

FastCampus Pytorch

Ch7. Natural Language Processing

HARRY KIM

Lecture Content

- 1 Natural Language Processing
- 2 Bag of Words
- 3 Word2Vec
- 4 Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental Analysis

■ 강의 자료

■ Books

- 핸드온 머신러닝 [오렐리앙 제롱, 2018]
- 머신러닝, 딥러닝 실전개발 입문 [쿠지라 히코우즈쿠에, 2017]

■ Online

- UVA DEEP LEARNING COURSE [University of Amsterdam, 2018]
- KoNLPy [<http://konlpy-ko.readthedocs.io/ko/v0.4.3/start/>]
- Convolutional Neural Networks for Sentence Classification[<https://www.aclweb.org/anthology/D14-1181>]

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

1. Natural Language Processing

Natural Language Processing

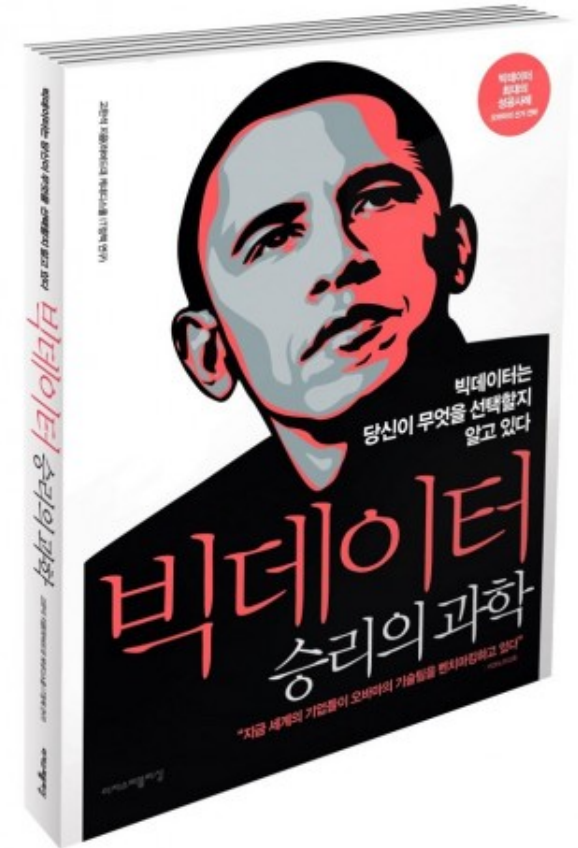
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- 자연어 처리(Natural Language Processing)
 - 인간이 사용하는 언어 데이터를 기반으로 정보를 추출하고 분석, 활용하는 기술
 - 응용 사례
 - 챗봇(Chat-bot)
 - 구글 번역(Google Translate)
 - 소셜 미디어 분석 (SNS Analysis)
 - 감성 분석 (Sentimental Analysis)
 - ...



Natural Language Processing

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- 자연어 처리(Natural Language Processing)
 - 인간의 언어 \neq 컴퓨터의 언어
 - 인간의 언어를 컴퓨터가 이해하기 위해서는 어떻게 해야하는가?
- “나는 학교에 갈 것이다”

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

1) 분석 단위의 설정 문제

- 공백으로 나눌 것인가?
- 따옴표, 쉼표와 같은 특수문자는 어떻게 처리할 것인가?
- 대문자, 소문자는 어떻게 다룰 것인가?
- 숫자는?

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

1) 분석 단위의 설정 문제 (토큰화 : Tokenize)

- 공백으로 나눌 것인가?
- 따옴표, 쉼표와 같은 특수문자는 어떻게 처리할 것인가?
- 대문자, 소문자는 어떻게 다룰 것인가?
- 숫자는?

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

- 1) 분석 단위의 설정 문제 (토큰화 : Tokenize)
- 2) 불필요한 단어의 제거
 - The, a, an, that 의미는 없으나 많이 쓰이는 것
 - 한글에서는 은, 는, 이, 가

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

- 1) 분석 단위의 설정 문제 (토큰화 : Tokenize)
- 2) 불필요한 단어의 제거 (스톱핑 : Stopping)
 - The, a, an, that 의미는 없으나 많이 쓰이는 것
 - 한글에서는 은, 는, 이, 가

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

- 1) 분석 단위의 설정 문제 (토큰화 : Tokenize)
- 2) 불필요한 단어의 제거 (스톱핑 : Stopping)
- 3) 단어의 다양한 형태 고려
 - Push, Pushing, Pushed... 의미는 같으나 형태가 다른 것
 - 수영하다. 수영했다. 수영할 것이다.

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

- 1) 분석 단위의 설정 문제 (토큰화 : Tokenize)
- 2) 불필요한 단어의 제거 (스톱핑 : Stopping)
- 3) 단어의 다양한 형태 고려 (스템밍 : Stemming)
 - Push, Pushing, Pushed... 의미는 같으나 형태가 다른 것
 - 수영하다. 수영했다. 수영할 것이다.

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

- 1) 분석 단위의 설정 문제 (토큰화 : Tokenize)
- 2) 불필요한 단어의 제거 (스톱핑 : Stopping)
- 3) 단어의 다양한 형태 고려 (스템밍 : Stemming)
- 4) 단어의 의미 고려
 - 의미를 고려하면 더 자세한 분석이 가능

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석에서의 문제점

- 1) 분석 단위의 설정 문제 (토큰화 : Tokenize)
- 2) 불필요한 단어의 제거 (스톱핑 : Stopping)
- 3) 단어의 다양한 형태 고려 (스템밍 : Stemming)
- 4) 단어의 의미 고려 (형태소 분석 : POS-tagging)
 - 의미를 고려하면 더 자세한 분석이 가능

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 자연어 분석 방법

- 1) 분석 단위의 설정 문제
 - > 토큰화 : **Tokenize**
- 2) 불필요한 단어의 제거
 - > 스톱핑 : **Stopping**
- 3) 단어의 다양한 형태 고려
 - > 스템밍 : **Stemming**
- 4) 단어의 의미 고려
 - > 형태소 분석 : **POS-tagging**

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 토큰화 (Tokenize)

- 문장을 어떻게, 얼마나 작은 요소(토큰, Token)로 분리할 것인가?
 - 일반적으로 단어 단위의 요소로 분리
- 초기 Information Retrieval System에서 사용된 기법
 - 길이가 3 이상인 글자 또는 숫자만 남김
 - 공백 글자 또는 다른 특수 글자들로 토큰화
 - 대문자는 소문자로 변화

Natural Language Processing

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 토큰화 (Tokenize)

- 문장을 어떻게, 얼마나 작은 요소(토큰, Token)로 분리할 것인가?
 - 일반적으로 단어 단위의 요소로 분리
- 초기 Information Retrieval System에서 사용된 기법
 - 길이가 3 이상인 글자 또는 숫자만 남김
 - 공백 글자 및 다른 특수 문자들로 토큰화
 - 대문자는 소문자로 변화
- "Hyunjun showed a 10% increase in game performance in 2018."
- [Hyunjun, showed, increase, global, game, performance, 2018]

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ 토큰화 (Tokenize)

- 초기 Information Retrieval System의 문제점 : 너무 많은 정보가 사라짐
 - 짧은 단어의 생략
 - 특수 문자의 생략
 - 숫자의 생략
- 이에 따라 다양한 토큰화 방법 파생됨
 - 특수한 특수문자는 따로 저장
 - 짧은 단어라도 명사로 판단되는 것은 남김
 - 숫자도 생략하지 않음
 - ...

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- 스톱핑 (Stopping)
 - 기능형 단어 (Function Words)
 - 관사, 전치사 등 (the, a, ...)
 - 생략해도 문제가 없는 단어
 - 본 분석에 의미가 없다고 판단되는 단어 = Stopword
 - 물론 상황에 따라 기능형 단어도 살려둘 수 있음
 - Stopword의 도입으로 분석의 정확도를 높이고, 속도도 높일 수 있음

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **스토핑 (Stopping)**
 - 기능형 단어 (Function Words)
 - 관사, 전치사 등 (the, a, ...)
 - 생략해도 문제가 없는 단어
 - 본 분석에 의미가 없다고 판단되는 단어 = Stopword
 - 물론 상황에 따라 기능형 단어도 살려둘 수 있음
 - Stopword의 도입으로 분석의 정확도를 높이고, 속도도 높일 수 있음
- **Stopword List : [a, the]**
- "A cat eat a mouse, a mouse eat the cheese"

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **스토핑 (Stopping)**
 - 기능형 단어 (Function Words)
 - 관사, 전치사 등 (the, a, ...)
 - 생략해도 문제가 없는 단어
 - 본 분석에 의미가 없다고 판단되는 단어 = Stopword
 - 물론 상황에 따라 기능형 단어도 살려둘 수 있음
 - Stopword의 도입으로 분석의 정확도를 높이고, 속도도 높일 수 있음
 - **Stopword List : [a, the]**
 - "A cat eat a mouse, a mouse eat the cheese"
 - [cat, eat, mouse, mouse, eat, cheese]

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- 스템밍 (Stemming)
 - 단어는 다양한 형태를 가짐
 - 복수형, 시제, 동명사 등
 - Dogs = Dog
 - 하지만 의미는 비슷
 - 스템머 (Stemmer) : 원래 단어의 접사를 제거
 - 유의미한 분석 결과 향상을 가져옴
 - 다양한 스템머의 존재 : 사전 기반, 알고리즘 기반...

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- 스템밍 (Stemming)
 - 단어는 다양한 형태를 가짐
 - 복수형, 시제, 동명사 등
 - Dogs = Dog
 - 하지만 의미는 비슷
 - 스템머 (Stemmer) : 원래 단어의 접사를 제거
 - 유의미한 분석 결과 향상을 가져옴
 - 다양한 스템머의 존재 : 사전 기반, 알고리즘 기반...
 - PorterStemmer : 가장 유명한 알고리즘 기반 스템머 (http://9ol.es/porter_js_demo.html)
 - [fishing, fished, fisher] → [fish, fish, fish]
 - [running, runs, run] → [run, run, run]

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **스테밍 (Stemming)**
 - 물론 단점이 존재
 - PorterStemmer : [lying, lie] → [ly, lie]
- **Lemmatization**
 - 단어의 기본형을 반환
 - EX) [lying(v), lie(v)] → [lie(v), lie(v)]
 - EX2) [booking(v), books(n)] → [book(v), book(n)]
 - 허나, 각 단어에 대한 품사 정보 및 방대한 단어장 필요

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **형태소 분석 (Part-Of-Speech Tagging)**
 - 단어를 분석할 때 동음이의어의 경우 문제가 될 수 있음
 - 또한 각 단어의 품사를 활용하면 보다 자세한 분석 가능
- “영화가 좋다”
 - 영화/Noun
 - 가/Josa
 - 좋다/Adjective
- Report/Noun
- Report/Verb

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

▪ EXTRA) N-gram

- 단어는 순서 집합체이므로, 단어 사이의 연결을 유지해야할 필요 있음
- 따라서, 단어를 n 개씩 묶어서 분석하여 단어의 순서 및 의미를 고려
- 유니그램 (unigram) : 단어 1개씩 분석
- 바이그램 (bigram) : 단어 2개씩 분석
- 트라이그램 (trigram) : 단어 3개씩 분석
- “나는 치킨을 먹었다”을 **Bigram**으로 표현
- [나는 치킨을, 치킨을 먹었다]

Natural Language Processing

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- 자연어 처리(Natural Language Processing)
 - 주로 5가지 기법 (토큰화, 스톱핑, 스테밍, 형태소 분석, N-gram)을 활용하여 전처리
 - 그렇다면, 문서(혹은 문장)를 전처리 한 후에는 어떻게 해야하는가?
 - = Word Embedding (단어를 벡터화하는 것)
 - Bag of Words
 - Word2Vec

NLP

Bag of Words

Word2Vec

**Sentimental
Analysis**

2. Bag of Words

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **Bag of Words (BoW)**
 - 문장을 숫자로 변경시키기 위해, 단순히 등장 횟수로 벡터화하는 것
 - 문장을 가방에 넣고 섞은 다음에, 단어의 개수를 통해 문장을 확인
 - 단어 사전을 만드는 것이라고 이해할 수 있음

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental Analysis

- **Bag of Words (BoW)**
 - "I like apples. My parents like apples too."
 - I : 1
 - like : 2
 - apples : 3
 - My : 4
 - Parents : 5
 - Too : 6

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental Analysis

- **Bag of Words (BoW)**
 - "I like apples. My parents like apples too."
 - I : 1 = [1,0,0,0,0,0]
 - like : 2 = [0,1,0,0,0,0]
 - apples : 3 = [0,0,1,0,0,0]
 - My : 4 = [0,0,0,1,0,0]
 - Parents : 5 = [0,0,0,0,1,0]
 - Too : 6 = [0,0,0,0,0,1]
- "I like apples. My parents like apples too." = [1, 2, 2, 1, 1, 1]

- **Bag of Words (BoW)**

- 문서(혹은 문장)의 특징을 분석하고 싶을 시, 각 문서(혹은 문장)의 특징을 추출하기 원함
- 하지만 아래와 같은 단어는 어떤 문서이든 자주 등장할 것임
 - “나는 ~ ”
 - “~ 입니다”
 - “~ 했다”
- 따라서 중요한 단어에 가중을 두기 위해, TF-IDF를 사용

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **TF-IDF**

- TF(Term Frequency) : 특정 단어가 문서 내에 등장하는 횟수
- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ TF-IDF

- TF(Term Frequency) : 특정 단어가 문서 내에 등장하는 횟수
- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

TF	나는	사랑을	했다	고기를	먹었다	숙제를
문서 1	1	1	1			
문서 2	1			1	1	
문서 3	1		1			1

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **TF-IDF**

- DF(Document Frequency) : 특정 단어가 총 문서 중 등장하는 문서의 개수
- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ TF-IDF

- DF(Document Frequency) : 특정 단어가 총 문서 중 등장하는 문서의 개수
- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

DF	나는	사랑을	했다	고기를	먹었다	숙제를
	3	1	2	1	1	1

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental Analysis

TF-IDF

- IDF(Inverse Document Frequency) : $\log(\text{전체 문서 개수} / \text{DF})$
- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

DF	나는	사랑을	했다	고기를	먹었다	숙제를
	3	1	2	1	1	1

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental Analysis

TF-IDF

- IDF(Inverse Document Frequency) : $\log(\text{전체 문서 개수} / \text{DF})$
- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

DF	나는	사랑을	했다	고기를	먹었다	숙제를
	3	1	2	1	1	1

IDF	나는	사랑을	했다	고기를	먹었다	숙제를
	0	$\log(3)$	$\log(3/2)$	$\log(3)$	$\log(3)$	$\log(3)$

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ TF-IDF

- 문서 1 : "나는 사랑을 했다"
- 문서 2 : "나는 고기를 먹었다"
- 문서 3 : "나는 숙제를 했다"

TF	나는	사랑을	했다	고기를	먹었다	숙제를
문서 1	1	1	1			
문서 2	1			1	1	
문서 3	1		1			1

IDF	나는	사랑을	했다	고기를	먹었다	숙제를
	0	$\log(3)$	$\log(3/2)$	$\log(3)$	$\log(3)$	$\log(3)$

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **TF-IDF**

- $TF-IDF = TF * IDF = TF * \log (\text{전체 문서 개수} / DF)$
- TF가 증가할수록 증가
- DF가 증가할수록 감소

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

TF-IDF

- TF-IDF = $TF * IDF = TF * \log(\text{전체 문서 개수} / DF)$
- TF가 증가할수록 증가
- DF가 증가할수록 감소

TF-IDF	나는	사랑을	했다	고기를	먹었다	숙제를
문서 1	0	$\log(3)$	$\log(3/2)$			
문서 2	0			$\log(3)$	$\log(3)$	
문서 3	0		$\log(3/2)$			$\log(3)$

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental Analysis

TF-IDF

- TF-IDF = $TF * IDF = TF * \log(\text{전체 문서 개수} / DF)$
- TF가 증가할수록 증가
- DF가 증가할수록 감소

TF-IDF	나는	사랑을	했다	고기를	먹었다	숙제를
문서 1	0	$\log(3)$	$\log(3/2)$			
문서 2	0			$\log(3)$	$\log(3)$	
문서 3	0		$\log(3/2)$			$\log(3)$

Bag of Words

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- Bag of Words (BoW) / TF-IDF
 - 장점
 - 문장을 쉽게 표현 가능
 - 코딩화하기 쉬움
 - 단점
 - 문장 내 순서의 무시
 - 단어의 의미 고려 불가 (동의어, 반의어 등)
 - 단어의 개수가 증가할 수록 연산량이 많아짐
 - 짧은 문장의 경우 Sparse해질 수 있음

NLP

Bag of Words

Word2Vec

**Sentimental
Analysis**

3. Word2Vec

NLP

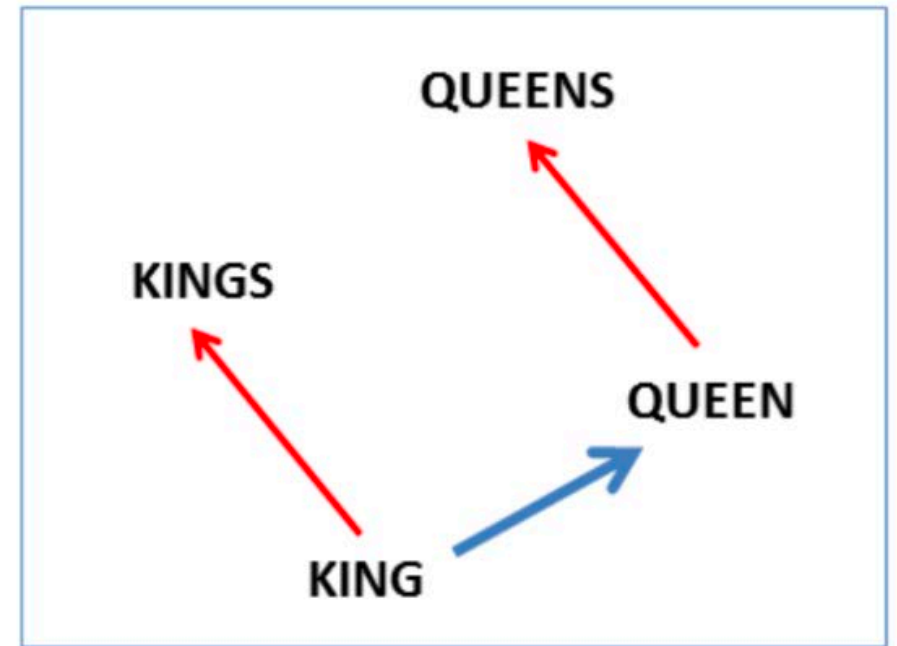
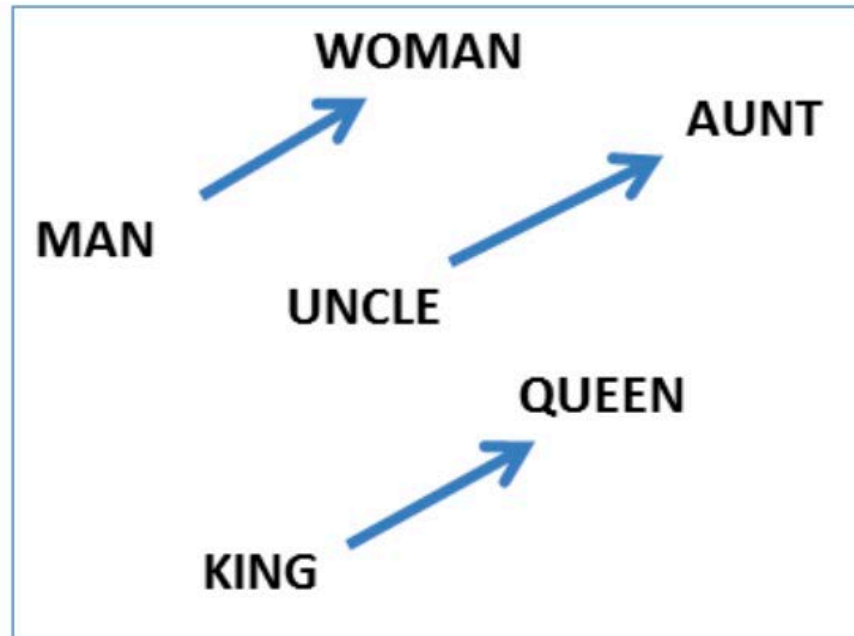
Bag of Words

Word2Vec

Sentimental
Analysis

■ Word2Vec

- 실제로 단어는 서로 간의 상관관계가 높음
- 동의어, 반의어 등 서로 간의 연관성이 있음
- 이는 벡터 연산과 흡사



Word2Vec

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **Word2Vec**
 - 실제로 단어는 서로 간의 상관관계가 높음
 - 동의어, 반의어 등 서로 간의 연관성이 있음
 - 이는 벡터 연산과 흡사

남자-여자+왕

QUERY

+남자/Noun +왕/Noun -여자/Noun

RESULT

국왕/Noun

한국-서울+도쿄

QUERY

+한국/Noun +도쿄/Noun -서울/Noun

RESULT

일본/Noun

<http://w.elnn.kr/search/>

NLP

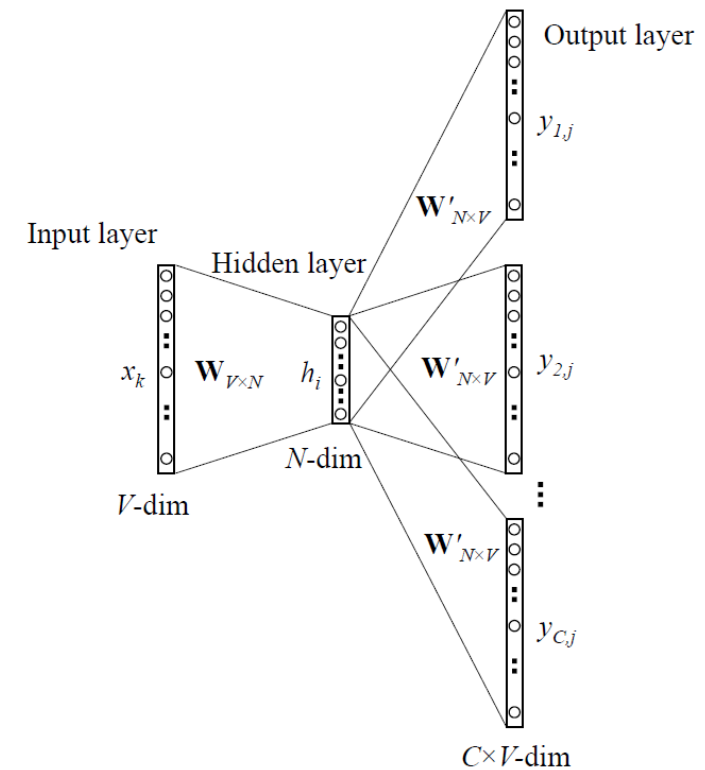
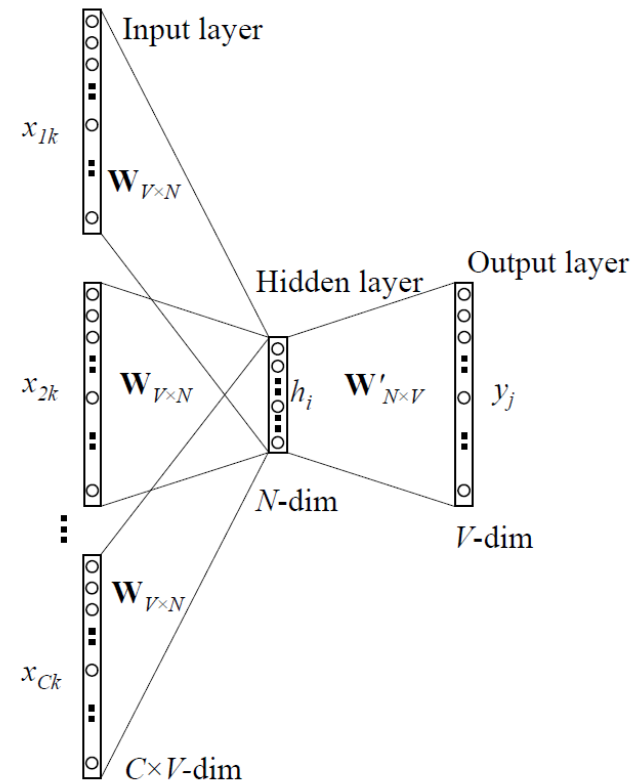
Bag of Words

Word2Vec

Sentimental Analysis

Word2Vec

- 2013, Google, Tomas Mikolov가 제안
- 크게 CBOW(Continuous Bag-of-Words) / Skip-gram으로 나뉨



NLP

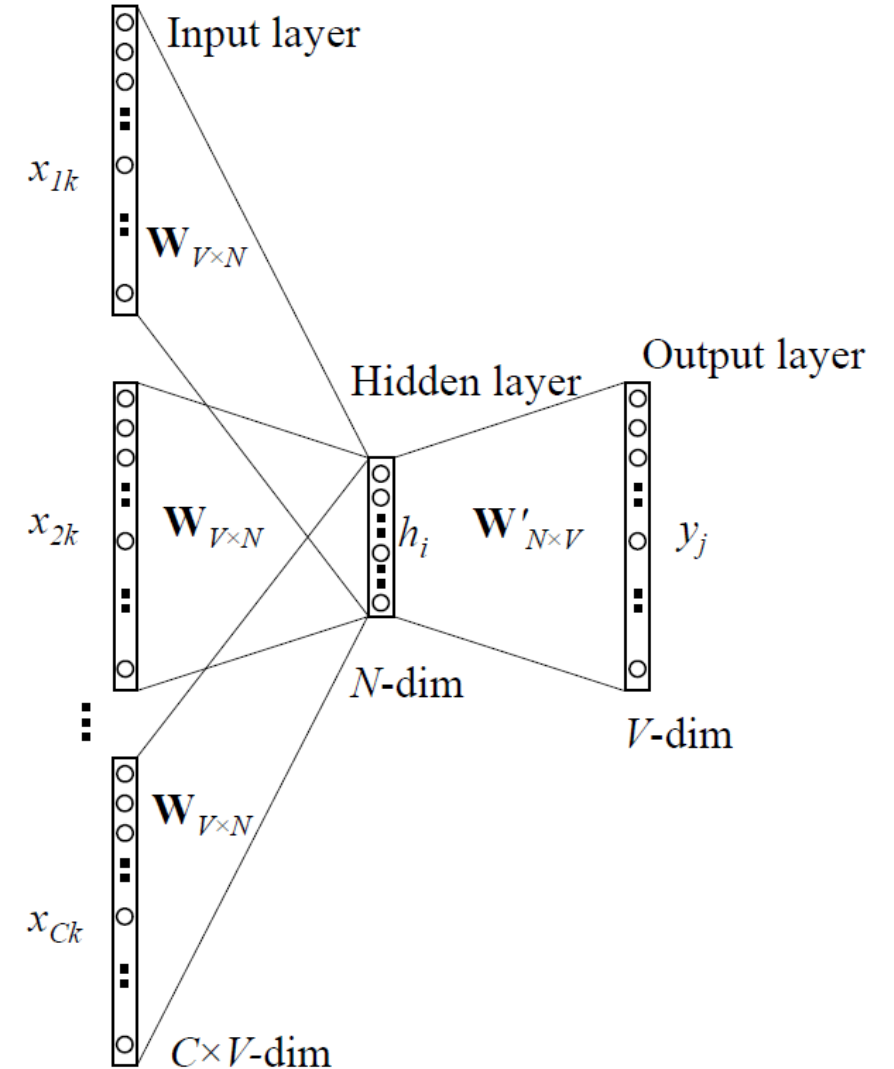
Bag of Words

Word2Vec

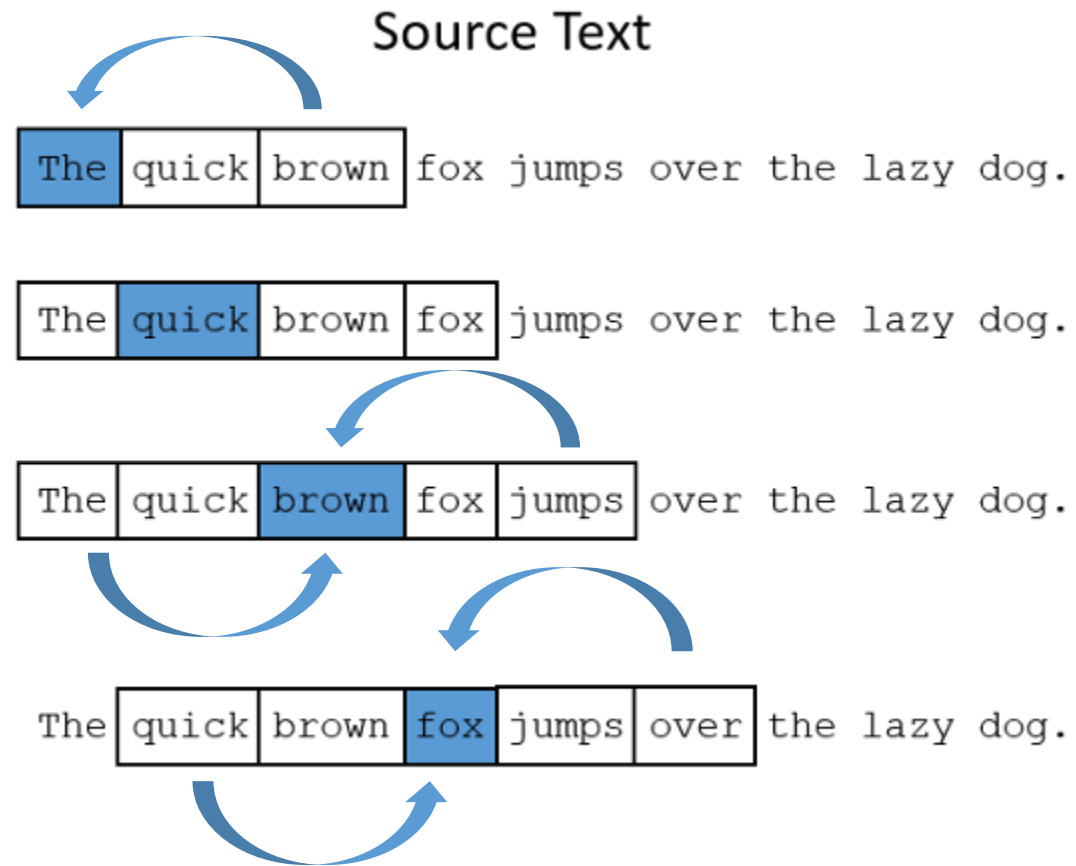
Sentimental Analysis

CBOW (Continuous Bag-of-Words)

- 컨텍스트로부터 단어를 예측
- "요즘 날씨가 더워서 ____ 많이 난다."
- 단어를 BOW화 하여 사용
- 주어진 단어 앞 뒤의 $N/2$ 개, 총 N 개의 입력
- 결과적으로 해당 단어를 예측
- 데이터가 적은 경우 적합



- CBOW (Continuous Bag-of-Words)



NLP

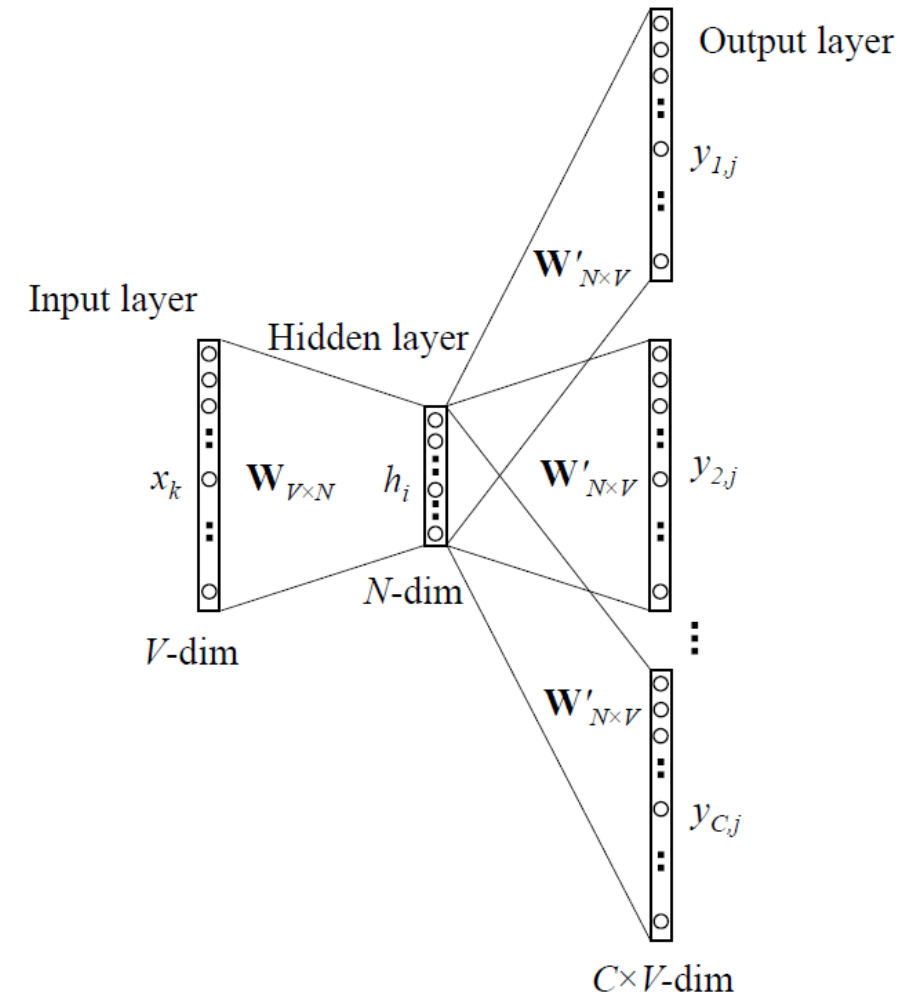
Bag of Words

Word2Vec

Sentimental
Analysis

■ Skip-Gram

- 단어로부터 컨텍스트를 예측
- " _____ 땀이 _____ "
- 주어진 단어 주위에서 샘플링
- 가까이 위치해있는 단어에 가중
- 결과적으로 단어의 분포를 예측
- 데이터가 많은 경우 적합



Word2Vec

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Skip-Gram

Source Text	Training Samples
<div> <div>The quick brown</div> fox jumps over the lazy dog. → </div>	(the, quick) (the, brown)
<div> <div>The quick brown fox</div> jumps over the lazy dog. → </div>	(quick, the) (quick, brown) (quick, fox)
<div> <div>The quick brown fox jumps</div> over the lazy dog. → </div>	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
<div> <div>The quick brown fox jumps over</div> the lazy dog. → </div>	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Word2Vec의 장점

- 위 모델들을 통해 학습한 가중치(W, W')의 행을 활용
 - 단어가 One-hot encoding임을 활용하면 유도 가능
 - Input Vector(W), Output Vector(W')
- V 차원의 단어를 N 차원의 벡터로 나타낼 수 있음
- 기존 Sparse하던 BOW와 달리, Dense하게 표현 가능
- 적은 차원으로 많은 단어 표현 가능
- 단어 간의 유사도 측정 가능

NLP

Bag of Words

Word2Vec

**Sentimental
Analysis**

4. Sentimental Analysis

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **Sentimental Analysis (감성 분석)**
 - 텍스트를 통해 텍스트에 담겨있는 주관적인 정보를 파악하는 것
 - 예시
 - 뉴스 댓글을 통한 특정 안건의 찬성도 파악
 - 특정 제품의 호감/비호감 파악
 - 영화에 대한 평가 파악
- 주로 긍정/부정/중립으로 텍스트를 분류
- 이를 위해 앞서 다룬 자연어 처리가 필수

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- Sentimental Analysis (감성 분석)

“영화 리뷰가 주어지면 평점을 예측하고 싶다!”

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Sentimental Analysis (감성 분석)



Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental Analysis

■ Sentimental Analysis (감성 분석)



★★★★★ 10 이 영화를 본 제가 별 10개를 받아야겠습니다
오레온(audi****) | 2017.06.28 03:01 | 신고

공감 13838 비공감 514

★★★★★ 10 나만 당할수는 없다!!
김용훈(abak****) | 2017.06.28 00:35 | 신고

공감 11119 비공감 692

★★★★★ 10 저는 10년째 불면증을 겪고있습니다 수면제없이 잠을 못잡니다..처음으로 수면제없이 상쾌한 잠을 잤습니다 값비싼 프로포폴대신 싸고 저렴한 "리얼"을 강력추천합니다
안녕ㅎ(jiwo****) | 2017.06.28 05:51 | 신고

공감 7945 비공감 334

★★★★★ 10 클레멘타인을 이을 영화
ㅇㅇ(dm dd****) | 2017.06.28 00:00 | 신고

공감 7496 비공감 690

★★★★★ 10 씨리얼 보는게 더 재미있어요
햇도그(qusg****) | 2017.06.28 01:22 | 신고

공감 6859 비공감 259

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Sentimental Analysis : 데이터 수집



★★★★★ 10 이 영화를 본 제가 별 10개를 받아야겠습니다

오레온(audj****) | 2017.06.28 03:01 | 신고

 공감 13838  비공감 514

★★★★★ 10 나만 당할수는 없다!!

김용훈(abak****) | 2017.06.28 00:35 | 신고

 공감 11119  비공감 692

★★★★★ 10 저는 10년째 불면증을 겪고있습니다 수면제없이 잠을 못잡니다..처음으로 수면제없이 상쾌한 잠을 잤습니다 값비싼 프로포폴대신 싸고 저렴한 "리얼"을 강력 추천합니다

안녕ㅎ(jiwo****) | 2017.06.28 05:51 | 신고

 공감 7945  비공감 334

★★★★★ 10 클레멘타인을 이을 영화

ㅇㅇ(dmdd****) | 2017.06.28 00:00 | 신고

 공감 7496  비공감 690

★★★★★ 10 씨리얼 보는게 더 재미있어요

햇도그(qusg****) | 2017.06.28 01:22 | 신고

 공감 6859  비공감 259

Reviews :

[“이 영화를 본 제가 별 10개를 받아야겠습니다”, “나만 당할 수는 없다!!”, “저는 10년째 불면증을 겪고 있습니다”, “클레멘타인을 이을 영화”, “씨리얼 보는게 더 재미있어요”]

Scores :

[10, 10, 10, 10, 10]

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- Sentimental Analysis : 데이터 전처리

Reviews :

["이 영화를 본 제가 별 10개를 받아야겠습니다", "나만 당할 수는 없다!!", "저는 10년째 불면증을 겪고 있습니다", "클레멘타인을 이을 영화", "씨리얼 보는게 더 재미있어요", "asdfas□
ㅈㄷㄱㅇㅇㄹ"]

Scores :

[10, 10, 10, ?, 10, 10]

Reviews :

["이 영화를 본 제가 별 10개를 받아야겠습니다", "나만 당할 수는 없다!!", "저는 10년째 불면증을 겪고 있습니다", "클레멘타인을 이을 영화", "씨리얼 보는게 더 재미있어요"]

Scores :

[10, 10, 10, 6, 10]



Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- Sentimental Analysis : 데이터 벡터화
 - 감성 분석 모델(NN)

$$\begin{array}{cc}
 \left[\begin{array}{c} [\text{내, 인생, 최고, 영화}] \\ \vdots \\ [\text{줄거리, 너무, 진부}] \end{array} \right] & \left[\begin{array}{c} 1 \\ \vdots \\ 0 \end{array} \right] \\
 \text{Data}(x) & \text{Label}(y)
 \end{array}$$

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Sentimental Analysis : 문자 숫자화

■ 감성 분석 모델(NN)

$$\begin{array}{cc}
 \begin{bmatrix} \text{[내, 인생, 최고, 영화]} \\ \vdots \\ \text{[줄거리, 너무, 진부]} \end{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix} \\
 \text{Data}(x) & \text{Label}(y)
 \end{array}
 \rightarrow
 \begin{array}{cc}
 \begin{bmatrix} [1, 0, 0, 1, \dots, 0, 0, 1] \\ \vdots \\ [0, 1, 1, 0, \dots, 0, 1, 1] \end{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix} \\
 \text{Data}(x) & \text{Label}(y)
 \end{array}$$

Sentimental Analysis

NLP

Bag of Words

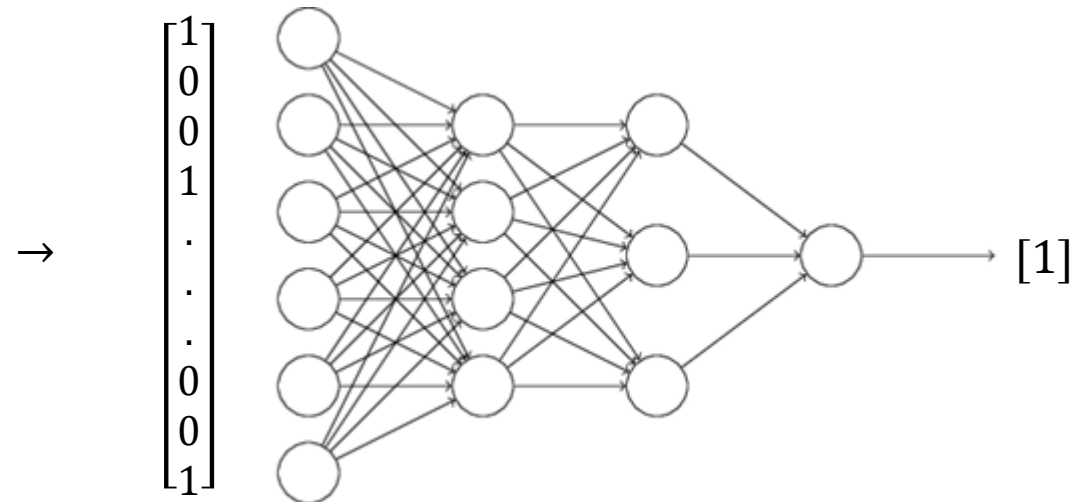
Word2Vec

Sentimental
Analysis

■ Sentimental Analysis : 모델 생성 및 학습

■ 감성 분석 모델(NN)

$$\begin{array}{cc}
 \begin{bmatrix} \text{[내, 인생, 최고, 영화]} \\ \vdots \\ \text{[줄거리, 너무, 진부]} \end{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix} \\
 \text{Data}(x) & \text{Label}(y)
 \end{array}
 \rightarrow
 \begin{array}{cc}
 \begin{bmatrix} [1, 0, 0, 1, \dots, 0, 0, 1] \\ \vdots \\ [0, 1, 1, 0, \dots, 0, 1, 1] \end{bmatrix} & \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix} \\
 \text{Data}(x) & \text{Label}(y)
 \end{array}$$



Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Sentimental Analysis (감성 분석)

■ 감성 분석 모델(CNN) : Convolutional Neural Networks for Sentence Classification(Yoon Kim)

■ "I like apples. My parents like apples too."

■ I : 1 = [1,0,0,0,0,0]

■ like : 2 = [0,1,0,0,0,0]

■ apples : 3 = [0,0,1,0,0,0]

■ My : 4 = [0,0,0,1,0,0]

■ Parents : 5 = [0,0,0,0,1,0]

■ Too : 6 = [0,0,0,0,0,1]

■ "I like apples. My parents like apples too." =

$$\begin{bmatrix} [1,0,0,0,0,0] \\ [0,1,0,0,0,0] \\ [0,0,1,0,0,0] \\ [0,0,0,1,0,0] \\ [0,1,0,0,0,0] \\ [0,0,0,0,1,0] \\ [0,0,0,0,0,1] \end{bmatrix}$$

Sentimental Analysis

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ Sentimental Analysis (감성 분석)

- 감성 분석 모델(CNN) : Convolutional Neural Networks for Sentence Classification(Yoon Kim)

$$\begin{bmatrix} \text{"내 인생 최고의 영화"} \\ \cdot \\ \cdot \\ \cdot \\ \text{"줄거리가 너무 진부"} \end{bmatrix} \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$Data(x)$

$Label(y)$

→

$$\begin{bmatrix} [1, 0, 0, 0, 0, 0] \\ [0, 1, 0, 0, 0, 0] \\ [0, 0, 1, 0, 0, 0] \\ [0, 0, 0, 1, 0, 0] \\ [0, 1, 0, 0, 0, 0] \\ [0, 0, 0, 0, 1, 0] \\ [0, 0, 0, 0, 0, 1] \end{bmatrix} \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

$Data(x)$

$Label(y)$

Sentimental Analysis

NLP

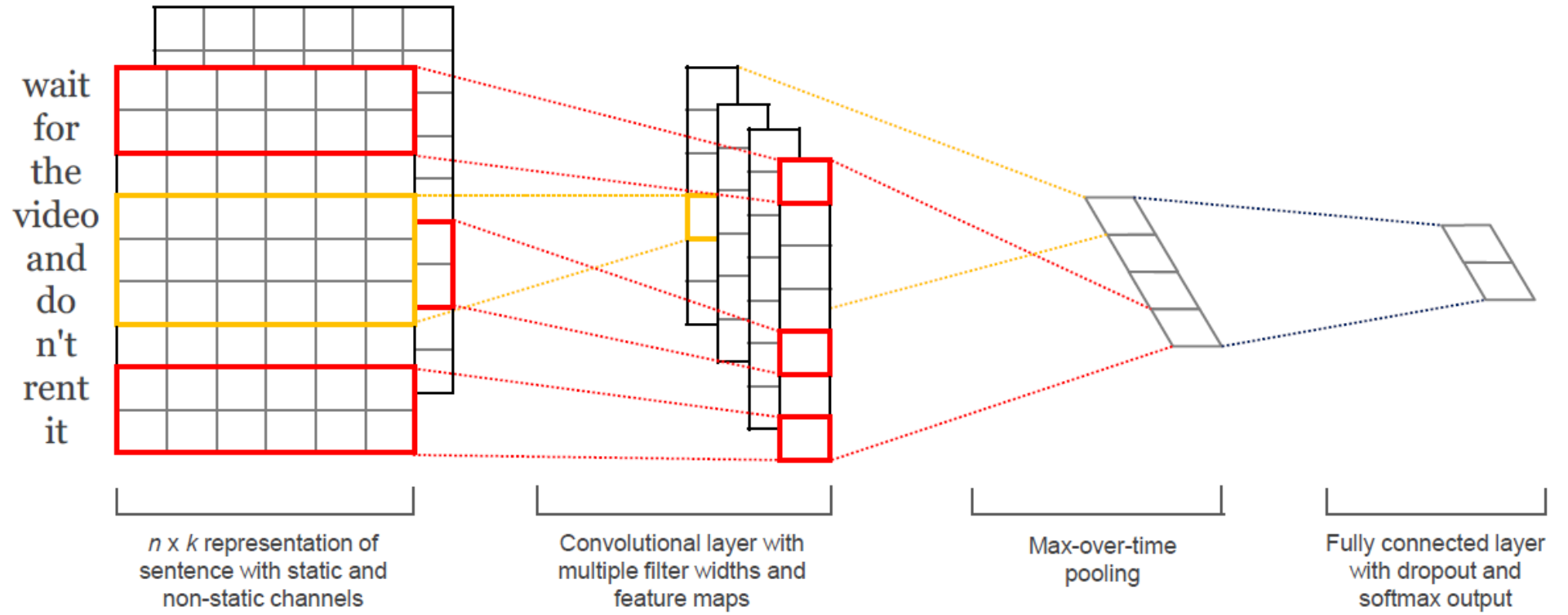
Bag of Words

Word2Vec

Sentimental Analysis

■ Sentimental Analysis (감성 분석)

- 감성 분석 모델(CNN) : Convolutional Neural Networks for Sentence Classification(Yoon Kim)



NLP

Bag of Words

Word2Vec

Sentimental
Analysis

<부록1> NLTK 설치 방법

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ NLTK

- 자연어 정보처리를 위한 파이썬 패키지 (<https://www.nltk.org/>)
- 오픈소스 소프트웨어

NLTK 3.3 documentation

[NEXT](#) | [MODULES](#) | [INDEX](#)

Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to [over 50 corpora and lexical resources](#) such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active [discussion forum](#).

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

[Natural Language Processing with Python](#) provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

TABLE OF CONTENTS

[NLTK News](#)

[Installing NLTK](#)

[Installing NLTK Data](#)

[Contribute to NLTK](#)

[FAQ](#)

[Wiki](#)

[API](#)

[HOWTO](#)

SEARCH

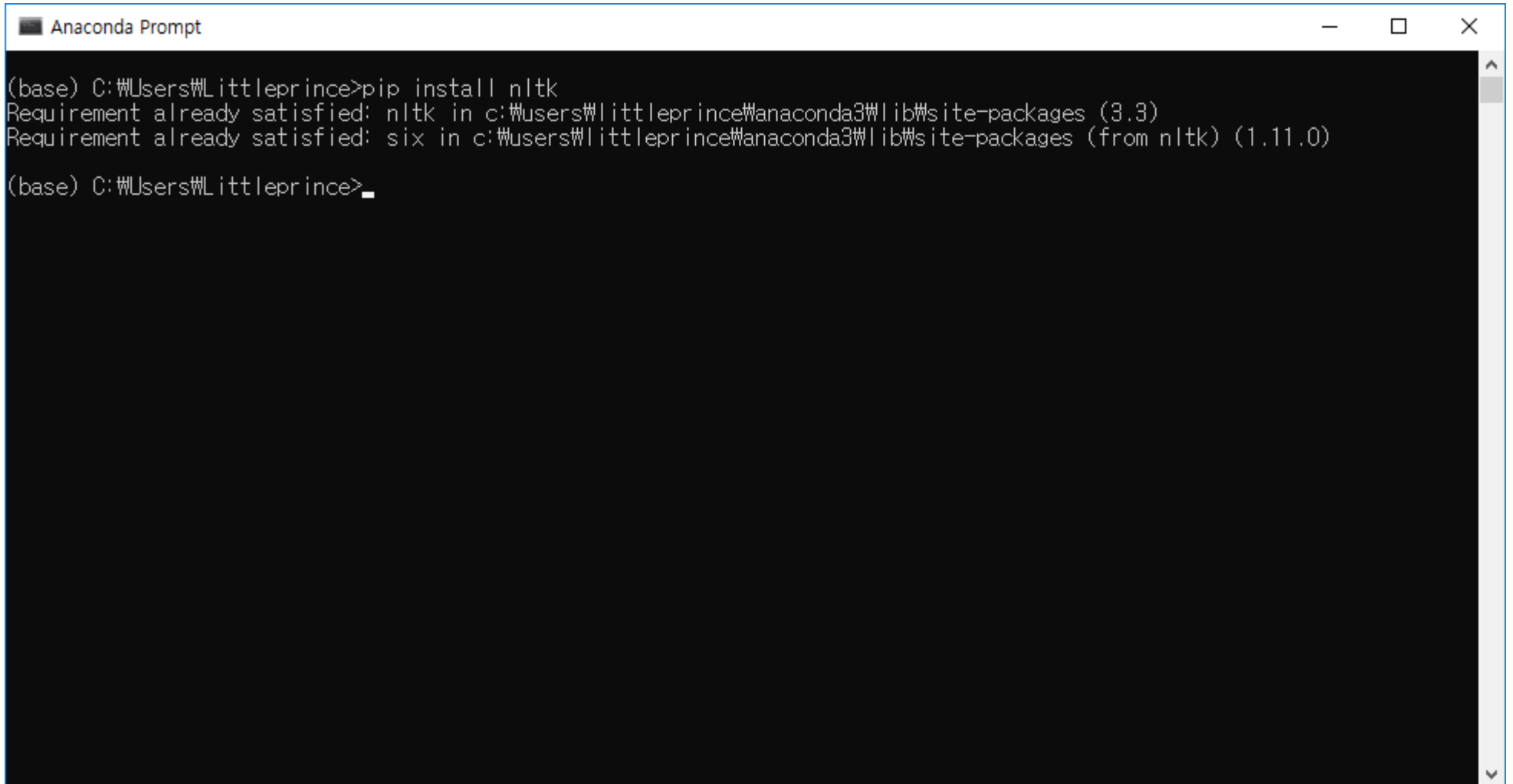
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- NLTK 설치 방법 (Window)



```
Anaconda Prompt

(base) C:\Users\Littleprince>pip install nltk
Requirement already satisfied: nltk in c:\users\littleprince\anaconda3\lib\site-packages (3.3)
Requirement already satisfied: six in c:\users\littleprince\anaconda3\lib\site-packages (from nltk) (1.11.0)

(base) C:\Users\Littleprince>_
```

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

<부록2> KoNLPy 설치 방법

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy

- 한국어 정보처리를 위한 파이썬 패키지 (<http://konlpy.org/ko/v0.4.3/>)
- 오픈소스 소프트웨어

저인의 어깨 위에 서기

아름답지만 다소 복잡하기도한 한국어는 전세계에서 13번째로 많이 사용되는 언어입니다. 복잡미묘한 한국어 텍스트에서 유용한 특성을 추출하기 위해 그 동안 수많은 한국어 정보처리 도구가 개발되기도 했습니다.

KoNLPy는 같은 기능을 하는 또 하나의 도구를 만들려는 것이 아닙니다. 그보다는, 현존하는 도구 위에 한 층을 쌓아 더 멀리 내다보려는 것입니다. 또한 KoNLPy는 파이썬 프로그래밍 언어로 사용할 수 있도록 만들어졌는데, 그것은 파이썬이 간결하고 우아한 문법구조, 강력한 스트링 연산 기능을 가지고 있을 뿐 아니라 크롤링, 웹프로그래밍, 그리고 데이터 분석을 수행할 수 있는 다양한 패키지를 사용할 수 있는 언어이기 때문입니다.

이 프로젝트에는 세 가지 철학이 있습니다:

- 사용법이 간단해야 한다.
- 누구나 쉽게 이용할 수 있어야 한다. [1]
- “인터넷 민주주의는 효과적이다.”

위의 항목 중 하나라도 어긋나는 것이 있다면 제보 부탁드립니다.

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- **KoNLPy 설치 방법 (맥 OS)**

- `pip3 install konlpy`

- **KoNLPy 설치 방법 (Ubuntu)**

- `sudo apt-get install g++ openjdk-7-jdk`
 - `sudo apt-get install python3-dev; pip3 install konlpy`

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

▪ KoNLPy 설치 방법 (Window)

- Java 설치
 - <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html>
 - 회원가입 후 메일 인증 / 자신의 운영체제 버전에 맞게 다운로드
- JAVA HOME 설정
 - https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/index.html
 - JAVA HOME 설정이 되어있지 않을 시, 실행 안 됨
- JPyype1을 다운로드 받고 설치
 - <https://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype>
 - > pip install --upgrade pip
 - > pip install JPype1-0.5.7-cp27-none-win_amd64.whl

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



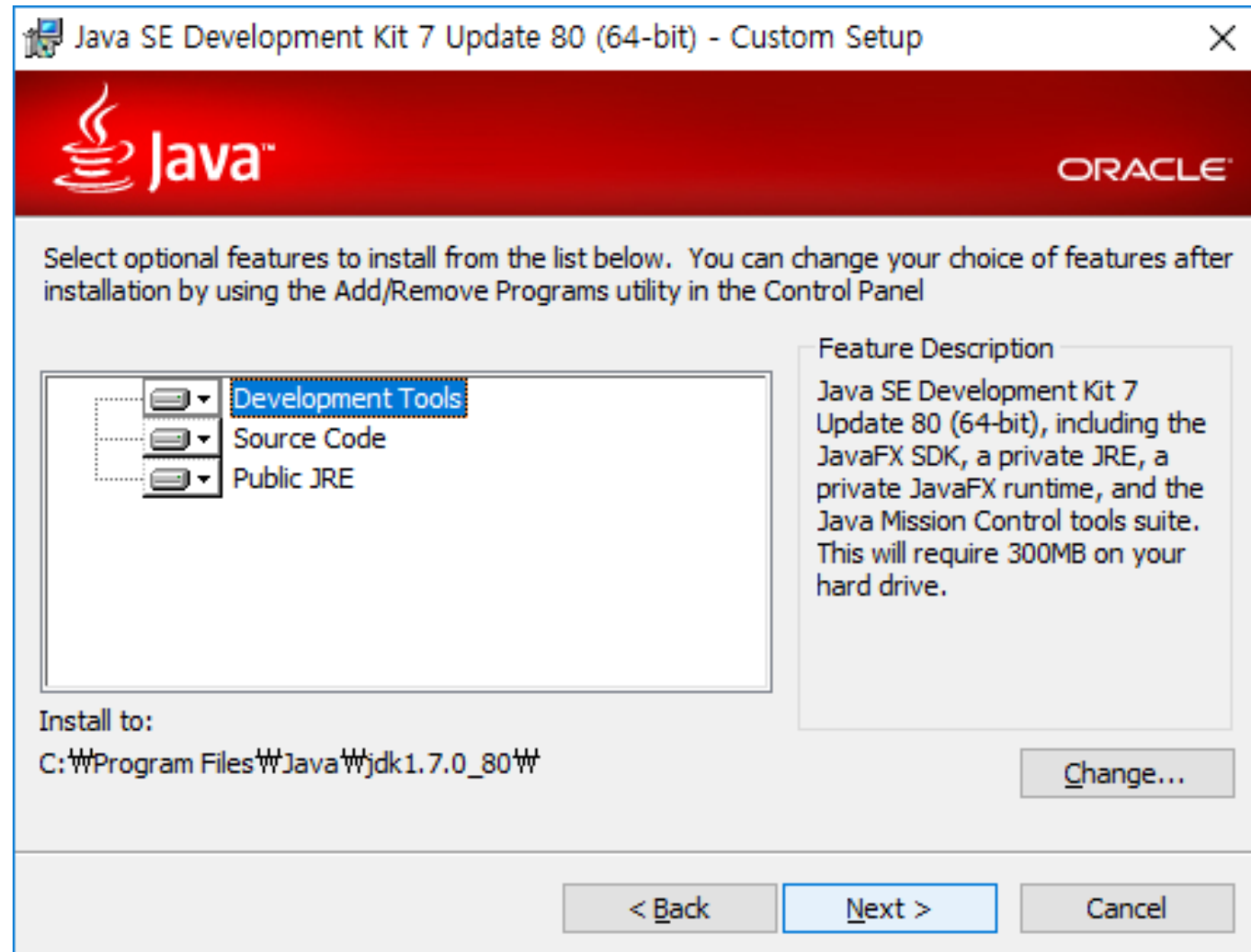
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



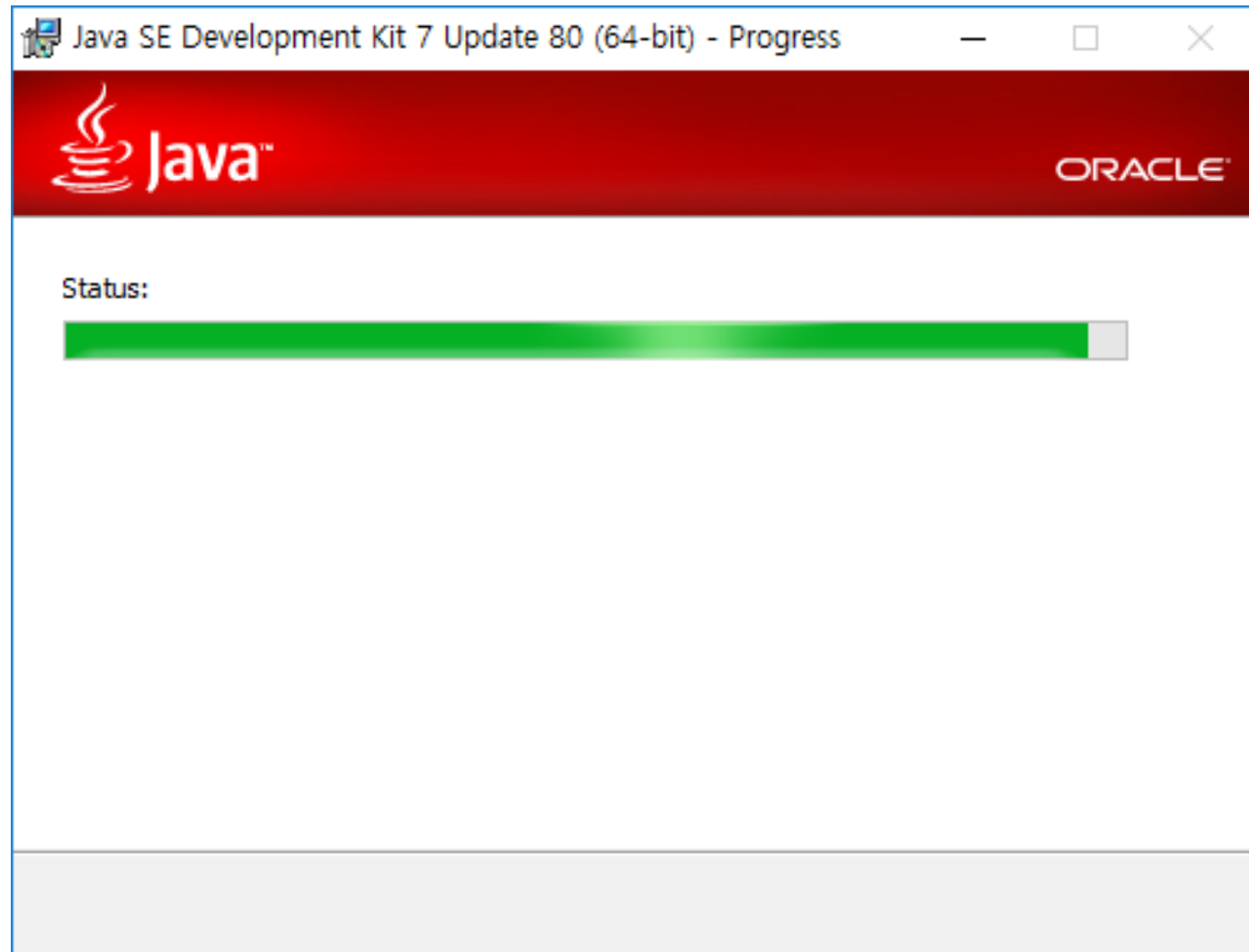
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- KoNLPy 설치 방법 (Window)



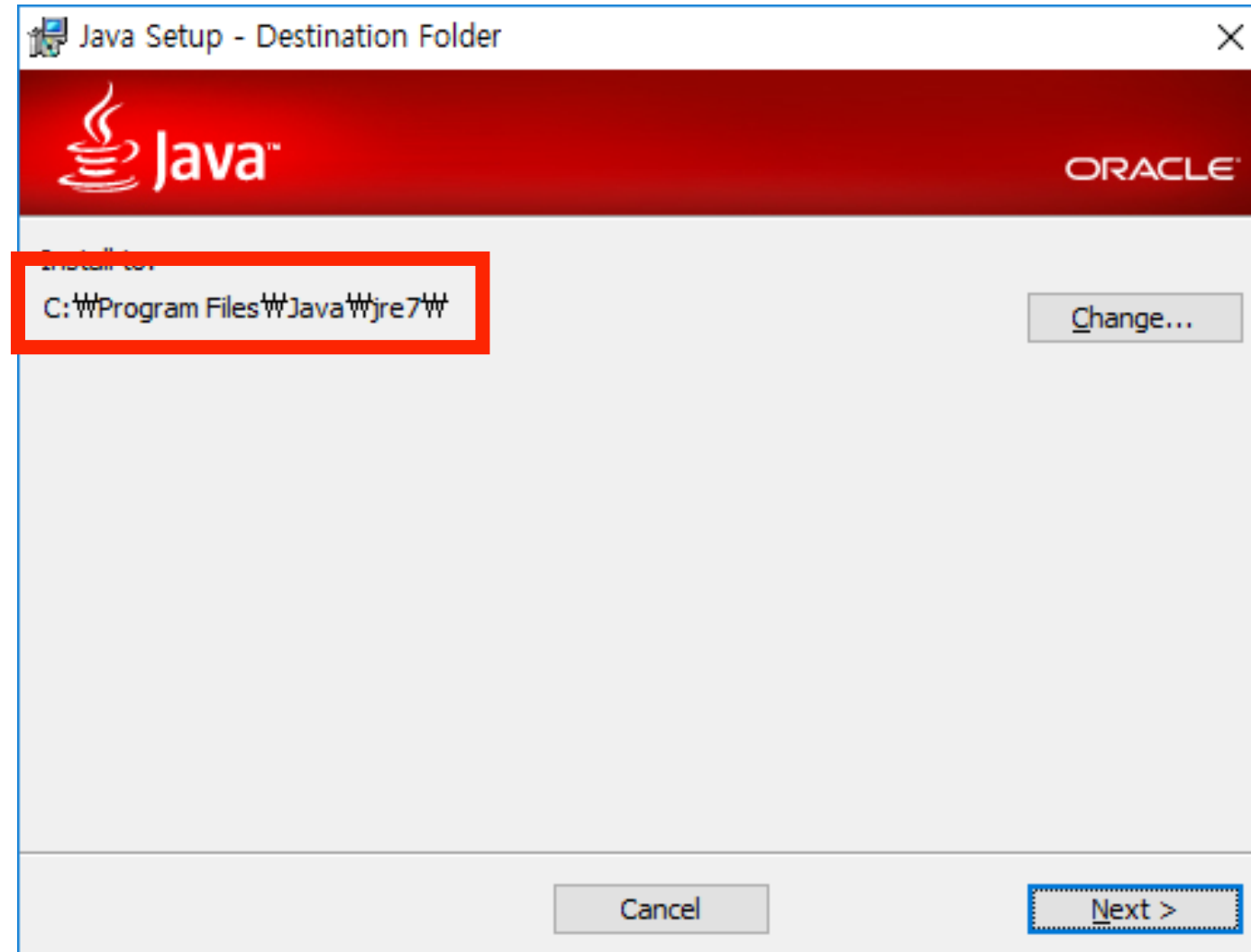
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

- KoNLPy 설치 방법 (Window)



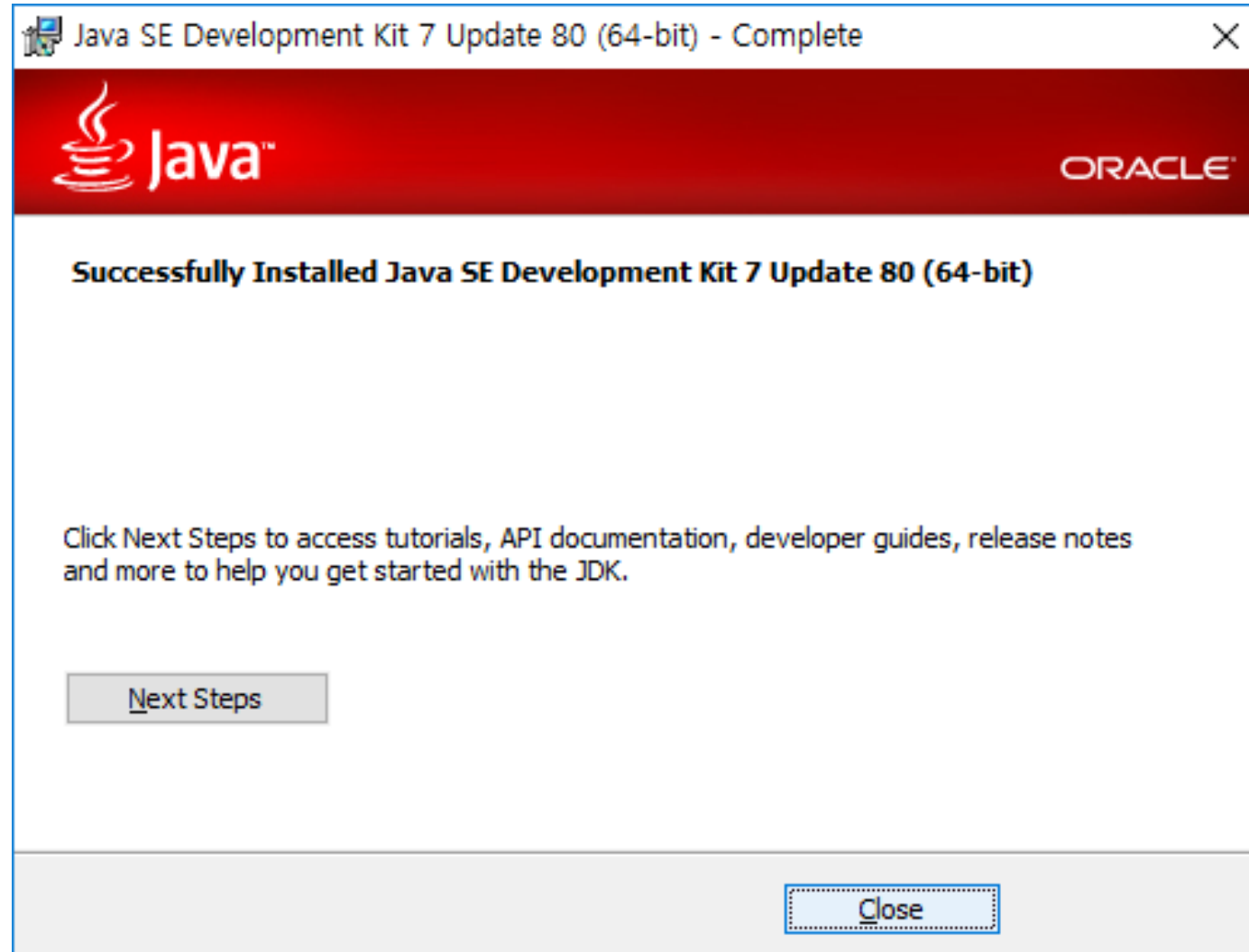
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



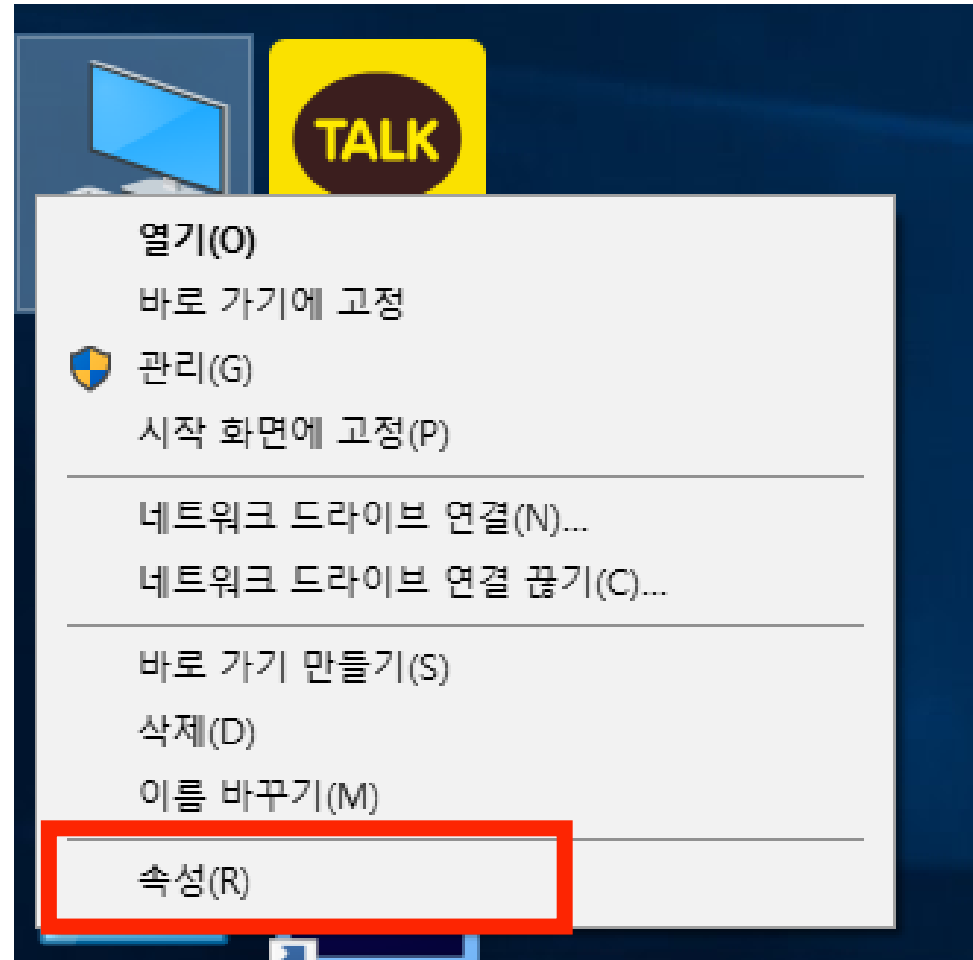
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



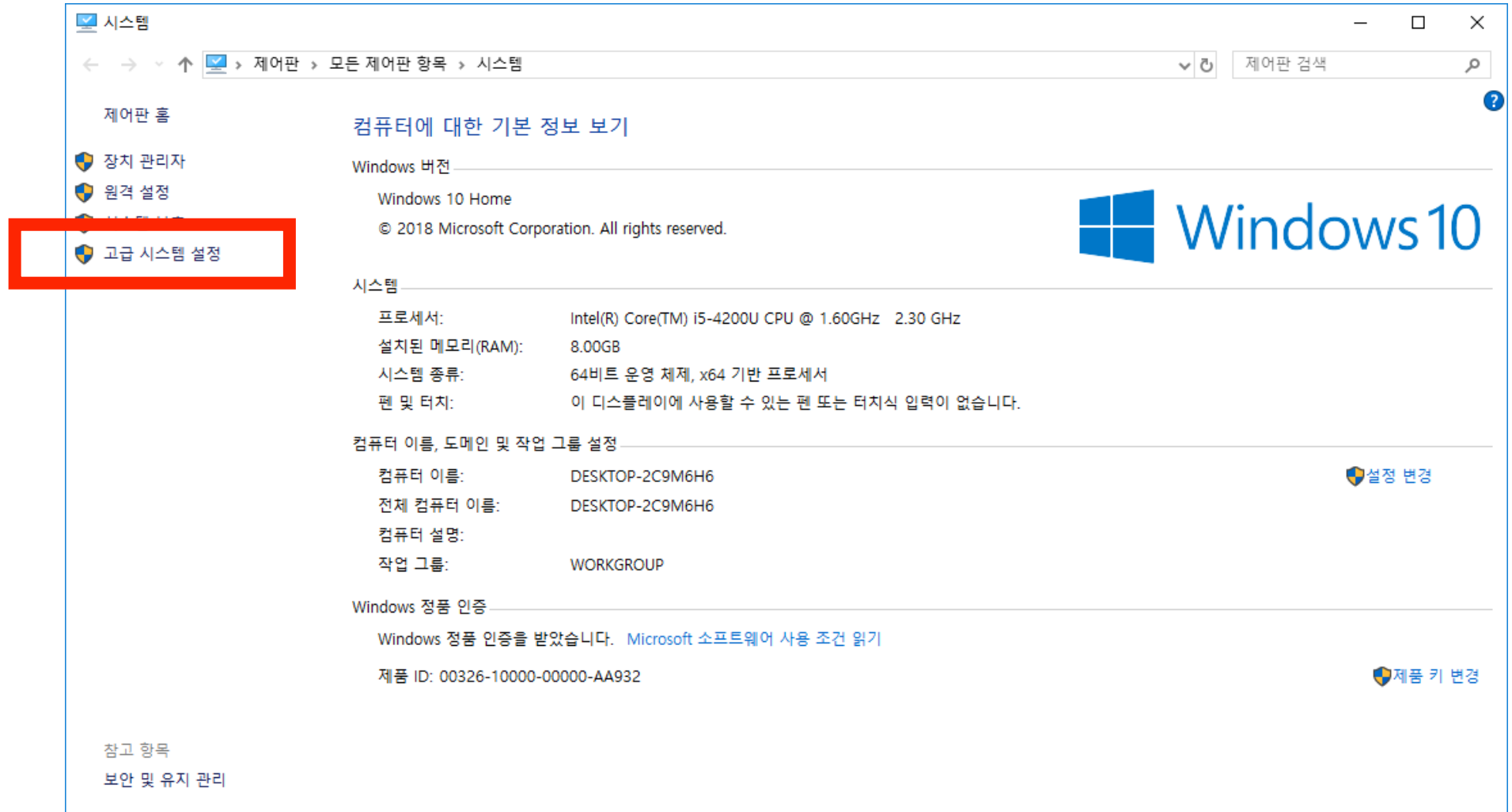
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



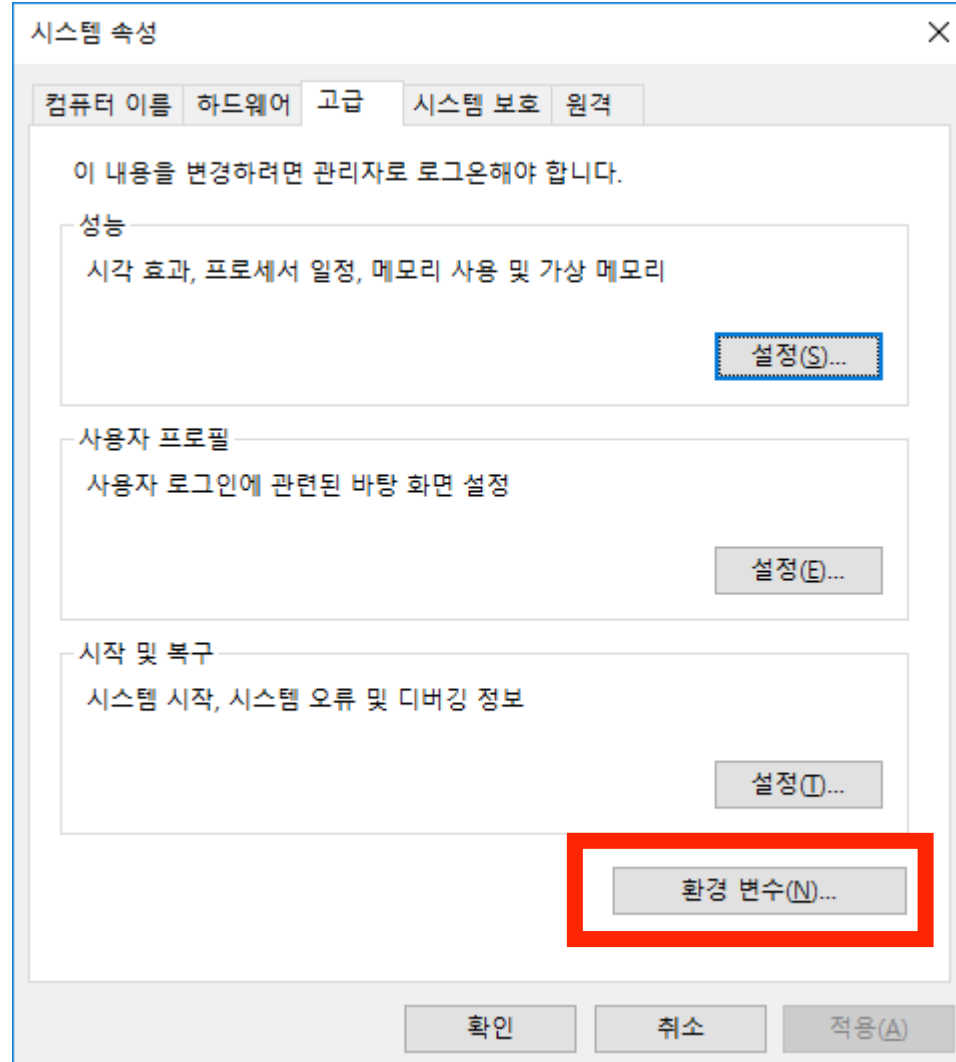
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



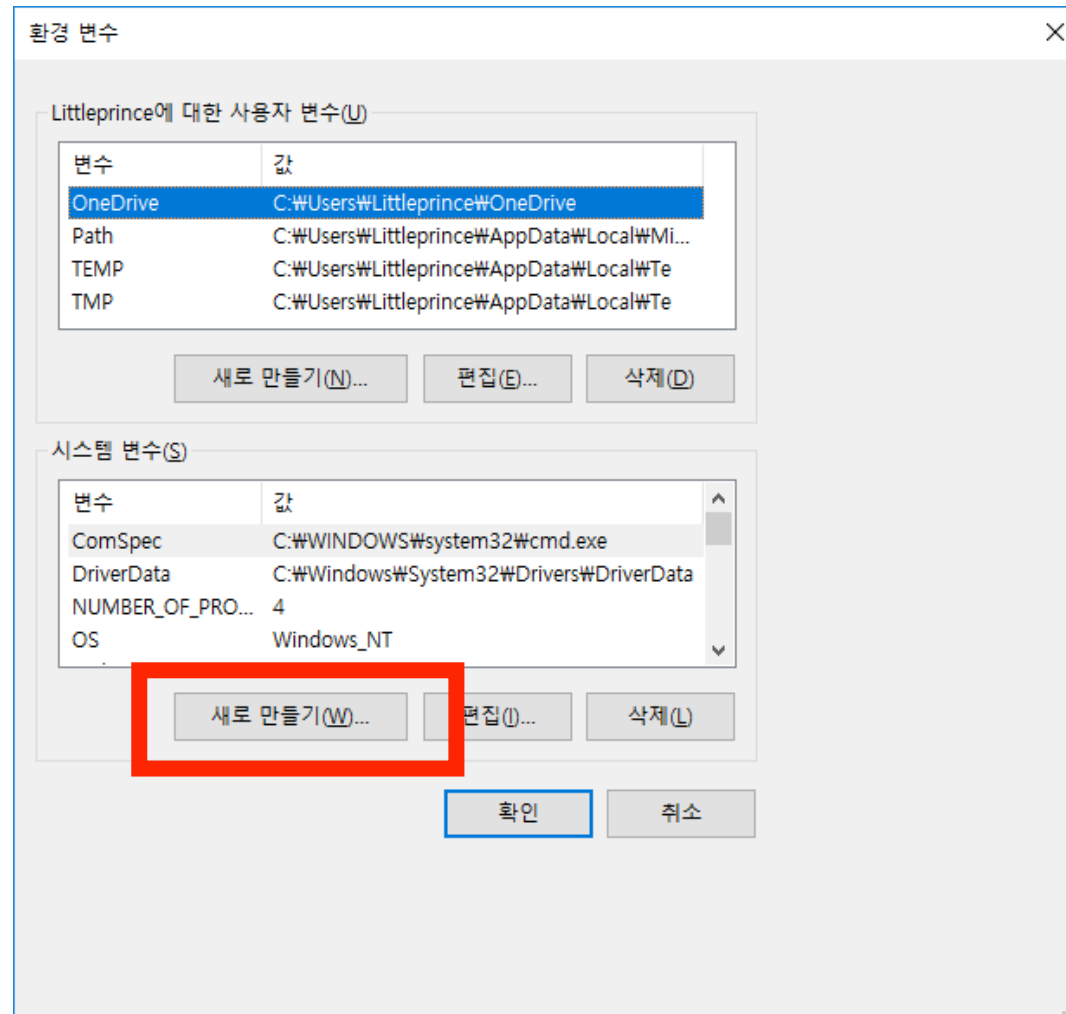
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



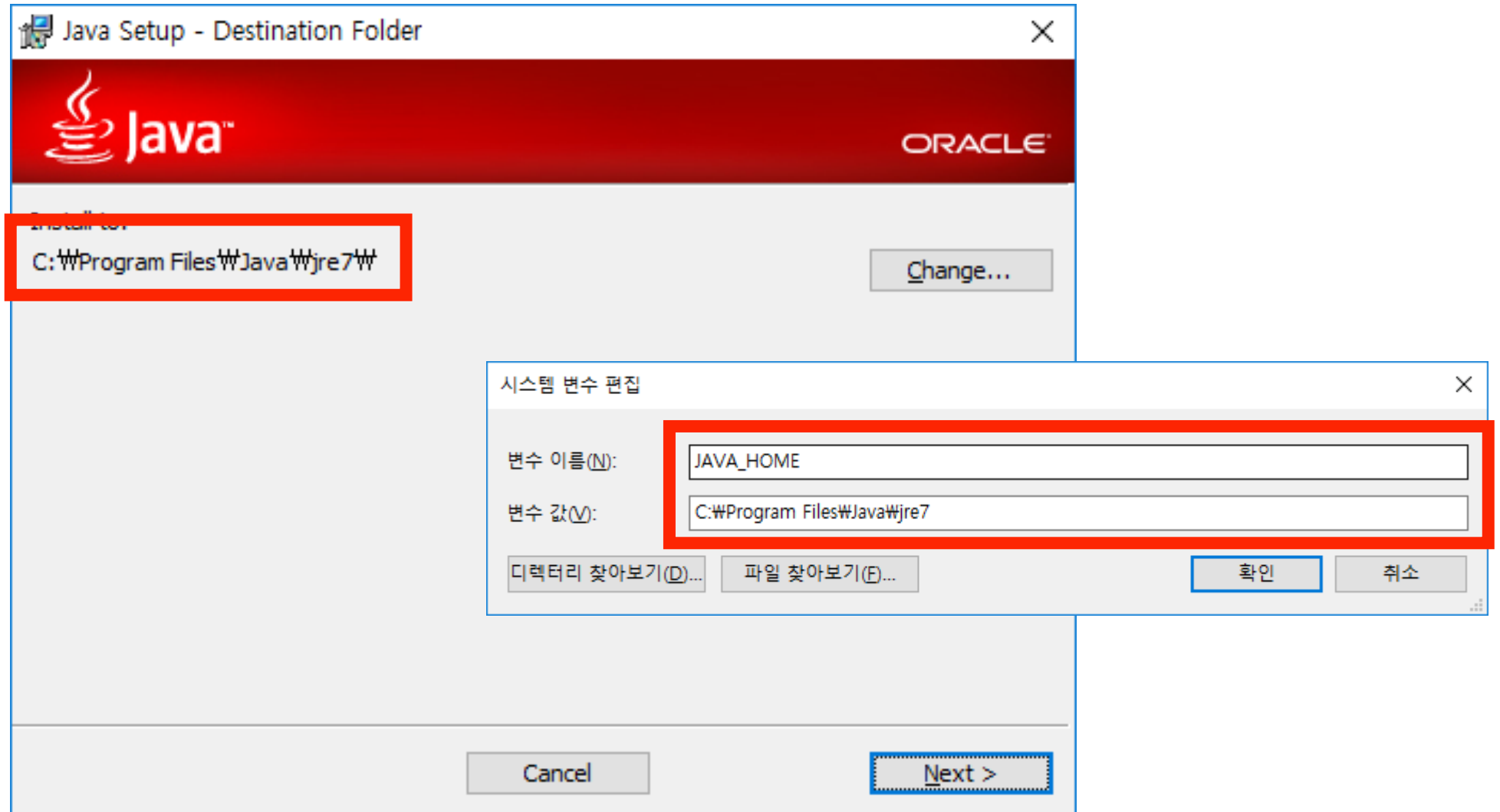
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



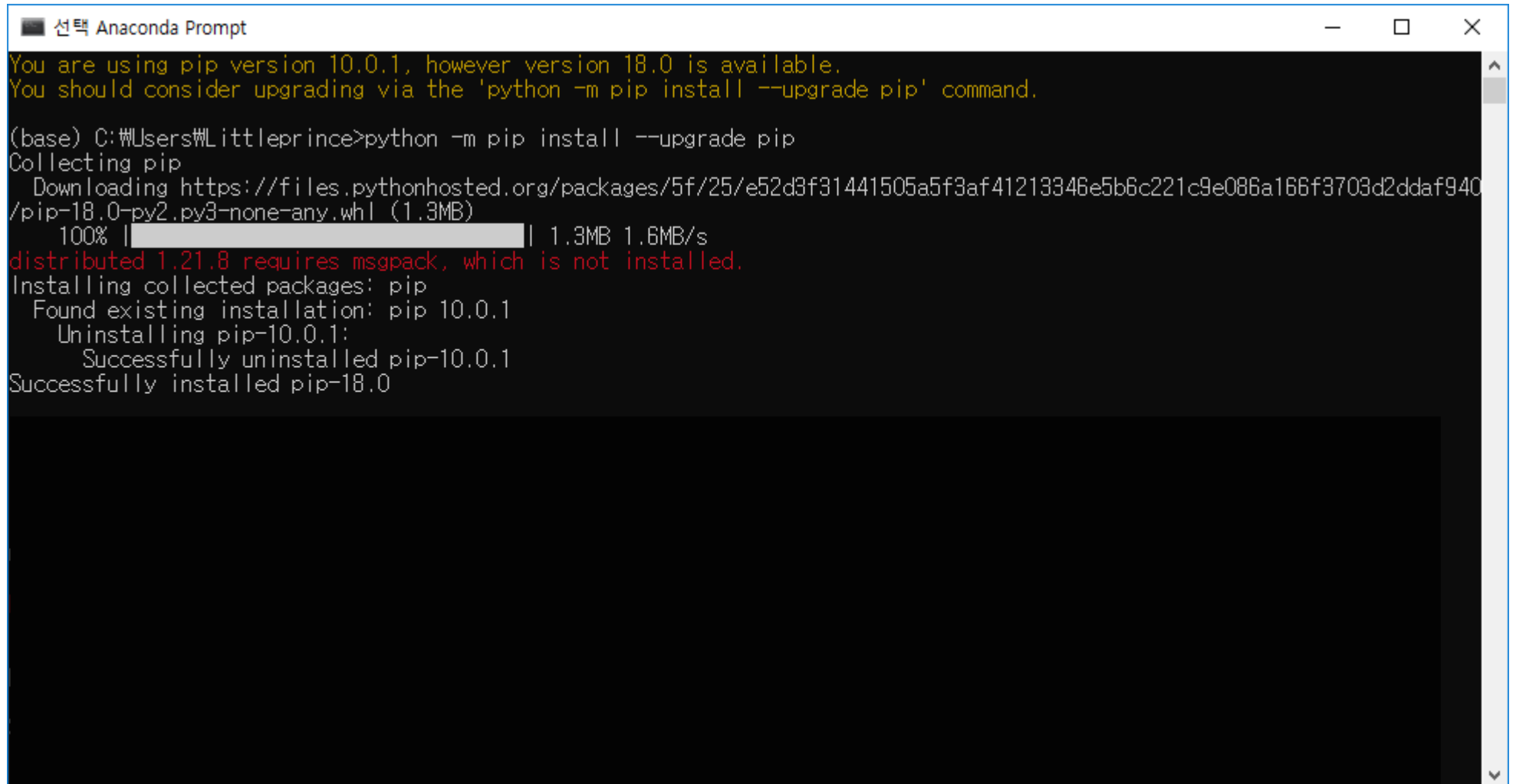
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



```
선택 Anaconda Prompt
You are using pip version 10.0.1, however version 18.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

(base) C:\Users\Littleprince>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/5f/25/e52d3f31441505a5f3af41213346e5b6c221c9e086a166f3703d2ddaf940/pip-18.0-py2.py3-none-any.whl (1.3MB)
    100% |#####| 1.3MB 1.6MB/s
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: pip
  Found existing installation: pip 10.0.1
    Uninstalling pip-10.0.1:
      Successfully uninstalled pip-10.0.1
Successfully installed pip-18.0
```

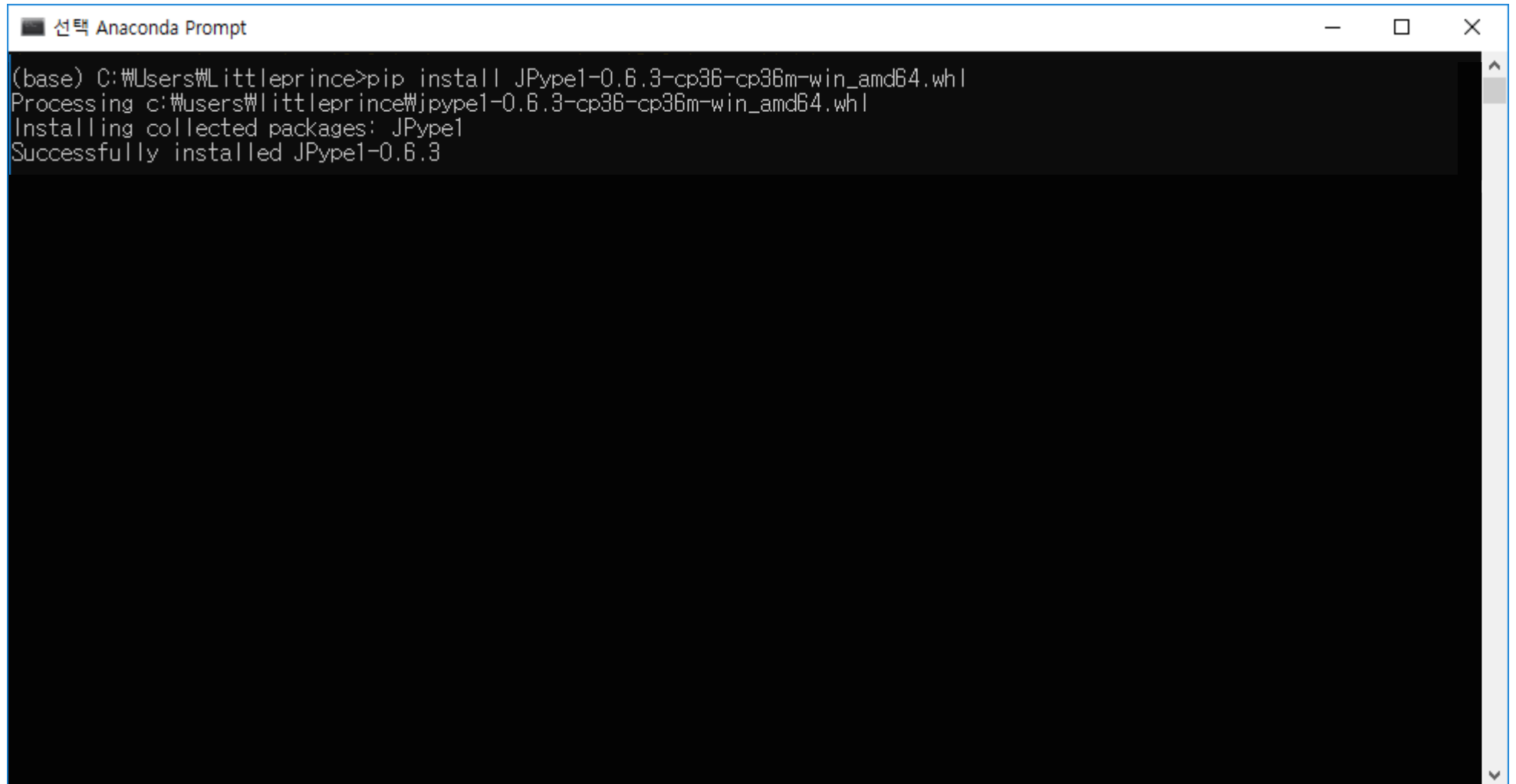
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



```

선택 Anaconda Prompt

(base) C:\Users\Littleprince>pip install JPype1-0.6.3-cp36-cp36m-win_amd64.whl
Processing c:\Users\Littleprince\JPype1-0.6.3-cp36-cp36m-win_amd64.whl
Installing collected packages: JPype1
Successfully installed JPype1-0.6.3
  
```

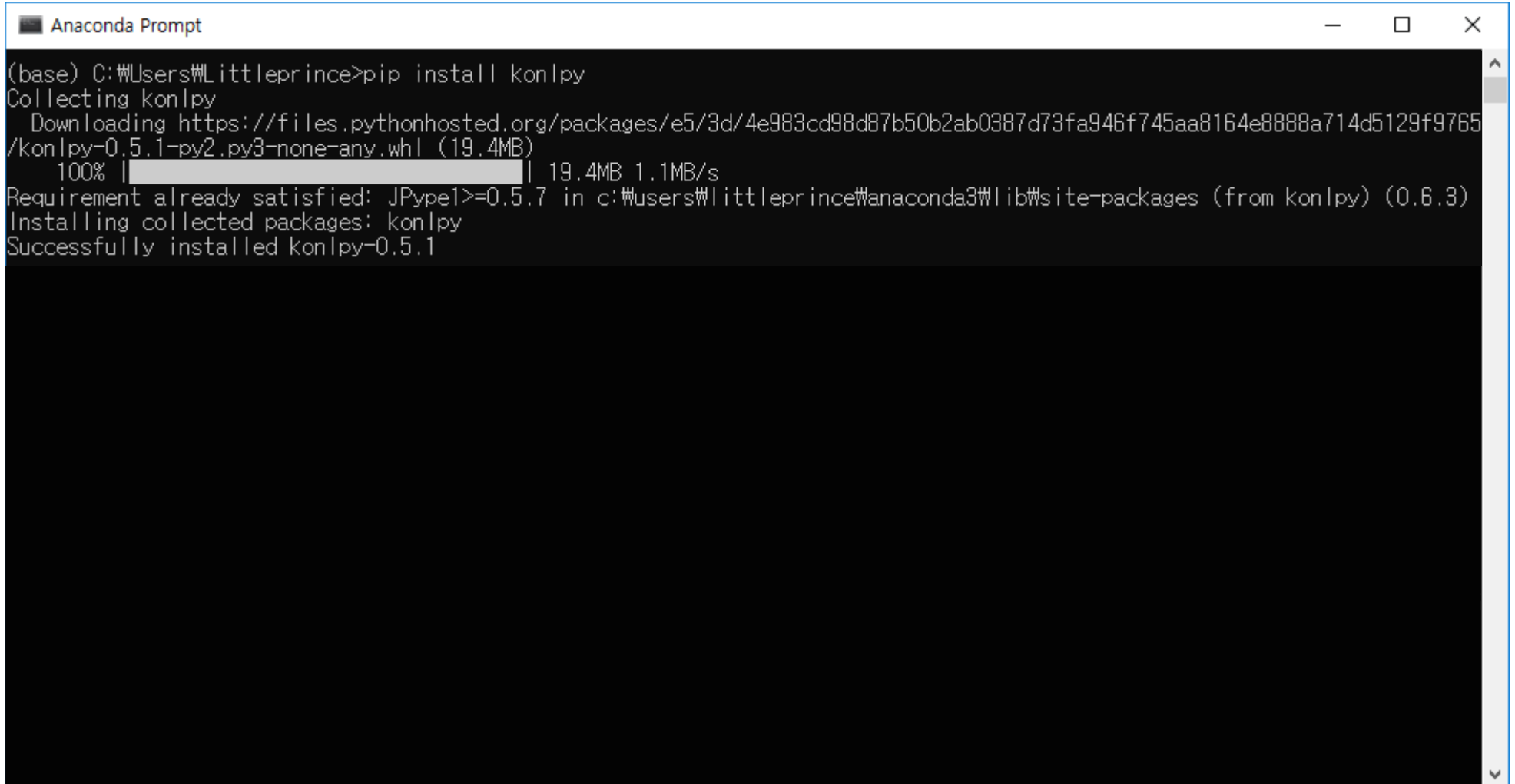
NLP

Bag of Words

Word2Vec

Sentimental
Analysis

■ KoNLPy 설치 방법 (Window)



```

Anaconda Prompt
(base) C:\Users\Littleprince>pip install konlpy
Collecting konlpy
  Downloading https://files.pythonhosted.org/packages/e5/3d/4e983cd98d87b50b2ab0387d73fa946f745aa8164e8888a714d5129f9765/konlpy-0.5.1-py2.py3-none-any.whl (19.4MB)
    100% |#####| 19.4MB 1.1MB/s
Requirement already satisfied: JPype1>=0.5.7 in c:\users\littleprince\anaconda3\lib\site-packages (from konlpy) (0.6.3)
Installing collected packages: konlpy
Successfully installed konlpy-0.5.1
  
```

NLP

Bag of Words

Word2Vec

Sentimental
Analysis

실 습