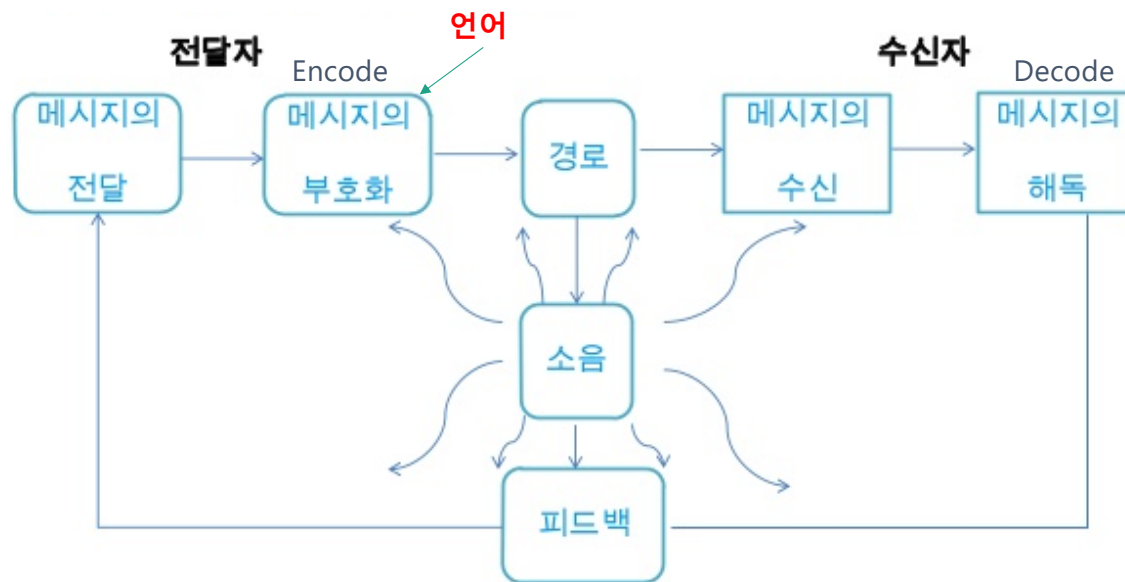


자연어란?

자연어 처리 (Natural Language Processing, NLP)

언어¹ 言語 🗣️ ★ +

명사 생각, 느낌 따위를 나타내거나 전달하는 데에 쓰는 음성, 문자 따위의 수단. 또는 그 음성이나 문자 따위의 사회 관습적인 체계.



자연어란?

자연어 처리 (Natural Language Processing, NLP)

자연 언어 自然言語 +

언어 일반 사회에서 **자연**히 발생하여 쓰이는 언어.



자연언어 : 한국어, 영어, 일본어



인공언어 : 프로그래밍 언어, 에스페란토어

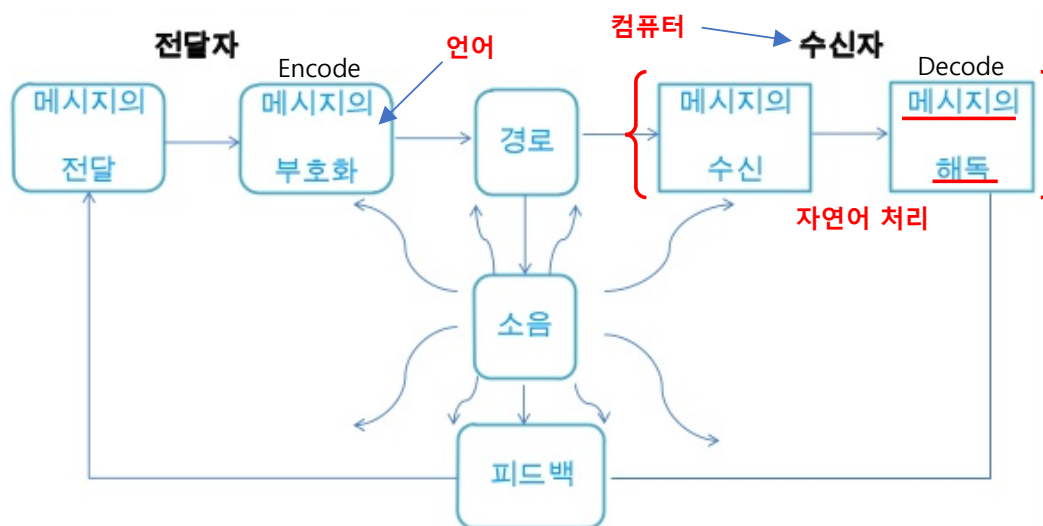
* Chisung Song, 시나브로 배우는 자연어처리 바벨피쉬 송치성
<https://www.slideshare.net/shuraba1/ss-56479835>

자연어 처리란?

자연어 처리 (Natural Language Processing, NLP)

자연어 처리 自然語處理 +

정보·통신 컴퓨터를 이용하여 인간 언어의 이해, 생성 및 분석을 다루는 인공지능 기술.



다양한 자연어 처리 기술

자연어 처리 (Natural Language Processing, NLP)

- 자연어 처리란, '자연어를 컴퓨터가 해독하고 그 의미를 이해하는 기술'

Symbolic approach

- 규칙/지식 기반 접근법

```
@Override
boolean isAnswerableQuestion(String messagePattern) {
    boolean isAnswerable = false;

    if (messagePattern.matches("ChannelNm(NOW)?(PROGRAM)?WHAT")) {
        // 국회TV에서 지금 뭐해?
        isAnswerable = true;
    } else if (messagePattern.matches("ChannelNm(NOW)?WHATPROGRAM")) {
        // 국회TV에서 지금 무슨 방송해?
        isAnswerable = true;
    } else if (messagePattern.matches("ChannelNm(NOW)?PROGRAMHOW")) {
        // 국회TV에서 지금 무슨 방송해?
        isAnswerable = true;
    }
    return isAnswerable;
}
```

100 원	100 (Number) + 원(G_ExchangeRateKRW : KRW=원)
100 달러	100(Number) + 달러(G_ExchangeRateKRW : USD=달러), S_UNIT_USD_money
100 m	100(Number) + m (S_UNIT_m_length)
100 미터	100(Number) + m (S_UNIT_m_length)

Statistical approach

- 확률/통계 기반 접근법
- TF-IDF를 이용한 키워드 추출
- TF (Term frequency): 단어가 문서에 등장한 개수
-> TF가 높을수록 중요한 단어
- DF (Document frequency): 해당 단어가 등장한 문서의 개수
-> DF가 높을수록 중요하지 않은 단어

$$w_{x,y} = tf_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y
 df_x = number of documents containing x
 N = total number of documents

* sujin oh, 실생활에서 접하는 빅데이터 알고리즘

https://www.slideshare.net/osujin121/ss-441864517from_action=save

자연어 처리의 단계

자연어 처리 (Natural Language Processing, NLP)

• 전처리

- 개행문자 제거
 - 특수문자 제거
 - 공백 제거
 - 중복 표현 제어 (ㅋㅋㅋㅋㅋ, ㅋㅋㅋㅋ, ...)
 - 이메일, 링크 제거
 - 제목 제거
 - 불용어 (의미가 없는 용어) 제거
 - 조사 제거
 - 띄어쓰기, 문장분리 보정
 - 사전 구축
- Tokenizing
 - Lexical analysis
 - Syntactic analysis
 - Semantic analysis

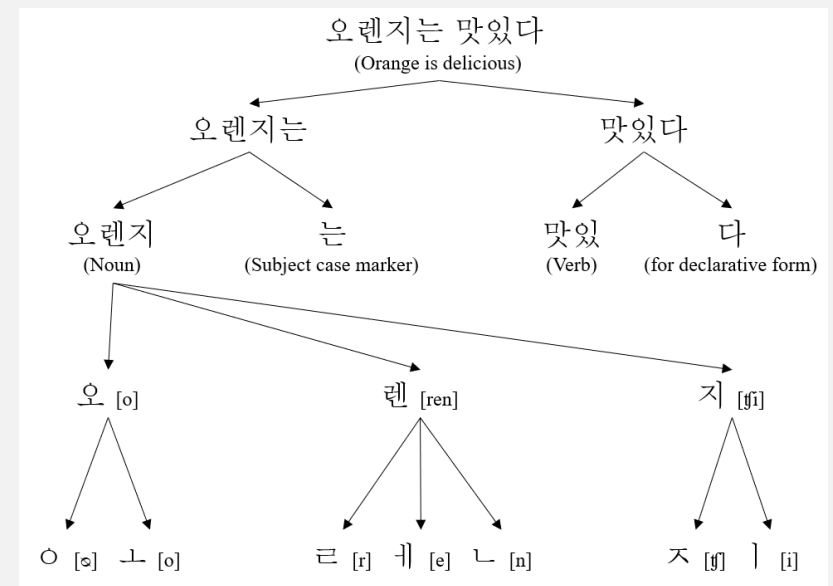
의심한그득+. + 앞으로는 사람들 많을 퇴근시간 말구
미리 가서 **사와야겠똥**⁹('ù'),
맥주로 터진 입, 청포도와 **고구농스틱**으로 달래봅니당
막상 먹다보니 양이 부족하진 않았는데 괜히 ...
먹을 때 많이 먹자며 배 터지기 직전까지 **남냐미:D**
식후땡 아슈크림⁹ (>_< ,)
브라우니쿠키 먹고 출근하세요 ♡
gs편의점에서 파는 **진 - 한** 브라우니 쿠키인데 **JMTgr**



자연어 처리의 단계

자연어 처리 (Natural Language Processing, NLP)

- 전처리
- Tokenizing
 - 자연어를 어떤 단위로 살펴볼 것인가
 - 어절 tokenizing
 - 형태소 tokenizing
 - n -gram tokenizing
 - WordPiece tokenizing
- Lexical analysis
 - 어휘 분석
 - 형태소 분석
 - 개체명 인식
 - 상호 참조
- Syntactic analysis
 - 구문 분석
- Semantic analysis
 - 의미 분석



전처리와 토큰나이징 실습

자연어 처리 (Natural Language Processing, NLP)

대소문자의 변환

함수	설명
upper()	모두 대문자로 변환
lower()	모두 소문자로 변환
capitalize()	문자열의 첫 문자를 대문자로 변환
title()	문자열에서 각 단어의 첫 문자를 대문자로 변환
swapcase()	대문자와 소문자를 서로 변환

편집, 치환

함수	설명
strip()	좌우 공백을 제거
rstrip()	오른쪽 공백을 제거
lstrip()	왼쪽 공백을 제거
replace(a, b)	a를 b로 치환

전처리와 토큰나이징 실습

자연어 처리 (Natural Language Processing, NLP)

분리, 결합

함수	설명
<code>split()</code>	공백으로 분리
<code>split('Wt')</code>	탭을 기준으로 분리
<code>' '.join(s)</code>	리스트 <code>s</code> 에 대하여 각 요소 사이에 공백을 두고 결합
<code>lines.splitlines()</code>	라인 단위로 분리

구성 문자열 판별

함수	설명
<code>isdigit()</code>	숫자 여부 판별
<code>isalpha()</code>	영어 알파벳 여부 판별
<code>isalnum()</code>	숫자 혹은 영어 알파벳 여부 판별
<code>islower()</code>	소문자 여부 판별
<code>isupper()</code>	대문자 여부 판별
<code>isspace()</code>	공백 문자 여부 판별
<code>startswith('hi')</code>	문자열이 hi로 시작하는지 여부 파악
<code>endswith('hi')</code>	문자열이 hi로 끝나는지 여부 파악

전처리와 토큰나이징 실습

자연어 처리 (Natural Language Processing, NLP)

검색

함수	설명
count('hi')	문자열에서 hi가 출현한 빈도 리턴
find('hi')	문자열에서 hi가 처음으로 출현한 위치 리턴, 존재하지 않는 경우 -1
find('hi', 3)	문자열의 index에서 3번부터 hi가 출현한 위치 검색
rfind('hi')	문자열에서 오른쪽부터 검사하여 hi가 처음으로 출현한 위치 리턴, 존재하지 않는 경우 -1
index('hi')	find와 비슷한 기능을 하지만 존재하지 않는 경우 예외발생
rindex('hi')	rfind와 비슷한 기능을 하지만 존재하지 않는 경우 예외발생
count('hi')	문자열에서 hi가 출현한 빈도 리턴

전처리와 토크나이징 실습

자연어 처리 (Natural Language Processing, NLP)

토큰화 (Tokenizing)

- 주어진 데이터를 토큰(Token)이라 불리는 단위로 나누는 작업
- 토큰이 되는 기준은 다를 수 있음 (어절, 단어, 형태소, 음절, 자소 등)

문장 토큰화 (Sentence Tokenizing)

- 문장 분리

단어 토큰화 (Word Tokenizing)

- 구두점 분리, 단어 분리
"Hello, World!" ->
"Hello", ",", "World", "!"

전처리와 토크나이징 실습

자연어 처리 (Natural Language Processing, NLP)

문장 토큰화(Sentence Tokenization. 문장 분리)

```
from nltk.tokenize import sent_tokenize  
  
text = "Hello, world. These are NLP tutorials."  
print(sent_tokenize(text))
```

전처리와 토큰나이징 실습

자연어 처리 (Natural Language Processing, NLP)

단어 토큰화(Word Tokenization. 단어 분리, 구두점 분리)

```
import nltk
from nltk import WordPunctTokenizer

nltk.download('punkt')

text = "Hello, world. These are NLP tutorials."
print(WordPunctTokenizer().tokenize(text))
```

전처리와 토큰나이징 실습

자연어 처리 (Natural Language Processing, NLP)

한국어 토큰화

- 영어는 New York과 같은 합성어 처리와 it's와 같은 줄임말 예외처리만 하면, 띄어쓰기를 기준으로 잘 동작하는 편
- 한국어는 조사나 어미를 붙여서 말을 만드는 교착어로, 띄어쓰기만으로는 부족
예시) he/him -> 그, 그가, 그는, 그를, 그에게
- 한국어에서는 조사, 어미를 분리하는 형태소 분석을 통하여 토큰화를 수행
예시) 안녕하세요 -> 안녕/NNG, 하/XSA, 세/EP, 요/EC







다양한 자연어 처리 Applications

자연어 처리 (Natural Language Processing, NLP)

- 문서 분류
- 문법, 오타 교정
- 정보 추출
- 음성 인식결과 보정
- 음성 합성 텍스트 보정
- 정보 검색
- 요약문 생성
- 기계 번역
- 질의 응답
- 기계 독해
- 챗봇
- 형태소 분석
- 개체명 분석
- 구문 분석
- 감성 분석
- 관계 추출
- 의도 파악

다양한 자연어 처리 Applications

자연어 처리 (Natural Language Processing, NLP)

데이터	분석	언어	지식	지능	엑소브레인
					
문서 필터 HTML 랩퍼 데이터셋목록 검색 데이터 조회 리소스 다운로드 온디맨드데이터 수집 데이터검색 데이터다운로드 데이터 업로드 데이터학습추천 데이터알림	트렌드(키워드) 분석 키워드 추출 연관 주제어 분석 토픽 트렌드 분석 감성분석 자동 분류 소셜 데이터(문서) 검색 오늘의 토픽 분석 인용구분석 관계도 분석 자동군집(v1) 자동군집(v2) 이슈감지	형태소분석 구문분석 개체명 인식 워드임베딩 언어 식별 띄어쓰기 신조어추출 오탈자교정 이벤트인식 시맨틱분석	RDF저장소 클래스검색 속성검색 인스턴스 검색 SP 조회 PO 조회 SO 조회 SPARQL Endpoint 질의 상의어, 하의어, 동의어, 유의어 의미망 경로 조회 데이터변환(Data2RDF) 지식변환 NLQ to SPARQL 지식임베딩	심층 질의응답 사용자의도이해(NLU) MRC 기반 질의응답 음성 질의 응답 음성 인식 음성 합성 이미지 분류 자동번역 이미지인식 이미지 캡셔닝 얼굴인식 객체인식 선호학습추천	Class 검색 Class 정보조회 Property 검색 Property 정보조회 Instance 검색 Instance 정보조회 Instance 시간정보조회 Instance 공간정보조회 Instance Type 추론 Relation Type 추론 시간 추론 1 시간 추론 2 Object 검색 Subject 검색 Property 검색

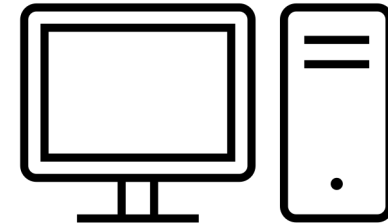
Reference: <https://github.com/MrBananaHuman/KorNlpTutorial>

다양한 자연어 처리 Applications

자연어 처리 (Natural Language Processing, NLP)

챗봇
+
음성 합성
+
감성 분류
+
개체명 인식
+
추천 시스템
+
기계 독해
+
지식 그래프
+
관계 추출
+
플러그인
⋮

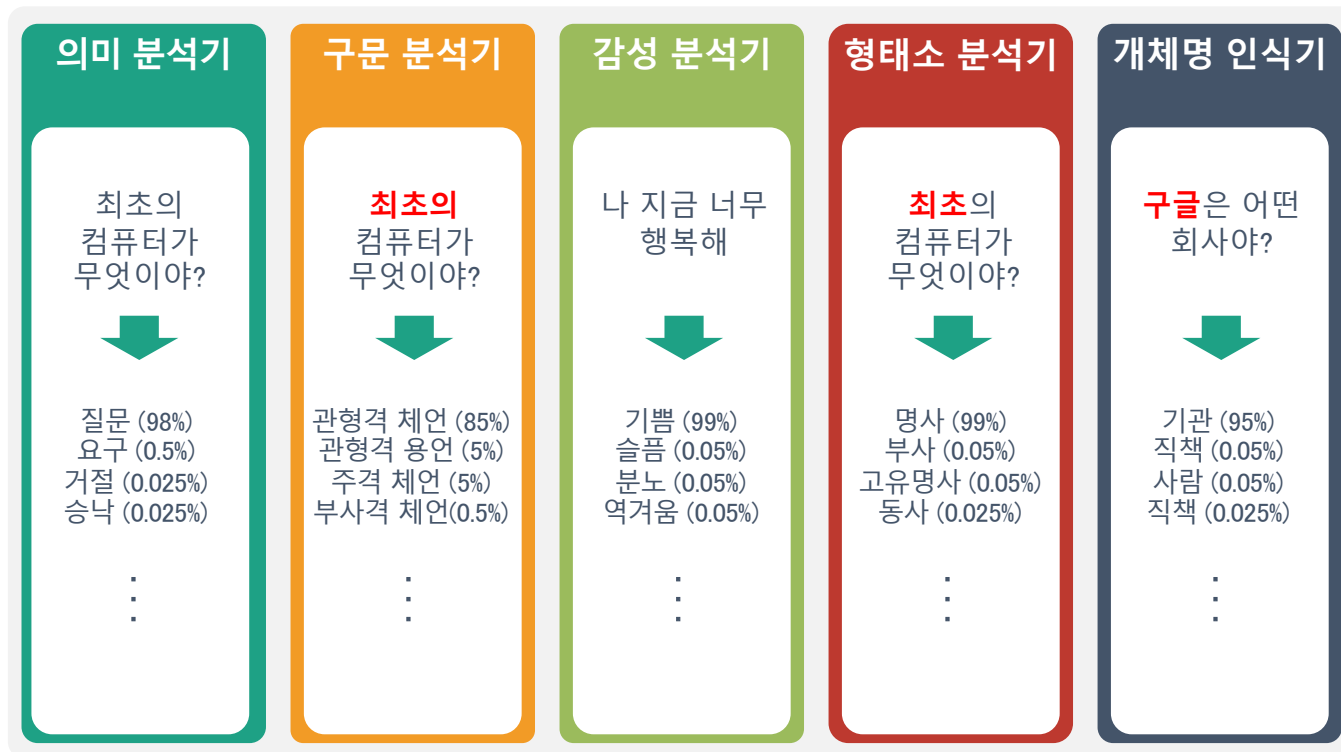
자그마치 9개 이상의 NLP 기술이 합쳐진 컴비네이션



다양한 자연어 처리 Applications

자연어 처리 (Natural Language Processing, NLP)

- 형태소 분석, 문서 분류, 개체명 인식 등, 대부분의 자연어 처리 문제는 '분류'의 문제



특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- ‘분류’를 위해선 데이터를 수학적으로 표현
- 먼저, 분류 대상의 특징 (Feature)을 파악 (Feature extraction)

분류 대상



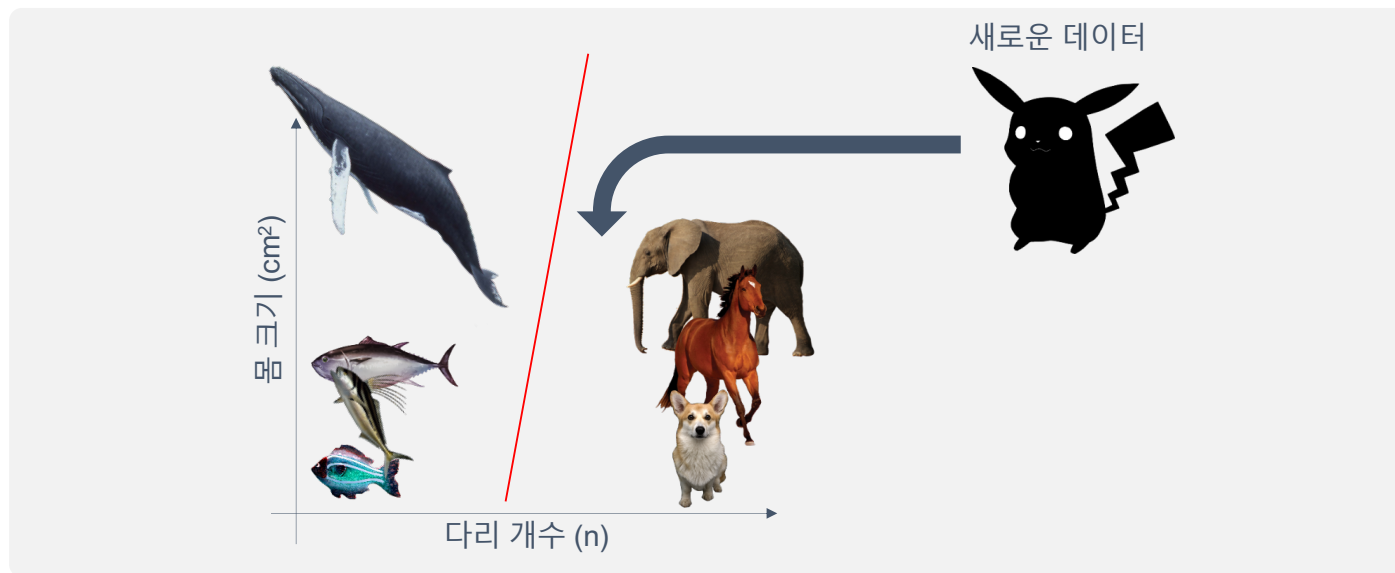
분류 대상의 특징

크기가 다양
다리의 개수가 다양

특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- 분류 대상의 특징 (Feature)를 기준으로, 분류 대상을 **그래프 위에 표현** 가능
- 분류 대상들의 경계를 수학적으로 나눌 수 있음 (Classification)
- 새로운 데이터 역시 특징을 기준으로 그래프에 표현하면, 어떤 그룹과 유사한지 파악 가능



자연어에서의 특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- 과거에는 사람이 직접 특징 (Feature)를 파악해서 분류
- 실제 복잡한 문제들에선 분류 대상의 특징을 사람이 파악하기 어려울 수 있음
- 이러한 특징을 컴퓨터가 스스로 찾고 (Feature extraction), 스스로 분류 (Classification) 하는 것이 '기계학습'의 핵심 기술

분류 대상	나는 대한민국에서 태어났다	최초의 컴퓨터는 누가 만들었어?
	이순신은 조선 중기의 무신이다	아이유 노래 틀어줘
	나 지금 너무 행복해	내일 날씨 알려줘
		지금 몇 시야?



분류 대상의 특징

자연어에서의 특징 추출과 분류

자연어 처리 (Natural Language Processing, NLP)

- 과거에는 사람이 직접 특징 (Feature)를 파악해서 분류
- 실제 복잡한 문제들에선 분류 대상의 특징을 사람이 파악하기 어려울 수 있음
- 이러한 특징을 컴퓨터가 스스로 찾고 (Feature extraction), 스스로 분류 (Classification) 하는 것이 '기계학습'의 핵심 기술

