자연처 처리 개요

명지대학교
MYONGJI UNIVERSITY

인공지능 자연어 처리 개요

자연어 처리의 개념

자연어란?

인간이 상대방과 의사소통을 위해 사용하는 언어

↓

한국어, 영어, 중국어, 스페인어 등

자연어란?

- 일상적으로 듣고, 쓰고, 말하고, 읽는 모든 언어
- 인간의 생각이나 감정 표현 수단
- 파이썬, C++ 등 인공 언어와 대비되는 개념

인공지능 자연어 처리 개요

자연어 처리의 개념

자연어 처리란?

인간의 자연어를 컴퓨터가 이해하고, 처리할 수 있도록 하는
인공지능의 한 분야

자연어 처리의 특징

- **언어 이해:** 인간 언어의 의미와 맥락 파악
- **인공지능 기반:** 머신러닝/딥러닝 알고리즘 활용
- **데이터 처리:** 텍스트 데이터 분석 및 패턴 인식
- **자동화:** 언어 관련 작업의 효율화

자연어 처리란? (Natural Language Processing, NLP)

- 인공지능의 한 분야
- 사람 언어 현상을 컴퓨터로 다루는 기술
- 최종목표: 컴퓨터가 사람 언어를 이해하고 문제 해결

인공지능 자연어 처리 개요

자연어 처리의 범주

자연어 처리(NLP, Natural Language Processing)란?

인간의 자연어를 컴퓨터가 이해하고, 처리할 수 있도록 하는 인공지능의 한 분야

자연어 생성 (NLG, Natural Language Generation)

컴퓨터 스스로 자연어를 생성하기 위한 것

자연어 이해 (NLU, Natural Language Understanding)

컴퓨터가 인간의 자연어를 이해하기 위한 것

NLU와 NLG는 챗봇(Chatbot) 엔진의 핵심구성 요소

인공지능 자연어 처리 개요

자연어 처리의 개념

NLP와 NLU, NLG의 관계 및 활용

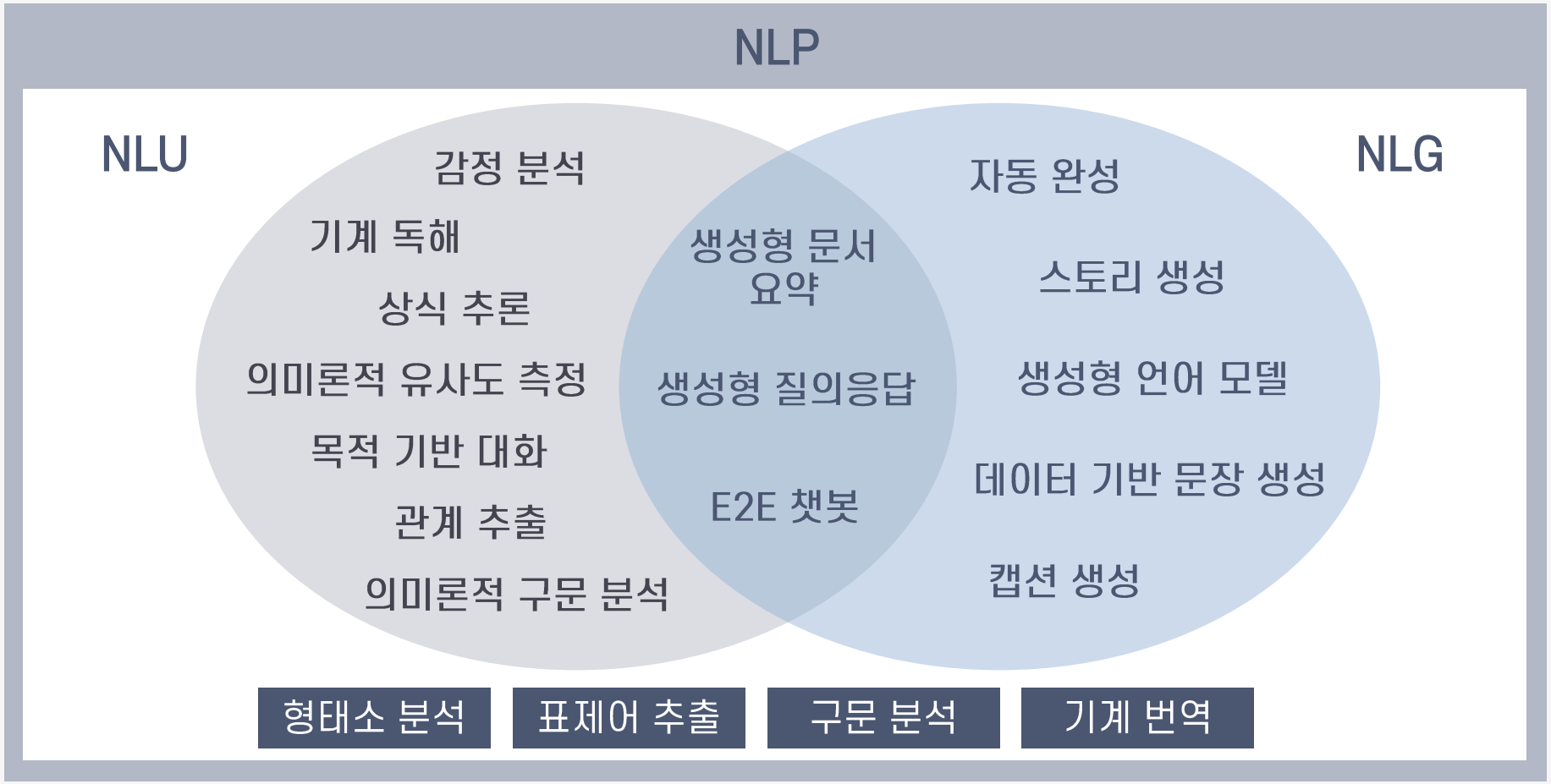
NLU	NLG
감정분석 및 기계 독해 등에 활용	자동완성 및 생성형 언어모델 등에 활용

NLU와 NLG 공통으로 생성형 문서요약이나 질의 응답에 활용

인공지능 자연어 처리 개요

자연어 처리의 개념

NLP와 NLU, NLG의 관계 및 활용



<출처: NLP 관계. https://www.kakaobrain.com/blog/118. 20. 9>

인공지능 자연어 처리 개요

자연어 처리의 역사

1940년대	실용적인 관점에서 이중어 사전(Bilingual Dictionary)을 이용한 기계번역 시도
1950년대	언어를 작은 의미소 단위로 분석, 형태소 추출 후 문장을 재구성하는 구조주의 방법론 적용
1960년대	SAD-SAM, BASEBALL, STUDENT, ELIZA 등 다양한 NLP 시스템이 개발되기 시작

<출처: ELIZA. https://en.wikipedia.org/wiki/ELIZA. 20. 9>

인공지능 자연어 처리 개요

자연어 처리의 역사

1970년대	구문분석과 의미분석, 어휘 사전을 활용하여 특정 분야에 특화된 NLP 연구가 활발하게 진행
1980년대	변형을 최소화한 통합기반문법(Unification-based Grammar) 부상
1990년대	말뭉치를 대규모로 구축하여 통계적으로 처리하는 NLP 방식이 발전
2000년대 이후	NLP 분야에 기계학습(Machine Learning)과 딥러닝(Deep Learning)을 본격적으로 적용

인공지능 자연어 처리 개요

자연어 처리의 역사

1970년대	구문분석과 의미분석, 어휘 사전을 활용하여 특정 분야에 특화된 NLP 연구가 활발하게 진행
1980년대	변형을 최소화한 통합기반문법(Unification-based Grammar) 부상
1990년대	말뭉치를 대규모로 구축하여 통계적으로 처리하는 NLP 방식이 발전
2000년대 이후	NLP 분야에 기계학습(Machine Learning)과 딥러닝(Deep Learning)을 본격적으로 적용

인공지능 자연어 처리 개요

자연어 처리의 역사

최신 자연어 처리 동향

- 최근 발전: CNN, RNN → 트랜스포머(Transformer) → 대규모 언어 모델(LLM)
- 시장 전망: 2024년 273억 달러 → 2037년 7,348억 달러(연평균 28.8% 성장)
- 주요 연구 방향: 다국어 모델 개발, 도메인 적응 기술, NLP와 컴퓨터 비전 및 강화학습 통합

인공지능 자연어 처리 개요

자연어 처리 기술의 발전

전통적인 접근법

- 알고리즘 기반 후보 생성 방식
- 여러 개의 후보 생성 후 확률적 방법으로 모호함 해소
- 형태소 분석: 어절로부터 모든 가능한 형태소 후보 생성
- 의존 구조 분석: 어절 간 관계를 문법으로 기술
- 애매성 해소: 대용량 통계데이터 기반 확률적 선택(HMM 등)

딥러닝 기술

- 2013년 word2vec 발전 이후 연구 활발화
- 현대 자연어 처리에 활용되는 주요 모델:
  - 합성곱 신경망(CNN)
  - 순환 신경망(RNN)
  - 트랜스포머(Transformer)
- 강화 학습 통한 최적화 시도
- 기존 방법 대비 문맥 이해와 의미 추출에서 우수한 성능

인공지능 자연어 처리 개요

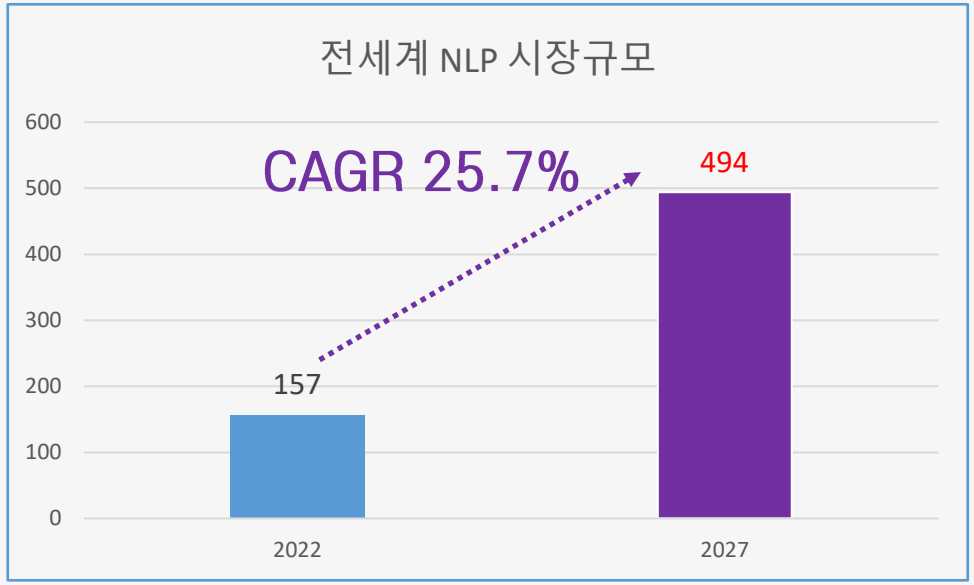
자연어 처리 시장 규모

전 세계 자연어 처리(NLP) 시장 규모는 급속히 성장할 것으로 전망

2022년 157억 달러에서 2027년에는 494억 달러로 성장할 전망

2022년부터 2027년까지 자연어 처리 시장의 연평균 성장률(CAGR)은 25.7%로 예측

(Natural Language Processing (NLP) Market - Global Forecast to 2027, MarketsandMarkets)

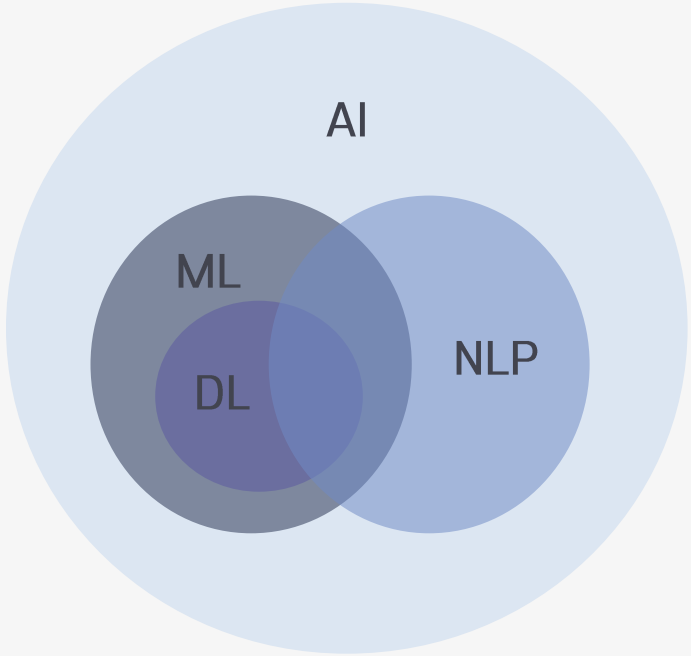


인공지능 자연어 처리 개요

인공지능과 자연어 처리

**인공지능(AI)이라는 큰 범주 내에 머신러닝이 있고,
딥러닝은 머신러닝 범주 내에 포함**

» 인공지능 자연어 처리(NLP)는 머신러닝과 딥러닝 내 일부 포함되어 있으면서, 별도의 영역도 포함



- AI 인공지능(Artificial intelligence)
- ML 머신러닝(Machine learning)
- NLP 자연어 처리(Natural Language Processing)
- DL 딥러닝(Deep learning)

인공지능 자연어 처리 개요

자연어 처리 과정

자연어처리(Natural Language Processing)은 데이터 전처리, 언어학적 분석, 특징 추출 및 표현, 모델 적용 및 분석 등 4단계 절차에 따라 수행

데이터 전처리

토큰화, 소문자 변환, 불용어 제거, 어간 추출 등 기초 작업

언어학적 분석

형태소, 구문, 의미, 담화 분석을 통한 구조 파악

특징 추출 및 표현

품사 태깅, 개체명 인식, 벡터 임베딩 등 수치화

모델 적용 및 분석

감정 분석, 주제 모델링, 기계 번역, 텍스트 요약 구현

자연어처리는 단계별로 데이터를 점진적으로 정제하고 구조화하여, 최종적으로 의미 있는 정보와 통찰을 얻는 과정

인공지능 자연어 처리 개요

자연어 처리 과정 - 데이터 전처리(Preprocessing)

토큰화

문장을 단어/구문 단위로 분리 → 분석의 기본 단위 설정

예: "자연어처리는 재미있다" → ["자연어처리는", "재미있다"]

불용어 제거

"은", "는", "이", "그" 등 의미 적은 단어 제거 → 데이터 크기 감소 → 분석 효율성 증대

데이터 전처리

언어 데이터를 기계가 처리하기 적합한 형태로 변환 → NLP 모델 성능과 품질의 핵심 결정 단계

소문자 변환

텍스트 소문자로 통일 → 대소문자 불일치 방지 → 일관된 처리 가능

예: "Apple"과 "apple" 동일 처리

어간/표제어 추출

다양한 단어 형태를 기본형으로 통일 → 동일 의미 다른 표현 통합 → 분석 정확도 향상

예: "먹었다", "먹습니다" → "먹다"

+ 영문 자연어 전처리 개요

정제(Cleaning)

- **특수문자 제거**
  - 특수문자 : '!"#$%&₩'()*+,-./:;<=>?@[₩₩]^_`{|}~’

- **대소문자 통일**
  - KOREA, Korea, korea → korea

+ 영문 자연어 전처리 개요

토큰화(Tokenization)

코퍼스(Corpus)에서 분리자(Separator)를 포함하지 않는 연속적인 문자열 단위로 분리

토큰화 단위에 따른 분류
- 문장(Sentence)단위 토큰화
- 단어(Word)단위 토큰화

단어 단위 토큰화 방법에 따른 분류
- 어절 단위 : 띄어쓰기 단위, 영문 적합
- 형태소 단위 : 한글 적합
- Subword : 형태소와 유사, 의미 대신 통계적 방법 적용

+ 영문 자연어 전처리 개요

불용어 제거(Stopword Elimination)

전치사, 관사 등 문장이나 문서의 특징을 표현하는데 불필요한 단어를 제거하는 단계

예) you, my, the, a, of, at 등

+ 영문 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

- 제거 대상 특수문자의 종류

```
import string

print(string.punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

+ 영문 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

- **영문 자연어 전처리를 위한 코퍼스(Corpus)**

> Beneath it were the words: “Stay Hungry. Stay Foolish.” It was their farewell message as they signed off. And I have always wished that for myself. And now, as you graduate to begin anew, I wish that for you.
>
> 출처 : 스티브 잡스, 2005, 스탠포드대학교 졸업식

+ 영문 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

정규식을 이용한 특수문자 제거

- re.sub('[^₩w₩.₩s]','',text)를 이용하여 '.'을 제외한 특수문자 제거

```
import re

cleaned_text = re.sub('[^\w\.\s]','',text)
print(cleaned_text)
```

```
Beneath it were the words Stay Hungry. Stay Foolish. It was their farewell message
as they signed off. Stay Hungry. Stay Foolish. And I have always wished that for myself. And now as
you graduate to begin a new I wish that for you.
```

+ 영문 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

정규식을 이용한 '₩n' 문자 제거

- re.sub('₩n','',cleaned_text)을 이용하여 '₩n' 문자 제거

```
cleaned_text = re.sub('\n','',cleaned_text)
print(cleaned_text)
```

```
Beneath it were the words Stay Hungry. Stay Foolish. It was their farewell message as they signed off.
```

+ 영문 자연어 전처리 적용

토큰화(Tokenization)

| nltk를 이용한 영문 토큰화

- **nltk를 이용한 영문 토큰화를 위해 punkt 모듈 다운로드 및 활용**

  - nltk punkt 모듈 다운로드

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

+ 영문 자연어 전처리 적용

토큰화(Tokenization)

| 문장 단위 토큰화

- nltk.sent_tokenize를 이용한 문장 단위 토큰화 수행

```
sent_tokens = nltk.sent_tokenize(cleaned_text)
print(sent_tokens)
```

```
['Beneath it were the words Stay Hungry.', 'Stay Foolish.', 'It was their farewell message as they signed off.']
```

+ 영문 자연어 전처리 적용

토큰화(Tokenization)

| 단어 단위 토큰화 - 어절

- nltk.work_tokenize를 이용한 단어 단위 토큰화 수행

```
tokens = nltk.word_tokenize(clean)
tokens
```

```
['Beneath',
 'it',
 'were',
 'the',
 'words',
 'Stay',
 'Hungry',
```

+ 영문 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| nltk패키지의 stopword를 이용한 불용어 제거

- **nltk패키지의 stopword 다운로드**

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

+ 영문 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| 영문 불용어 로드 및 출력

- nltk패키지의 stopword 중 영문 불용어를 로드하고 출력

```
from nltk.corpus import stopwords
stop = stopwords.words('english')
print(stop)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
```

+ 영문 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| nltk패키지의 stopword를 이용한 불용어 제거

- **nltk패키지의 stopword와 python List Comprehension을 이용한 불용어 제거**

```
tokens = [token for token in tokens if token not in stop]
print(tokens)
```

```
['Beneath', 'words', 'Stay', 'Hungry', '.', 'Stay', 'Foolish', '.',
```

+ 영문 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| nltk패키지의 stopword를 이용한 불용어 제거

- 불용어 제거 후 단어의 길이가 2글자 이하인 단어 제거

```
tokens = [token for token in tokens if len(token) >= 3]
print(tokens)
```

```
['Beneath', 'words', 'Stay', 'Hungry', 'Stay', 'Foolish', 'farewell',
```

+ 영문 자연어 전처리 적용

소문자화(Lower Capitalization)

| 정제(Cleaning) 과정 중 하나인 소문자화

- **lower( ) 함수와 List Comprehension을 이용한 소문자화**

```
print('소문자화 수행전 : ', tokens)
tokens = [token.lower() for token in tokens]
print('소문자화 수행후 : ', tokens)
```

```
소문자화 수행전 :  ['Beneath', 'words', 'Stay', 'Hungry', 'Stay', 'Foolish',
소문자화 수행후 :  ['beneath', 'words', 'stay', 'hungry', 'stay', 'foolish',
```

+ 한글 자연어 전처리 개요

토큰화(Tokenization)

코퍼스(Corpus)에서 분리자(Separator)를 포함하지 않는 연속적인 문자열 단위로 분리

토큰화 단위에 따른 분류
- 문장(Sentence)단위 토큰화
- 단어(Word)단위 토큰화

단어 단위 토큰화 방법에 따른 분류
- 어절 단위 : 띄어쓰기 단위, 영문에 적합
- 형태소 단위 : 한글에 적합
- Subword : 형태소와 유사, 의미 대신 통계적 방법 적용

+ 한글 자연어 전처리 개요

불용어 제거(Stop Word Elimination)

문장이나 문서의 특징을 표현하는데 불필요한 단어를 제거하는 단계
불용어 사전을 활용하여 제거, 불용어 사전 관리

+ 한글 자연어 전처리 적용

정제(Cleaning)

| 특수문자 제거

- 제거 대상 특수문자의 종류 - 영문과 동일

```
import string

print(string.punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

+ 한글 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

- **한글 자연어 전처리를 위한 코퍼스(Corpus)**

> 그 밑에는 "계속 배고픔을 느끼세요. 계속 바보로 남으세요" 라는 문구가 새겨져 있었습니다.
> 그들이 전한 마지막 인사말이었습니다. 계속 배고픔을 느끼세요, 계속 바보로 남으세요. 그리고 저는
> 항상 제 자신이 그렇길 바랬습니다. 이제는 졸업을 하고 새로운 출발을 하는 여러분에게 바라는 바 입니다.
>
> 출처 : 스티브 잡스, 2005, 스탠포드대학교 졸업식

+ 한글 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

- **정규식을 이용한 특수문자 제거**

  - re.sub('[^₩w₩.₩s]','',text)를 이용하여 '.'을 제외한 특수문자 제거

```
import string
import re

cleaned_text = re.sub('[^\w\s]','',text)
print(cleaned_text)
```

```
그 밑에는 계속 배고픔을 느끼세요 계속 바보로 남으세요 라는 문구가 새겨져 있었습니다
그들이 전한 마지막 인사말이었습니다 계속 배고픔을 느끼세요 계속 바보로 남으세요 그리고 저는
항상 제 자신이 그렇길 바랬습니다 이제는 졸업을 하고 새로운 출발을 하는 여러분에게 바라는 바 입니다
```

+ 한글 자연어 전처리 적용

정제(Cleaning)
| 특수문자 제거

- **정규식을 이용한 '₩n' 문자 제거**
  - re.sub('₩n','',cleaned_text)을 이용하여 '₩n' 문자 제거

```
cleaned_text = re.sub('\n',' ',cleaned_text)
print(cleaned_text)
```

```
그 밑에는 계속 배고픔을 느끼세요 계속 바보로 남으세요 라는 문구가 새겨져 있었습니다
```

+ 한글 자연어 전처리 적용

정제(Cleaning)

| 띄어쓰기가 안되어 있는 경우 띄어쓰기 적용

- **띄어쓰기가 적용되지 않은 상태로 코퍼스 변환**

  - 정규식을 이용하여 띄어쓰기가 적용되지 않은 상태로 코퍼스 변환

```
non_space = re.sub('\s', '', text)
print(non_space)
```

```
그밑에는"계속배고픔을느끼세요.계속바보로남으세요"라는문구가새겨져있었습니다.
```

+ 한글 자연어 전처리 적용

정제(Cleaning)

| 띄어쓰기가 안되어 있는 경우 띄어쓰기 적용

- **띄어쓰기 적용을 위한 PyKoSpacing 설치**

  - PyPi.org에 등록되지 않은 상태이므로 git을 이용한 설치

```
!pip install git+https://github.com/haven-jeon/PyKoSpacing.git
```

```
Collecting git+https://github.com/haven-jeon/PyKoSpacing.git
  Cloning https://github.com/haven-jeon/PyKoSpacing.git to /tmp/pip-req-build-fk4ufrer
  Running command git clone -q https://github.com/haven-jeon/PyKoSpacing.git /tmp/pip-req-build-fk4ufrer
```

+ 한글 자연어 전처리 적용

정제(Cleaning)

| 띄어쓰기가 안되어 있는 경우 띄어쓰기 적용

PyKoSpacing을 이용한 띄어쓰기 적용

- spacing( ) 함수를 이용하여 띄어쓰기 적용

```
from pykospacing import Spacing

spacing = Spacing()
new_sent = spacing(non_space)
print(new_sent)
```

```
그 밑에는 "계속 배고픔을 느끼세요. 계속 바보로 남으세요"라는 문구가 새겨져 있었습니다.
```

+ 한글 자연어 전처리 적용

토큰화(Tokenization)

| 한글 문장 단위 토큰화

- **한글 문장 단위 토큰화를 위해 kss 모듈 설치 및 활용**

  - 한글 문장 단위 토큰화 적용을 위해 kss 모듈 설치

```
!pip install kss
```

```
Collecting kss
  Downloading kss-3.2.0.tar.gz (42.4 MB)
     |████████████████████████████████████████|
```

+ 한글 자연어 전처리 적용

토큰화(Tokenization)

| 한글 문장 단위 토큰화

- **kss 모듈을 활용한 한글 문장단위 토큰화 적용**

  - kss.split_sentences( )를 이용하여 한글 문장단위 토큰화 적용

```
import kss

sent_tokens = kss.split_sentences(cleaned_text)
print(sent_tokens)
```

```
['그밑에는계속배고픔을느끼세요', '계속바보로남으세요라는문구가새겨져있었습니다',
```

+ 한글 자연어 전처리 적용

토큰화(Tokenization)

| 한글 단어 단위 토큰화

- **한글 단어단위 토큰화 적용 - 어절 토큰화**

```
def tokenizer(words):
  tokens = words.split()
  return tokens

tokens = tokenizer(cleaned_text)
print(tokens)
```

```
['그', '밑에는', '계속', '배고픔을', '느끼세요', '계속', '바보로', '남으세요',
```

+ 한글 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| 한글 불용어 사전 작성 및 업로드

- **한글 불용어 사전 작성 및 stopword_dict.csv로 저장**

	A
1	stopword
2	그
3	라는
4	바
5	입니다
6	하고

+ 한글 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| 한글 불용어 사전 작성 및 업로드

- **한글 불용어 사전 업로드**

```
from google.colab import files
import os

data_dir = 'data'

if not os.path.exists(data_dir):
  os.mkdir(data_dir)
os.chdir(data_dir)
files.upload()
os.chdir('..')
```

파일 선택 선택된 파일 없음 Cancel upload

'파일 선택' 클릭 후
stopword_dict.csv 파일 선택

+ 한글 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| 한글 불용어 사전 작성 및 업로드

- **pandas 패키지를 이용하여 업로드된 불용어 사전 출력**

```
import pandas as pd

stop = pd.read_csv('./data/stopword_dict.csv', encoding='cp949')
print(stop[:10])
```

```
  stopword
0        그
1       라는
2        바
3      입니다
4       하고
```

+ 한글 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| 한글 불용어 사전 작성 및 업로드

- pandas 패키지를 이용하여 업로드된 불용어 사전 리스트로 변환

```
print(list(stop.stopword))
```

```
['그', '라는', '바', '입니다', '하고']
```

To. 김복주 교수

아래와 같은 한기대 자문
촬영하실 때, 짧게 언급해

(검토의견) stop.s
csv파일의 h
대응됨을 설명

+ 한글 자연어 전처리 적용

불용어 제거(Stopword Elimination)

| 불용어 사전을 활용한 불용어 제거

- **파이썬 List Comprehension을 이용하여 불용어 제거**

```
tokens = [token for token in tokens if token not in list(stop.stopword)]
print(tokens)
```

```
['밑에는', '계속', '배고픔을', '느끼세요', '계속', '바보로', '남으세요', '문구가',
```

- 끝 -