20기 정규세션

ToBig's 19기 강의자
윤세휘

Week 7:
NLP Basic

Contents

Unit 01 | NLP Introduction

자연어 (natural language) : 컴퓨터에서 사용하는 프로그램 작성 언어 또는 기계어와 구분하기 위해 인간이 일상생활에서 의사 소통을 위해 사용하는 언어를 가리키는 말.

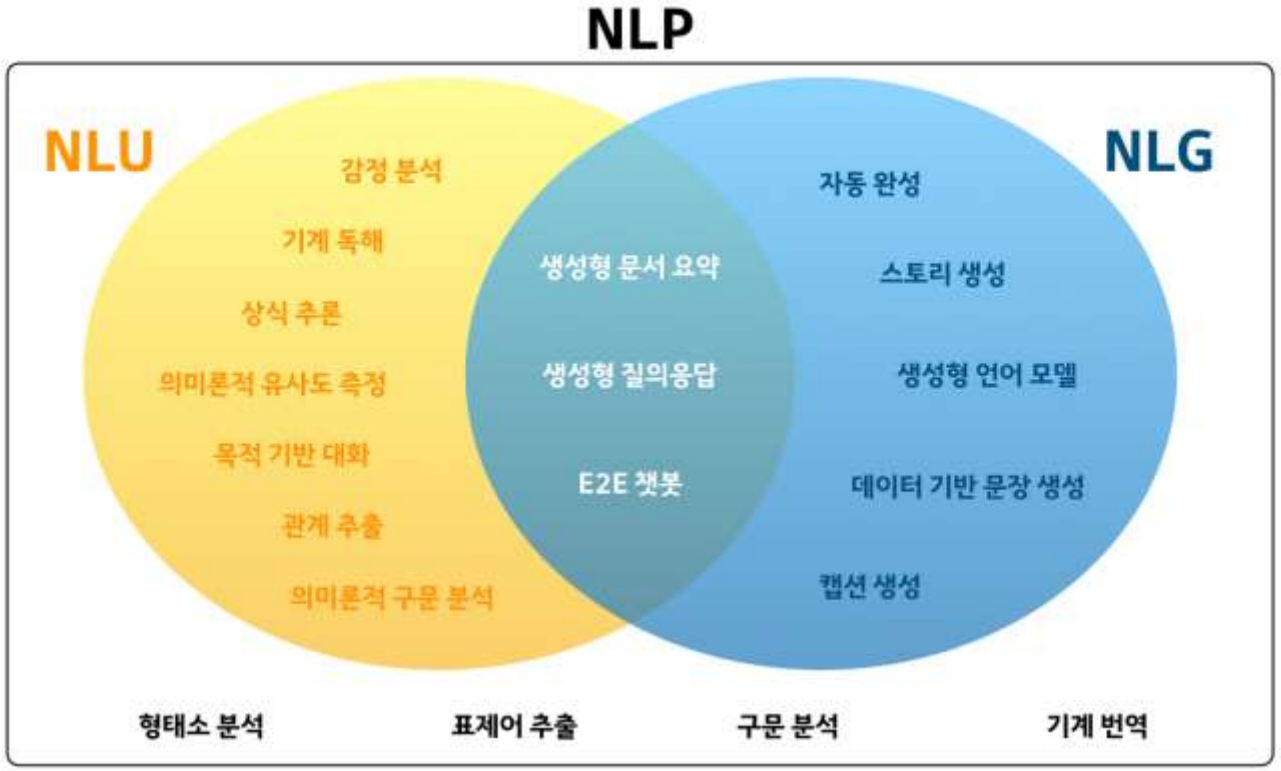
자연어 처리 (natural language processing) : 컴퓨터가 자연어를 처리하는 일.

<대표적인 NLP 태스크>

- 음성 인식 Speech Recognition
- 번역 Translation
- 요약 Text Summarization
- 분류 Text Classification

Unit 01 | NLP Introduction

자연어 이해 (Natural Language Understanding;NLU) : 자연어의 의미를 모델이 이해하도록 하는 것
자연어 생성 (Natural Language Generation;NLG) : 자연어를 모델이 생성하도록 하는 것



<텍스트 전처리 과정>

- **토큰화 (tokenization) : 텍스트를 특정 기준에 맞게 나누는 작업. 주로 단어 단위로 나누는 단어 토큰화(word tokenization)를 의미한다.**

- **단어 임베딩 (word embedding) : 토큰화 후, 토큰들을 모델이 이해할 수 있도록 수치화하는 작업.**

Contents

<토큰화 (Tokenization) 방법>

- 토큰화 라이브러리

- NLTK : 교육용으로 개발된 자연어 처리 및 문서 분석용 파이썬 패키지
  - 토큰화, 품사태깅, stemming(어간추출) 등의 기능 제공
- spaCy : 자연어 처리를 위한 Python 기반의 오픈 소스 라이브러리
  - 토큰화, 품사태깅, 구문분석 등 다양한 기능 제공

- KoNLPy : 한글 형태소 분석 라이브러리
  - Hannanum
  - Kkma
  - Komoran
  - Mecab
  - Okt

● 토큰화 라이브러리

1) NLTK

```
from nltk.tokenize import word_tokenize
print('단어 토큰화1 :',word_tokenize("Don't be fooled by the dark sounding name, Mr. Jone's Orphanage is as cheery as cheery goes for a pastry shop."))
```

"**Don't** be fooled by the dark sounding name, Mr. **Jone's** Orphanage is as cheery as cheery goes for a pastry shop."

단어 토큰화1 : [**'Do', "n't",** 'be', 'fooled', 'by', 'the', 'dark', 'sounding', 'name', ',', 'Mr.', **'Jone', "'s"**, 'Orphanage', 'is', 'as', 'cheery', 'as', 'cheery', 'goes', 'for', 'a', 'pastry', 'shop', '.']

- **토큰화 라이브러리**

1) NLTK

```
from nltk.tokenize import WordPunctTokenizer
print('단어 토큰화2 :',WordPunctTokenizer().tokenize("Don't be fooled by the dark sounding name, Mr. Jone's Orphanage is as cheery as cheery goes for a pastry shop."))
```

"**Don't** be fooled by the dark sounding name, Mr. **Jone's** Orphanage is as cheery as cheery goes for a pastry shop."

단어 토큰화2 :[**'Don', "'", 't**', 'be', 'fooled', 'by', 'the', 'dark', 'sounding', 'name', ',', 'Mr', '.', **'Jone', "'", 's**', 'Orphanage', 'is', 'as', 'cheery', 'as', 'cheery', 'goes', 'for', 'a', 'pastry', 'shop', '.']

● 토큰화 라이브러리

2) KoNLPy

- **Hannanum :** 띄어쓰기가 없는 문장은 분석 품질이 좋지 않다. 정제된 언어가 사용되지 않는 문서에 대한 형태소 분석 정확도가 높지 않은 문제점이 있다.

- **Komoran :** 여러 어절을 하나의 품사로 분석 가능함으로써 형태소 분석기의 적용 분야에 따라 공백이 포함된 고유명사(영화 제목, 음식점명, 노래 제목, 전문 용어 등)를 더 정확하게 분석할 수 있다.

- **Kkma :** 띄어쓰기 오류에 덜 민감한 한글 형태소 분석기이다. 분석시간이 Knlpy 중에서 가장 오래 걸린다. 정제된 언어가 사용되지 않는 문서에 대한 형태소 분석 정확도가 높지 않은 문제가 있다.

- **Mecab :** Okt 공개 전 띄어쓰기에서 가장 좋은 성능과 속도 정확도 모두 좋은 성능을 보여 주었다. 미등록어 처리, 동음이의어 처리에 한계가 있다.

- **Okt :** 미등록어 처리 문제점, 동음이의어 처리문제, 분석 범주(사용가능한 태그)가 다른 형태소 분석기들에 적음. 유일하게 stemming 기능을 제공한다.

Contents

Unit 03 | Word Embedding

<Word Embedding>

- 단어 임베딩 **(Word Embedding)**이란, 자연어처리에서 사람이 쓰는 자연어를 기계가 이해할 수 있도록 단어를 수치화하여 벡터로 바꾸는 과정이다. 단어 벡터화라고도 한다.

<Word Embedding 방법>

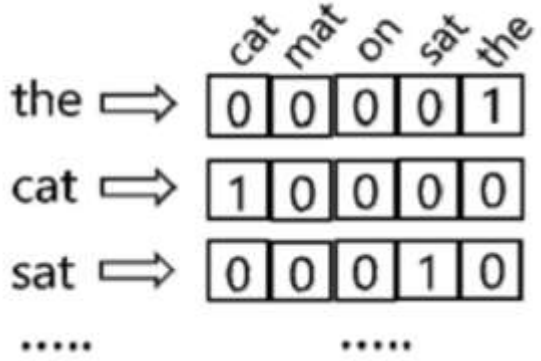
- One-hot encoding
- Word2Vec
- ELMo
- Transfer Learning

<Word Embedding 방법>

- **One-hot encoding** : 표현하고 싶은 단어의 인덱스에 1의 값을 부여하고, 다른 인덱스에는 0을 부여하는 단어의 벡터 표현 방식. 단어 집합의 크기가 벡터의 차원이 된다.

- 장점 : 가장 간단한 단어 임베딩 방법이다.
- 단점 : 단어 개수가 늘어남에 따라 벡터 차원이 커진다.
  맥락 정보를 반영하지 못한다.
  단어 간 유사도를 파악할 수 없다.

Example) The cat sat on the mat.

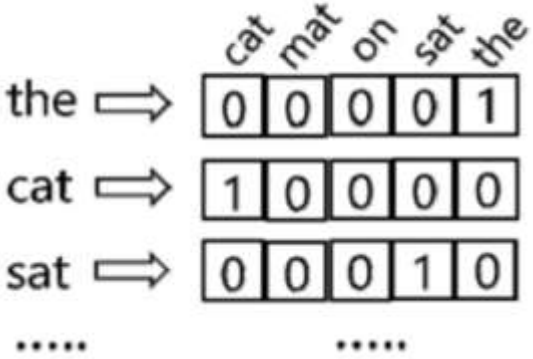


Unit 03 | Word Embedding

<Word Embedding 방법>

- **one-hot encoding**으로 얻은 **one-hot vector**는 단어의 대부분이 **0**으로 표현되는 희소 표현**(sparse representation)**이다.

Example) The cat sat on the mat.

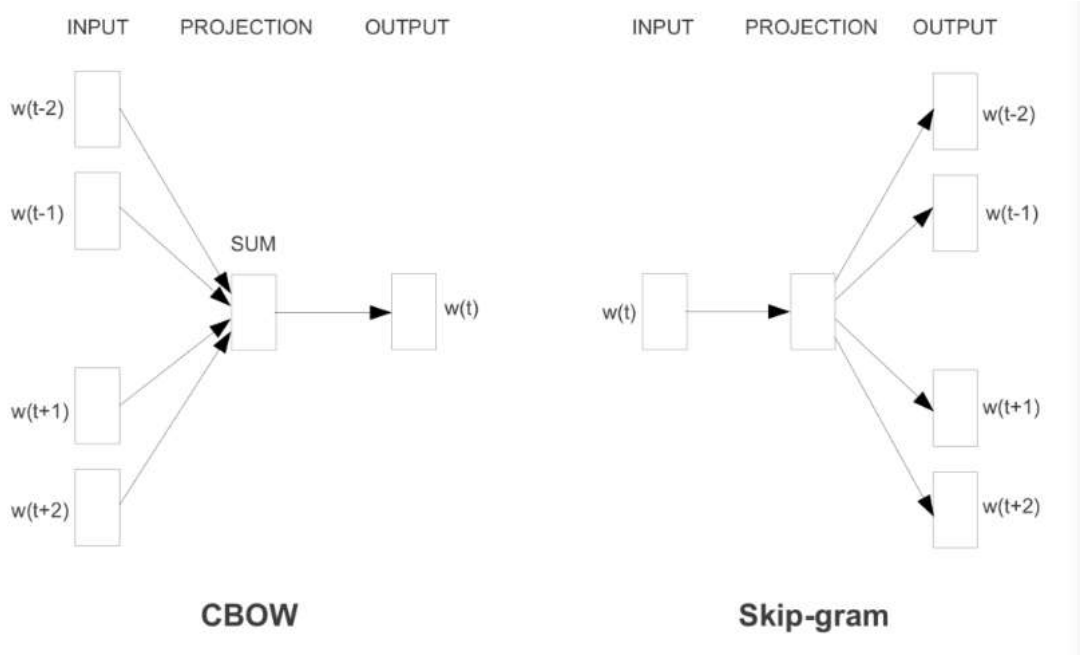


- 분산 표현**(distributed representation)**은 '비슷한 문맥에서 등장하는 단어들은 비슷한 의미를 가진다' 라는 분포가설을 가정하에 만들어진다.
- 분포가설에 따라서 텍스트의 단어들을 벡터화한다면 해당 단어 벡터들은 유사한 벡터값을 가지게 된다. 분산 표현은 분포 가설을 이용하여 텍스트를 학습하고, 단어의 의미를 벡터의 여러 차원에 분산하여 표현한다.
- 따라서 저차원의 벡터로 단어를 표현할 수 있게 된다.

<Word Embedding 방법>

- **Word2Vec :** 분산 표현의 대표적인 방법으로, **CBOW**와 **Skip-gram** 2가지 학습 방식으로 나뉜다.

- CBOW: 주변에 있는 단어들을 입력으로 중간에 있는 단어들을 예측
- Skip-Gram: 중간에 있는 단어들을 입력으로 주변 단어들을 예측



<Word Embedding 방법>

- **Word2Vec :** 분산 표현의 대표적인 방법으로, **CBOW**와 **Skip-gram** 2가지 학습 방식으로 나뉜다.

- CBOW: 주변에 있는 단어들을 입력으로 중간에 있는 단어들을 예측
- Skip-Gram: 중간에 있는 단어들을 입력으로 주변 단어들을 예측
- 전반적으로 Skip-gram이 CBOW보다 성능이 좋다고 알려짐.

CBOW

The fat cat ___ on the mat

중심 단어 / 주변 단어

중심 단어	주변 단어
The fat cat sat on the mat	[1, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0]
The fat cat sat on the mat	[0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0]
The fat cat sat on the mat	[0, 0, 1, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0]
The fat cat sat on the mat	[0, 0, 0, 1, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 1, 0]	[0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 1]	[0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]

Skip-gram

The ___ ___ sat ___ ___ mat

중심 단어 / 주변 단어

The fat cat sat on the mat

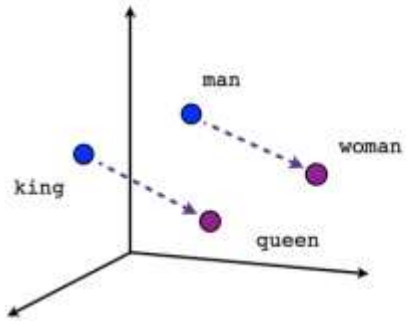
The fat cat sat on the mat

중심 단어	주변 단어
cat	The
cat	Fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the

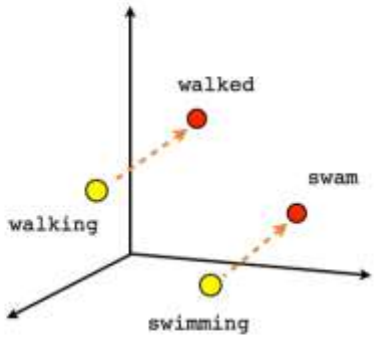
<Word Embedding 방법>

- **Word2Vec**

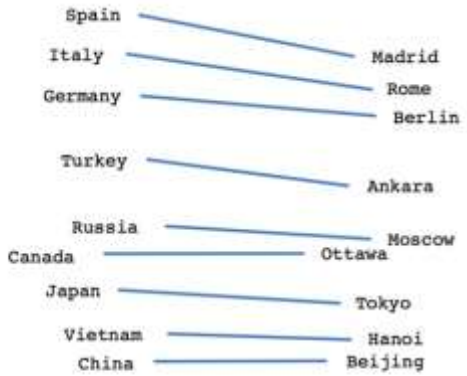
- 장점 : 단어 간 유사도 계산 가능, 벡터 연산을 통해 관계 추론 가능
- 단점 : 단어의 Subword Information 무시
  Out of Vocabulary (OOV) 문제 - 코퍼스를 통해서 학습이 되지 않은 단어에 대한 정보가 없음



Male-Female



Verb tense

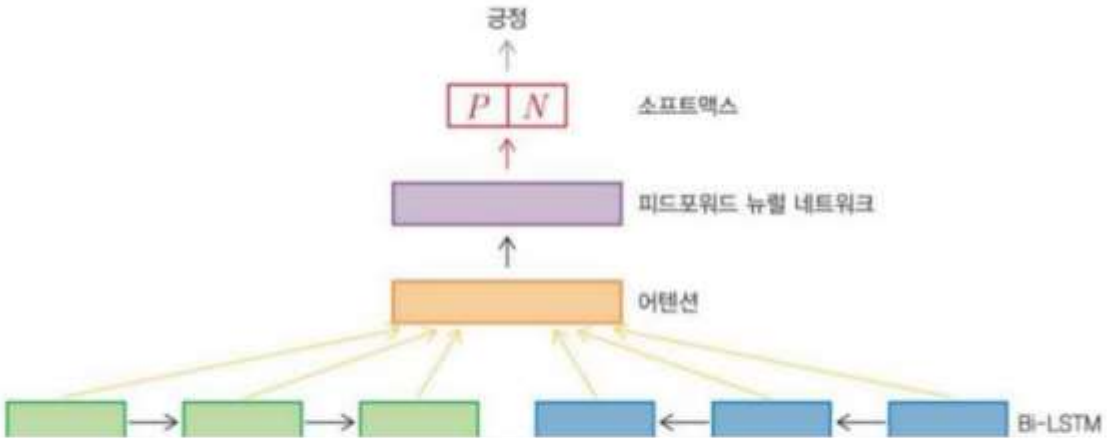


Country-Capital

<Word Embedding 방법>

- **전이학습 (Transfer Learning) : 임베딩을 다른 딥러닝 모델의 입력값으로 쓰는 기법**

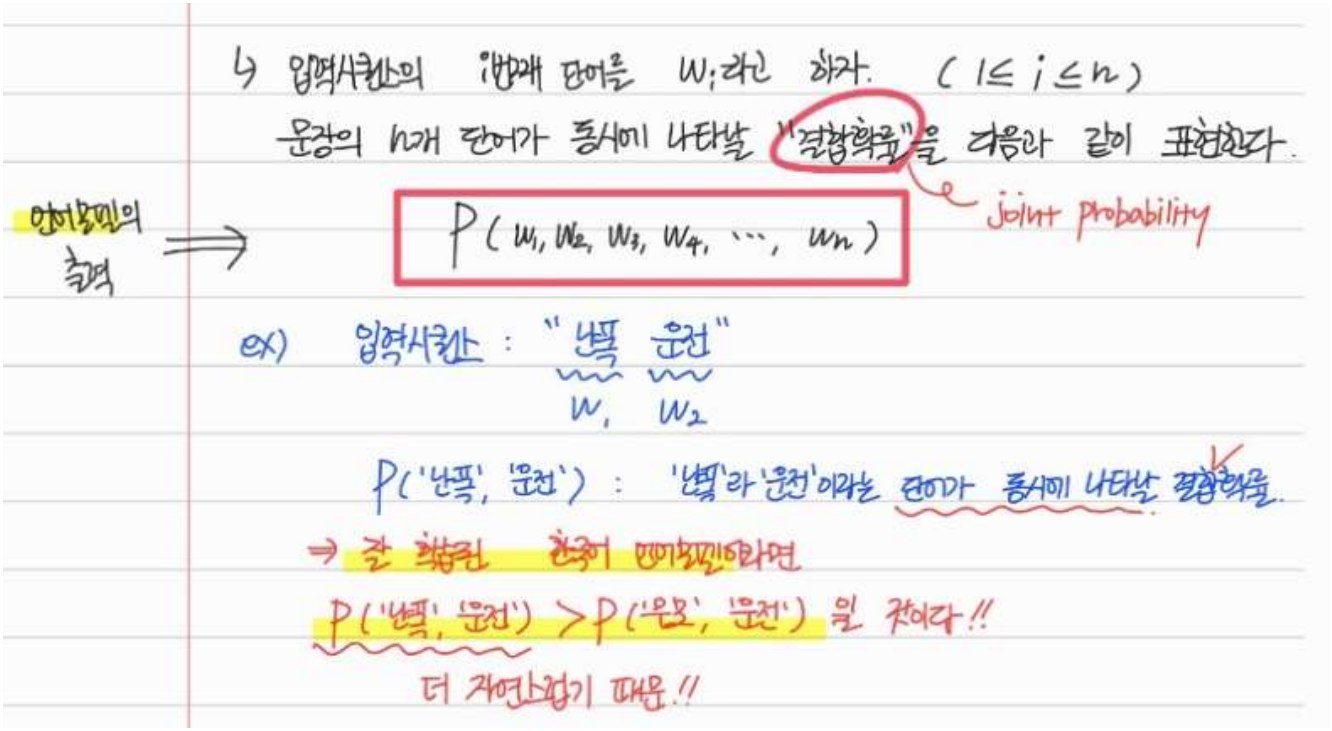
- 대규모 말뭉치를 활용해 임베딩을 미리 만들어 놓고, 이 임베딩을 입력값으로 쓰는 전이 학습 모델은 문서 분류같은 자연어처리 태스크를 빠르게 잘할 수 있게 된다.
- 적은 학습데이터 일 때 유용함
- 임베딩이 중요한 이유! 임베딩의 품질이 좋을수록 더 효율적이고 좋은 성능으로 모델 학습이 가능해짐.



Contents

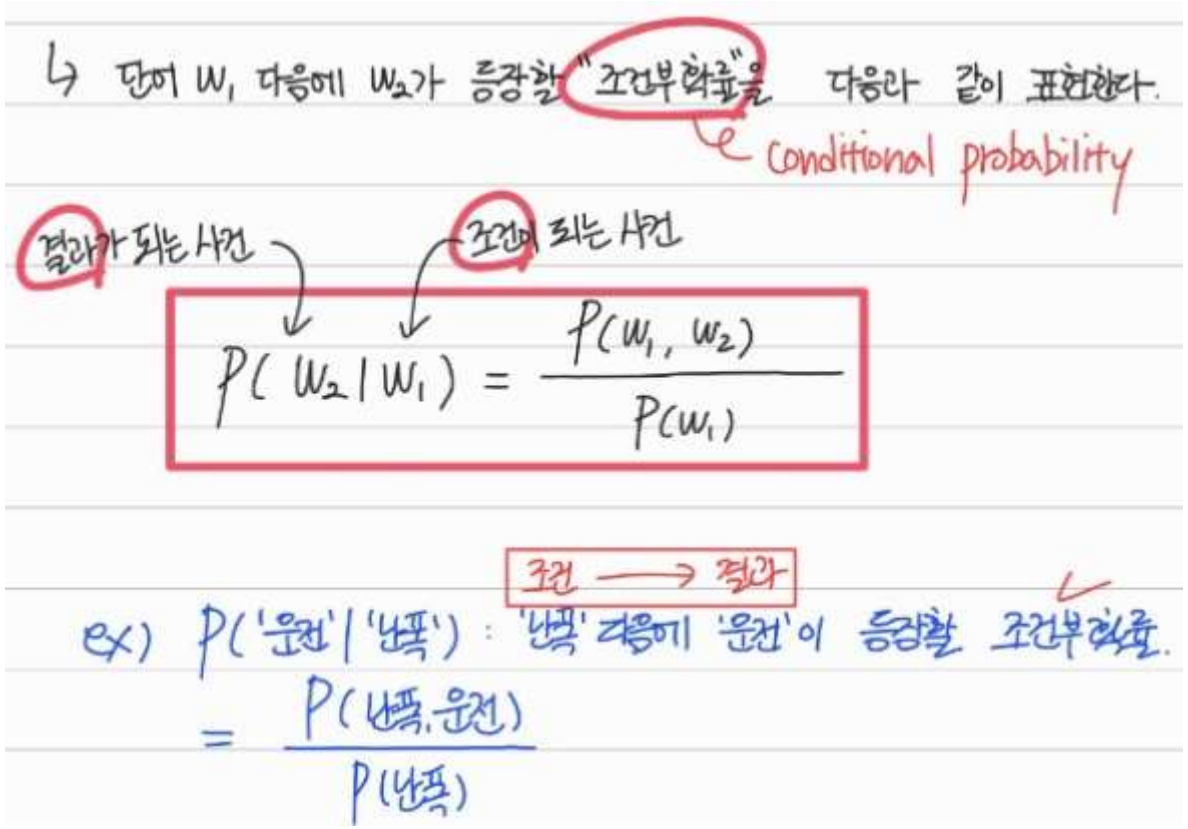
Unit 04 | Language Model

언어모델 (Language Model) : 단어시퀀스에 확률을 부여하는 모델
= 단어시퀀스를 입력받아 해당 시퀀스가 얼마나 그럴듯한지 확률을 출력하는 모델

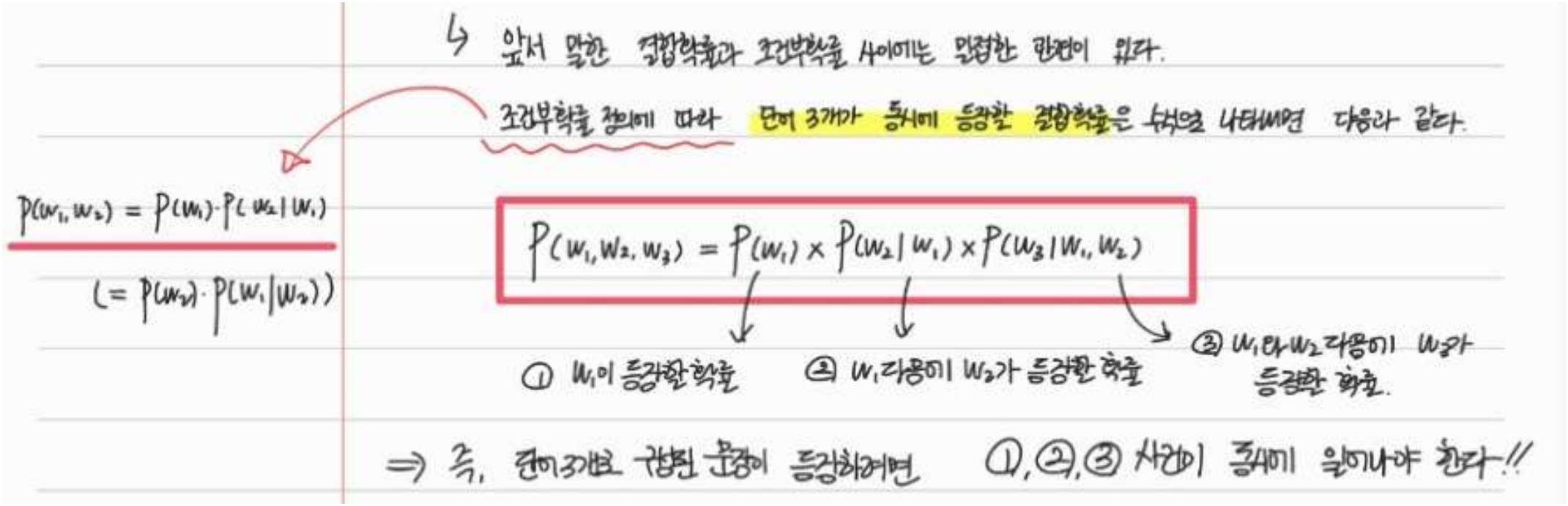


Unit 04 | Language Model

언어모델 (Language Model) : 단어시퀀스에 확률을 부여하는 모델
= 단어시퀀스를 입력받아 해당 시퀀스가 얼마나 그럴듯한지 확률을 출력하는 모델



언어모델 (Language Model) : 단어시퀀스에 확률을 부여하는 모델
= 단어시퀀스를 입력받아 해당 시퀀스가 얼마나 그럴듯한지 확률을 출력하는 모델

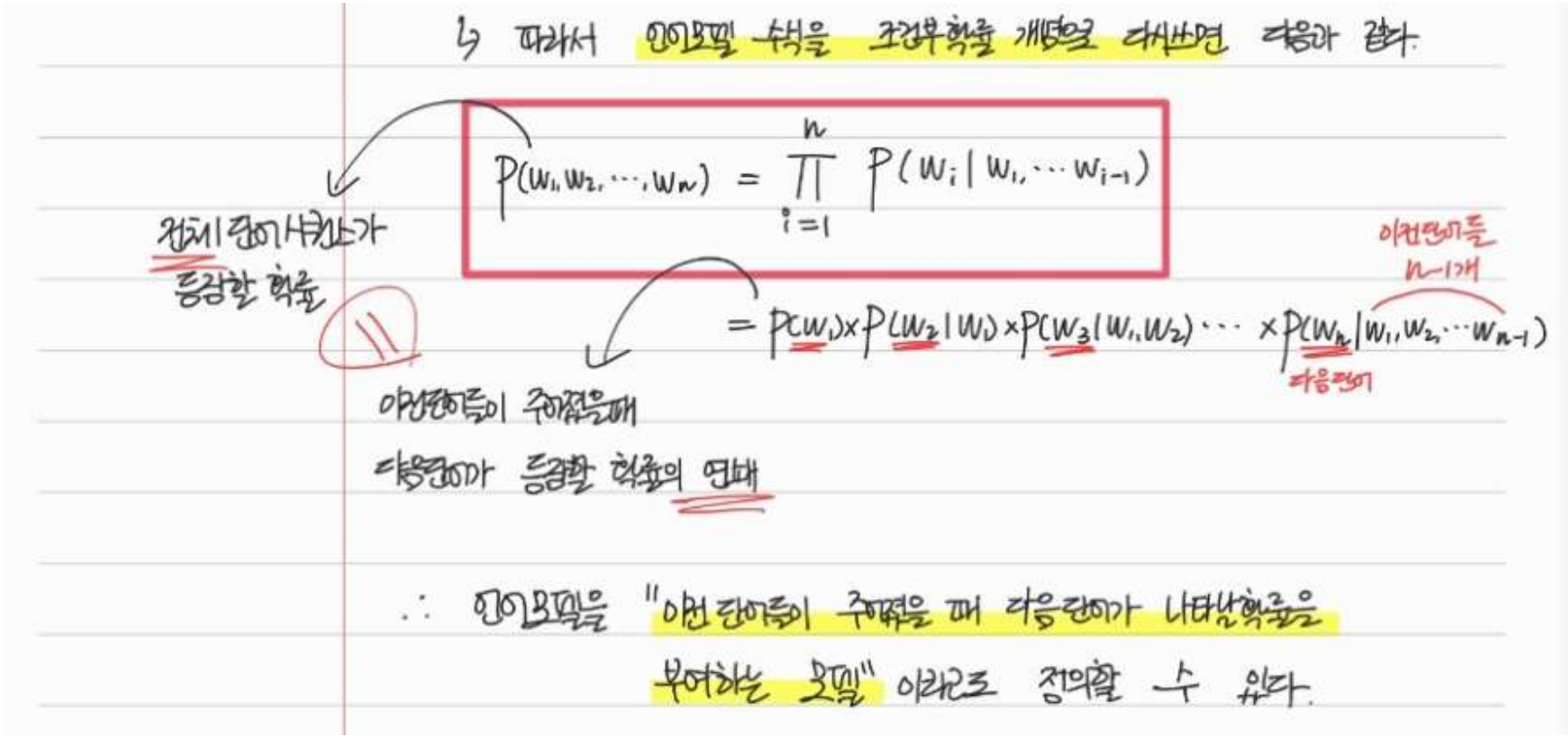


Unit 04 | Language Model

언어모델 (Language Model) : 단어시퀀스에 확률을 부여하는 모델

= 단어시퀀스를 입력받아 해당 시퀀스가 얼마나 그럴듯한지 확률을 출력하는 모델

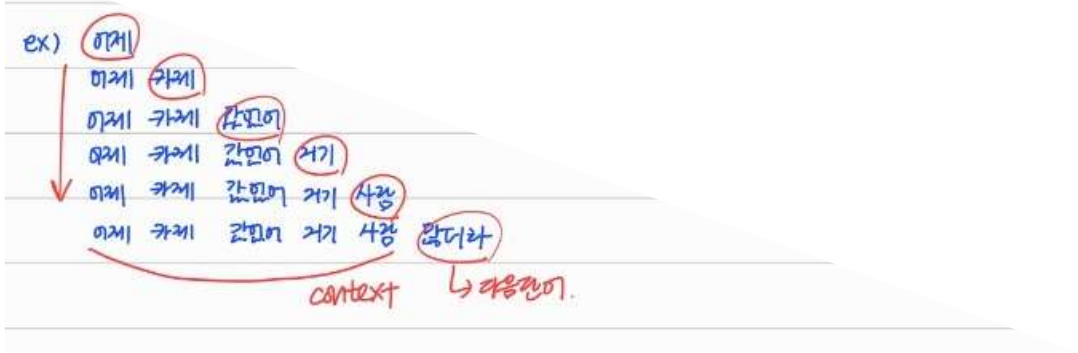
= 이전단어들이 주어졌을때 다음단어가 나타날 확률을 부여하는 모델



Unit 04 | Language Model

언어모델의 목표는 **임의의 단어시퀀스가 해당 언어에서 얼마나 자연스러운지 이해**하고 있는 언어모델을 구축하는 것!

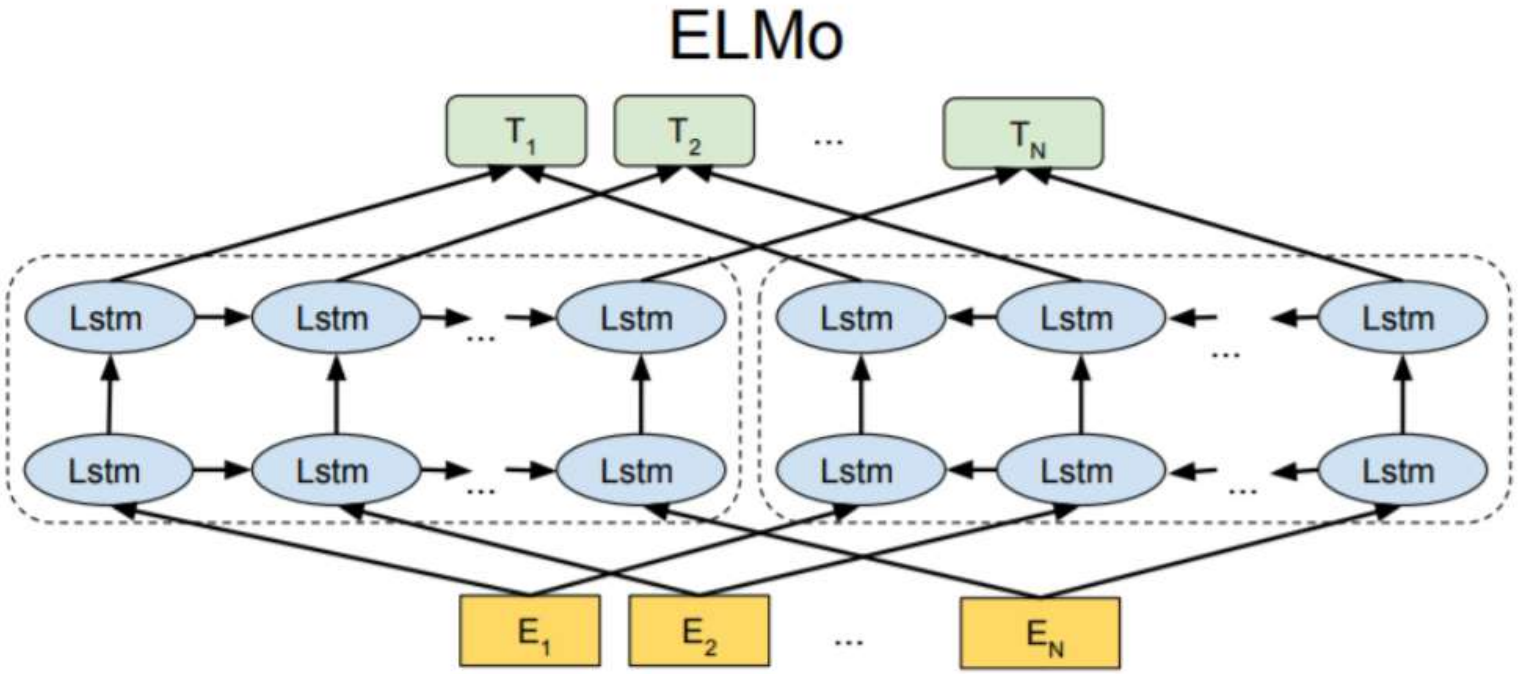
순방향 언어모델 (left-to-right LM): 이전 단어들이 context로 주어졌을 때 다음단어를 예측한다.



역방향 언어모델 (right-to-left LM): 문장의 뒤부터 앞으로 계산한다.

<Word Embedding 방법>

ELMo : 양방향 언어 모델을 활용하여 문맥을 반영한 단어 임베딩 방법
-> 동음이의어에 대해 서로 다른 벡터로 임베딩할 수 있게 됨!



Homework

과제 수행 후 Week7_NLPBasic_Assignment.ipynb 파일 제출

데이터 : 스팸 분류 영어 데이터

1. **Tokenization (소개한 라이브러리 외에도 사용 가능, 토큰화 외의 전처리 추가 가능)**
2. **임베딩 (One-hot encoding, Word2Vec, transfer learning등)**
3. **유의미한 해석 도출 (단어 유사도, word cloud 생성, 이진 분류 모델 학습, 그래프 해석 등)**

* 토크나이저 및 임베딩 모델 선택 과정, 인사이트 해석을 주석으로 달아주세요

참고

- https://wikidocs.net/21698
- https://ratsgo.github.io/nlpbook/docs/language_model/semantics/
- https://bkshin.tistory.com/entry/NLP-12-%EA%B8%80%EB%A1%9C%EB%B8%8CGloVe

감사합니다