INNOCUS

파이썬을 활용한

빅데이터 기반
뉴스 트렌트 분석 1일차

부제 : 텍스트 마이닝

목 차

Ⅰ. 빅데이터 개요

1 빅데이터란?

Ⅰ. 빅데이터 개요

조직의 내외부에 존재하는 **다양한 형태의 데이터**를 **수집, 처리, 저장**하여
목적에 맞게 분석함으로써 해당분야의 **필요지식을 추출**하고,
전략적 **의사결정**에 활용하거나, **비즈니스 모델**, 또는 **서비스 모델**의
개발 및 개선, 수행 기술 또는 행위

1 빅데이터란?

Ⅰ. 빅데이터 개요

Volume

데이터 규모
(Peta, Zetta)

Velocity

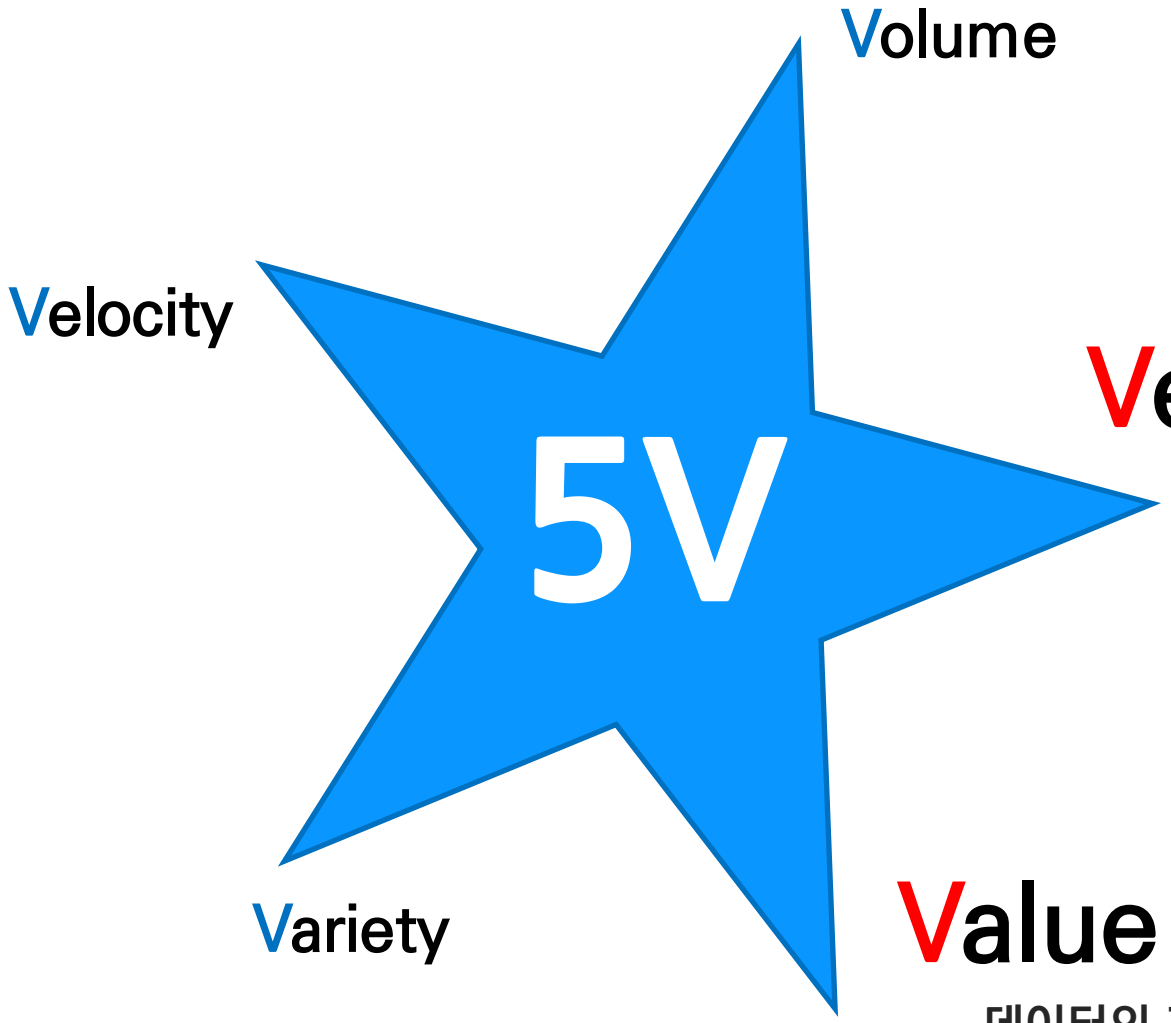
데이터 처리 속도
(실시간)

Variety

데이터의 다양성
(정형,비정형)

1 빅데이터란?

Ⅰ. 빅데이터 개요



Veracity

데이터의 정확성
(신뢰)

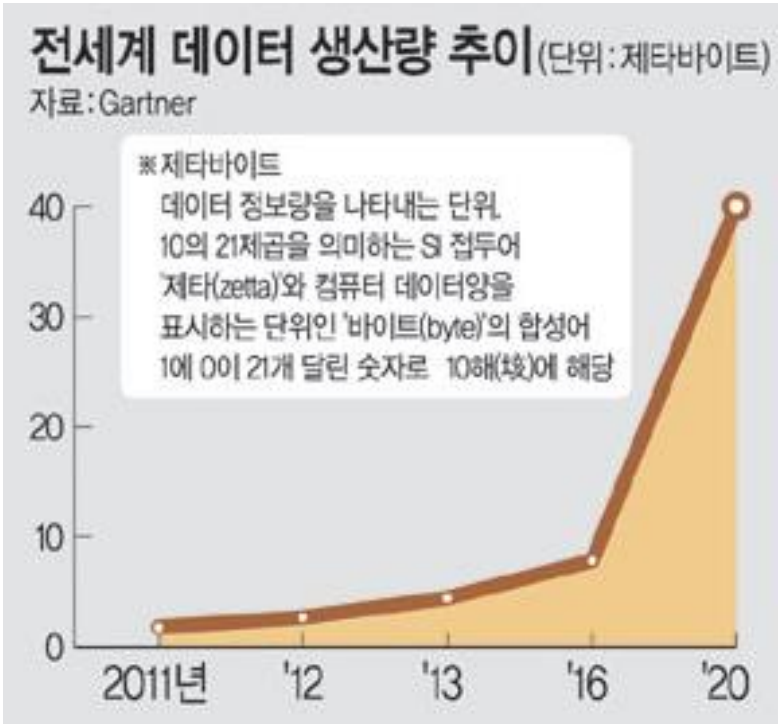
Value

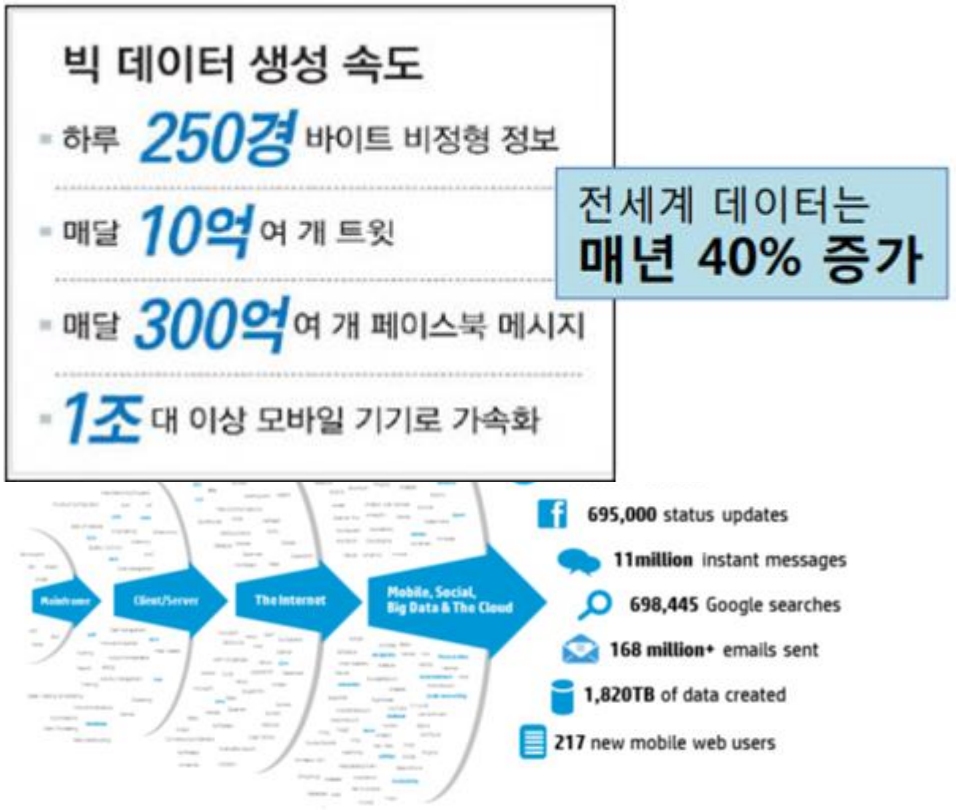
데이터의 가치
(목적)

2 빅데이터 등장 배경

Ⅰ. 빅데이터 개요

매년 데이터 생성량 40% 증가





2 빅데이터 등장 배경

Ⅰ. 빅데이터 개요

저렴한 메모리 비용으로 빅데이터 저장 비용 감소

2002년

펜티엄4 2.4GHz, 256MB RAM
80GB HDD, GeForce2 MX400

= 130만원

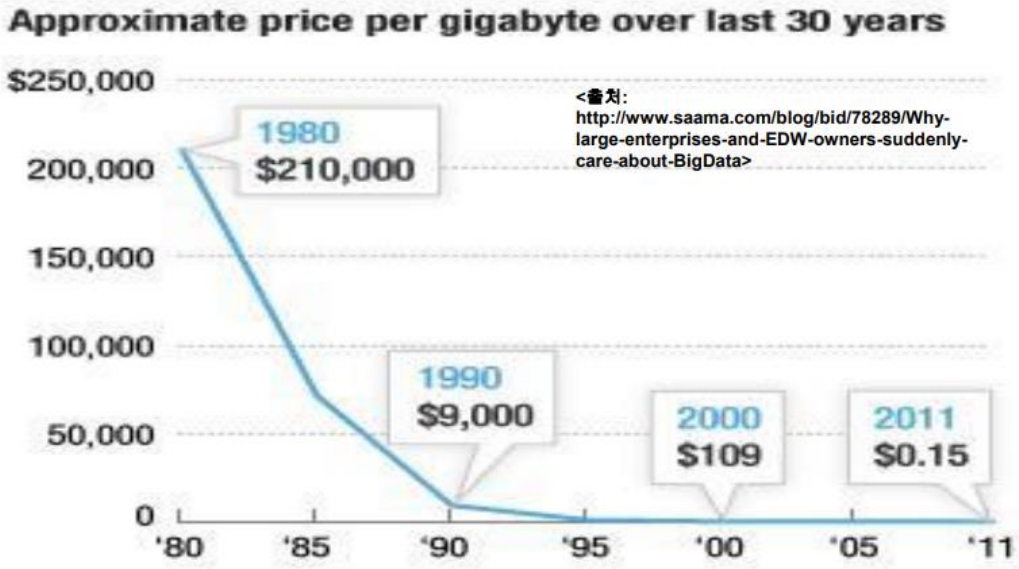
초코 다이제스티브 2,600통

2014년

Core i7-4770, 8GB RAM
2TB HDD, GeForce GTX770

= 124만원

다이제 초코 496통

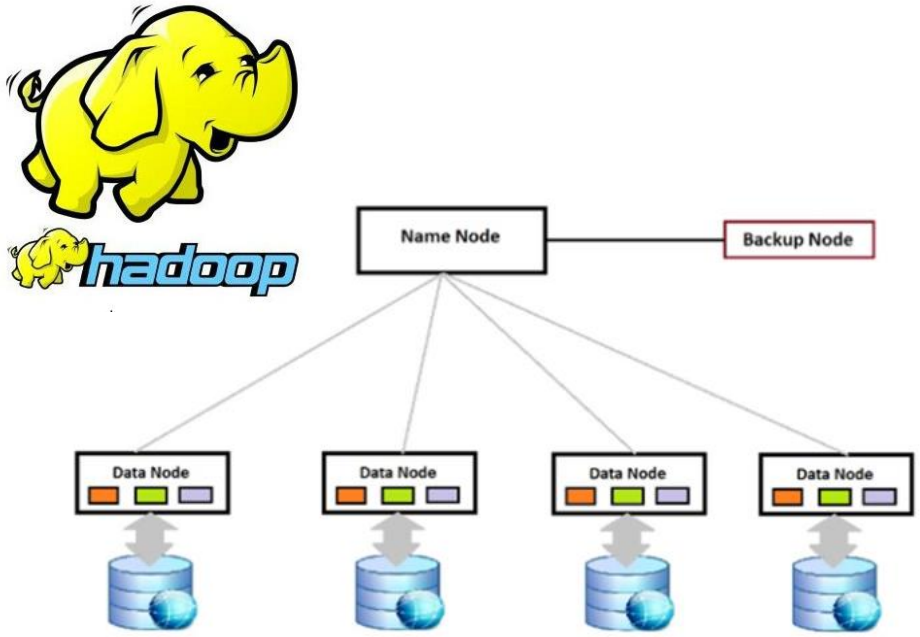


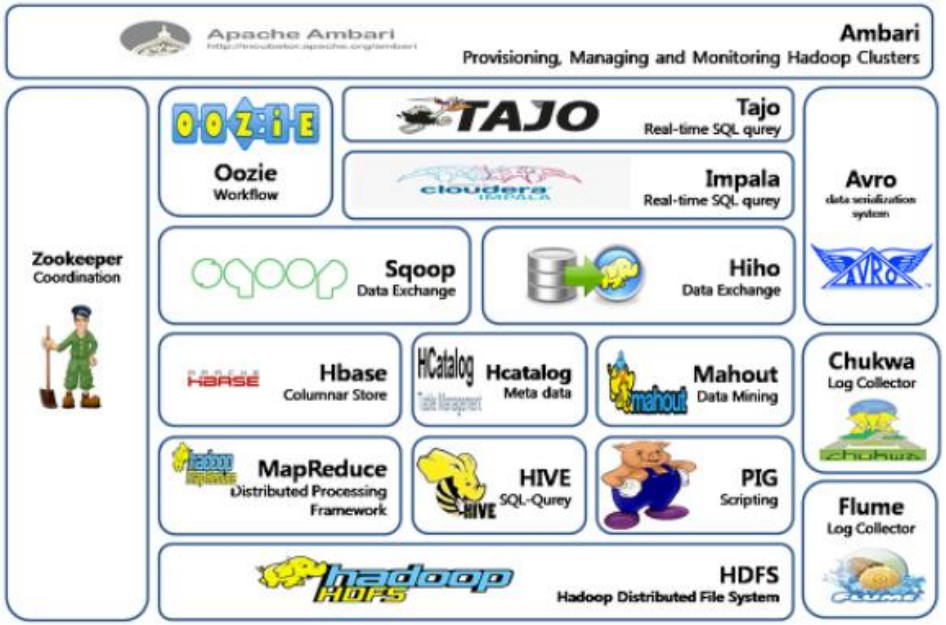
<출처:
http://www.saama.com/blog/bid/78289/Why-large-enterprises-and-EDW-owners-suddenly-care-about-BigData>

2 빅데이터 등장 배경

Ⅰ. 빅데이터 개요

분산처리 기술 발전을 통한 이전보다 빠른 빅데이터 처리





분산처리 기술 등장
(HDFS; Hadoop Distributed FileSystem)

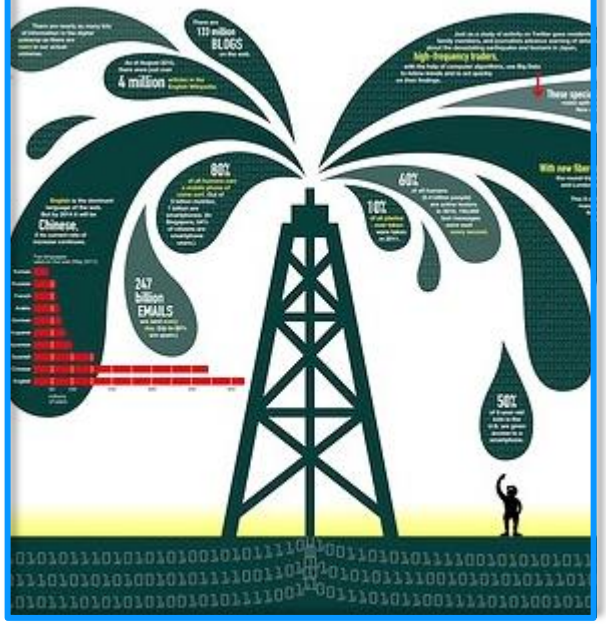
▶

하둡 에코시스템 발전

2 빅데이터 등장 배경

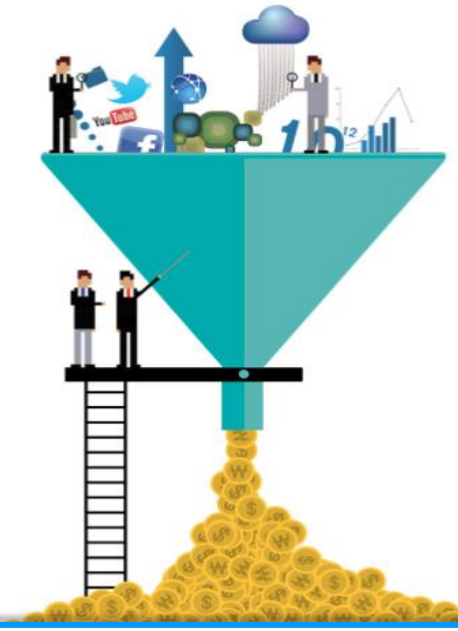
Ⅰ. 빅데이터 개요

빅데이터는
미래 경쟁력을
좌우하는 21세기의 원유
- Gartner



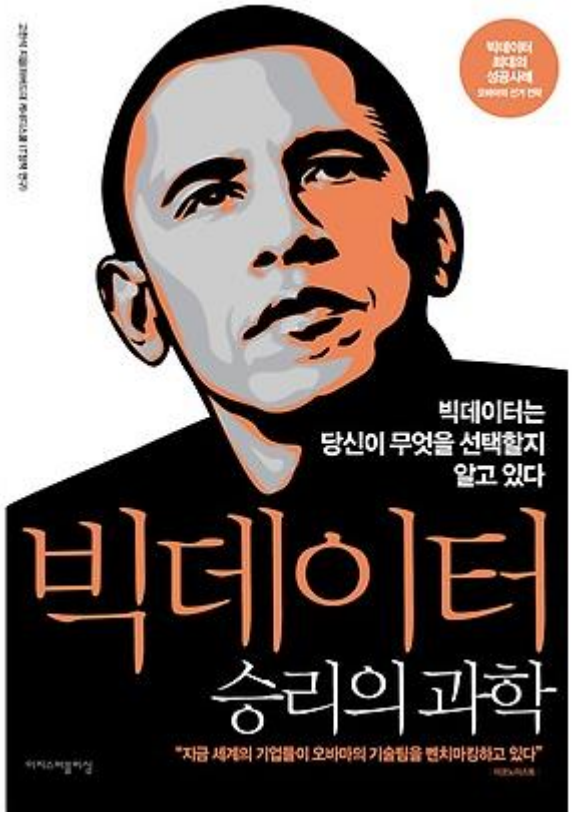
빅데이터는
혁신, 경쟁과 생산성의
차세대 첨단 주자
- McKinsey

빅데이터는
화폐나 금처럼
새로운 자산
- Davos Forum



3 빅데이터 활용사례

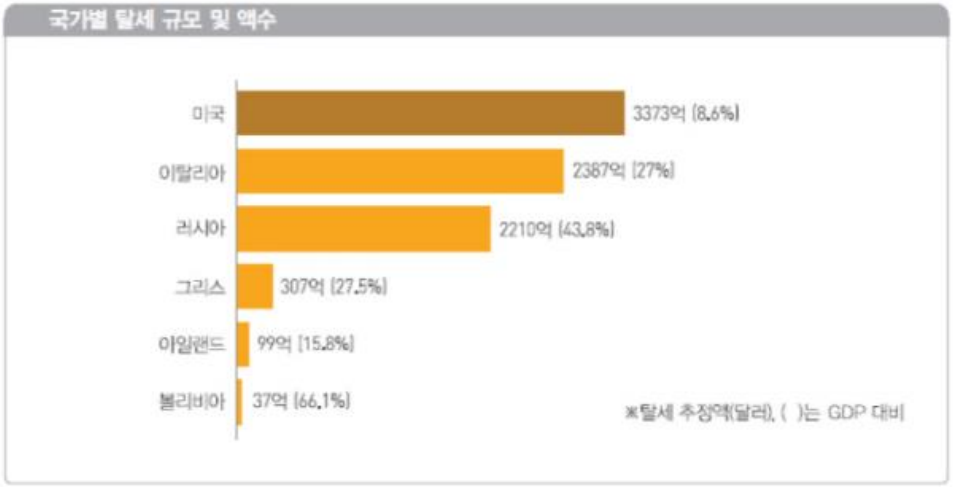
Ⅰ. 빅데이터 개요



2008년 미국 대통령 선거에서 **빅데이터팀**을 운영한 버락 오바마

3 빅데이터 활용사례

Ⅰ. 빅데이터 개요



자료 : 포린폴리시

탈세 및 사기 범죄 예방시스템을 통해 미국 국세청은

연간 **3450억달러(약 388조원)효과**가 있을 것으로 예측

3 빅데이터 활용사례

Ⅰ. 빅데이터 개요

아마존의
주문 예측을 통한
배송 시스템

구글의
자율 주행자동차

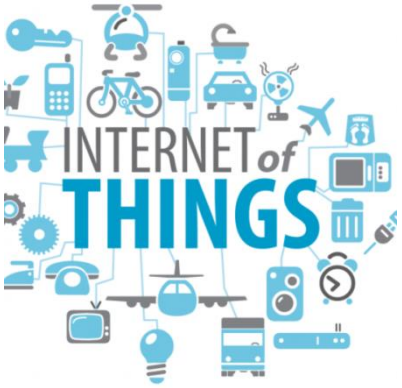
4 빅데이터 기반한 기술 융합

Ⅰ. 빅데이터 개요

빅데이터

IoT

다양하고 많은
데이터 수집

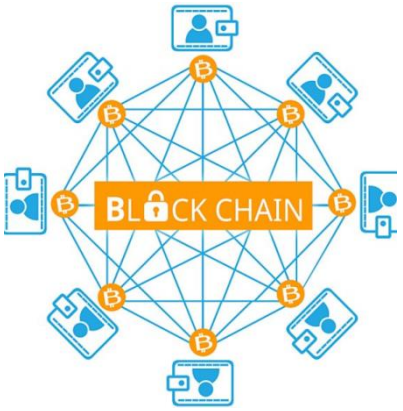


클라우드

빠른 데이터 처리
안정적인 데이터 저장

블록체인

높은 데이터 신뢰성
제공 데이터 거래



AI

데이터 학습을 통한
데이터의 활용

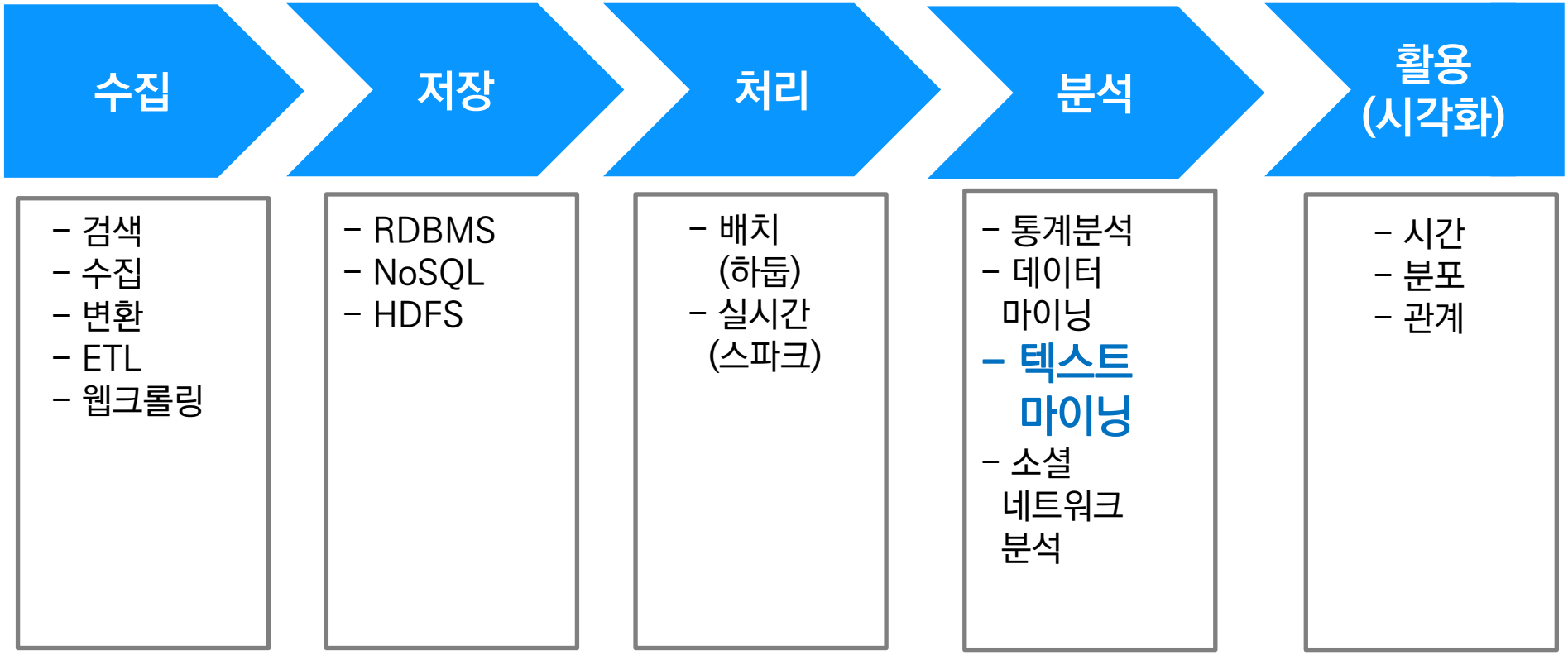
4차 산업 혁명

Ⅱ. 텍스트마이닝 개요

1 텍스트마이닝이란?

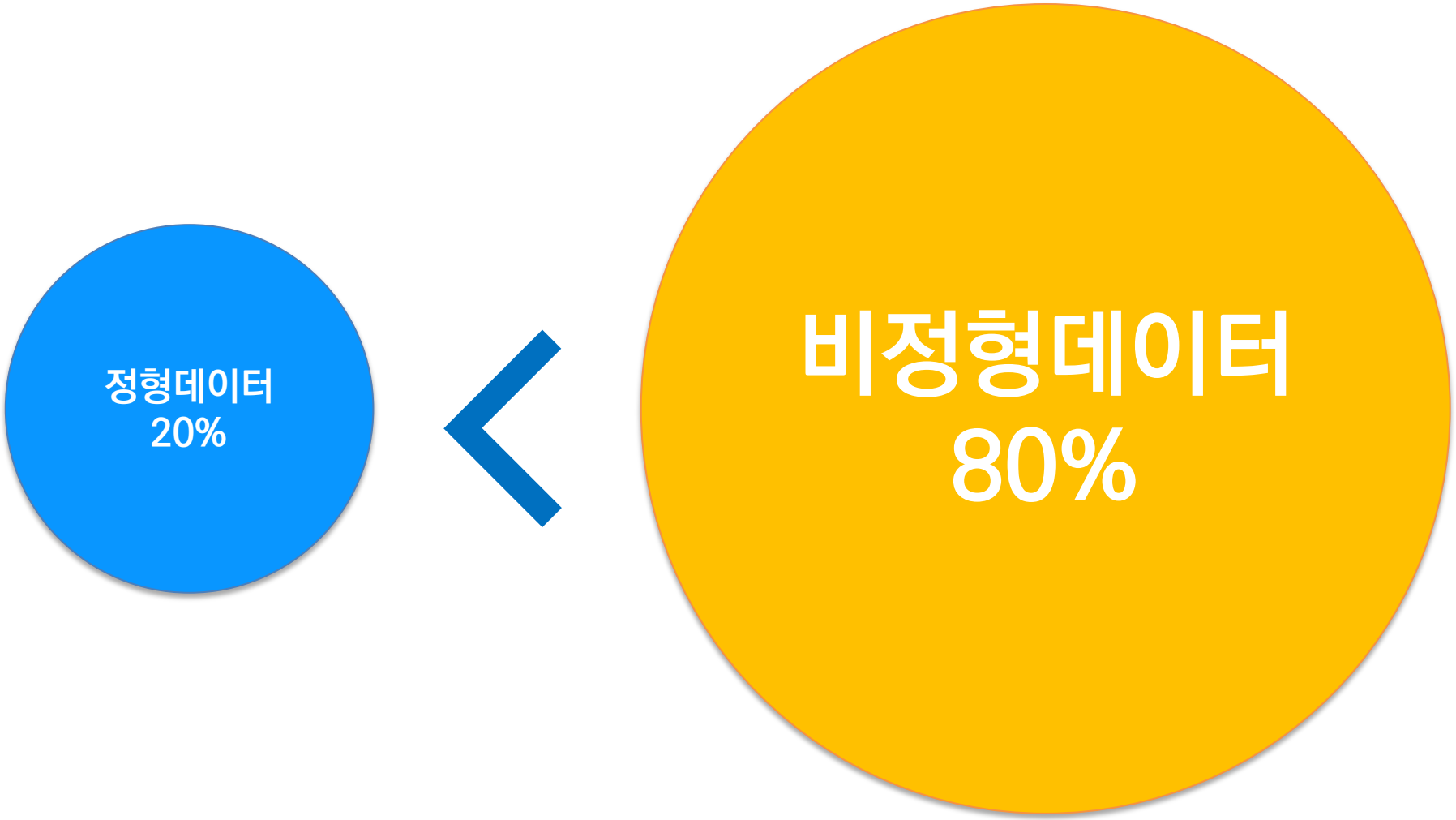
Ⅱ. 텍스트마이닝 개요

ㅇ 빅데이터 활용 절차



1 텍스트마이닝이란?

Ⅱ. 텍스트마이닝 개요



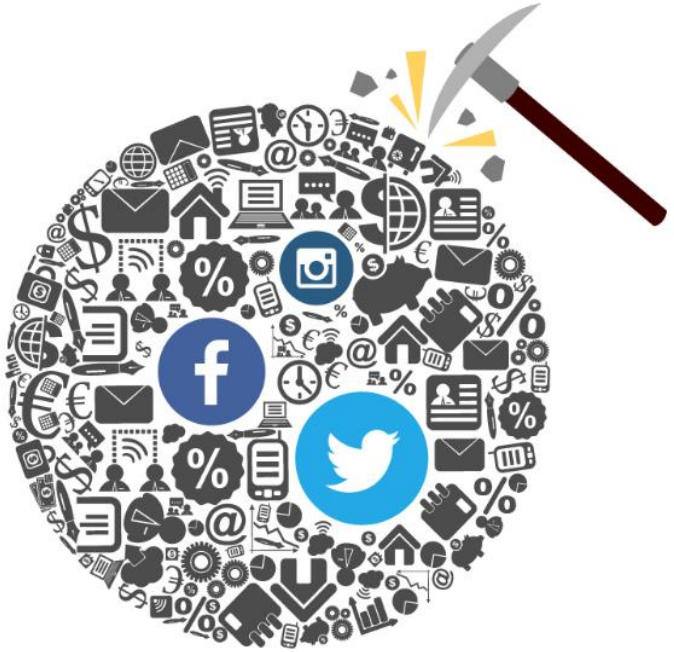
1 텍스트마이닝이란?

Ⅱ. 텍스트마이닝 개요

비정형화 텍스트에서
자연어처리 (NLP: Natural Language Processing)를 통해
유용한 **패턴**이나 **정보**를 찾는 것

1 텍스트마이닝 활용분야

Ⅱ. 텍스트마이닝 개요



- 문서 연관 분석
- 추천
- 문서 분류
- 문서 요약
- 챗봇
- 번역
- 음성 인식

2 텍스트마이닝을 어렵게 하는 문제들

Ⅱ. 텍스트마이닝 개요

- 추상적인 개념을 표현함에 있어서 모호함
- 개념들 간에 미묘하고 수많은 조합이 존재
- 유사한 개념(동의어, 유의어)를 표현하기 위한 다양한 방법 존재
- 텍스트 데이터의 고차원성 문제
- 개념을 시각화하기 어려움

3 자연어 처리 기술의 변화

Ⅱ. 텍스트마이닝 개요

자연어 처리
NLP(Natural Language Processing)

통계 기반의
자연어처리

자연어 이해
NLU(Natural Language Understanding)

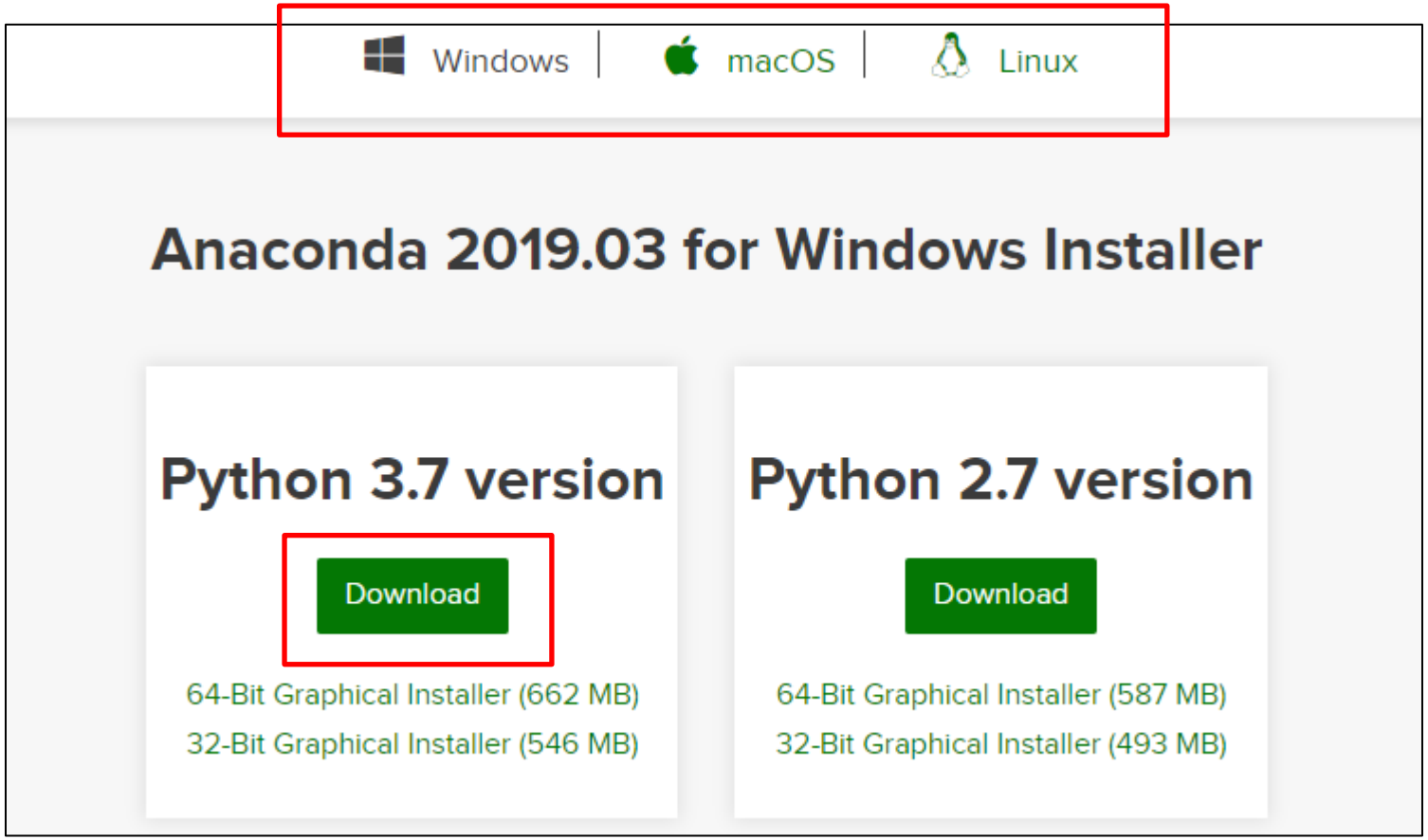
딥러닝 기반의
자연어처리

Ⅲ. 개발환경 구축

1 파이썬 배포판 (Anaconda) 설치

Ⅲ. 개발환경 구축

https://www.anaconda.com/distribution/#download-section

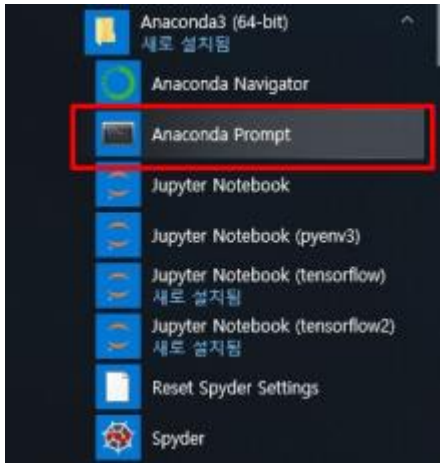


- 다운로드 완료 후 관리자 권한으로 실행 -> 설치

2 가상 파이썬 환경 생성

Ⅲ. 개발환경 구축

- 관리자 권한으로 아나콘다 프롬프트 (Anaconda Prompt) 실행




```
관리자: Anaconda Prompt

(C:₩ProgramData₩Anaconda3) C:₩WINDOWS₩system32>
```

- 가상 환경 목록 보기
  - 처음 실행한 경우 base 항목만 표시됨

```
관리자: Anaconda Prompt

(C:₩ProgramData₩Anaconda3) C:₩WINDOWS₩system32>conda info --envs
# conda environments:
#
root                  *  C:₩ProgramData₩Anaconda3
```

2 가상 파이썬 환경 생성

Ⅲ. 개발환경 구축

- 가상 환경 만들기

```
C:\WINDOWS\system32>conda create --name pyenv3 python=3.7
Fetching package metadata ...........
Solving package specifications: .

Package plan for installation in environment
C:\ProgramData\Anaconda3\envs\pyenv3:

The following NEW packages will be INSTALLED:

    certifi:         2016.2.28-py36_0
    pip:             9.0.1-py36_1
    python:          3.6.2-0
    setuptools:      36.4.0-py36_1
    vc:              14-0
    vs2015_runtime:  14.0.25420-0
    wheel:           0.29.0-py36_0
    wincertstore:    0.2-py36_0

Proceed ([y]/n)? y
```

2 가상 파이썬 환경 생성

Ⅲ. 개발환경 구축

- 가상 환경 만들기 (계속)

```
# To activate this environment, use:
# > activate pyenv3
#
# To deactivate an active environment, use:
# > deactivate
#
# * for power-users using bash, you must source
#□□□□
```

- 설치된 가상 환경 확인 (가상 환경 목록 보기)

```
관리자: Anaconda Prompt

(C:₩ProgramData₩Anaconda3) C:₩WINDOWS₩system32>conda info --envs
# conda environments:
#
pyenv3                   C:₩ProgramData₩Anaconda3₩envs₩pyenv3
root                  *  C:₩ProgramData₩Anaconda3
```

2 가상 파이썬 환경 사용

Ⅲ. 개발환경 구축

- 가상 파이썬 환경 접속

```
관리자: Anaconda Prompt

(C:₩ProgramData₩Anaconda3) C:₩WINDOWS₩system32>activate pyenv3

(pyenv3) C:₩WINDOWS₩system32>
```

- 명령 프롬프트에서 대화형 프로그램 환경 실행

```
관리자: Anaconda Prompt - python
(pyenv3) C:₩WINDOWS₩system32>python
Python 3.6.2 |Continuum Analytics, Inc.| (default, Jul 20 2017, 12:30:02) [MSC v.1900 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello, python")
Hello, python
>>>
```

- 종료는 exit() 또는 quit() 함수 호출 또는 ctrl+z 입력

2 가상 파이썬 환경에 모듈 설치

Ⅲ. 개발환경 구축

- 가상 파이썬 환경 접속

```
관리자: Anaconda Prompt

(C:₩ProgramData₩Anaconda3) C:₩WINDOWS₩system32>activate pyenv3

(pyenv3) C:₩WINDOWS₩system32>
```

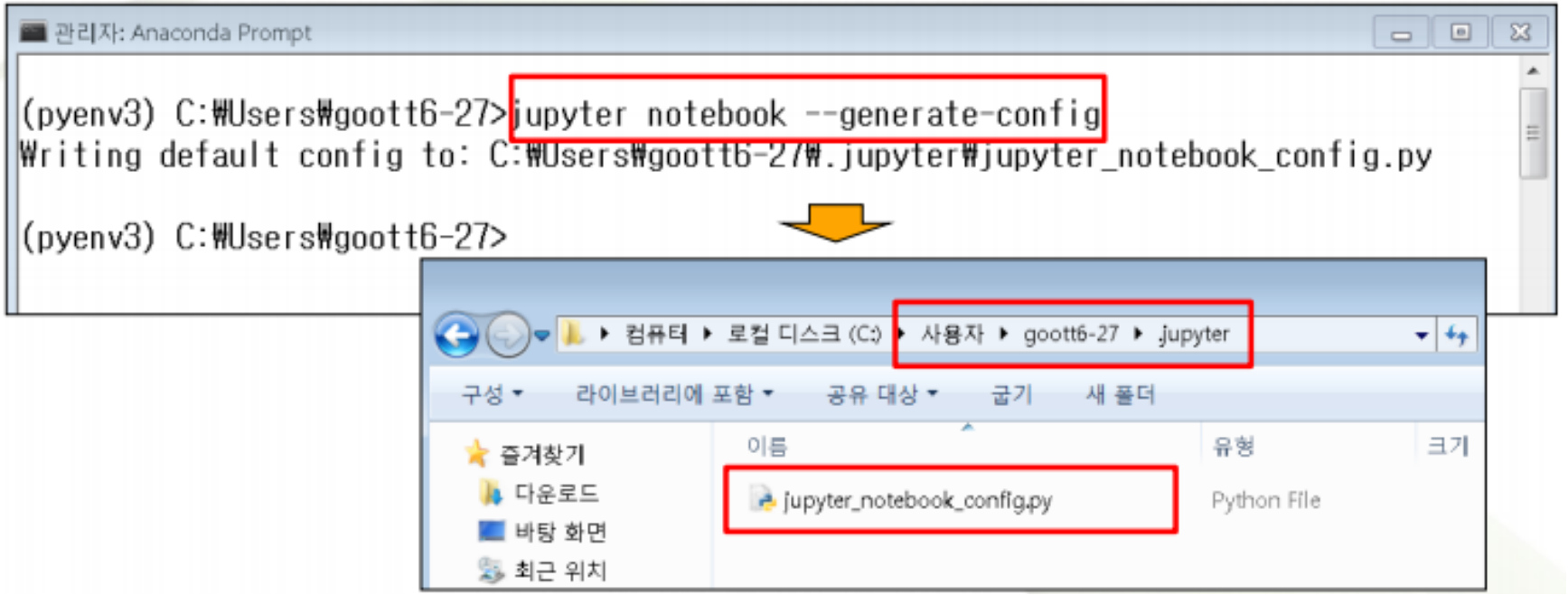
- 대화형 프로그램 개발을 위한 Jupyter Notebook 모듈 설치

```
(pyenv3) C:₩WINDOWS₩system32>conda install jupyter
Fetching package metadata .............
Solving package specifications: .

Package plan for installation in environment C:₩ProgramData₩Anaconda3₩envs₩pyenv3:

The following NEW packages will be INSTALLED:

    jupyter: 1.0.0-py36_3

Proceed ([y]/n)? y
```

2 대화형 실행 환경

Ⅲ. 개발환경 구축

- notebook 설정 파일 만들기 (선택적 – 명령행에서 직접 이동할 수 있음)



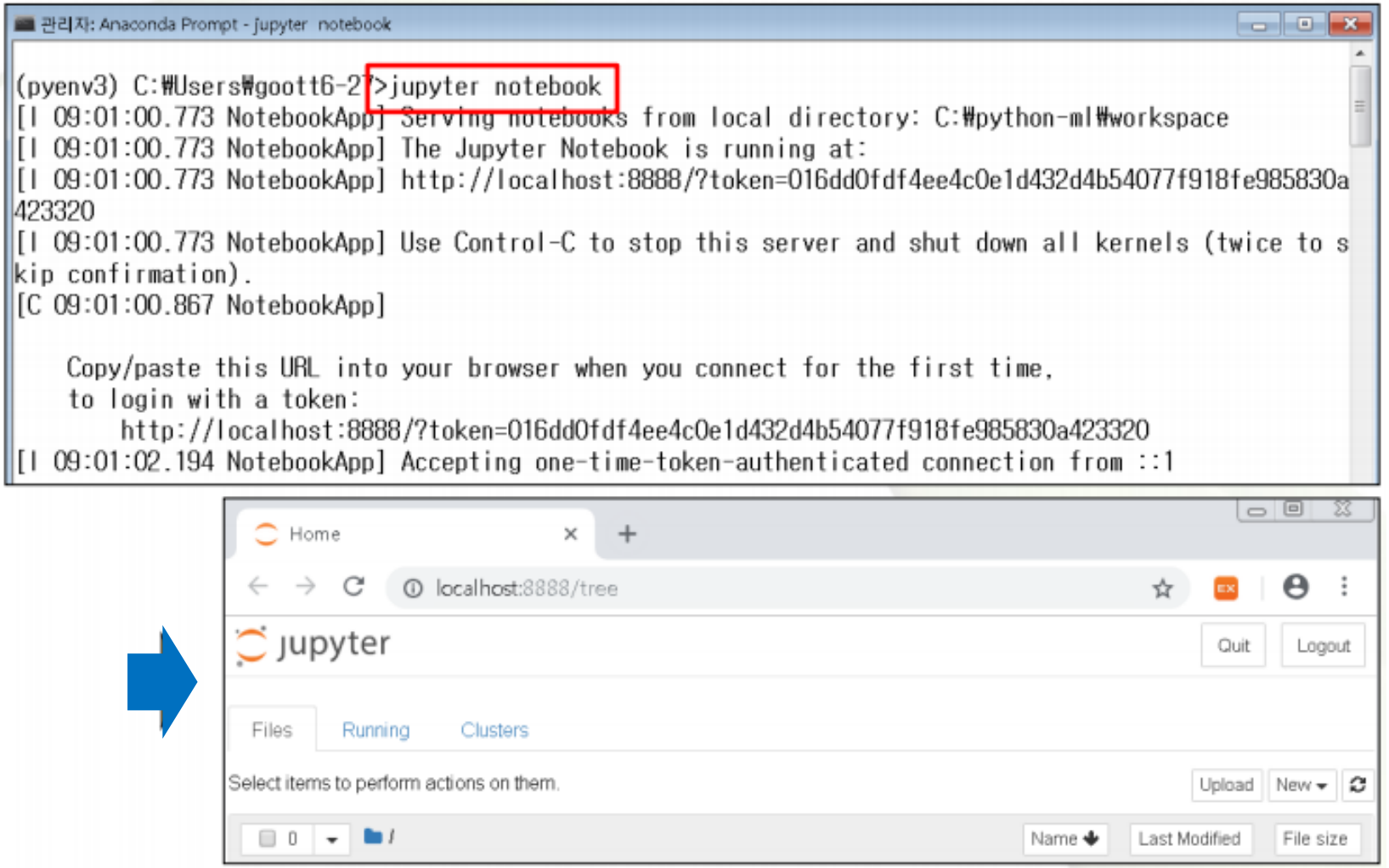
- notebook 설정 파일 수정 (선택적 – 명령행에서 직접 이동할 수 있음)

```
259
260  ## The directory to use for notebooks and kernels.
261  c.NotebookApp.notebook_dir = r'C:\python-ml\workspace'
```

2 대화형 실행 환경

Ⅲ. 개발환경 구축

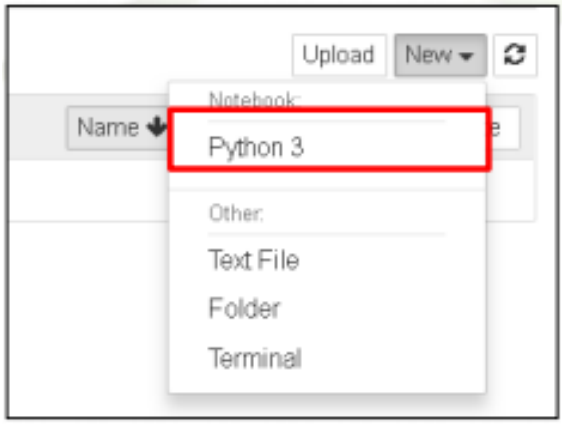
- notebook 시작



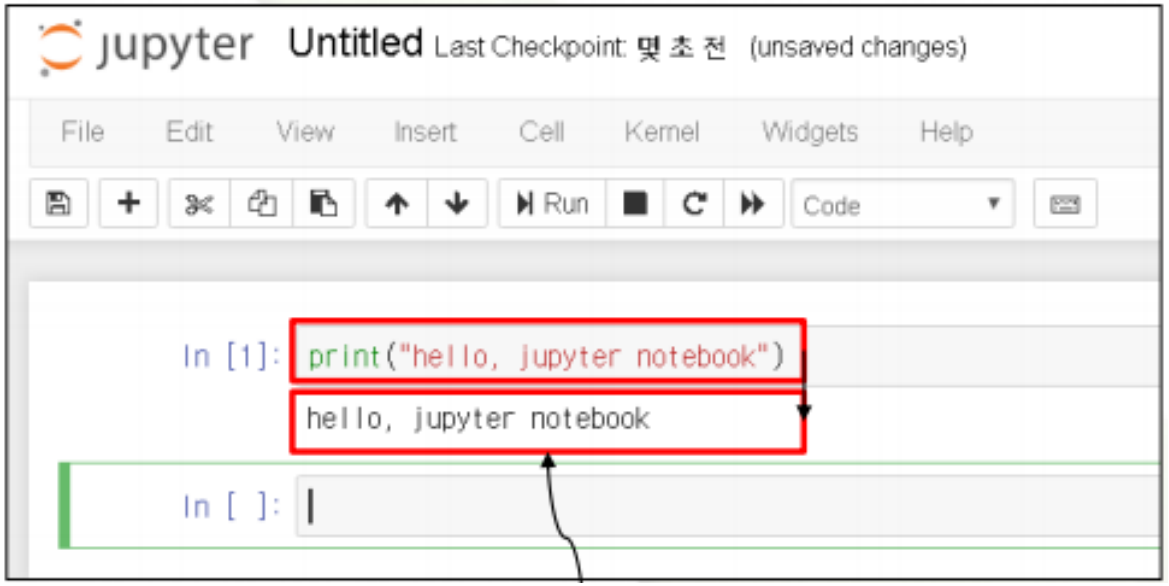
2 대화형 실행 환경

Ⅲ. 개발환경 구축

- 작업 파일 만들기



- 명령어 입력 후
  shift + enter 또는 ctrl + enter 를 통해 실행



Ⅳ. 자연어처리 기초

1 One-hot-encoding

Ⅳ. 자연어처리 기초

- 문자를 숫자로 바꾸는 기법 중 가장 대표적인 방법
- 어휘 사전 (vocabulary) : 분석할 텍스트에 포함된 모든 단어를 중복없이 리스트로 만든 것
- 특정 단어를 단어집합의 리스트 위치에 1로 나머지는 0으로 표현함

Family is an important thing.

-	Family	is	not	an	important	thing
Family	1	0	0	0	0	0
is	0	1	0	0	0	0
.....						
important	0	0	0	0	1	0
thing	0	0	0	0	0	1

2 BoW (Bag of Words)

Ⅳ. 자연어처리 기초

- 광범위하게 사용되는 간결하고 효과적인 텍스트 표현 기법
- 빈도수 기반의 방법론
- 구조와 상관없이 단어의 출현 횟수만 계산
  - 장, 문장, 서식 등 입력 텍스트의 구조 대부분은 유실
  - 텍스트에 각 단어가 얼마나 많이 나타나는지 계산

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

2 BoW (Bag of Words)

Ⅳ. 자연어처리 기초

- Bag of Words 처리 단계

토큰화 (Tokenization) — 각 문서를 공백, 구두점 등을 단어(토큰)로 분리

▼

어휘 사전 구축 — 모든 문서에 나타난 모든 단어를 모으고 번호 부여

▼

인코딩 — 어휘 사전의 단어가 각 문서마다 몇 번 나타나는지 계산

3 문서 단어 행렬 (Document-Term Matrix, DTM)

- 문서 단어 행렬 (DTM)
  - 서로 다른 문서들의 BoW 들을 결합한 표현 방법

문서1 : 먹고 싶은 사과
문서2 : 먹고 싶은 바나나
문서3 : 길고 노란 바나나 바나나
문서4 : 저는 과일이 좋아요

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

3 문서 단어 행렬 (Document-Term Matrix, DTM)

- 희소 행렬

대부분이 0 으로 채워진 문서 단어 행렬(DTM)

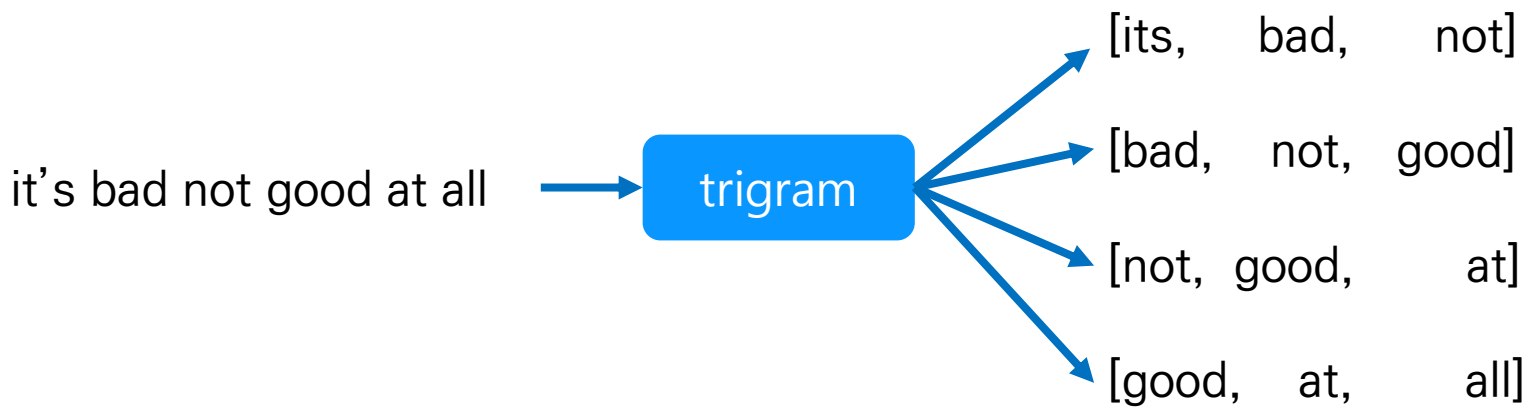
-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

4 n-gram (여러 단어로 만든 BOW)

Ⅳ. 자연어처리 기초

- BoW 는 단어의 순서를 고려하지 않기 때문에 문맥에 따른 의미 차이를 반영하지 못함
  - it's bad, not good at all == it's good, not bad at all

- 연속된 2개 이상의 토큰을 함께 고려해서 문맥 반영
  - unigram(1개), bigram(2개), trigram(3개), 4-grams, n-gram



5 전처리 - 토큰화(Tokenization)

Ⅳ. 자연어처리 기초

- 토큰화(Tokenization)
  - 주어진 코퍼스(corpus) 에서 토큰(token)이라 불리는 단위로 나누는 작업.
  - 토큰은 의미가 있는 단위를 말하며, 일반적으로 단어, 문장 단위로 토큰화를 함
- 토큰화에서 고려해야할 사항
  - 단순 띄어쓰기로 하면 안된다

    ex) “New York”
  - 구두점이나 특수 문자를 기준으로 단순 토큰화를 하면 안된다

    ex) “don’t” , “36.5” , “2018.12.31” , “AT&T”
- 한국어에서의 토큰화의 어려움
  - 한국어는 교착어
  - 영어와 달리 조사가 존재

6 전처리 - 정제(Cleaning)

Ⅳ. 자연어처리 기초

- 정제(Cleaning) : 갖고 있는 말뭉치(Corpus) 로 부터 노이즈 데이터를 제거
- 자연어가 아니면 아무 의미도 갖지 않는 글자를 제거

  ex) 특수문자
- 불용어(stopword) 제거 : 분석 목적에 필요 없는 단어를 제거

  ex) 관사(영어), 조사(한글) , 접미사, 주제와 연관성이 없는 단어
- 빈도수가 적은 단어 제거

  ex) 총 10만개의 문서 중에 5번 등장하는 단어가 있다면 제거
- 길이가 짧은 단어 제거

  ex) 일반적으로 영어 단어의 평균길이는 6~7, 한국어는 2~3 그 이하의 단어는 제거

  ex) it, at, to, on, in, by

  ex) 한글의 경우 토큰화의 한계로 한글자가 많이 나오는 경우 발생

7 전처리 - 정규화(Normalization)

Ⅳ. 자연어처리 기초

- 정규화(Normalization) : 표현 방법이 다른 단어들을 통합시켜서 같은 단어로 만드는 것
  - 영어의 경우 대소문자를 통합
  - 의미가 같은 단어는 하나의 단어로 통일
- 표제어 추출 (Lemmatization) : 표제어는 '기본 사전형 단어 '
  - 표제어 추출은 단어의 기본형태를 찾아 가는 것

    ex) am, are, is -〉 be, has -〉 have
  - 단어의 형태소를 파악해서 어간(stem)과 접사(affix)로 구분함

    ex) cats -〉 cat(어간) 과 -s(접사) 로 구분,
  - 표제어 사전이 구축 필요
- 어간 추출 (Stemming) : 단순 규칙에 따라 어미를 자르거나 치환하는 것

  ex) having -〉 hav, copy -〉 copi

실 습