LangCon 2019

트랜스퍼 러닝과 텍스트 문서 분류

고재선

소개

고재선

- 통신공학 전공, 법학전문대학원 졸업
- 대학원(법학박사) 과정
- 2014년부터 변호사로 근무
- 관심분야 : 특허법, 디지털포렌식, 자연어처리

소개

데이터가 부족할 때는,

트랜스퍼 러닝을 한 번 생각해보자.

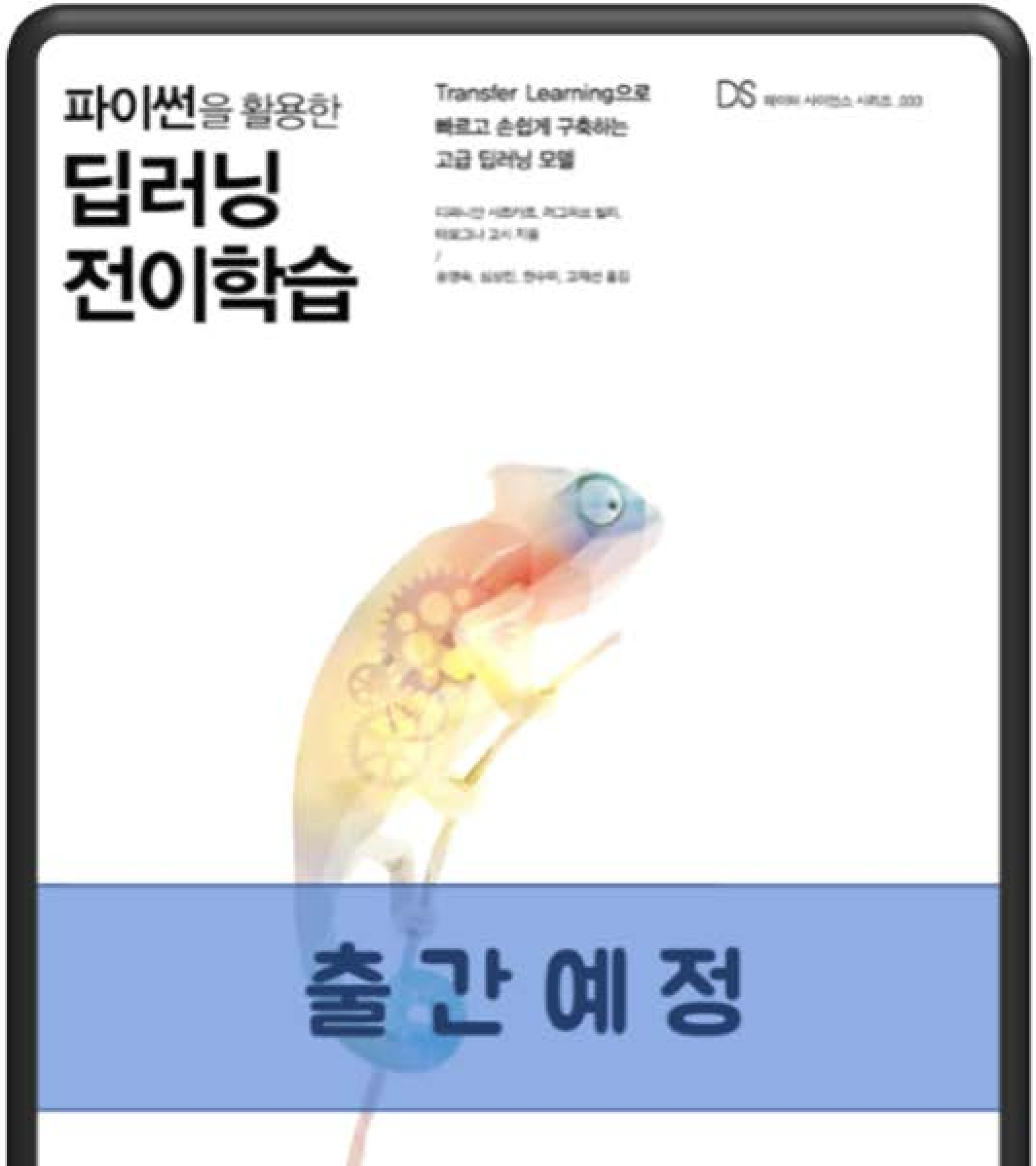
목차

1. 트랜스퍼 러닝

2. 워드 임베딩/CNN 문서 모델

3. 트랜스퍼 러닝 예제

4. 요약 및 결론



파이썬을 활용한
딥러닝
전이학습

Transfer Learning으로
빠르고 손쉽게 구축하는
고급 딥러닝 모델

디파난잔 사르카르, 라그하브 발리,
타모그나 고시 지음
/
송경록, 심상진, 한수미, 고복선 옮김

DS 데이터 사이언스 시리즈 _033

출간예정

* 이하의 그림 및 내용들은 국내 번역출간예정인 Dipanjan Sarkar, Raghav Bali, Tamoghna Ghosh가 저술한, “Hands-On Transfer Learning with Python”의 내용을 주로 인용하였습니다.

트랜스퍼 러닝(Transfer Learning)?

≒하나의 설정에서 배운 무엇인가를,

다른 설정에서도 일반화할 수 있도록 활용하는 환경*

* 이안 굿펠로,요슈아 벤지오,에런 쿠빌 공저, 심층학습(Deep Learning), 류광 역

트랜스퍼 러닝(Transfer Learning)?

≒하나의 설정에서 배운 무엇인가를,

다른 설정에서도 일반화할 수 있도록 활용하는 환경*

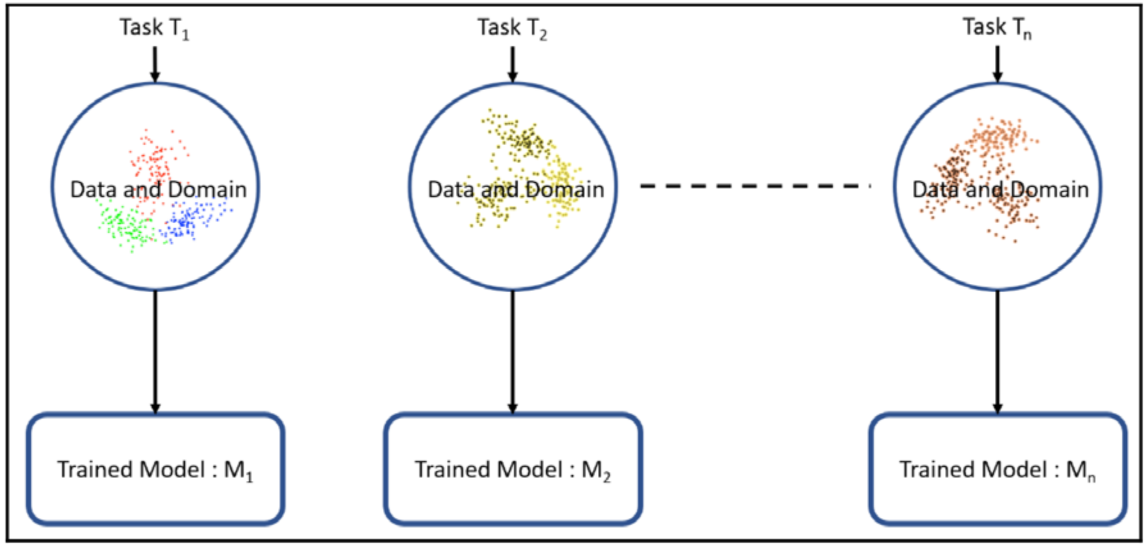
≒다른 분야의 학습 모델을 가져와 유사한 분야에서 적용하는 것

* 이안 굿펠로,요슈아 벤지오,에런 쿠빌 공저, 심층학습(Deep Learning), 류광 역

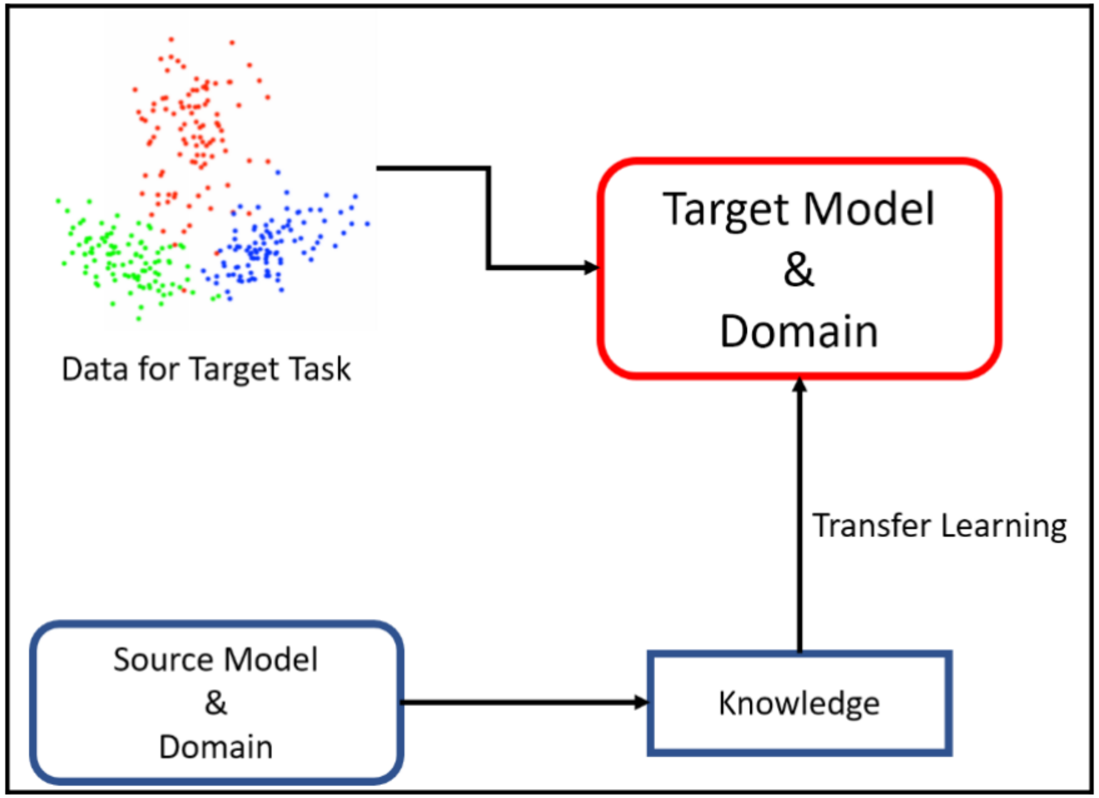
1. 트랜스퍼 러닝

Language Conference 2019

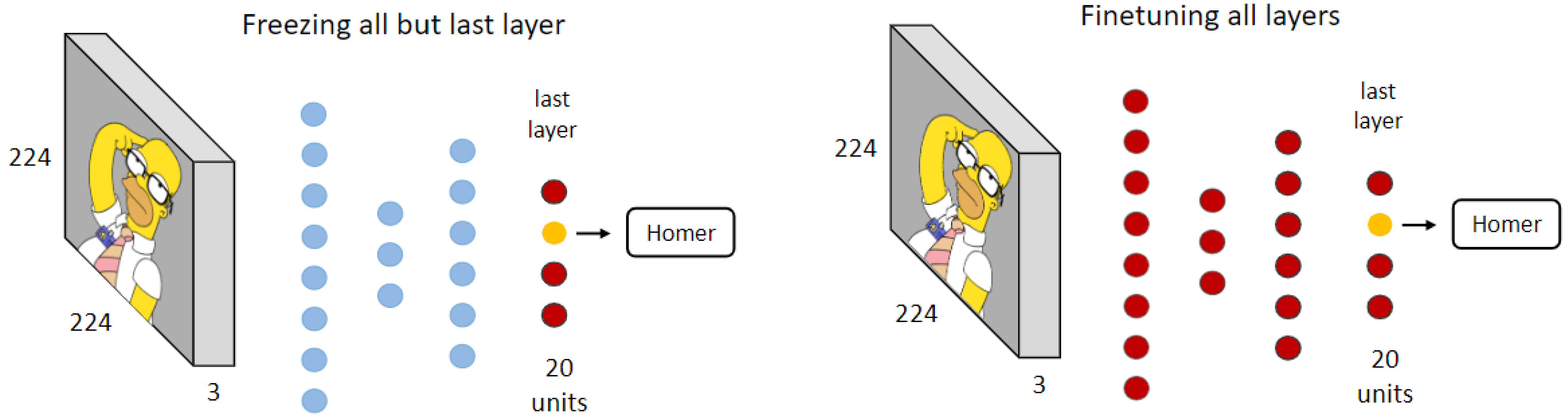
기존 머신 러닝



트랜스퍼 러닝



트랜스퍼 러닝



*Miguel González-Fierro, A Gentle Introduction To Transfer Learning For Image Classification

트랜스퍼 러닝을 사용하는 이유?

1. 성능의 향상
2. 모델 개발/학습 시간 단축

영상(CV) 분야의 트랜스퍼 러닝?

대량의 이미지 데이터 셋으로

학습시킨 모델을 사용하여

구체적인 문제들을 해결

*http://www.image-net.org

자연어 처리의 트랜스퍼 러닝은?

- 워드 임베딩을 중심으로 논의
- 최근 ELMO, BERT 등의 사전 학습 모델 등장

목차

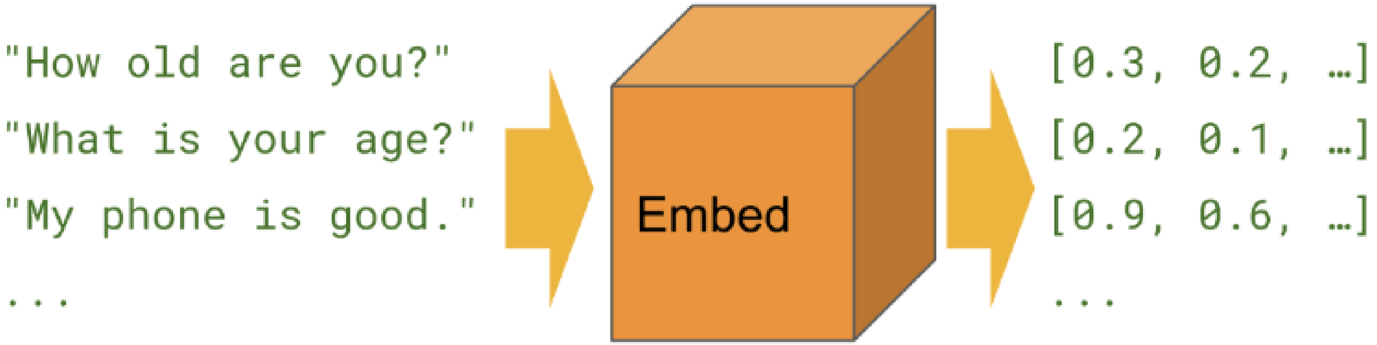
1. 트랜스퍼 러닝

2. 워드 임베딩/문서 분류 모델

3. 트랜스퍼 러닝 예제

4. 요약 및 결론

임베딩?



```
"How old are you?"
"What is your age?"
"My phone is good."
...
```

```
[0.3, 0.2, …]
[0.2, 0.1, …]
[0.9, 0.6, …]
...
```

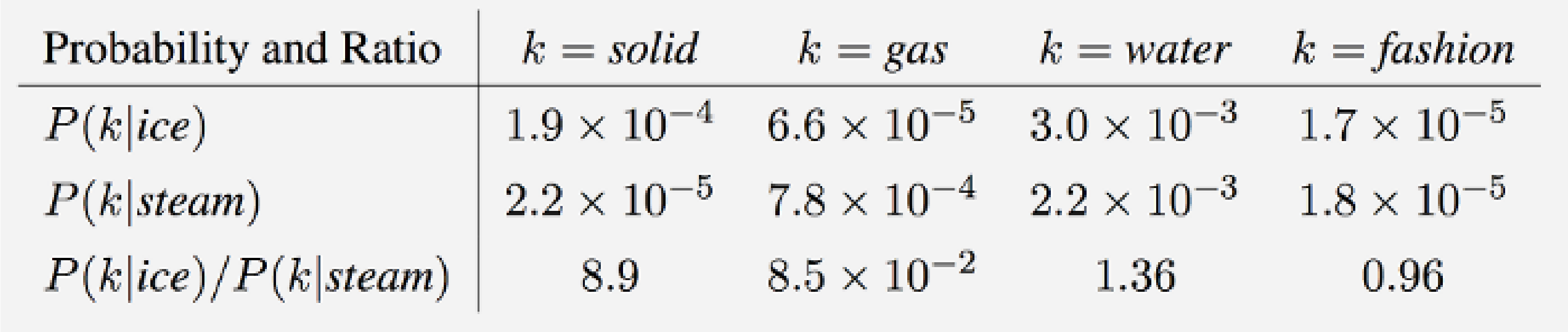
- 워드 임베딩 : 단어를 실수 벡터 값으로 맵핑시키는 것
- 어떻게 맵핑?

*https://www.learnopencv.com/universal-sentence-encoder/

워드 임베딩 모델 : Word2vec, Glove

- Word2vec : 문장 내 단어들의 위치를 기반으로 학습

- Glove : <u>전체 단어들의 통계 </u>정보(동시출현확률)를 사용



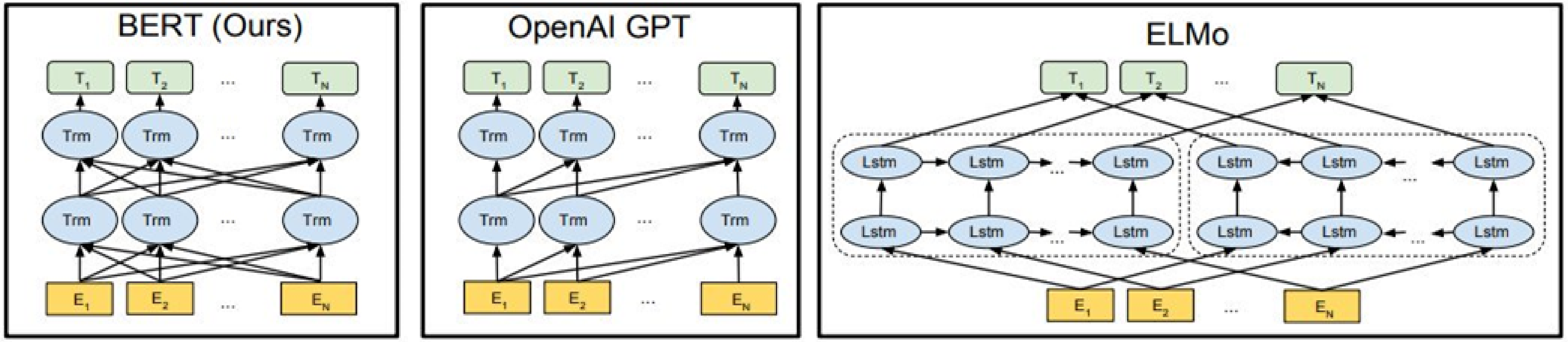
Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

*Jeffrey Pennington, Richard Socher, Christopher D. Manning, GloVe: Global Vectors for Word Representation

워드 임베딩 모델 : ELMO, BERT

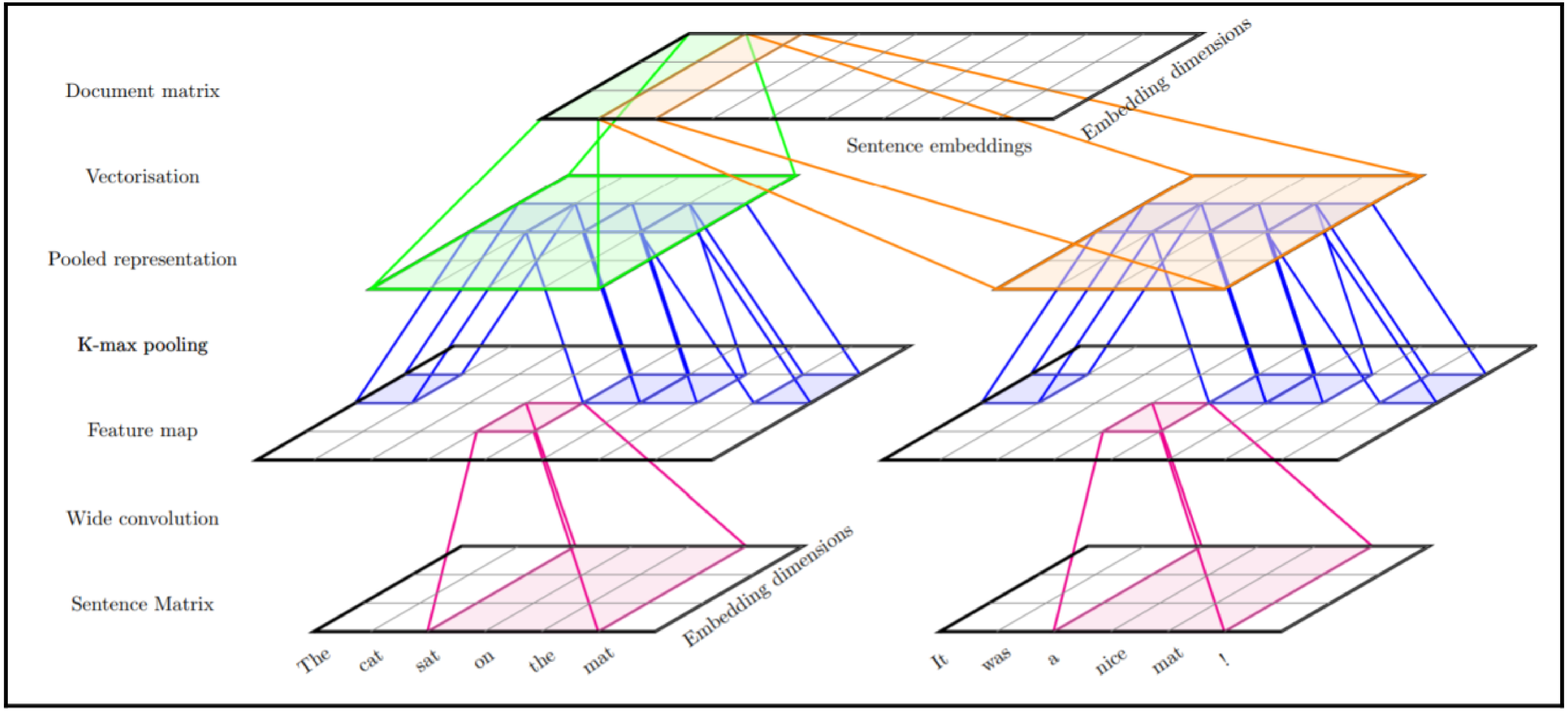
- 문맥에 따라 같은 단어라도 다른 벡터로 표현
  (Word2vec 에서의 다의어, 동음이의어 문제)

- 대량의 텍스트 데이터를 미리 학습하는 모델



* Jacob Devlin, et all, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

CNN 문서 모델 *



* Misha Denil, et all, Modelling, Visualising and Summarising Documents with a Single Convolutional Neural Network, 2014

CNN 문서 모델*

- 인풋 레이어 : 워드 임베딩
- 워드 임베딩 -> 문장 임베딩 -> 문서 임베딩
- 문장과 문서의 길이가 다를 수 있으므로,
  - 0으로 패딩 or 자르기

* Misha Denil, et all, Modelling, Visualising and Summarising Documents with a Single Convolutional Neural Network, 2014

목차

1. 트랜스퍼 러닝

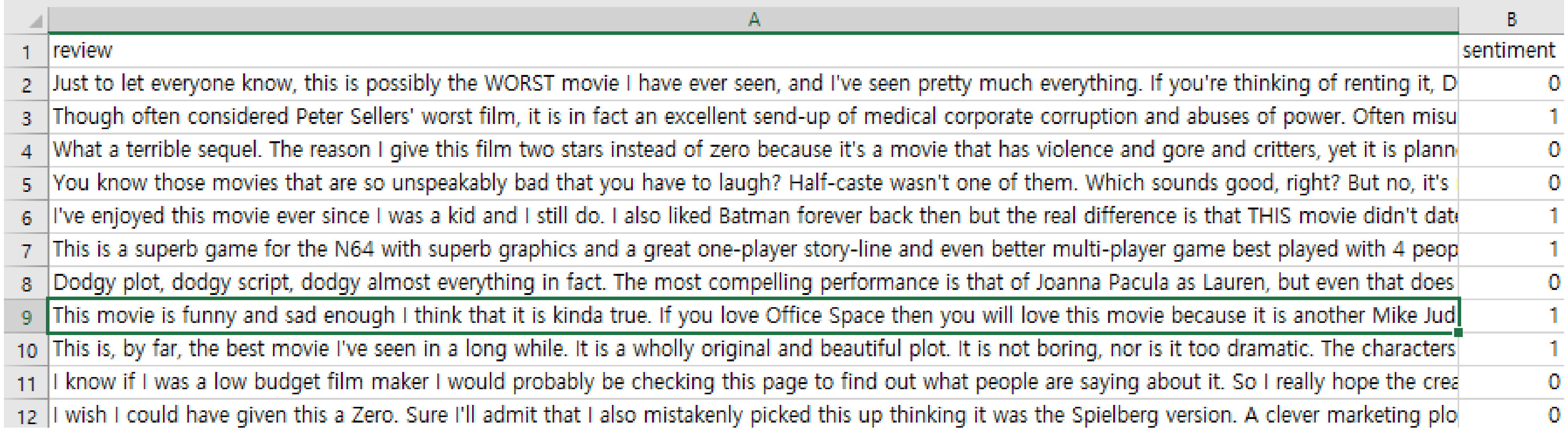
2. 워드 임베딩/문서 분류 모델

3. <u>트랜스퍼 러닝 예제</u>

4. 요약 및 결론

IMDB 영화 리뷰 – 긍정/부정 분류

- <u>트레이닝 데이터 25,000</u>개, 테스트 데이터 25,000개

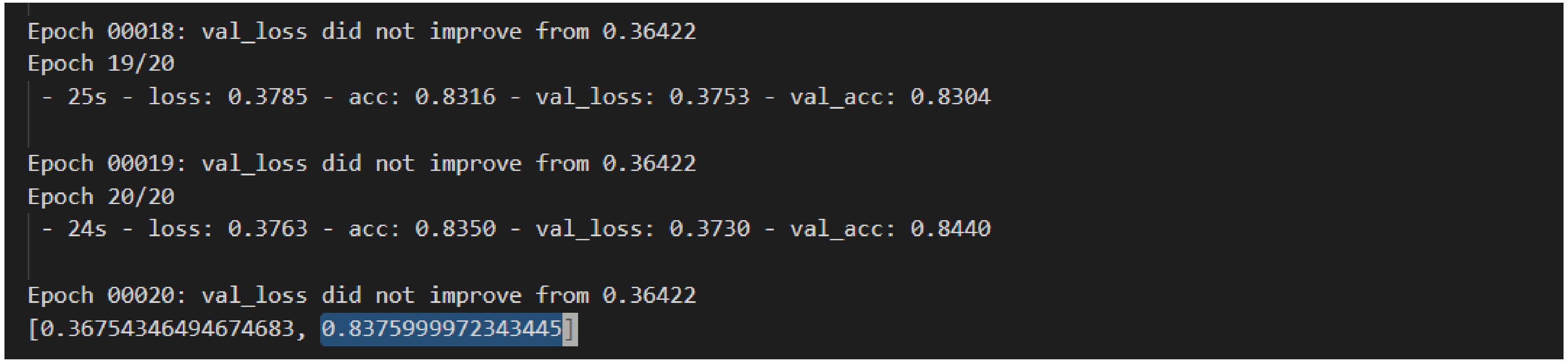


	A	B
1	review	sentiment
2	Just to let everyone know, this is possibly the WORST movie I have ever seen, and I've seen pretty much everything. If you're thinking of renting it, D	0
3	Though often considered Peter Sellers' worst film, it is in fact an excellent send-up of medical corporate corruption and abuses of power. Often misu	1
4	What a terrible sequel. The reason I give this film two stars instead of zero because it's a movie that has violence and gore and critters, yet it is plann	0
5	You know those movies that are so unspeakably bad that you have to laugh? Half-caste wasn't one of them. Which sounds good, right? But no, it's	0
6	I've enjoyed this movie ever since I was a kid and I still do. I also liked Batman forever back then but the real difference is that THIS movie didn't dat	1
7	This is a superb game for the N64 with superb graphics and a great one-player story-line and even better multi-player game best played with 4 peop	1
8	Dodgy plot, dodgy script, dodgy almost everything in fact. The most compelling performance is that of Joanna Pacula as Lauren, but even that does	0
9	This movie is funny and sad enough I think that it is kinda true. If you love Office Space then you will love this movie because it is another Mike Jud	1
10	This is, by far, the best movie I've seen in a long while. It is a wholly original and beautiful plot. It is not boring, nor is it too dramatic. The characters	1
11	I know if I was a low budget film maker I would probably be checking this page to find out what people are saying about it. So I really hope the crea	0
12	I wish I could have given this a Zero. Sure I'll admit that I also mistakenly picked this up thinking it was the Spielberg version. A clever marketing plo	0

*http://ai.stanford.edu/~amaas/data/sentiment/

IMDB 영화 리뷰 – 긍정/부정 분류

- 트레이닝 데이터 25,000개, 테스트 데이터 25,000개
- 사전 학습된 Glove 벡터(Wikipedia 2014 + Gigaword 5)
- 약 83.7%



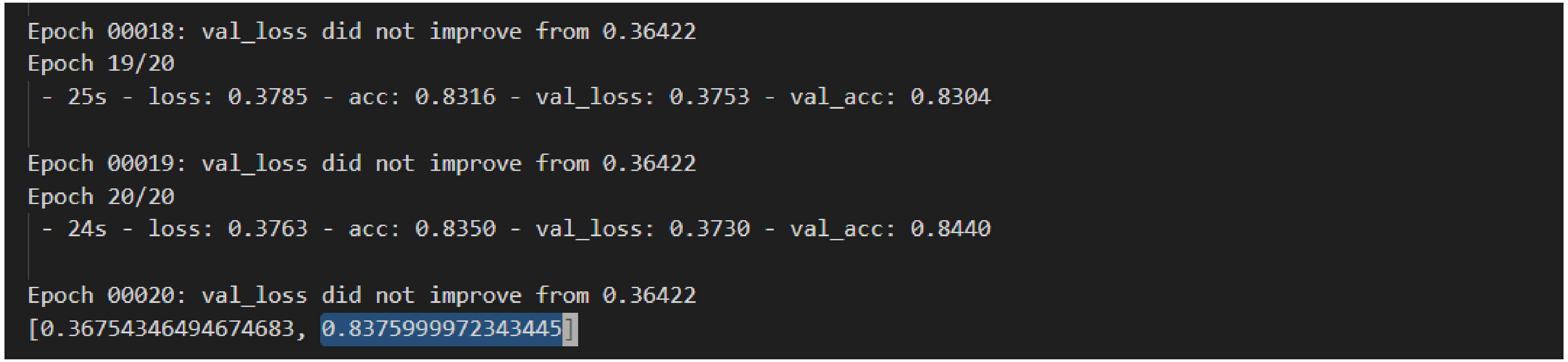
```
Epoch 00018: val_loss did not improve from 0.36422
Epoch 19/20
 - 25s - loss: 0.3785 - acc: 0.8316 - val_loss: 0.3753 - val_acc: 0.8304

Epoch 00019: val_loss did not improve from 0.36422
Epoch 20/20
 - 24s - loss: 0.3763 - acc: 0.8350 - val_loss: 0.3730 - val_acc: 0.8440

Epoch 00020: val_loss did not improve from 0.36422
[0.36754346494674683, 0.8375999972343445]
```

IMDB 영화 리뷰 – 긍정/부정 분류

- 트레이닝 데이터 25,000개, 테스트 데이터 25,000개

- 사전 학습된 Glove 벡터(Wikipedia 2014 + Gigaword 5)

- 약 83.7%

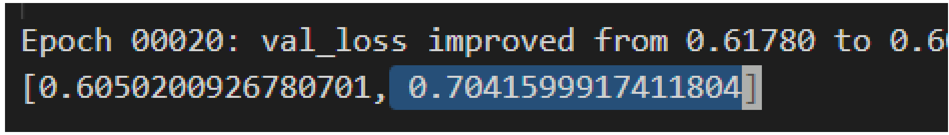


```
Epoch 00018: val_loss did not improve from 0.36422
Epoch 19/20
 - 25s - loss: 0.3785 - acc: 0.8316 - val_loss: 0.3753 - val_acc: 0.8304

Epoch 00019: val_loss did not improve from 0.36422
Epoch 20/20
 - 24s - loss: 0.3763 - acc: 0.8350 - val_loss: 0.3730 - val_acc: 0.8440

Epoch 00020: val_loss did not improve from 0.36422
[0.36754346494674683, 0.8375999972343445]
```

3. 트랜스퍼 러닝 예제

만약 트레이닝 데이터가 1,250개만 있다면? (5%)

- 적은 데이터 -> 성능이 안나옴[=70%]

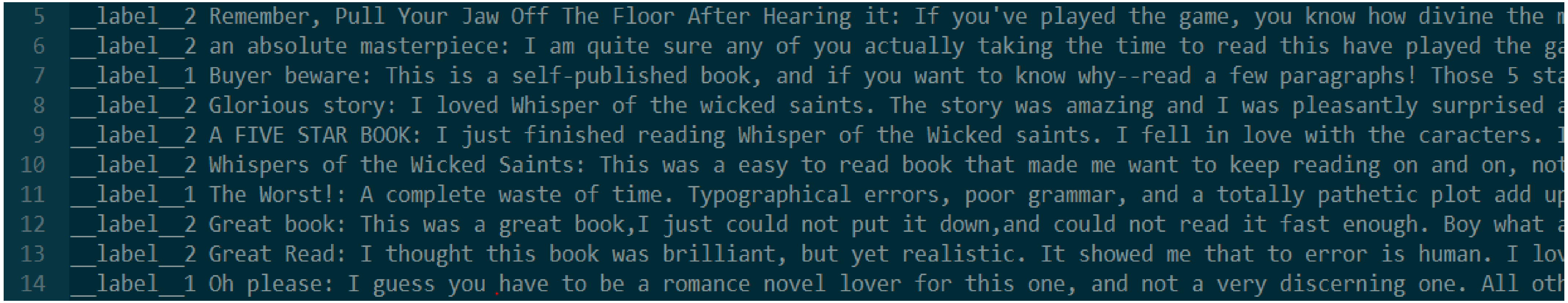


```
Epoch 00020: val_loss improved from 0.61780 to 0.6
[0.6050200926780701, 0.7041599917411804]
```

- 이 경우 트랜스퍼 러닝을 고려해 볼 수 있음
- 영화평과 유사한 상품 구매 평가 데이터!

아마존 제품 구매 평가 - 긍정/부정 분류

- 학습용 데이터 360만개, 테스트용 데이터 40만개



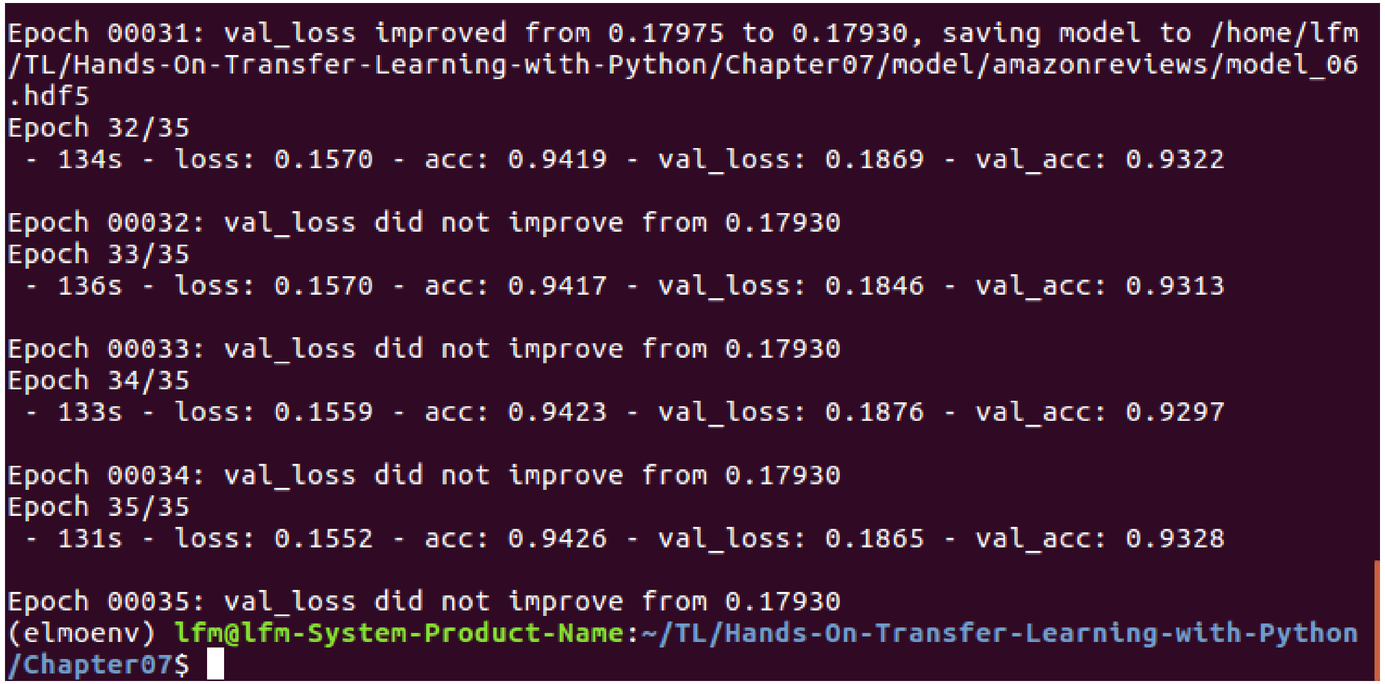
```
__label__2 Remember, Pull Your Jaw Off The Floor After Hearing it: If you've played the game, you know how divine the m
__label__2 an absolute masterpiece: I am quite sure any of you actually taking the time to read this have played the ga
__label__1 Buyer beware: This is a self-published book, and if you want to know why--read a few paragraphs! Those 5 sta
__label__2 Glorious story: I loved Whisper of the wicked saints. The story was amazing and I was pleasantly surprised a
__label__2 A FIVE STAR BOOK: I just finished reading Whisper of the Wicked saints. I fell in love with the caracters. I
__label__2 Whispers of the Wicked Saints: This was a easy to read book that made me want to keep reading on and on, not
__label__1 The Worst!: A complete waste of time. Typographical errors, poor grammar, and a totally pathetic plot add up
__label__2 Great book: This was a great book,I just could not put it down,and could not read it fast enough. Boy what a
__label__2 Great Read: I thought this book was brilliant, but yet realistic. It showed me that to error is human. I lov
__label__1 Oh please: I guess you have to be a romance novel lover for this one, and not a very discerning one. All oth
```

*https://www.kaggle.com/bittlingmayer/amazonreviews

아마존 제품 구매 평가 - 긍정/부정 분류

- 학습용 데이터 360만개, 테스트용 데이터 40만개

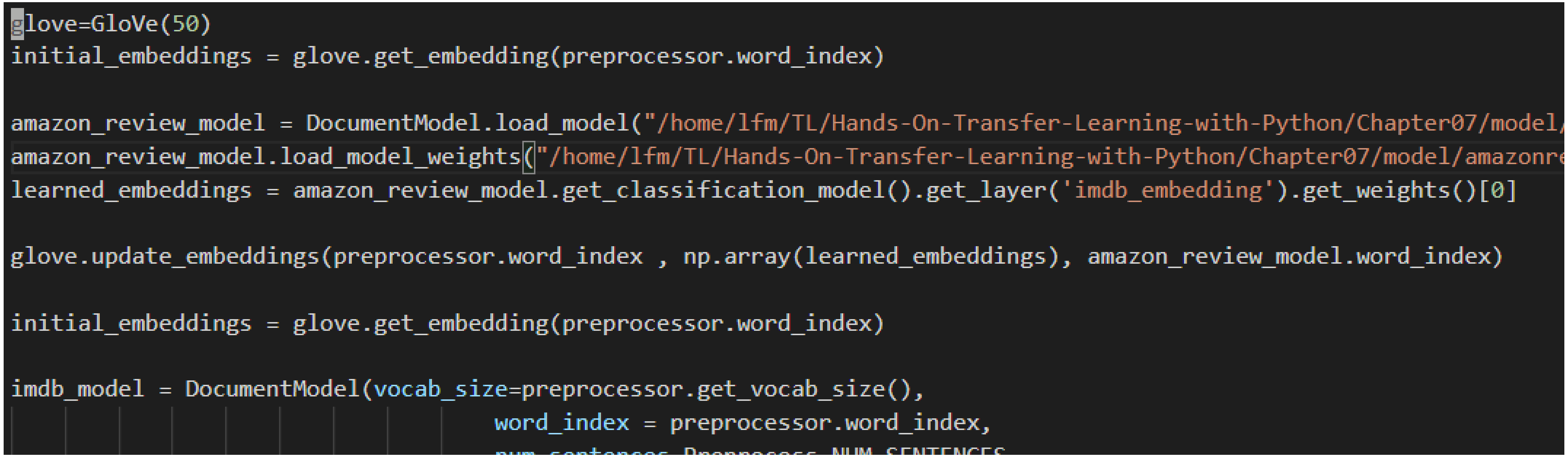
- 샘플 20만개 학습



```
Epoch 00031: val_loss improved from 0.17975 to 0.17930, saving model to /home/lfm
/TL/Hands-On-Transfer-Learning-with-Python/Chapter07/model/amazonreviews/model_06
.hdf5
Epoch 32/35
 - 134s - loss: 0.1570 - acc: 0.9419 - val_loss: 0.1869 - val_acc: 0.9322

Epoch 00032: val_loss did not improve from 0.17930
Epoch 33/35
 - 136s - loss: 0.1570 - acc: 0.9417 - val_loss: 0.1846 - val_acc: 0.9313

Epoch 00033: val_loss did not improve from 0.17930
Epoch 34/35
 - 133s - loss: 0.1559 - acc: 0.9423 - val_loss: 0.1876 - val_acc: 0.9297

Epoch 00034: val_loss did not improve from 0.17930
Epoch 35/35
 - 131s - loss: 0.1552 - acc: 0.9426 - val_loss: 0.1865 - val_acc: 0.9328

Epoch 00035: val_loss did not improve from 0.17930
(elmoenv) lfm@lfm-System-Product-Name:~/TL/Hands-On-Transfer-Learning-with-Python
/Chapter07$
```

아마존->IMDB 트랜스퍼 러닝

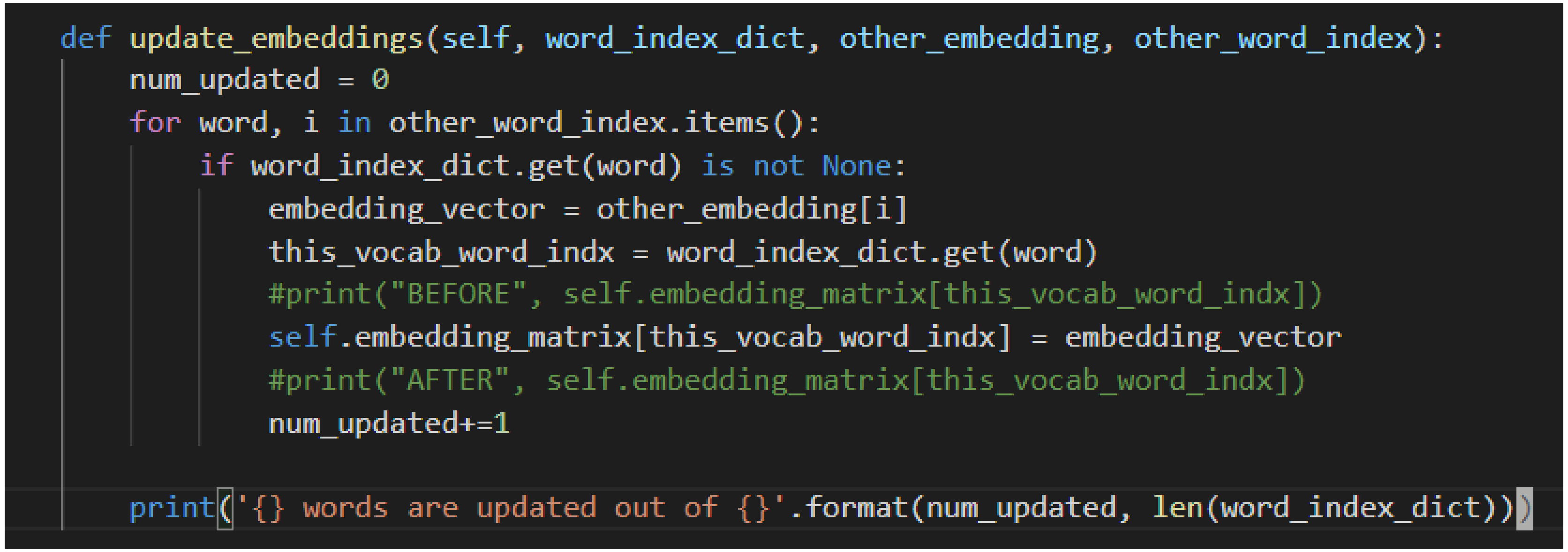
- 구매평으로 업데이트된 Glove 임베딩 + 1,250개 데이터 학습



```
glove=GloVe(50)
initial_embeddings = glove.get_embedding(preprocessor.word_index)

amazon_review_model = DocumentModel.load_model("/home/lfm/TL/Hands-On-Transfer-Learning-with-Python/Chapter07/model/
amazon_review_model.load_model_weights("/home/lfm/TL/Hands-On-Transfer-Learning-with-Python/Chapter07/model/amazonre
learned_embeddings = amazon_review_model.get_classification_model().get_layer('imdb_embedding').get_weights()[0]

glove.update_embeddings(preprocessor.word_index , np.array(learned_embeddings), amazon_review_model.word_index)

initial_embeddings = glove.get_embedding(preprocessor.word_index)

imdb_model = DocumentModel(vocab_size=preprocessor.get_vocab_size(),
                                    word_index = preprocessor.word_index,
                                    num_sentences=Preprocess.NUM_SENTENCES
```

아마존->IMDB 트랜스퍼 러닝

- 구매평으로 업데이트된 Glove 임베딩 + 1,250개 데이터 학습

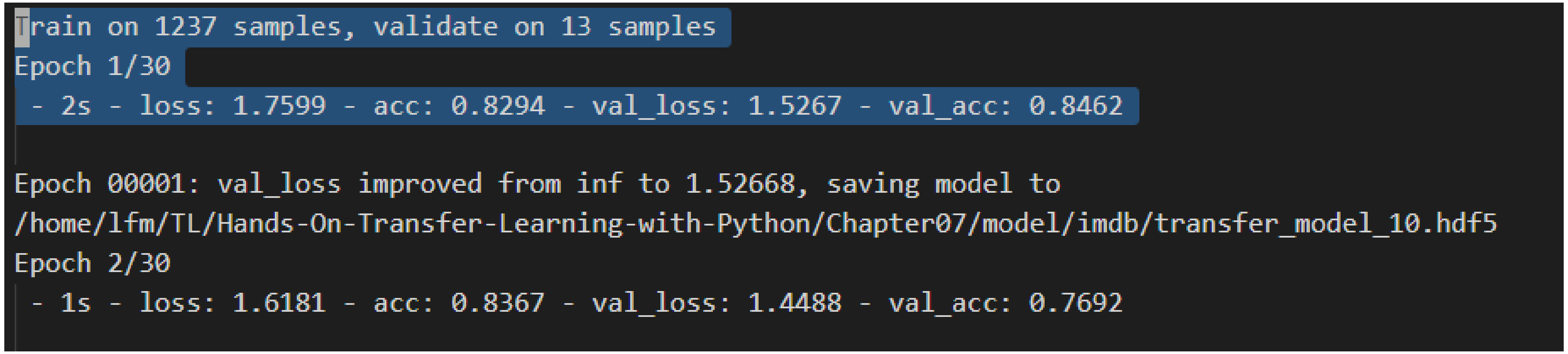


```python
def update_embeddings(self, word_index_dict, other_embedding, other_word_index):
    num_updated = 0
    for word, i in other_word_index.items():
        if word_index_dict.get(word) is not None:
            embedding_vector = other_embedding[i]
            this_vocab_word_indx = word_index_dict.get(word)
            #print("BEFORE", self.embedding_matrix[this_vocab_word_indx])
            self.embedding_matrix[this_vocab_word_indx] = embedding_vector
            #print("AFTER", self.embedding_matrix[this_vocab_word_indx])
            num_updated+=1

    print('{} words are updated out of {}'.format(num_updated, len(word_index_dict)))
```

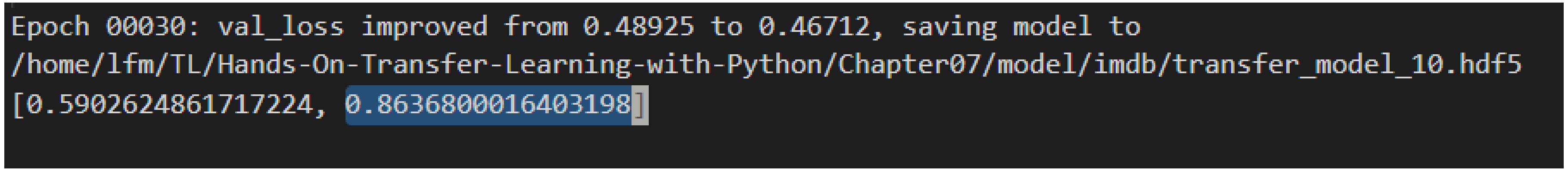
아마존->IMDB 트랜스퍼 러닝

- 구매평으로 업데이트된 Glove 임베딩 + 1,250개 데이터 학습

- 86.3%!



```
Train on 1237 samples, validate on 13 samples
Epoch 1/30
 - 2s - loss: 1.7599 - acc: 0.8294 - val_loss: 1.5267 - val_acc: 0.8462

Epoch 00001: val_loss improved from inf to 1.52668, saving model to
/home/lfm/TL/Hands-On-Transfer-Learning-with-Python/Chapter07/model/imdb/transfer_model_10.hdf5
Epoch 2/30
 - 1s - loss: 1.6181 - acc: 0.8367 - val_loss: 1.4488 - val_acc: 0.7692
```

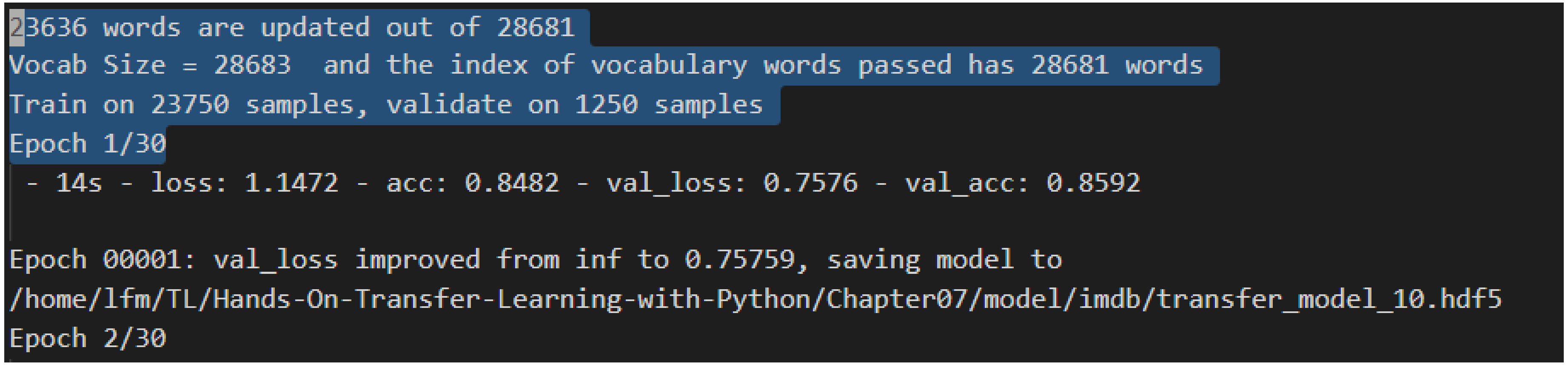


```
Epoch 00030: val_loss improved from 0.48925 to 0.46712, saving model to
/home/lfm/TL/Hands-On-Transfer-Learning-with-Python/Chapter07/model/imdb/transfer_model_10.hdf5
[0.5902624861717224, 0.8636800016403198]
```

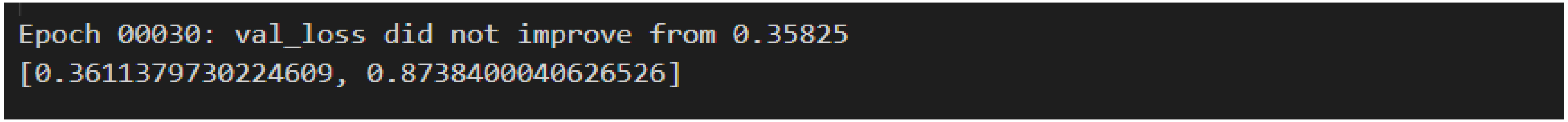
아마존->IMDB 트랜스퍼 러닝

- 업데이트된 임베딩 + 25,000개 데이터 학습

- 87.3%!!



```
23636 words are updated out of 28681
Vocab Size = 28683  and the index of vocabulary words passed has 28681 words
Train on 23750 samples, validate on 1250 samples
Epoch 1/30
 - 14s - loss: 1.1472 - acc: 0.8482 - val_loss: 0.7576 - val_acc: 0.8592

Epoch 00001: val_loss improved from inf to 0.75759, saving model to
/home/lfm/TL/Hands-On-Transfer-Learning-with-Python/Chapter07/model/imdb/transfer_model_10.hdf5
Epoch 2/30
```



```
Epoch 00030: val_loss did not improve from 0.35825
[0.3611379730224609, 0.8738400040626526]
```

트랜스퍼 러닝 결과

IMDB 5%	IMDB 100%	AMAZON ->IMDB(5%)	AMAZON ->IMDB (100%)
70%	83%	**86.3%**	**87.3%**

목차

1. 트랜스퍼 러닝

2. 워드 임베딩/문서 분류 모델

3. 트랜스퍼 러닝 예제

4. **요약 및 결론**

4. 요약 및 결론

학습에 필요한 데이터가 부족하거나,

성능 향상이 필요할 때,

트랜스퍼러닝 고려해볼 수도 있다.