

RNN을 이용한 챗봇

KB-KAIST AI 집중교육과정
딥러닝 실습 B

실습자료

실습조교: 정승준, 변준영
(KAIST Computational Intelligence Lab.)

목차

1. 실습 환경 구성 및 설명
2. Chat-bot 실제 동작 해보기
3. 데이터셋 바꾸어보기
4. Context Vector 바꾸어보기
5. In-class question #1 & #2
6. 빅데이터(긴 대화)에 적용하기

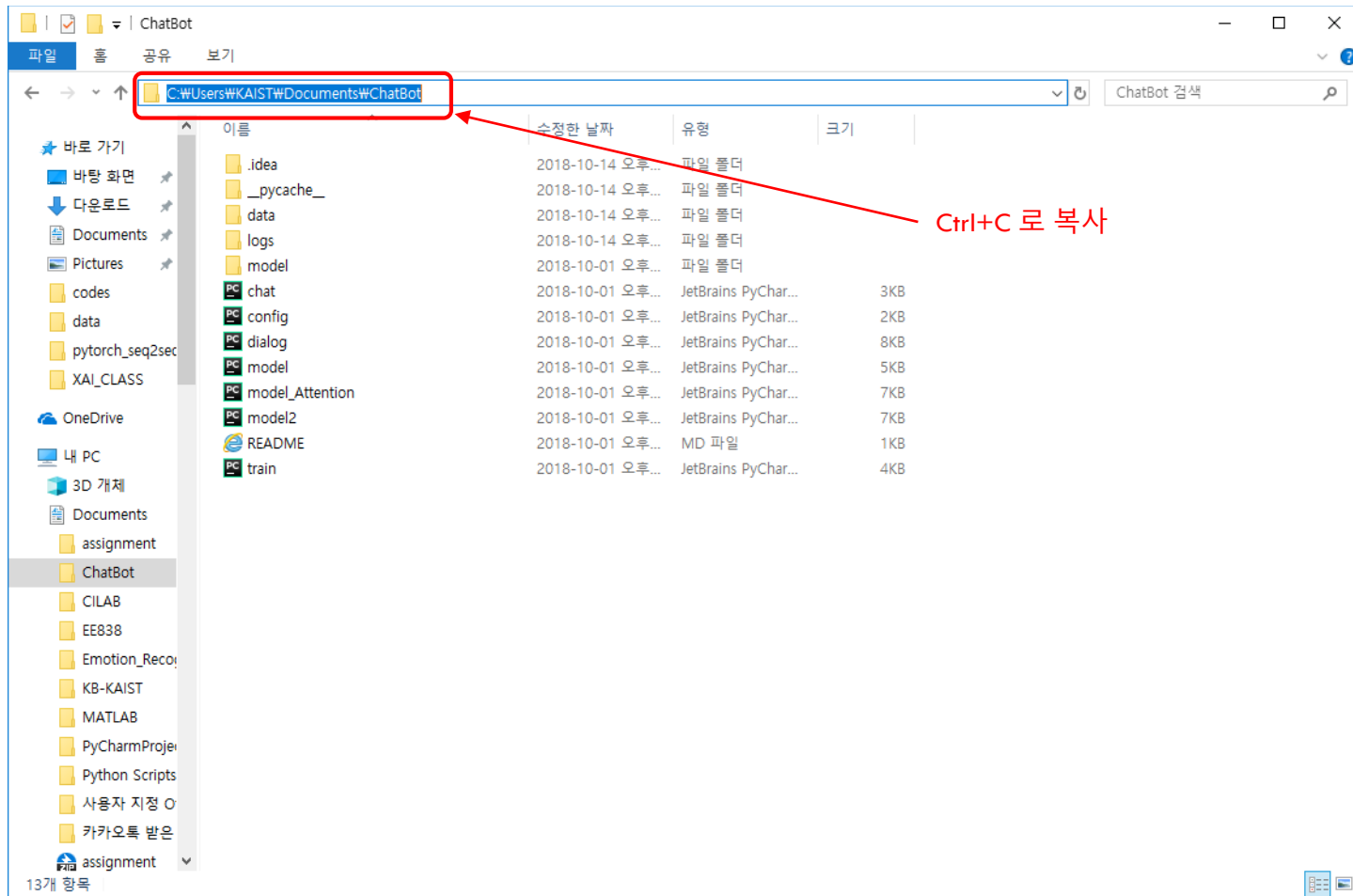
Anaconda Prompt 실행 및 동작 위치
설정

실습 환경 구성 및 설명

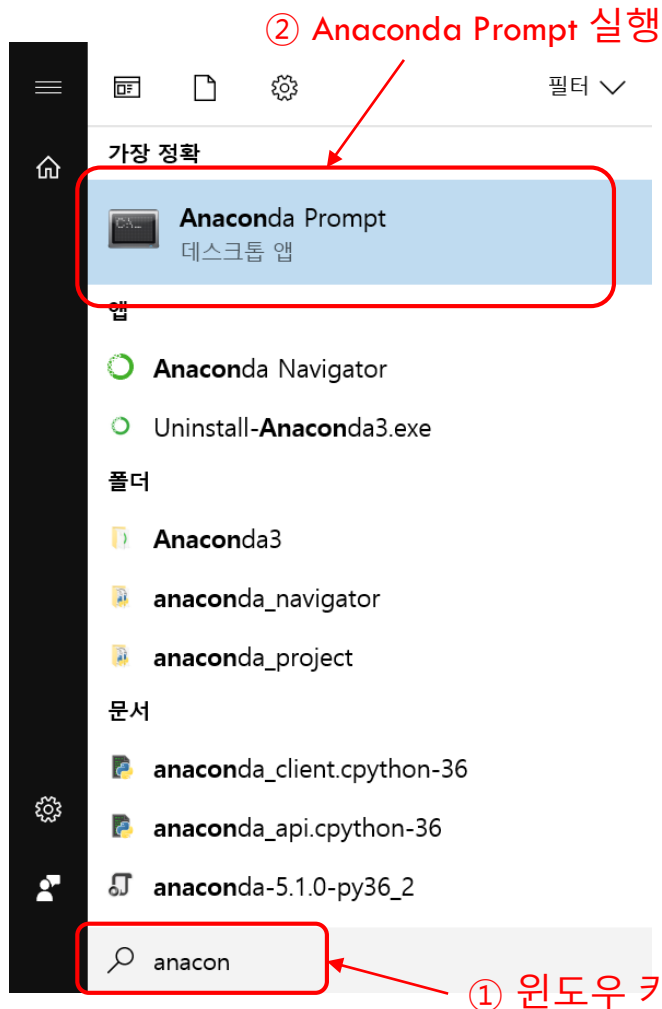
실습 환경 구성

1. 네트워크 설계 및 실습 실행을 위한 코드
 - Neural Network의 구조 및 Test를 위한 전반적인 내용을 설계하는 부분
2. 해당 코드의 실행을 위한 Anaconda prompt
 - 1에서 작성한 코드를 실행

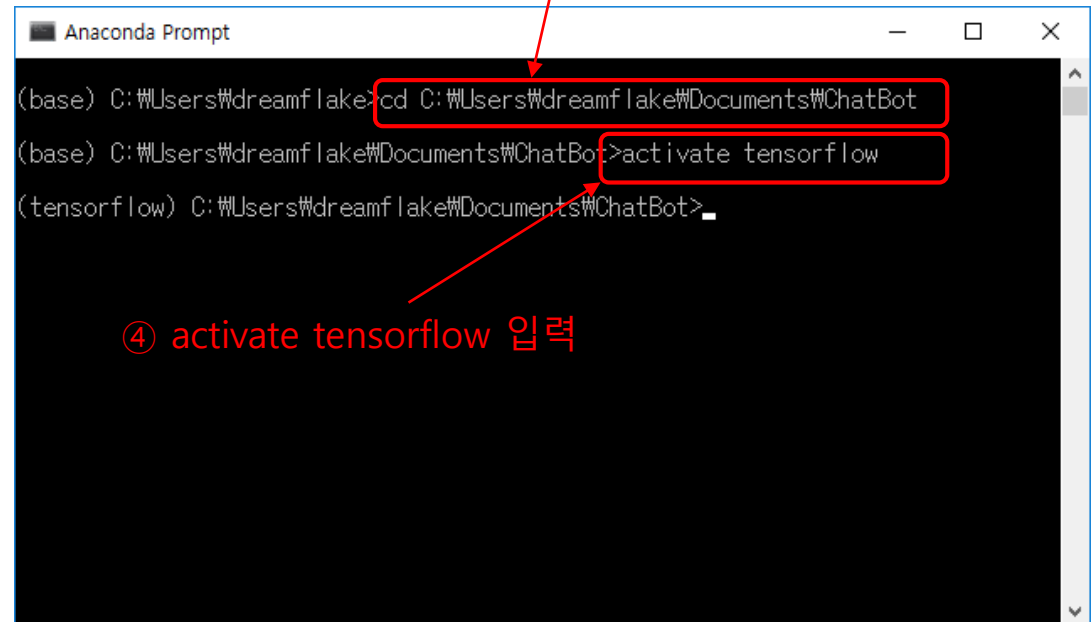
해당 실습 파일 위치 확인



Anaconda Prompt 동작 위치 설정



③ cd 입력 후 마우스 우클릭 (복사 된 주소 붙여넣기)

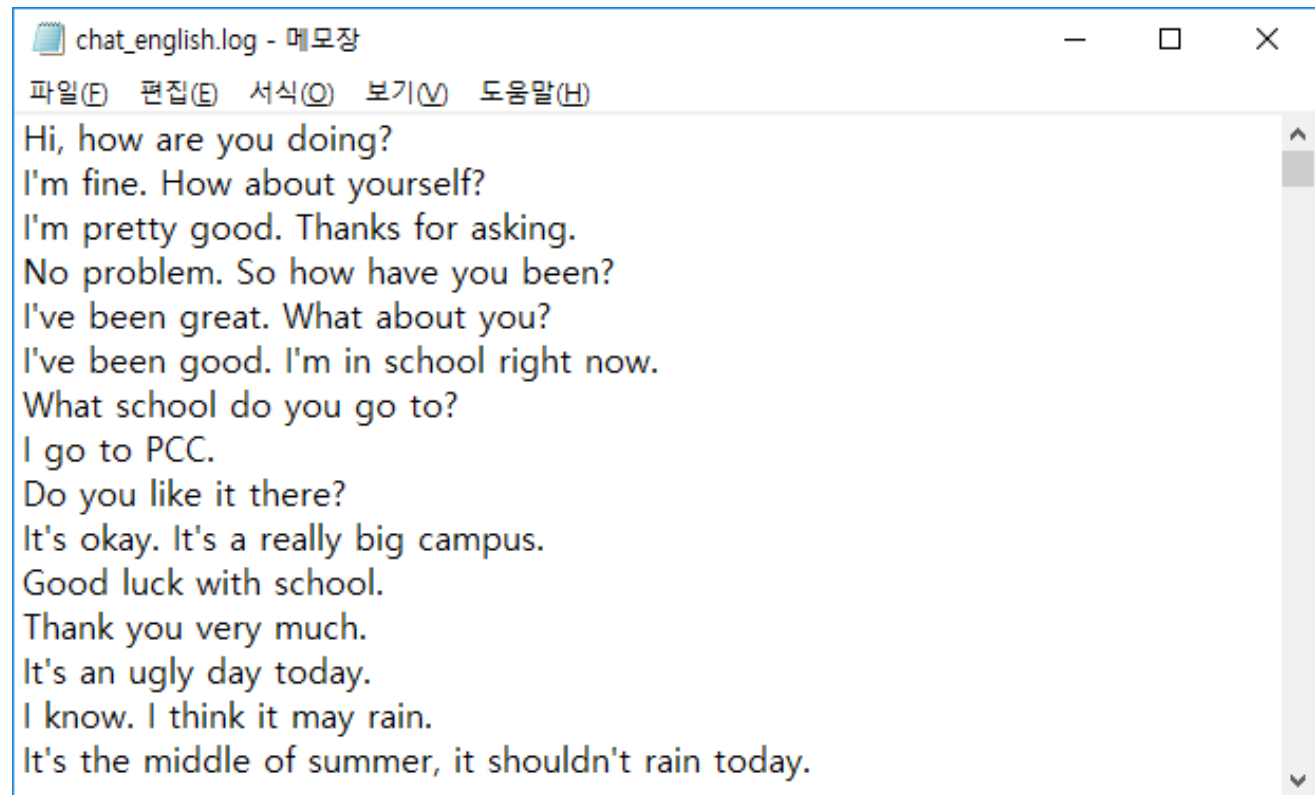


④ activate tensorflow 입력

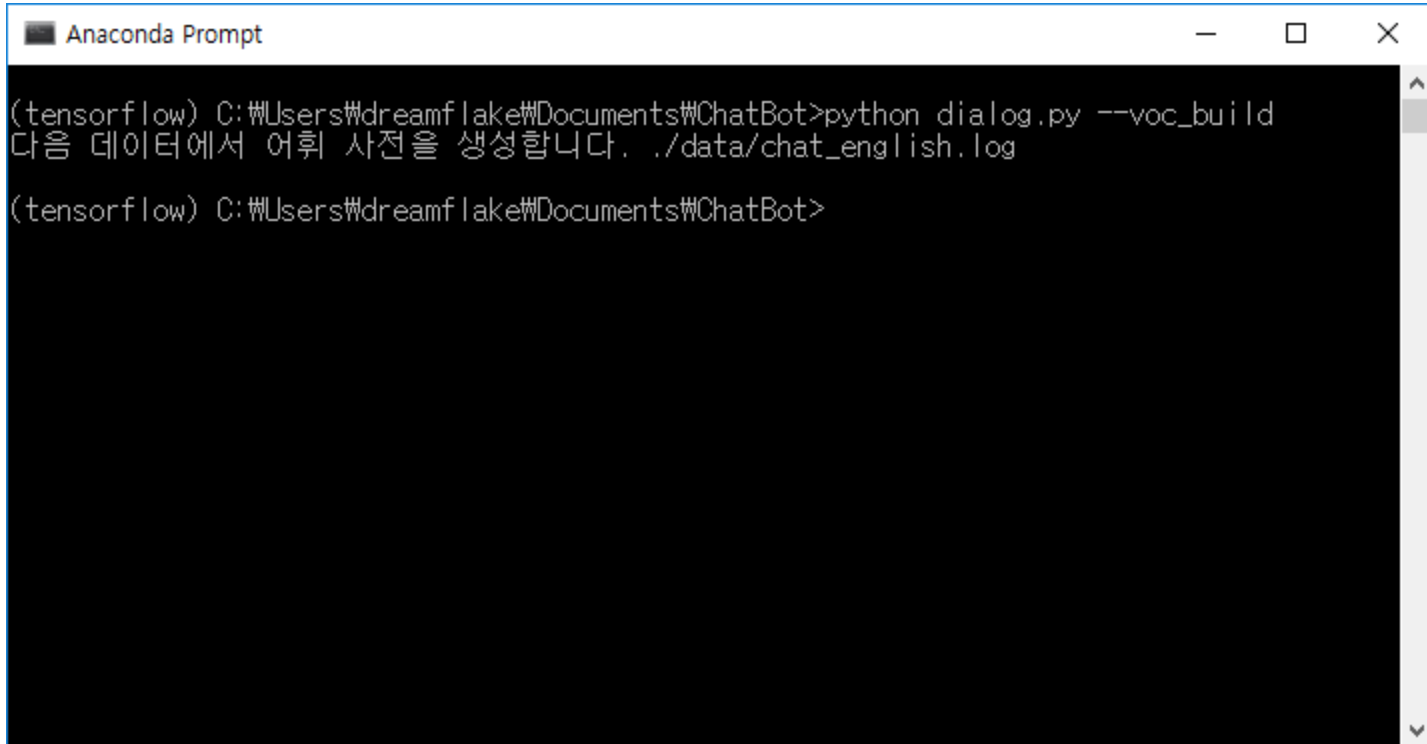


Chat-bot 실제 동작 해보기

영어 대화 데이터셋



어휘 파일 생성하기



```
Anaconda Prompt
(tensorflow) C:\Users\dreamflake\Documents\ChatBot>python dialog.py --voc_build
다음 데이터에서 어휘 사전을 생성합니다. ./data/chat_english.log
(tensorflow) C:\Users\dreamflake\Documents\ChatBot>
```

생성된 어휘 파일 확인하기

이 파일을 열 때 사용할 앱을 선택하세요.



워드패드

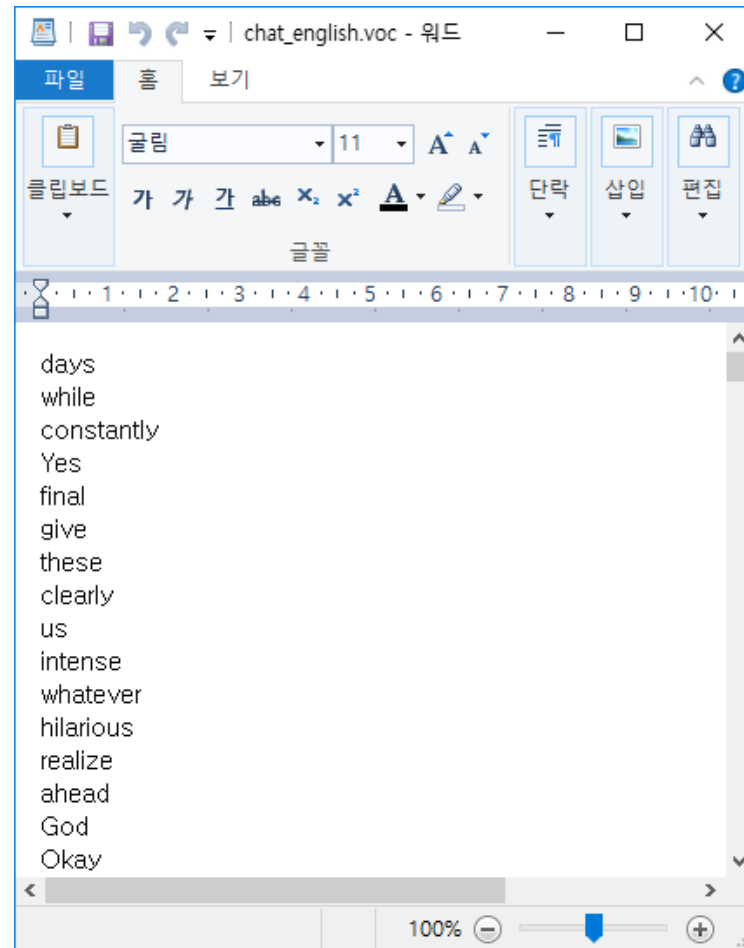


Microsoft Store에서 앱 찾기

[추가 앱 ↓](#)

☒ 항상 이 앱을 사용하여 .voc 파일 열기

확인



챗봇 학습하기

`python train.py --train`

학습을 위해 프롬프트에서 위 명령어를 입력해보세요.

```
선택 Anaconda Prompt - python train.py --train
다음 데이터에서 어휘 사전을 생성합니다. ./data/chat_english.log
(tensorflow) C:\Users\dreamflake\Documents\ChatBot>python train.py --train
2018-10-16 13:58:06.186771: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2018-10-16 13:58:06.519355: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:1356] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:01:00.0
totalMemory: 11.00GiB freeMemory: 9.10GiB
2018-10-16 13:58:06.528737: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:1435] Adding visible gpu devices: 0
2018-10-16 13:58:07.091864: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:923] Device interconnect StreamExecutor with strength 1 edge matrix:
2018-10-16 13:58:07.097654: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:929] 0
2018-10-16 13:58:07.101412: I T:\src\github\tensorflow\tensorflow\core\common_runtime\gpu\gpu_device.cc:942] 0:  N
새로운 모델을 생성하는 중 입니다.
Step: 000001 cost = 6.546778
Step: 000002 cost = 6.544735
```

챗봇 학습하기

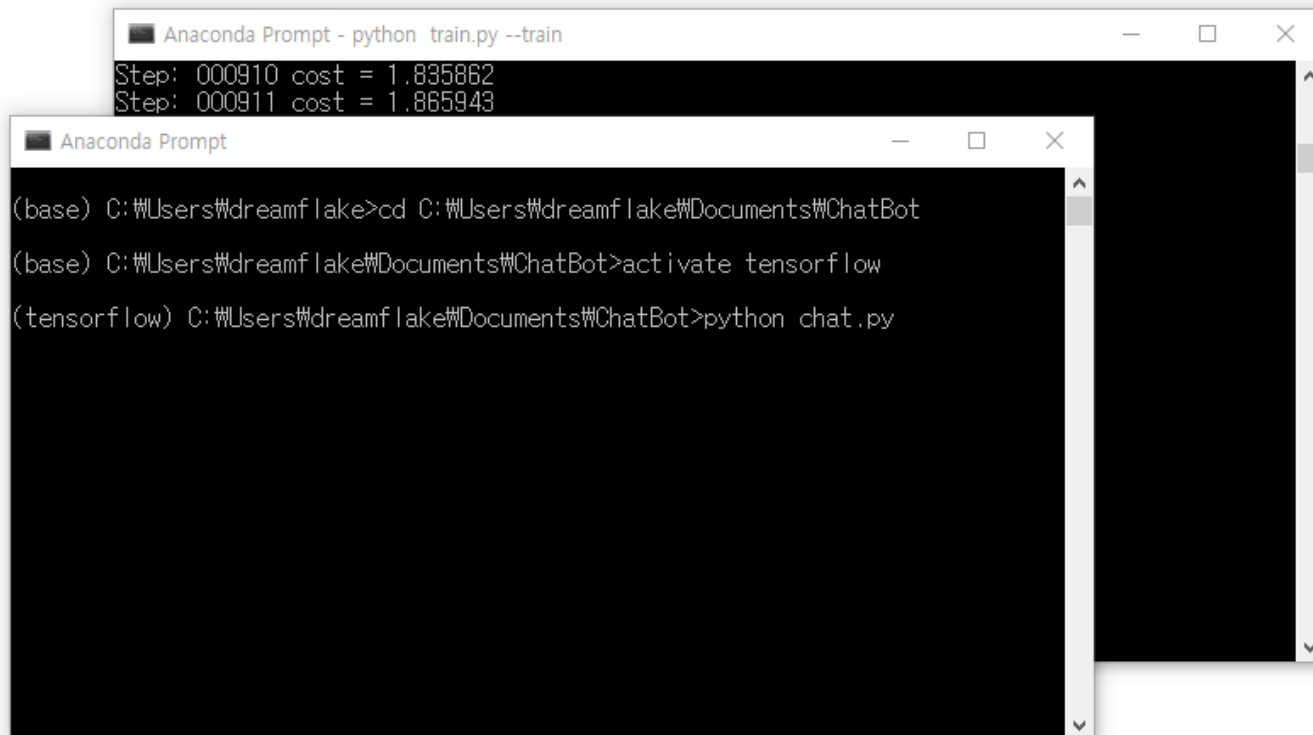
충분한 학습을 위해서는 20분 이상의 시간이 소요됩니다.
중간중간 모델이 저장됩니다.

```
선택 Anaconda Prompt - python train.py --train
Step: 000486 cost = 2.692885
Step: 000487 cost = 2.232306
Step: 000488 cost = 2.249778
Step: 000489 cost = 2.374797
Step: 000490 cost = 2.235514
Step: 000491 cost = 2.153999
Step: 000492 cost = 2.075456
Step: 000493 cost = 2.699585
Step: 000494 cost = 2.217270
Step: 000495 cost = 2.208056
Step: 000496 cost = 2.327363
Step: 000497 cost = 2.217830
Step: 000498 cost = 2.140596
Step: 000499 cost = 2.050863
Step: 000500 cost = 2.659835
Saving CheckPoint...
Step: 000501 cost = 2.213343
Step: 000502 cost = 2.221890
Step: 000503 cost = 2.328224
Step: 000504 cost = 2.203200
Step: 000505 cost = 2.136858
Step: 000506 cost = 2.055910
```

챗봇 학습하기

학습을 진행하는 도중에도 저장된 모델을 테스트할 수 있습니다.

새로운 Anaconda Prompt 창을 열어서 앞 슬라이드에 나온 Anaconda 동작 위치 설정 방법을 따라하신 다음에, `python chat.py`를 입력해보세요.



```
Anaconda Prompt - python train.py --train
Step: 000910 cost = 1.835862
Step: 000911 cost = 1.865943

Anaconda Prompt
(base) C:\Users\dreamflake>cd C:\Users\dreamflake\Documents\ChatBot
(base) C:\Users\dreamflake\Documents\ChatBot>activate tensorflow
(tensorflow) C:\Users\dreamflake\Documents\ChatBot>python chat.py
```

챗봇 학습하기

학습 초반에는 챗봇이 학습이 잘 되지 않은 상태이므로 전혀 이상한 답이 나옵니다.

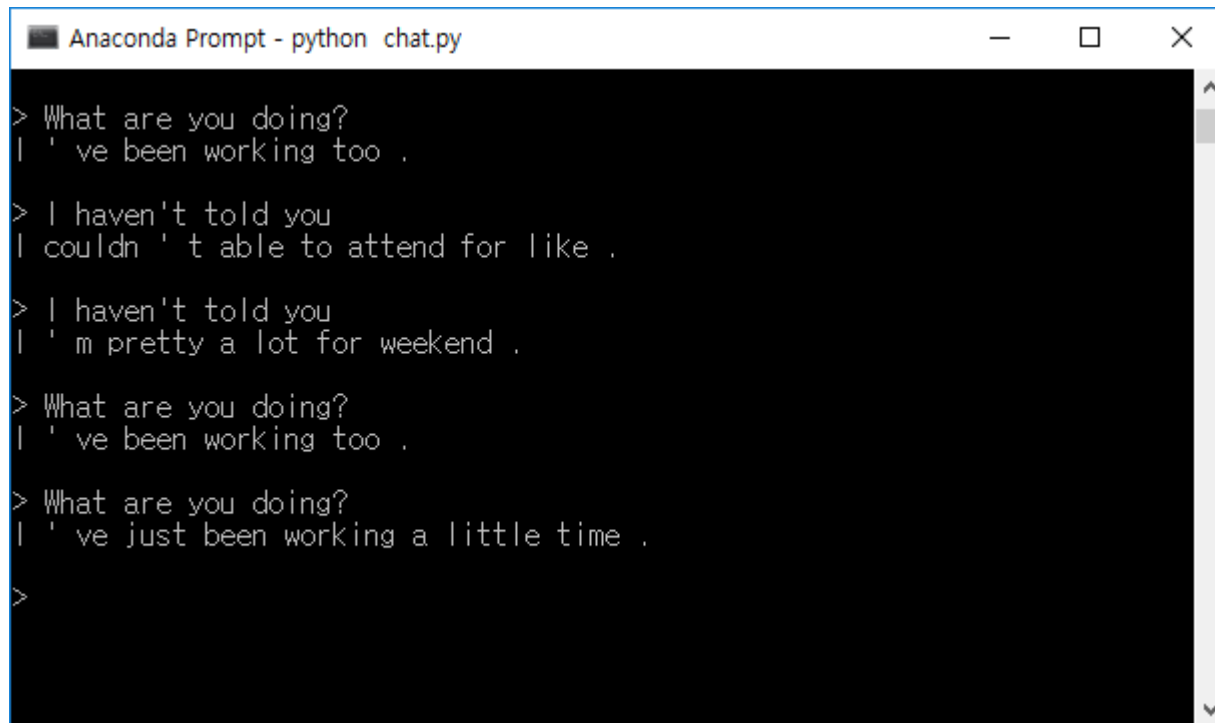
```
Anaconda Prompt - python chat.py
_runtime\gpu\gpu_device.cc:1356] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:01:00.0
totalMemory: 11.00GiB freeMemory: 9.10GiB
2018-10-16 14:04:13.552559: I T:\src\github\tensorflow\tensorflow\core\common
_runtime\gpu\gpu_device.cc:1435] Adding visible gpu devices: 0
2018-10-16 14:04:14.205757: I T:\src\github\tensorflow\tensorflow\core\common
_runtime\gpu\gpu_device.cc:923] Device interconnect StreamExecutor with stren
gth 1 edge matrix:
2018-10-16 14:04:14.213076: I T:\src\github\tensorflow\tensorflow\core\common
_runtime\gpu\gpu_device.cc:929] 0
2018-10-16 14:04:14.217955: I T:\src\github\tensorflow\tensorflow\core\common
_runtime\gpu\gpu_device.cc:942] 0: N
./model_english\conversation.ckpt-1000
> How are you?
| ' ' ' to to to to to .

> What's your name?
| ' ' ' to to to to .

> I don't know.
```

챗봇 학습하기

충분한 학습이 이뤄지고 나서 챗봇을 테스트해보면
이전보다 나은 답변을 하는 것을 볼 수 있습니다.



```
Anaconda Prompt - python chat.py

> What are you doing?
I 've been working too .

> I haven't told you
I couldn 't able to attend for like .

> I haven't told you
I 'm pretty a lot for weekend .

> What are you doing?
I 've been working too .

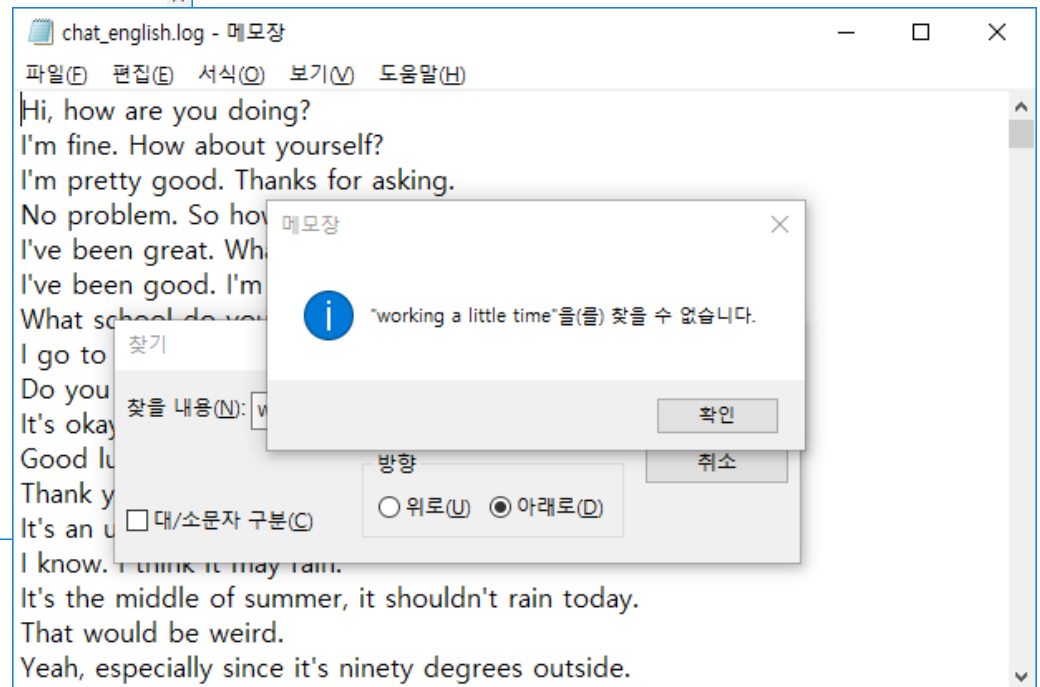
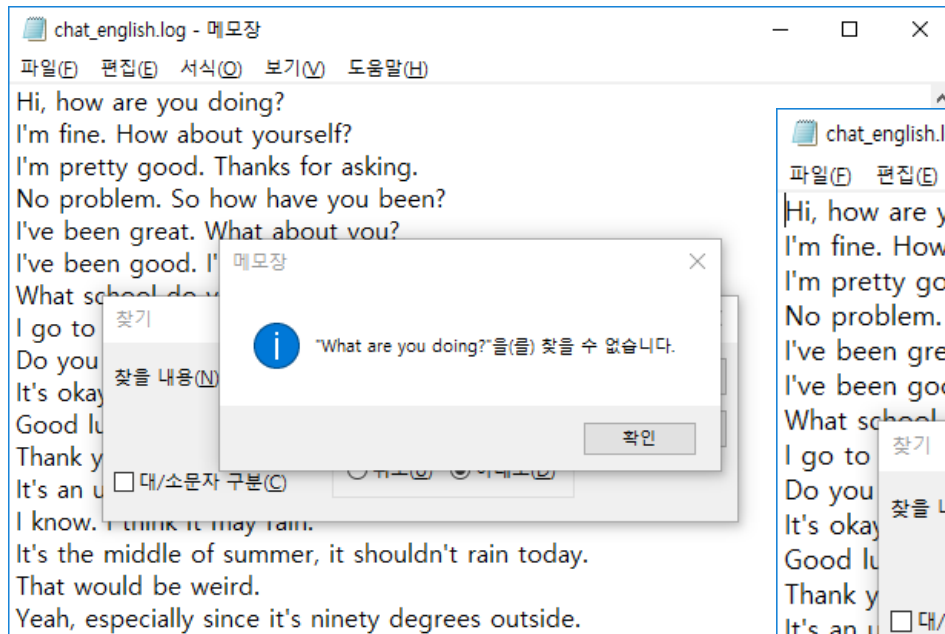
> What are you doing?
I 've just been working a little time .

>
```

같은 질문에도 다른 답변이 나올 수 있습니다.

챗봇 학습하기

질문도, 답변도, 학습 데이터셋에는 존재하지 않는 새로운 문장이 될 수 있습니다.



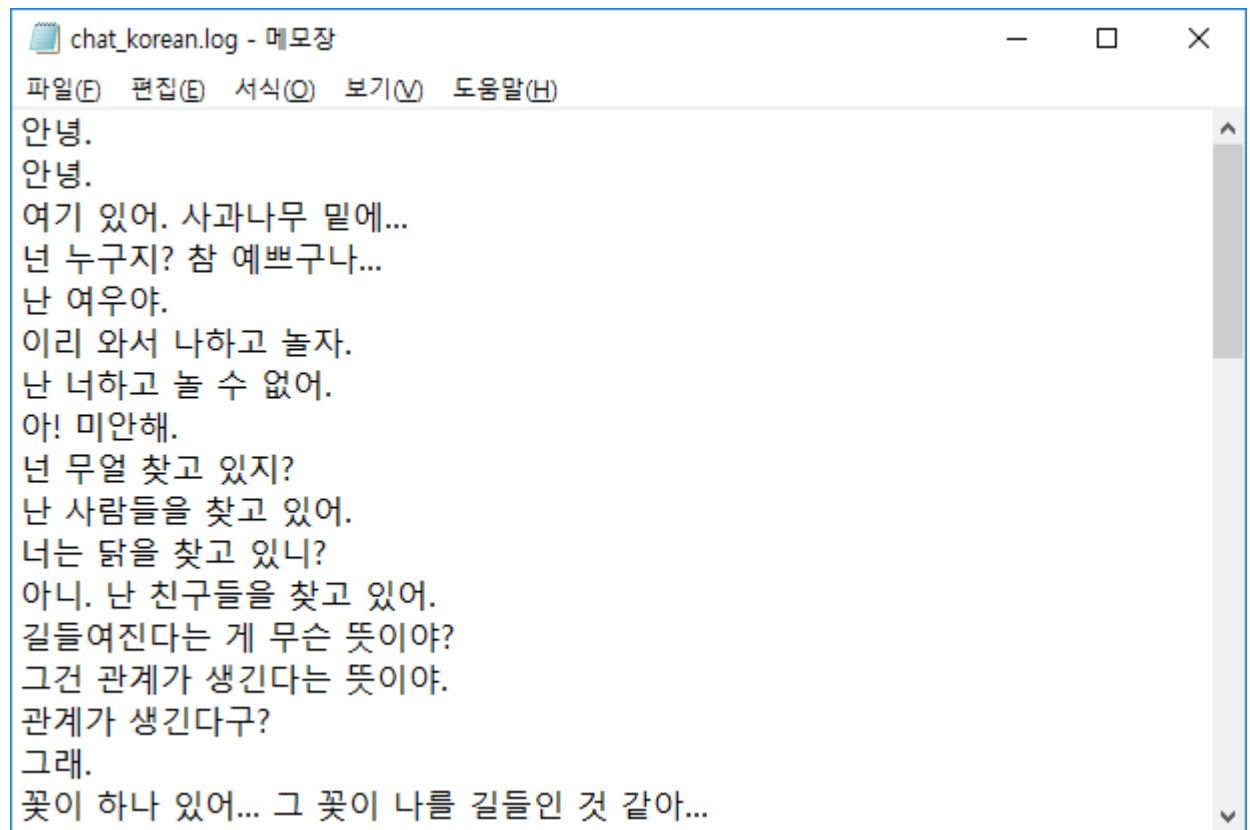


사용자 지정 대화록 학습시키기

데이터셋 바꾸어보기

한글 대화 데이터셋

한글 데이터셋은 어린왕자에서 발췌한 54줄의 문장으로 되어 있습니다.



```
chat_korean.log - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
안녕.
안녕.
여기 있어. 사과나무 밑에...
넌 누구지? 참 예쁘구나...
난 여우야.
이리 와서 나하고 놀자.
난 너하고 놀 수 없어.
아! 미안해.
넌 무얼 찾고 있지?
난 사람들을 찾고 있어.
너는 닭을 찾고 있니?
아니. 난 친구들을 찾고 있어.
길들여진다는 게 무슨 뜻이야?
그건 관계가 생긴다는 뜻이야.
관계가 생긴다구?
그래.
꽃이 하나 있어... 그 꽃이 나를 길들인 것 같아...
```

한글 대화 사용하기



config.py 더블클릭

한글 대화 사용하기

```
config.py x chat.py x C:\#chat.py x model.py x train.py x dialog.py x app.py x session.py x mod

No Python interpreter configured for the project

1  import tensorflow as tf
2
3  tf.app.flags.DEFINE_string("log_dir", "./logs", "로그를 저장할 폴더")
4  tf.app.flags.DEFINE_string("ckpt_name", "conversation.ckpt", "체크포인트 파일명")
5  tf.app.flags.DEFINE_boolean("train", False, "학습을 진행합니다.")
6  tf.app.flags.DEFINE_boolean("test", True, "테스트를 합니다.")
7  tf.app.flags.DEFINE_boolean("data_loop", True, "작은 데이터셋을 실험해보기 위해 사용합니다.")
8  tf.app.flags.DEFINE_integer("batch_size", 100, "미니 배치 크기")
9  tf.app.flags.DEFINE_integer("epoch", 1500, "총 학습 반복 횟수")
10
11  ##### 데이터셋을 변경할 때의 도움말 #####
12  # 1. 아래의 데이터셋 경로 변경
13  tf.app.flags.DEFINE_string("train_dir", "./model_english", "학습한 신경망을 저장할 폴더")
14  tf.app.flags.DEFINE_string("data_path", "./data/chat_english.log", "대화 파일 위치")
15  tf.app.flags.DEFINE_string("voc_path", "./data/chat_english.voc", "어휘 사전 파일 위치")
16  #####
17  # 2. 데이터셋으로부터 어휘 파일 생성
18  # python dialog.py --voc_build
19  # 3. 챗봇 모델 학습
20  # python train.py --train
21  # 4. 학습된 모델 테스트
22  # python chat.py
```

english 부분을 korean으로 모두 바꿔주세요

```
##### 데이터셋을 변경할 때의 도움말 #####
# 1. 아래의 데이터셋 경로 변경
tf.app.flags.DEFINE_string("train_dir", "./model_korean", "학습한 신경망을 저장할 폴더")
tf.app.flags.DEFINE_string("data_path", "./data/chat_korean.log", "대화 파일 위치")
tf.app.flags.DEFINE_string("voc_path", "./data/chat_korean.voc", "어휘 사전 파일 위치")
#####
```

한글 대화 사용하기

어휘 파일 생성과 학습을 위해 아래와 같이 명령어를 입력해보세요.
테스트 방법은 영어 데이터셋에서 했던 방법과 동일합니다.

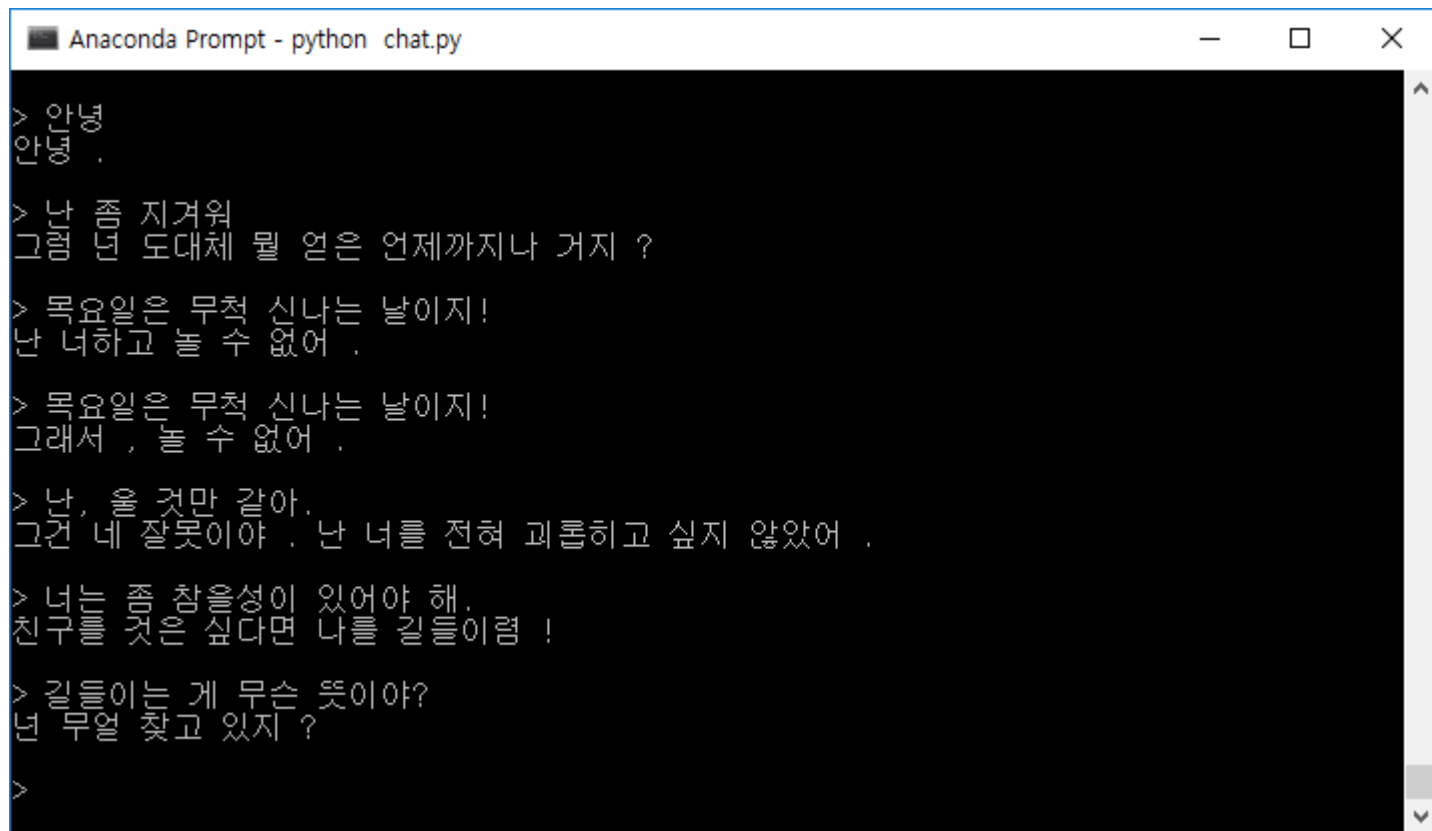
```
선택 Anaconda Prompt - python train.py --train

(tensorflow) C:\Users\dreamflake\Documents\ChatBot>python dialog.py --voc_build
다음 데이터에서 어휘 사전을 생성합니다. ./data/chat_korean.log

(tensorflow) C:\Users\dreamflake\Documents\ChatBot>python train.py --train
2018-10-16 15:57:30.846399: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_fea
ture_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not c
ompiled to use: AVX2
2018-10-16 15:57:31.179287: I T:\src\github\tensorflow\tensorflow\core\common_runtime\g
pu\gpu_device.cc:1356] Found device 0 with properties:
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:01:00:0
totalMemory: 11.00GiB freeMemory: 9.10GiB
2018-10-16 15:57:31.188627: I T:\src\github\tensorflow\tensorflow\core\common_runtime\g
pu\gpu_device.cc:1435] Adding visible gpu devices: 0
2018-10-16 15:57:31.876908: I T:\src\github\tensorflow\tensorflow\core\common_runtime\g
pu\gpu_device.cc:923] Device interconnect StreamExecutor with strength 1 edge matrix:
2018-10-16 15:57:31.882632: I T:\src\github\tensorflow\tensorflow\core\common_runtime\g
pu\gpu_device.cc:929] 0
2018-10-16 15:57:31.886387: I T:\src\github\tensorflow\tensorflow\core\common_runtime\g
pu\gpu_device.cc:942] 0: N
새로운 모델을 생성하는 중 입니다.
Step: 000001 cost = 5.123963
```

한글 대화 사용하기

한글 데이터셋은 영어 데이터셋에 비해 크기가 작아 금방 학습됩니다.
한글 데이터셋에 대해서도 테스트를 해보세요.



```
Anaconda Prompt - python chat.py

> 안녕
안녕 .

> 난 좀 지겨워
그럼 넌 도대체 뭘 얻은 언제까지나 거지 ?

> 목요일은 무척 신나는 날이지!
난 너하고 놀 수 없어 .

> 목요일은 무척 신나는 날이지!
그래서 , 놀 수 없어 .

> 난 , 올 것만 같아 .
그건 네 잘못이야 . 난 너를 전혀 괴롭히고 싶지 않았어 .

> 너는 좀 참을성이 있어야 해 .
친구를 것은 싶다면 나를 길들이렴 !

> 길들이는 게 무슨 뜻이야?
넌 무얼 찾고 있지 ?

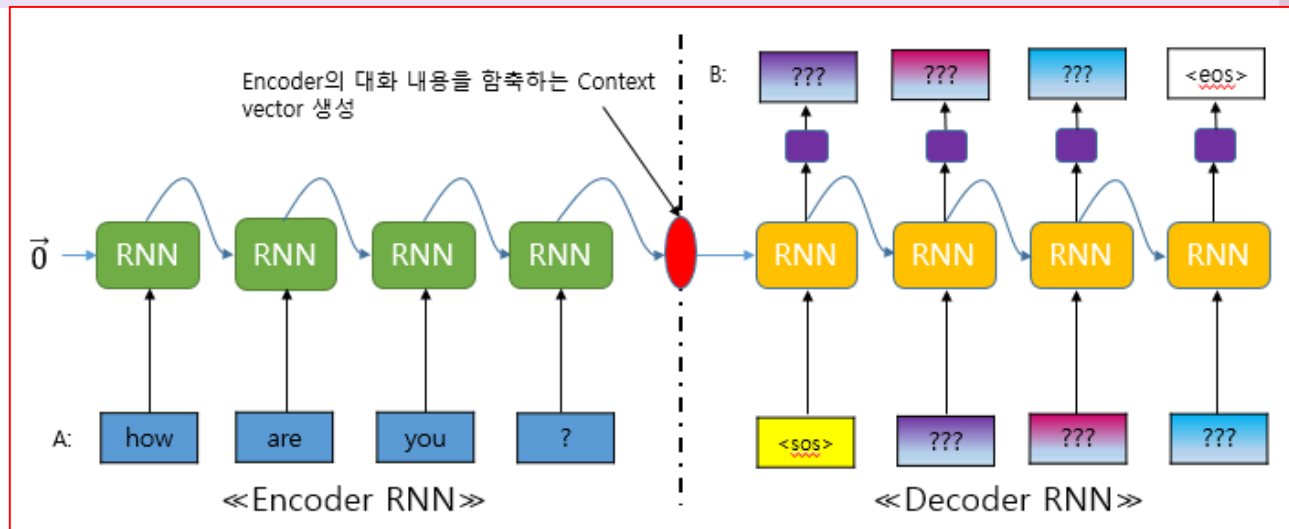
>
```



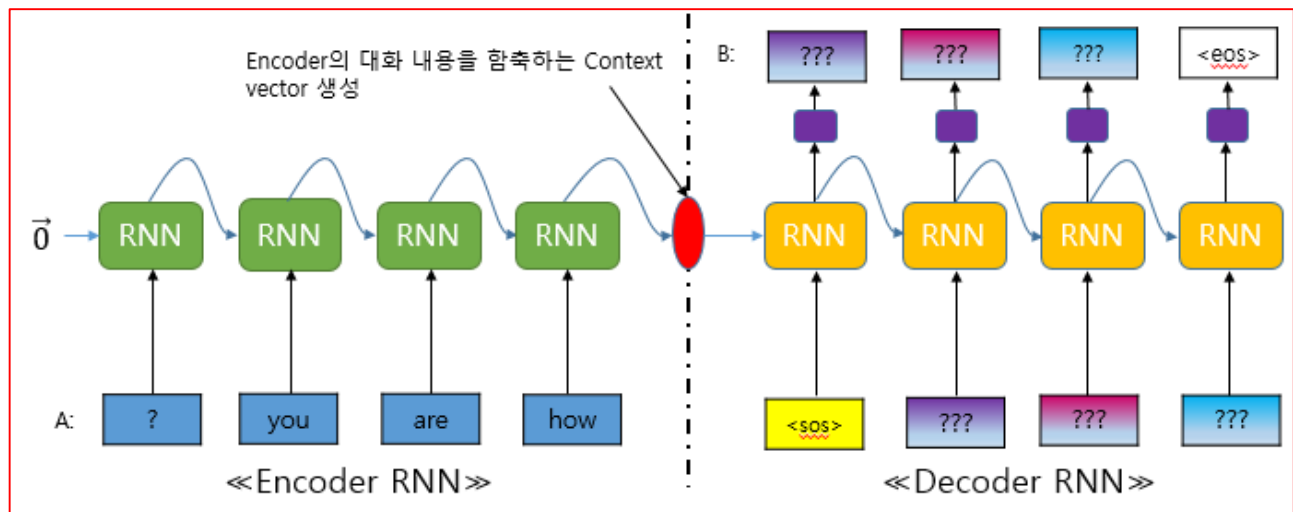
Context Vector 바꾸어보기

Context Vector 복습

1. 정방향 Encoding



2. 역방향 Encoding



Context Vector 설정하기

model.py 파일을 열어보세요.

```
config.py × chat.py × model.py × train.py × dialog.py ×
No Python interpreter configured for the project

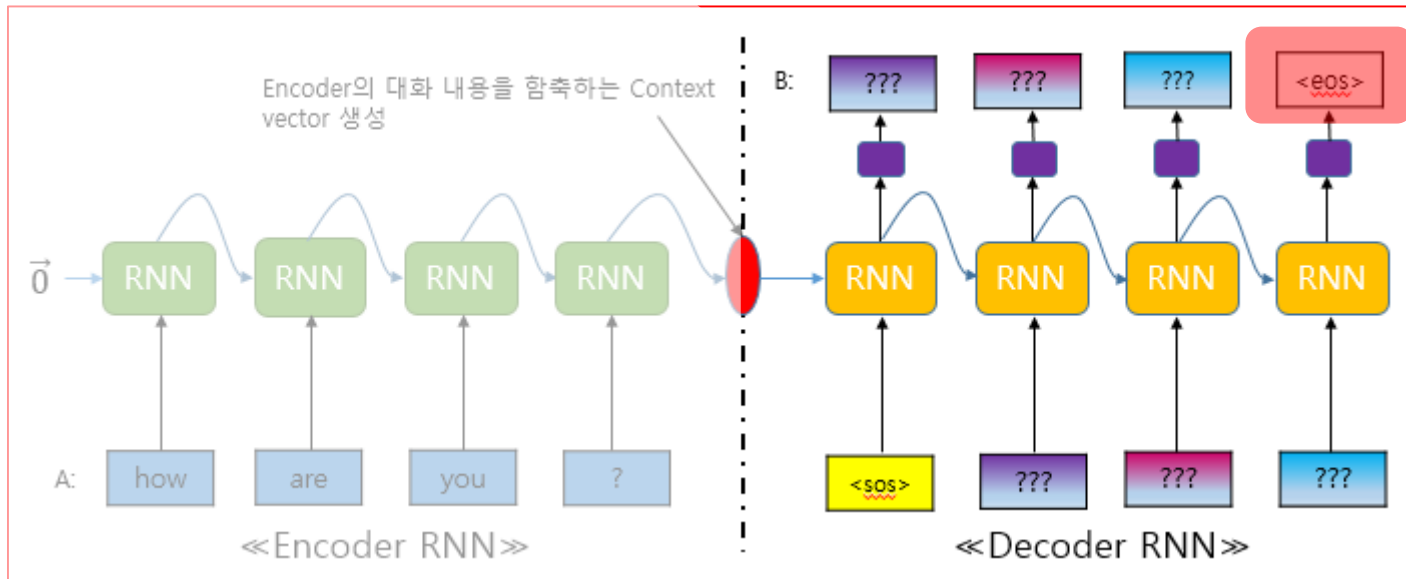
37
38 with tf.variable_scope('encode_forward'):
39     enc_forward_outputs, enc_states_forward_final = tf.nn.dynamic_rnn(enc_cell, self.enc_input, dtype=tf.float32)
40
41 with tf.variable_scope('encode_backward'):
42     enc_backward_outputs, enc_states_backward_final = tf.nn.dynamic_rnn(enc_cell, self.enc_input_reverse, dtype=tf.float32)
43
44 enc_states = []
45 enc_states_forward = enc_forward_outputs[0]
46 enc_states_backward = enc_backward_outputs[0]
47
48 for i, item in enumerate(enc_states_forward):
49     enc_states.append(tf.contrib.rnn.LSTMStateTuple(tf.concat((item[0], enc_states_backward[i][0]), axis=2),
50                                                         tf.concat((item[1], enc_states_backward[i][1]), axis=2)))
51
52 #####
53 # Decoder에 처음으로 들어갈 context vector의 값을 설정해 보세요
54 # initial_state= 다음에 enc_states_forward_final이 들어가면 정방향,
55 # enc_states_backward_final이 들어가면 역방향 입력이 들어갑니다.
56 with tf.variable_scope('decode'):
57     outputs, dec_states = tf.nn.dynamic_rnn(dec_cell, self.dec_input, dtype=tf.float32,
58                                             initial_state=enc_states_backward_final)
59 #####
```

정방향 Encoding 된
Context Vector

역방향 Encoding 된
Context Vector

위의 Context Vector
중 선택해서 기입

Test할 때의 Decoder 복습



EOS(End of Sequence)가 Output으로
나올 때 까지 문장 생성

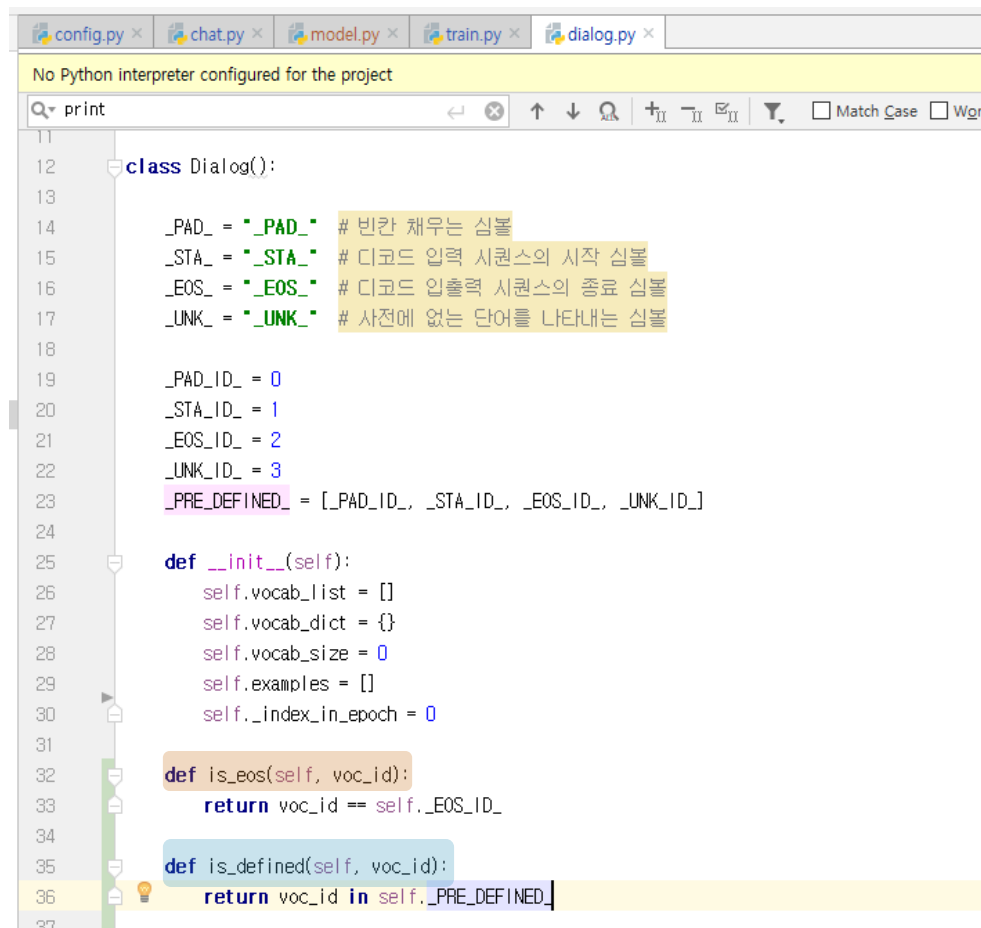
Test를 위한 Decoder 부분 살펴보기

- ▶ chat.py 파일을 열어보세요.

```
52                                     FLAGS.max_decode_len)
53
54     return self.model.predict(self.sess, [enc_input], [dec_input])
55
56     #####
57     def _get_replay(self, msg): # 실제 Reply(응답) 구성하는 부분
58         enc_input = self.dialog.tokenizer(msg)
59         enc_input = self.dialog.tokens_to_ids(enc_input)
60         dec_input = []
61
62         curr_seq = 0
63         for i in range(FLAGS.max_decode_len):
64             outputs = self._decode(enc_input, dec_input)
65             if self.dialog.is_eos(outputs[0][curr_seq]):
66                 break
67             elif self.dialog.is_defined(outputs[0][curr_seq]) is not True:
68                 dec_input.append(outputs[0][curr_seq])
69                 curr_seq+=1
70
71         reply = self.dialog.decode([dec_input], True)
72
73         return reply
74     #####
```

Test를 위한 Decoder 부분 살펴보기

- ▶ dialog.py 파일을 참고합니다.



```
11
12 class Dialog():
13
14     _PAD_ = "_PAD_" # 빈칸 채우는 심볼
15     _STA_ = "_STA_" # 디코드 입력 시퀀스의 시작 심볼
16     _EOS_ = "_EOS_" # 디코드 입출력 시퀀스의 종료 심볼
17     _UNK_ = "_UNK_" # 사전에 없는 단어를 나타내는 심볼
18
19     _PAD_ID_ = 0
20     _STA_ID_ = 1
21     _EOS_ID_ = 2
22     _UNK_ID_ = 3
23     _PRE_DEFINED_ = [_PAD_ID_, _STA_ID_, _EOS_ID_, _UNK_ID_]
24
25     def __init__(self):
26         self.vocab_list = []
27         self.vocab_dict = {}
28         self.vocab_size = 0
29         self.examples = []
30         self._index_in_epoch = 0
31
32     def is_eos(self, voc_id):
33         return voc_id == self._EOS_ID_
34
35     def is_defined(self, voc_id):
36         return voc_id in self._PRE_DEFINED_
37
```

Question #1

정방향? 역방향?

▶ 지금까지 실습에서는 역방향 Context Vector를 사용했습니다.

그 이유는 영어의 경우 "What..? How..?" 처럼 문장의 맨 앞에서 중요한 단어가 나오므로 역방향 Context Vector가 문장의 중요한 의미를 파악하기에 더 유리하기 때문입니다.

과연 한글의 경우에도 역방향 Context Vector가 더 좋을까요?

★ 이전 슬라이드에 나온 방법을 참고하여 순방향 Context Vector로 전환하여 학습 & 테스트를 진행해보시고 두 가지 방식을 비교하여 느낀 점을 적어주세요.

순방향 Context Vector를 사용하는 모델을 새로 학습하기 위해 먼저 config.py에서 model_korean을 model_korean_forward로 바꿔주세요.



Most Frequent Word를 이용한 학습

빅 데이터에 (긴 대화록)에 적용

빅 데이터로의 적용으로의 한계

- ▶ 상대적으로 매우 많은 단어 수로 인한 메모리 오류 발생
 - 해결방법: 가장 자주 사용되는 단어 몇 개만 임의로 선택하여 사용

Most Frequent Word 사용

PC config	2018-10-14 오후...	JetBrains PyChar...	2KB
PC dialog	2018-10-14 오후...	JetBrains PyChar...	8KB
PC model	2018-10-14 오후...	JetBrains PyChar...	5KB

```
172 def build_vocab(self, data_path, vocab_path):
173     with open(data_path, 'r', encoding='utf-8') as content_file:
174         content = content_file.read()
175         words = self.tokenizer(content)
176         #counter=collections.Counter(words)
177         #words=counter.most_common(5000)
178         pdb.set_trace()
179         #words=[i[0] for i in words]
180         words = list(set(words))
181
182     with open(vocab_path, 'w', encoding='utf-8') as vocab_file:
183         for w in words:
184             vocab_file.write(w + '\n')
```

```
172 def build_vocab(self, data_path, vocab_path):
173     with open(data_path, 'r', encoding='utf-8') as content_file:
174         content = content_file.read()
175         words = self.tokenizer(content)
176         counter=collections.Counter(words)
177         words=counter.most_common(10000)
178         #pdb.set_trace()
179         words=[i[0] for i in words]
180         words = list(set(words))
181
182     with open(vocab_path, 'w', encoding='utf-8') as vocab_file:
183         for w in words:
184             vocab_file.write(w + '\n')
```

가장 자주 사용되는 단어
10000개 사용

Question #2

Word Embedding의 사용

- ▶ 앞 단계에서는 가장 자주 사용되는 단어만 사용하여서 메모리 오류 현상을 방지하였습니다. 하지만 Word Embedding을 사용하여서도 이 문제를 해결할 수 있습니다.

★ Word Embedding을 사용할 때 메모리 오류 현상을 방지할 수 있는 이유를 작성하여 주세요.

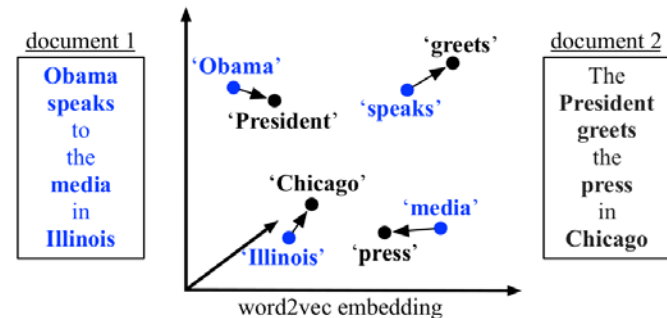
Try to build a lower dimensional embedding

Vocabulary:
Man, woman, boy,
girl, prince,
princess, queen,
king, monarch



	Femininity	Youth	Royalty
Man			
Woman			
Boy			
Girl			
Prince			
Princess			
Queen			
King			
Monarch			

@shane_a_lynn | @TeamEdgier



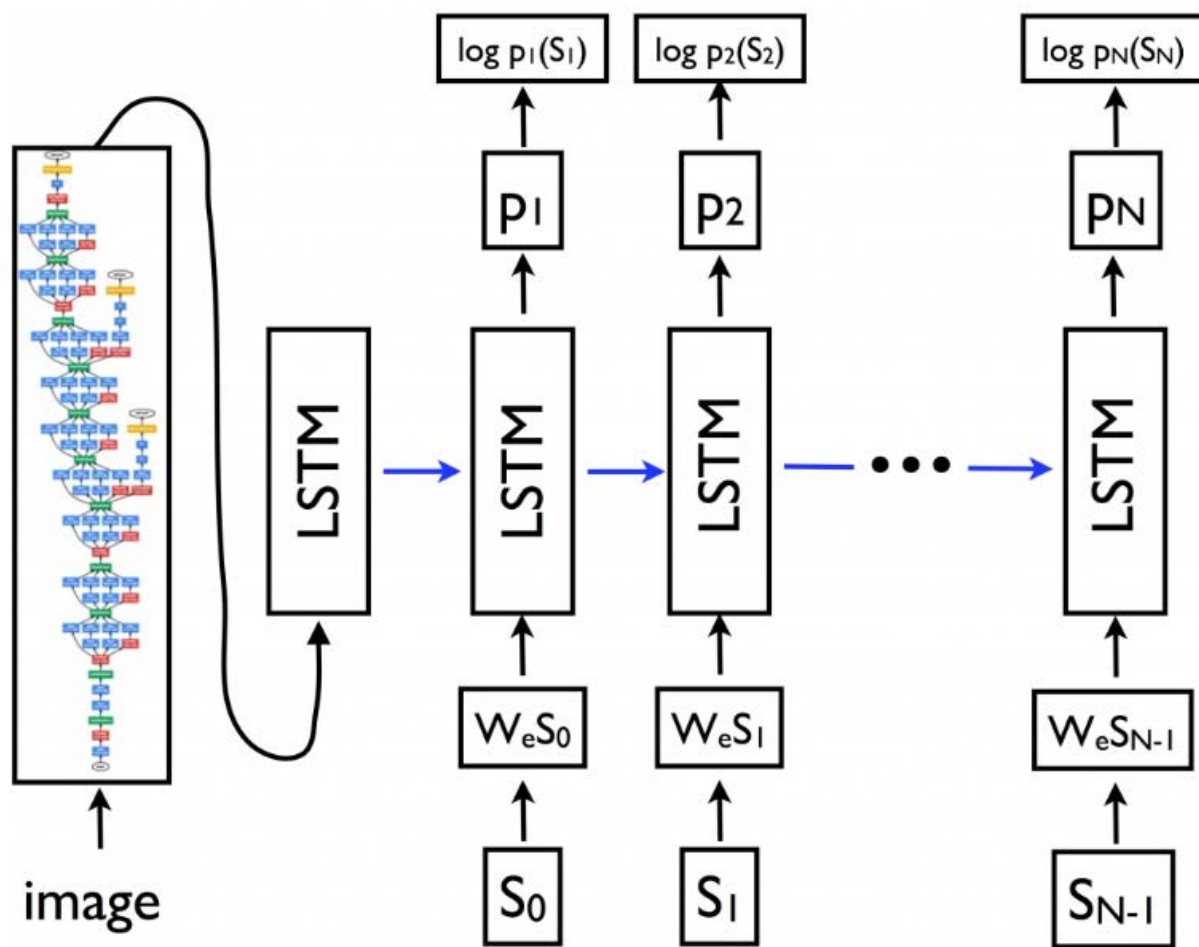
22

★ 그런데 Word Embedding은 챗봇의 성능을 향상시키는 데에도 도움을 줍니다. 왜 그렇게 될지 아래의 예를 보고 이유를 적어주세요.

문장1: 도로 위의 빨간 자동차를 찾으세요.

문장2: 길가에 붉은색 차가 있나요?

Context Vector를 영상과 결합한 예



<Image Captioning>

Context Vector를 영상과 결합한 예

A person skiing down a snow covered slope.



A group of giraffe standing next to each other.



감사합니다.