 마음을 어렴풋이 느끼게 해 준다. 우리 딸은 이 노래를 듣기 시작한 지 8년 만에 가족이 있었던 프랑스로 이민을 떠났다. 그리고 프랑스에서 우리 셋은 더 프랑스로 이민을 떠났고 아이들은 프랑스에서 지내고 있었다. 그 사이에도 프랑스는 늘 바빴고 지금도 여전히, 가족 여행도 다녔던 우리 셋은 프랑스를

3. 작품 소개



3. 작품 소개 - 줄거리

기	승	전	결
혼자 해외로 여행을 감, 평소 그 나라의 문화와 역사를 체험하고 존중해야 함을 알고 있었음.	여행지에서 새로운 사람을 만나 새로운 문화를 경험하며 초반에 당황함	다름에 대한 이해와 존중, 함께 더불어 사는 세상임을 몸소 체험함	서로 다른 문화, 인종, 성별을 존중하고, 모두 **함께** 하나가 되는 세상을 기대함

“주제”
혼자보다는 함께

제3회 대학생 AI BOOKATHON 대회

3. 작품 소개 - 퇴고

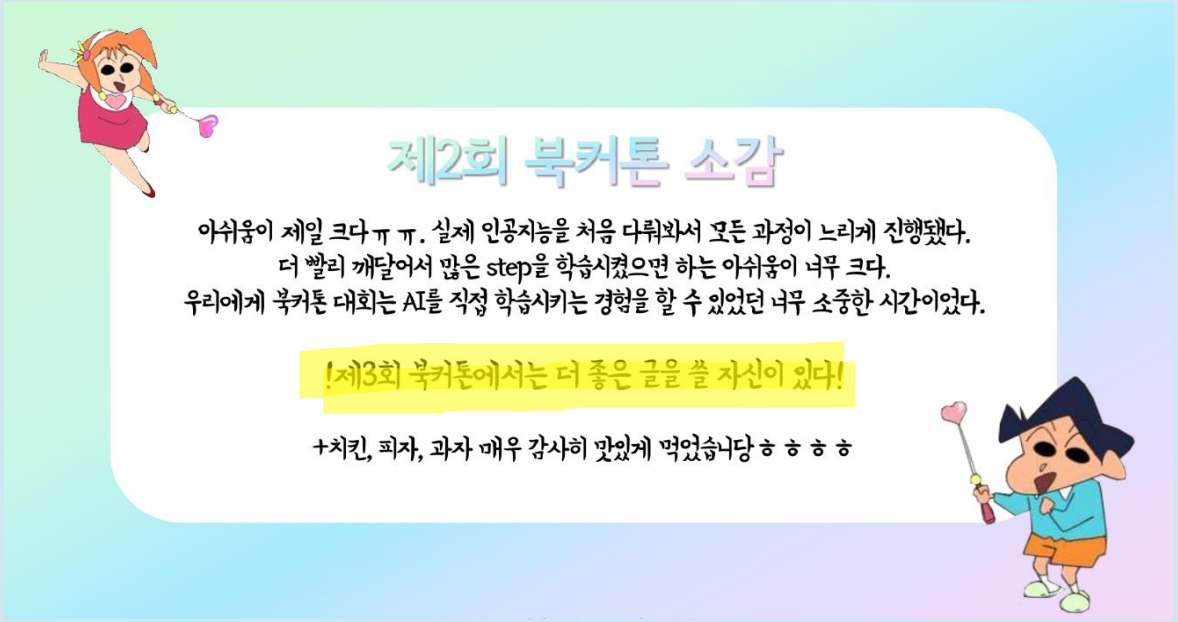
그래서 나는 유럽으로 떠났다. 처음에는 내가. 그 후로도 유럽 여행에 대한 로망을 놓지 않았다. 나는 유럽에 대한 로망을 하나씩 적어갔다. 그리고 그 로망을 실현하는 데 필요한 정보를 하나씩 적어갔다. 유럽여행을 계획 중이었던 사람들은 꼭 필요한 정보들을 정리했다. 유럽으로 가겠다는 용기와 계획을 하고 있었지만 ~~어디로 가든지 일정이 맞지 않아 포기하고 싶었던 적이 더 많았던 것 같다. 그 이후로 여행에 대해 조금씩 눈이 가고 가고 싶었다.~~ 그런데 유럽여행은 계획을 세우는 것만큼이나 많은 시간을 할애해야 하는 것 같다. 그 와중에도 내가 가 볼 수 있었던 유럽은 어디일까?

그래서 나는 유럽으로 떠났다. 처음에는 내가. 그 후로도 유럽 여행에 대한 로망을 놓지 않았다. 나는 유럽에 대한 로망을 하나씩 적어갔다. 그리고 그 로망을 실현하는 데 필요한 정보를 하나씩 적어갔다. 유럽여행을 계획 중이었던 사람들은 꼭 필요한 정보들을 정리했다. 유럽으로 가겠다는 용기와 계획을 하고 있었지만 유럽여행은 계획을 세우는 것만큼이나 많은 시간을 할애해야 하는 것 같다. 그 와중에도 내가 가볼 수 있었던 유럽은 어디일까?

3. 작품 소개 – Best Sentence

유럽여행을 갔을 때, 한 나라에만 흥미로운 게 아니었다. 여러 나라를 방문했을 때 그 나라의 문화와 역사를 체험하는 것은 다른 나라에서는 절대 시도하지 못하는 경험이다.

4. 참가 소감



제2회 북커톤 참가때보다 훨씬 많은 코드를 수정하고 많은 step을 학습시킬 수 있었다.
함수를 추가하고, 전처리를 더 많이 하고자 노력했다.
더 오래 학습하지 못하고 조급하게 변화를 준 것이 아쉽다.

좋은 기회 주셔서 감사합니다!

감사합니다

제3회
AI
BOOKATHON