

Natural Language Processing & Word Embeddings

Word representation

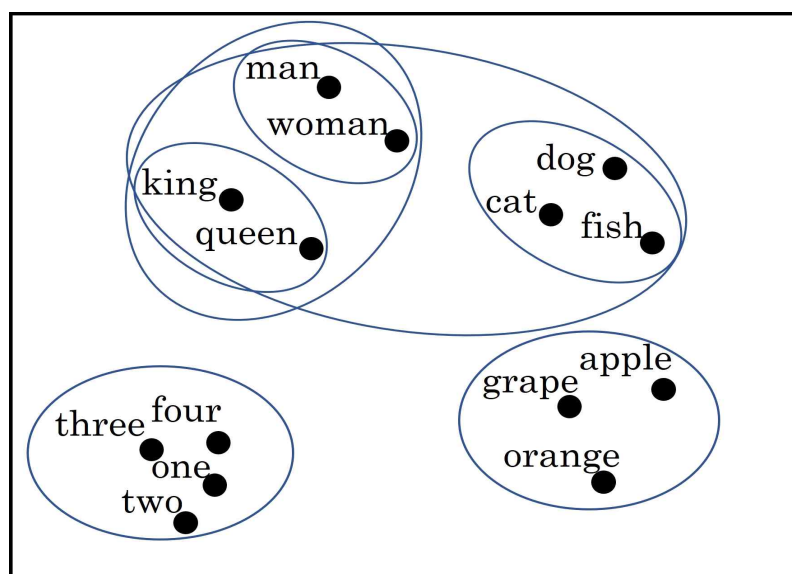
| | | | | | | |
|---|--|---|--|--|---|--|
| $V = [a, aaron, \dots, zulu, \langle \text{UNK} \rangle]$ | | | | | | |
| 1-hot representation | | | | | | |
| $\begin{matrix} \text{Man} \\ (5391) \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \\ O_{5391} \end{matrix}$ | $\begin{matrix} \text{Woman} \\ (9853) \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \\ O_{9853} \end{matrix}$ | $\begin{matrix} \text{King} \\ (4914) \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \\ O_{4914} \end{matrix}$ | $\begin{matrix} \text{Queen} \\ (7157) \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \\ O_{7157} \end{matrix}$ | $\begin{matrix} \text{Apple} \\ (456) \\ \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ O_{456} \end{matrix}$ | $\begin{matrix} \text{Orange} \\ (6257) \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \\ O_{6257} \end{matrix}$ | <p>I want glass of orange ____ .</p> <p>I want glass of apple ____ .</p> |

Featurized representation: Word Embedding

| | Man (5391) | Woman (9853) | King (4914) | Queen (7157) | Apple (456) | Orange (6257) |
|----------|---------------|-----------------|----------------|-----------------|----------------|------------------|
| Gender | -1 | 1 | -0.95 | 0.97 | 0.00 | 0.01 |
| Royal | 0.01 | 0.02 | 0.93 | 0.95 | -0.01 | 0.00 |
| Age | 0.03 | 0.02 | 0.7 | 0.69 | 0.03 | -0.02 |
| Food | 0.09 | 0.01 | 0.02 | 0.01 | 0.95 | 0.97 |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| | e_{5391} | e_{9853} | e_{4914} | e_{7157} | e_{456} | e_{6257} |

Visualizing word embedding

[van der Maaten and Hinton., 2008. Visualizing data using t-SNE]



t-SNE: A non-linear dimensionality reduction technique

for word embeddings, actually need BRNN, not just unidirectional RNN.

Transfer learning and word embeddings

1. Learn word embeddings from large text corpus.(1–100B words)
(Or download pre-trained embedding online.)
2. Transfer embedding to new task with smaller training set.
(say, 100k words)
3. Optional: Continue to finetune the word embeddings with new data.

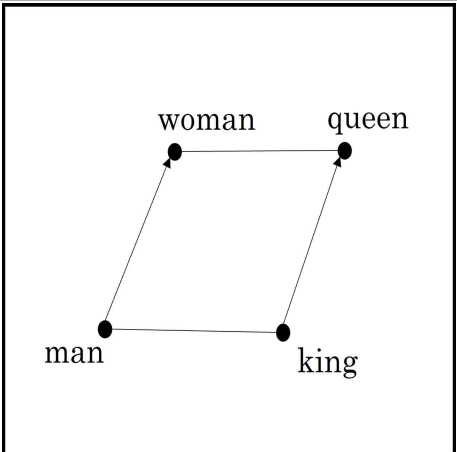
Properties of word embeddings

Analogies

[Mikolovet. al., 2013, Linguistic regularities in continuous space word representations]

| | |
|---|--|
| Man → Woman as King → ? | |
| $e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ | $e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ |
| $e_{man} - e_{woman} \approx e_{king} - e_{?} \rightarrow ? = \text{queen}$ | |

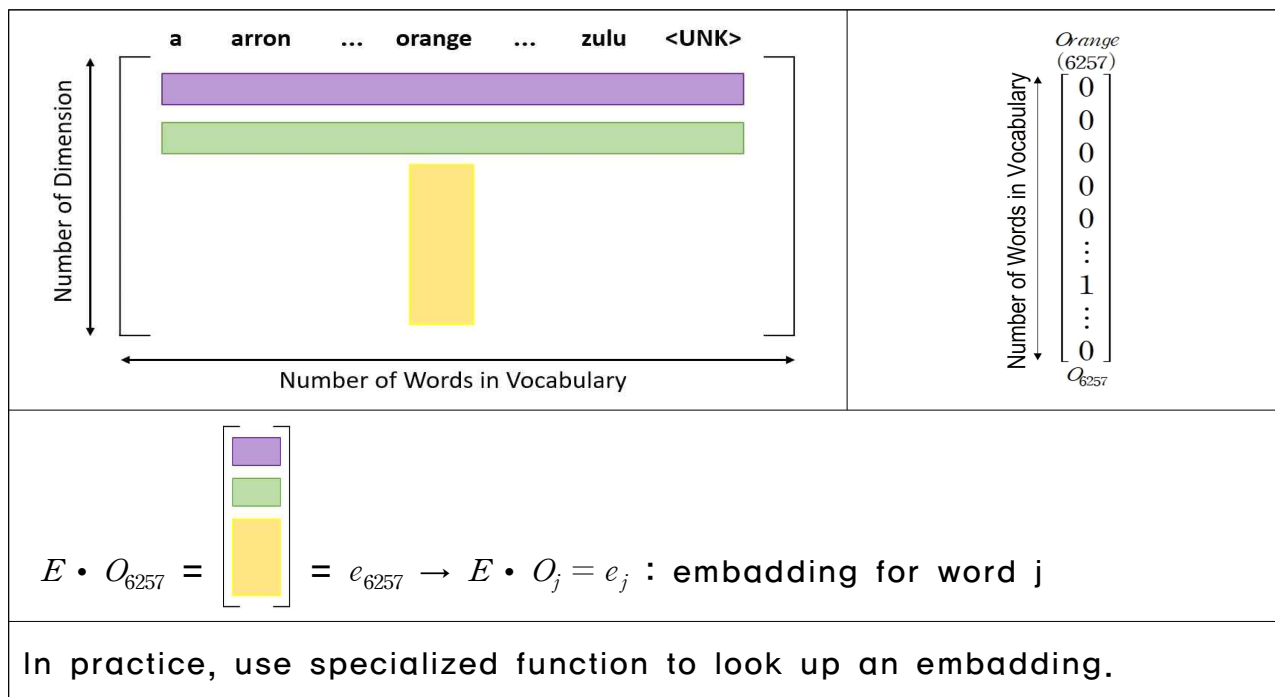
Analogies using word vectors

| | |
|---|--|
|  | $e_{man} - e_{woman} \approx e_{king} - e_{?}$ <p>find word w: $\operatorname{argmax}_w \operatorname{sim}(e_w, e_{king} - e_{man} + e_{woman})$</p> |
|---|--|

cosine similarity

$$\operatorname{sim}(e_w, e_{king} - e_{man} + e_{woman}) \rightarrow \operatorname{sim}(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

Embedding matrix



Learning word embeddings

Neural language model

[Bengio et. al., 2003, A neural probabilistic language model]

e_j 를 linear model과 activation function에 넣는다.

Context/Target pairs

Context: Last 4 words
 4 words on left & right
 Last 1 word
 Nearby 1 word \rightarrow Skip-grams

a glass of orange __?__ to go along with
 orange __?__
 glass of orange __?__에서 glass __?__

Word2Vec

Skip-grams

[Mikolov et. al., 2013. Efficient estimation of word representations in vector space.]

I want a glass of orange juice to go along with my cereal.

| Context | Target |
|---------|--------|
| orange | juice |
| orange | glass |
| orange | my |

Model

Voab size = 10,000k

Content $c \rightarrow$ Target t

$$O_c \rightarrow E \rightarrow e_c = E \cdot O_c \rightarrow \text{Softmax} \rightarrow \hat{y}$$

$$\text{Softmax: } p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{100000} e^{\theta_j^T e_c}}, \theta_t = \text{parameter associated with output}$$

$$y = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

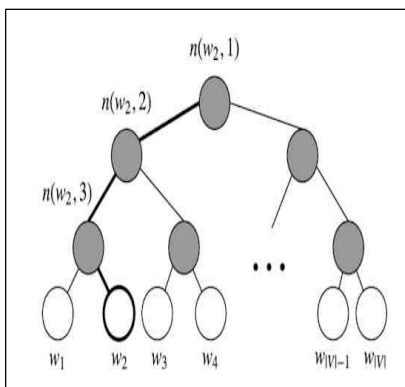
$$L(\hat{y}, y) = - \sum_{i=1}^{10000} y_i \log \hat{y}_i$$

Problems with softmax classification

$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{100000} e^{\theta_j^T e_c}}: \text{softmax는 데이터가 많을수록 분모 계산에 많은 시간이 소요된다.}$$

해결 방법: Hierarchical softmax classification

만 개의 행으로 분류하는 대신 한 번에 만 개의 행으로 분류하는 것



실제는 트리 모양이 대칭이지 않고
흔하게 사용하는 단어는 상위에 있고 덜 흔한 단어는 밑에 있다.

새롭게 나타나는 문제: 자주 사용하는 단어를 더 쉽게 접근할 수 있어

의미 없는 단어를 많이 본다. ex) the, of, a, and, too 등이 있다.

How to sample the context c ?

Negative sampling

[Mikolovet. al., 2013.

Distributed representation of words and phrases and their compositionality]

| I want a glass of orange juice to go along with my cereal. | | | |
|--|---------------|-----------------|--------------------|
| x | | y | |
| Context c | Word t | Target y | |
| orange | juice | 1 | Positive sample |
| orange | king | 0 | K Negative samples |
| orange | book | 0 | |
| orange | the | 0 | |
| orange | of | 0 | |

⇒ K+1 binary classification problem: 0 or 1

K: negative sample의 개수

K = 5 ~ 20: smaller datasets

K = 2 ~ 5: larger datasets

negative sampling은 random word in dictionary

positive와 negative word의 distribution을 구별하도록 sampling한다.

$$\text{Softmax: } p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{100000} e^{\theta_j^T e_c}}, \text{ \# words in vocabulary} = 10,000$$

$$p(y=1|c, t) = \sigma(\theta_t^T, e_c), \text{ t: Target, c: Context}$$

Selecting negative examples

중간 부분에 있는 단어들의 표본을 채취하는 방법이 있다. 문장에 대한 경험적 빈도에 따라 표본을 내는 것이다. 얼마나 자주 나타나는지에 따라 표본을 채취한다. 하지만 문제는 the, of, and,... 와 같은 단어들이 많이 나오는 것이다.

이 문제를 저자는 Softmax를 보정하여 해결했다. 정확히 해결 가능한지는 확실치 않다.

$$p(w_i) = \frac{f(w_i)^{\frac{3}{4}}}{\sum_{j=1}^{10000} f(w_j)^{\frac{3}{4}}}, \text{ \# words in vocabulary} = 10,000$$

GloVe(global vectors for word representation)

GloVe word vectors

[Pennington et. al., 2014. GloVe: Global vectors for word representation]

| |
|--|
| I want a glass of orange juice to go along with my cereal. |
| c, t $X_{ij} = \# \text{ times } j \text{ appears in content of } i, i \in c. j \in t$ $X_{ij} = X_{ji}$ |

Model

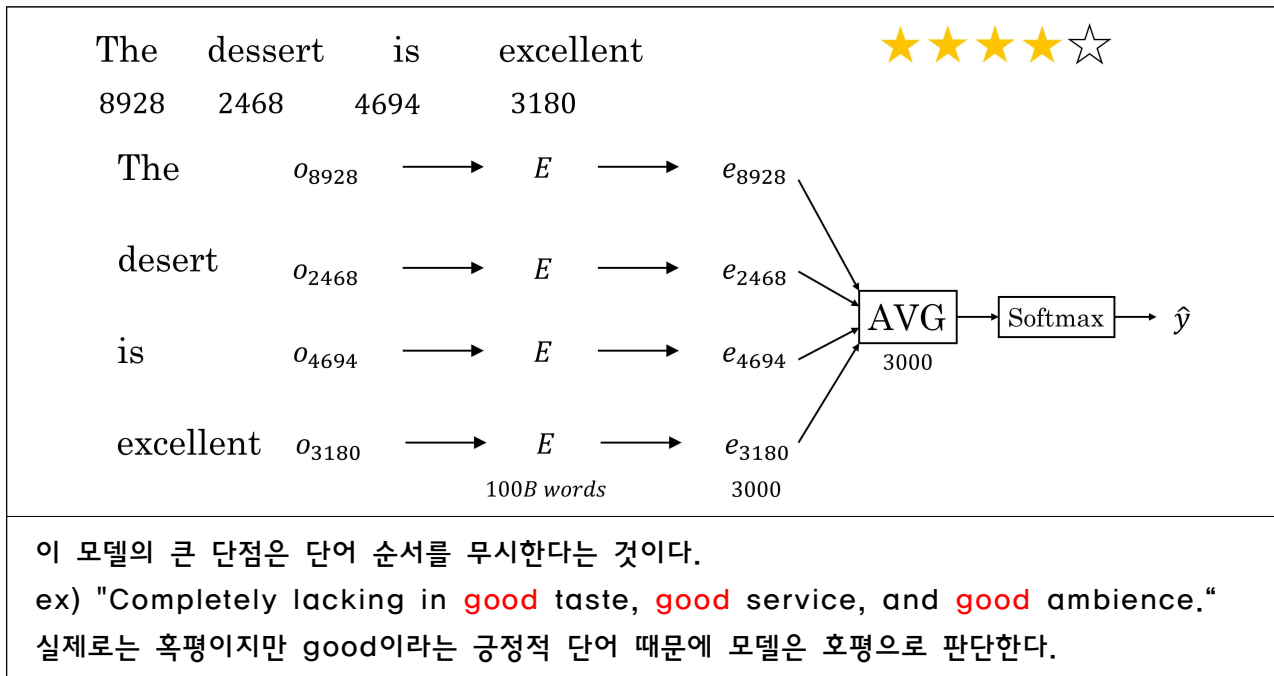
| |
|--|
| $\text{minimize } \sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij})(\theta_i^T e_j + b_i + b_j' - \log X_{ij})^2,$ $i \in c. j \in t, \# \text{ words in vocabulary} = 10,000$ |
| $\log X_{ij}$; how related i and j. i와 j가 얼마나 자주 서로에게 발생하는지 측정 |
| $f(X_{ij})$: weighting term. $X_{ij} = 0$ 이 돼서 $\log X_{ij} = -\infty$ 되는 것을 방지. $0 \log 0 = 0$ this, is, of, a와 같이 지나치게 많이 나오는 단어는 $f(X_{ij})$ 가 작고 durian처럼 많이 나오지 않는 단어는 $f(X_{ij})$ 가 비교적 크다. |

특징

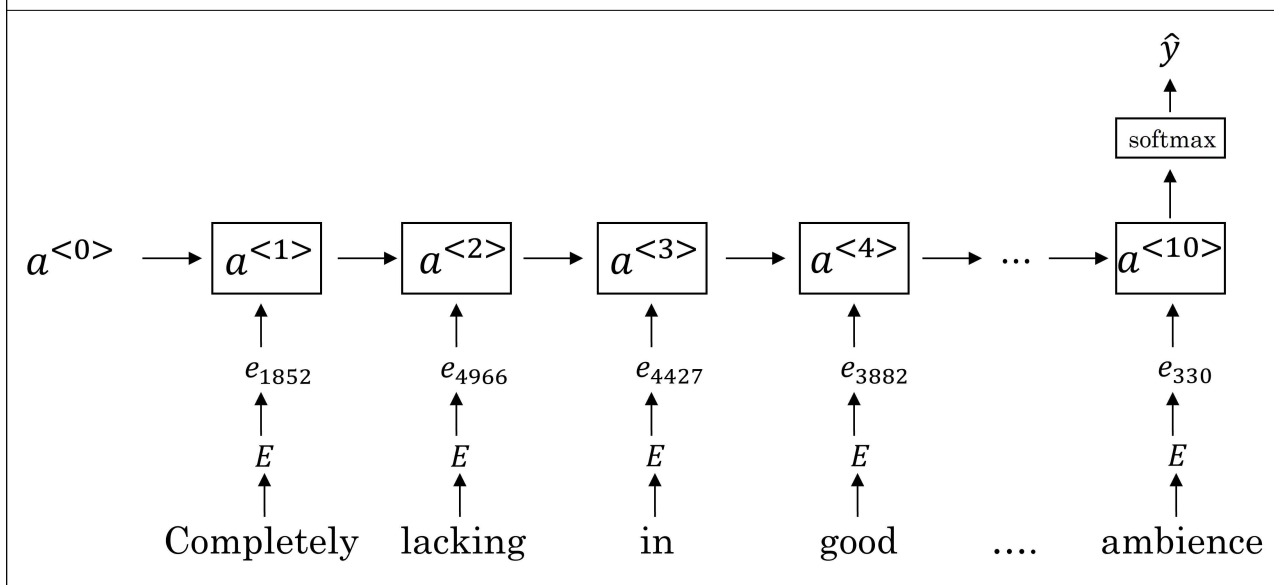
| |
|---|
| 1. θ_i, e_j are symmetric(대칭). $\Rightarrow e_w^{(final)} = \frac{e_w + \theta_w}{2}$ |
| 2. $\sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij})(\theta_i^T e_j + b_i + b_j' - \log X_{ij})^2$ 에서 $(A\theta_i)^T (A^{-T}e_j) = \theta_i^T A^T A^{-T}e_j = \theta_i^T e_j$ 선형대수학적 연산으로 식이 간단해졌다. 하지만 기능을 나타내는 축을 쉽게 해석하기는 어렵다. 어느 특성을 사용했고 연관되어 있는지 알 수 없다. 이러한 선형 연산에도 불구하고 parallelogram map은 여전히 쓸 수 있다. feature의 임의 선형 변형에도 불구하고 figure analogies에 대한 parallelogram 학습은 작동한다. |

Sentiment classification

리뷰를 보고 별점을 매기는 모델과 같다.(Many-to-One)



해결 방법: RNN for sentiment classification



Debiasing word embeddings

[Bolukbasi et. al., 2016. Man is to computer programmer

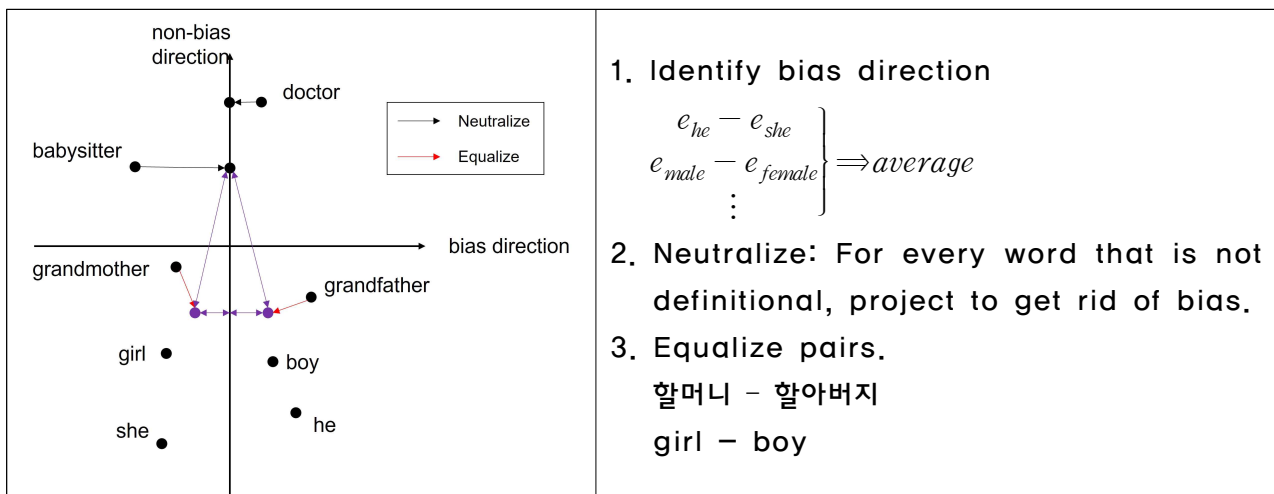
as woman is to homemaker? Debiasing word embeddings]

The problem of bias in word embeddings

여기서 bias는 이전에 배웠던 bias와 variance 문제의 bias가 아니라 사회적 편견(bias)이다. 예로 들어 남자는 프로그래머 여자는 주부와 같은 편견을 말한다.

Word embeddings can reflect gender, ethnicity, age, sexual orientation, and other biases of the text used to train the model.

Addressing bias in word embedding



가로 방향은 편견과 관련 있고 (편향 방향. bias)

세로 방향은 관련 없다. (비편향 방향. non-bias)

bias direction은 1차원보다 높을 수 있다.

사실 average보다는 SVU(singular value decomposition, 특이값 분해)라 불리는 더 복잡한 알고리즘을 사용한다.

SVU uses ideas similar to the pc(the principle component) analysis algorithm

의사나, 베이비시터와 같은 중성 단어의 bias를 없앤다.

그러나 할머니 할아버지, boy와 girl 같은 단어는 아니다. 단어 자체에 성이 들어있기 때문이다.

선형대수학적 특징은 할머니와 할아버지의 non-bias direction 축과의 거리는 같고 bias를 없앤 의사의 좌표와도 거리가 같다.

Neutralize 할 단어 선택 방법:

수염은 사실 중성적 단어지만 통계적으로 여자보다는 남자에게 많고 더 가깝다. 이런 단어들에 대해 저자는 어떤 단어가 definitional 한지, 어떤 단어가 성별과 관련되었는지, 어떤 단어가 그렇지 않은지에 대해 classifier를 훈련했다. 그 결과 영어의 대부분 단어가 definitional 하지 않다는 것이 밝혀졌고 대부분 단어는 성별이 definition에 속하지 않았다. 그래서 pair를 맞춰야 하는 단어들도 사실 적다.