Language Model and Word Embeddings

Seoul National University

Saerom Park

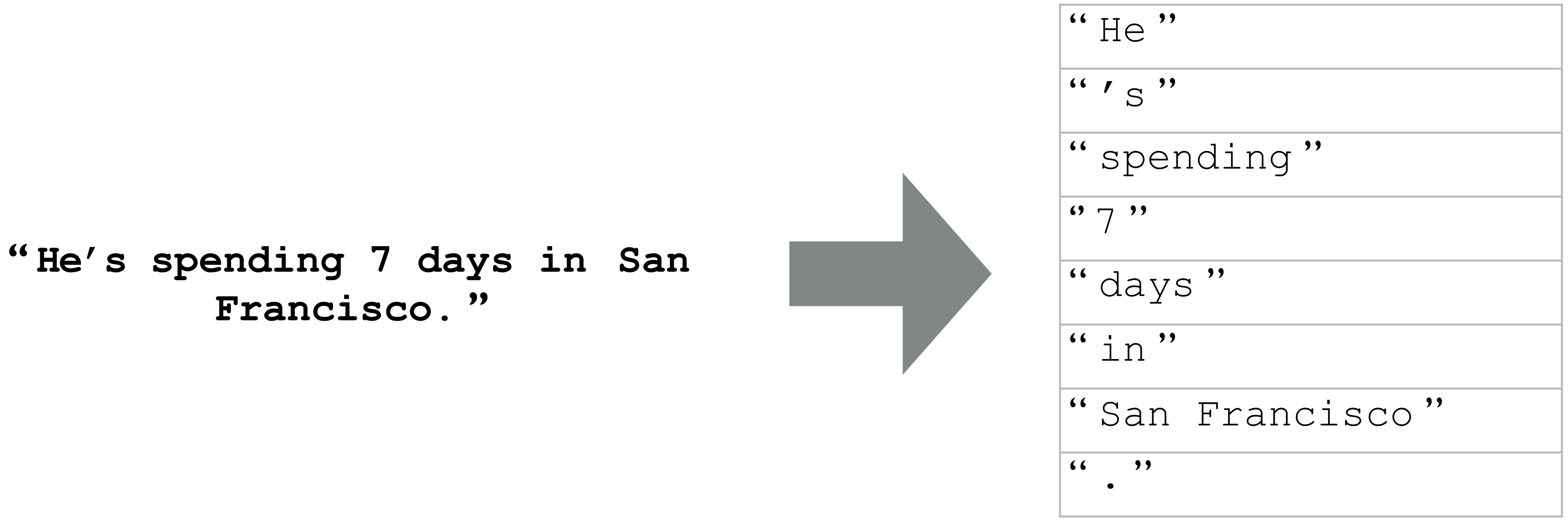
drsaerompark@gmail.com

Lecture Overview

- Preprocessing
- Word embeddings

Tokenization

- Typical preprocessing steps of text data
  - Tokenize text (from a long string to a list of token strings)



  - For many datasets, this has already been done for you
  - Splitting into tokens based on spaces and separating punctuation is good enough in English or French

Lemmatization

- **Lemmatize tokens**
  - Put into standard form

“He”	“he”
“’s”	“be”
“spending”	“spend”
“7”	”NUMBER”
“days”	“day”
“in”	“in”
“San Francisco”	“San Francisco”
“.”	“.”

  - The specific lemmatization will depend on the problem we want to solve
    - ✓ we can remove variations of words that are not relevant to the task at hand

vocabulary

- word to unique ID
  - First, construct dictionary (vocabulary)
  - Maps lemmatized words to a unique ID (position of word in dictionary)
- Selection of vocabulary
  - Pick most frequent words
  - Ignore uninformative words from a user-defined short list
    - ✓ ex. "the", "a", etc.
- All words not in the vocabulary will be mapped to a special "out-of-vocabulary" ID

vocabulary

- Example

Vocabulary

Word	*w*
“the”	1
“and”	2
“dog”	3
“.”	4
“OOV”	5

“the”
“cat”
“and”
”the”
“dog”
“play”
“.”

$\longrightarrow$

1
5
2
1
3
5
4

One-hot Encoding

- From its word ID, we get a basic representation of.a word through the one-hot encoding of the ID
  - The one-hot vector of an ID is a vector filled with 0s, except for a 1 at the position associated with the ID
    - ✓ Ex: for vocabulary size D=10, the one-hot vector of word ID w=4 is
      - ✓ e(w) = [0 0 0 1 0 0 0 0 0 0]
  - A one-hot encoding makes no assumption about word similarity
    - ✓ 두 개의 단어가 같으면 거리가 = 0
    - ✓ 두 개의 단어가 서로 다르면 거리가 무조건 = 2

- 단점
  - Word similarity 를 잘 나타내지 못함
  - One-hot representation has very high-dimension

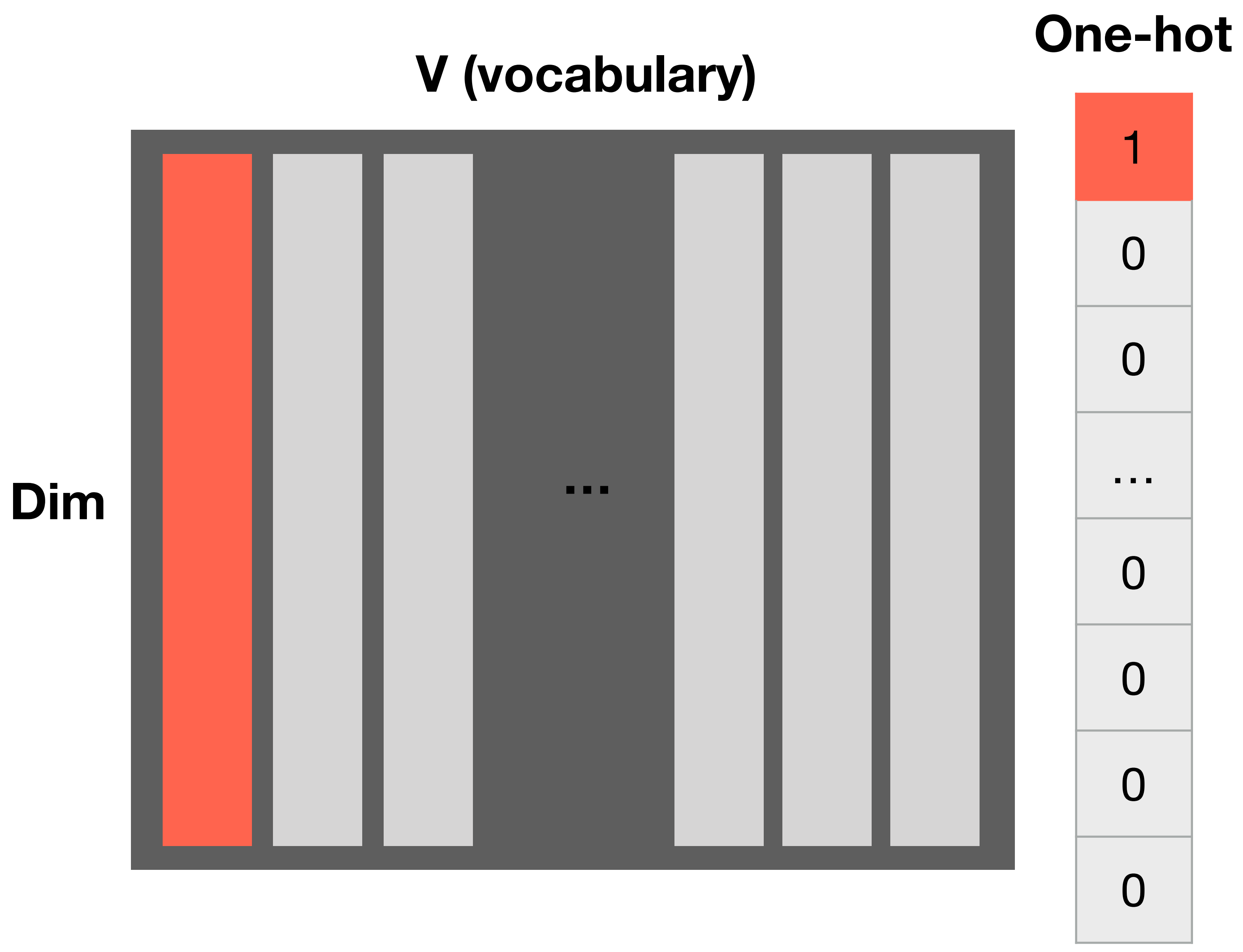
Word Embeddings

- Learn a continuous representation of words
  - 즉 각 단어의 representation 을 학습가능한 파라미터로 생각하여 학습함
  - 앞에서 언급한 one-hot encoding 의 한계 극복 가능

Word	*w*	*C*(*w*)
“ the ”	1	[0.6762, -0.9607, 0.3626, -0.2410, 0.6636]
“ a ”	2	[0.6859, -0.9266, 0.3777, -0.2140, 0.6711]
“ have	3	[0.1656, -0.1530, 0.0310, -0.3321, -0.1342]
“ be ”	4	[0.1760, -0.1340, 0.0702, -0.2981, -0.1111]
“ cat ”	5	[0.5896, 0.9137, 0.0452, 0.7603, -0.6541]
“ dog ”	6	[0.5965, 0.9143, 0.0899, 0.7702, -0.6392]
“ car ”	7	[-0.0069, 0.7995, 0.6433, 0.2898, 0.6359]
...	...	...

- How to do it?
  - word embedding 은 one-hot encoding 을 input 으로 하는 NN의 weight matrix 학습하는 것과 같음

Word Embeddings



One-hot
V (vocabulary)
Dim
...
1
0
0
...
0
0
0
0