단어 임베딩(Embedding)

박경규

Look Dick Look look at Jane
see Jane laugh and play
이봐 딕, 제인 좀 봐
즐겁게 놀고 있는 제인을 보라구

문장 토큰화(Tokenizing)

Look Dick Look look at Jane see Jane laugh and play

Look Dick Look see pretty Jane

look dick look look at jane see jane laugh and play

look dick look see pretty jane

단어사전(Vocab) 만들기

look dick look look at jane see jane laugh and play

look dick look see pretty jane

단어사전(Vocab)	
look	**0**
dick	**1**
at	**2**
jane	**3**
see	**4**
laugh	**5**
and	**6**
play	**7**
pretty	**8**

단어를 숫자로 치환

look	dick	look	look	at	jane	see	jane	laugh	and	play
0	1	0	0	2	3	4	3	5	6	7

단어사전(Vocab)	
look	0
dick	1
at	2
jane	3
see	4
laugh	5
and	6
play	7
pretty	8

[0, 1, 0 , 0, 2, 3, 4, 3, 5, 6]

입력 데이터 길이 맞추기

딥러닝 모델에 입력으로 사용하기 위해서는 길이를 같게 해 주어야 합니다.

Look Dick Look look at Jane see Jane laugh and play

Look Dick Look see pretty Jane

[0, 1, 0 , 0, 2, 3, 4, 3, 5, 6]
[0, 1, 0 , 4, 8, 3, 0, 0, 0, 0]

단어의 차원

너무 큰 차원의 데이터를 학습할 때, 차원의 저주(curse of dimensionality)에 빠질 수 있습니다.

look	dick	look	look	at	jane	see	jane	laugh	and	play
0	1	0	0	2	3	4	3	5	6	7

look	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]
dick	[0, **1**, 0, 0, 0, 0, 0, 0, 0]
look	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]
look	[**1**, 0, 0, 0, 0, 0, 0, 0, 0]
at	[0, 0, **1**, 0, 0, 0, 0, 0, 0]
jane	[0, 0, 0, **1**, 0, 0, 0, 0, 0]
see	[0, 0, 0, 0, **1**, 0, 0, 0, 0]
Jane	[0, 0, 0, **1**, 0, 0, 0, 0, 0]
laugh	[0, 0, 0, 0, 0, **1**, 0, 0, 0]
and	[0, 0, 0, 0, 0, 0, **1**, 0, 0]
play	[0, 0, 0, 0, 0, 0, 0, **1**, 0]

실제 Corpus에서는 단어가 많으므로,
단어의 차원이 1,000차원 이상이 됩니다.

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0 0, 0, 0, 0, 0, 0, 0, 0 0, 0, 0, 0, 0, 0, 0,
0 0, 0, 0, 0, 0, 0, 0, 0………………………………
……………………………………………………………
……………………………………………………………
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Tokenizer

Tokenizer의 fit_on_texts로 학습할 문장에 대하여 토큰화를 진행합니다.

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding,


vocab_size = 1000
oov_token  = "<OOV>"

tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_token)
tokenizer.fit_on_texts(train_sentences)
```

Embedding

Embedding은 큰 차원을 줄여주는 역할을 하며, Sparsity문제를 해결합니다.

```
vocab_size = 1000
embedding_dim  = 16

Embedding(vocab_size, embedding_dim, input_length=max_length)
```

```
<tf.Tensor: shape=(16,), dtype=float32,
numpy= array([-0.01208562, -0.02042891, 0.00930309, -
0.01072387, 0.00621265, -0.03476477, 0.02996606, 0.02715156, -
0.04755617, -0.03230421, -0.00678114, 0.04859651, 0.03986677,
0.01091999, -0.03999164, -0.01312722], dtype=float32)>
```

Thank you

kgpark88@gmail.com