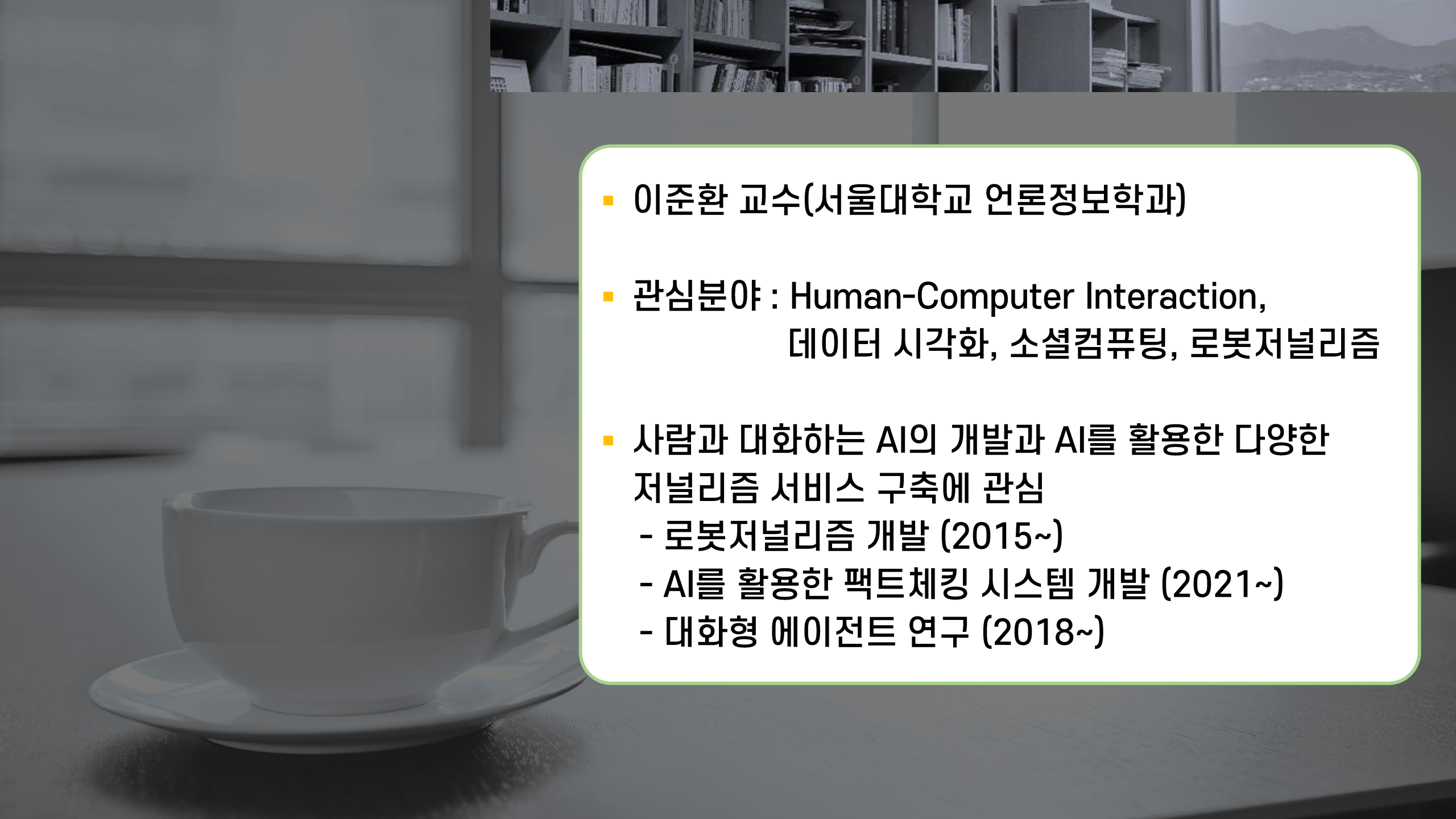


- 
- 이준환 교수(서울대학교 언론정보학과)
 - 관심분야 : Human-Computer Interaction,
데이터 시각화, 소셜컴퓨팅, 로봇저널리즘
 - 사람과 대화하는 AI의 개발과 AI를 활용한 다양한
저널리즘 서비스 구축에 관심
 - 로봇저널리즘 개발 (2015~)
 - AI를 활용한 팩트체크 시스템 개발 (2021~)
 - 대화형 에이전트 연구 (2018~)



어떤 일을 해왔는가...

- 소셜 컴퓨팅 (Social Computing)
 - 데이터를 통해 사용자를 관찰하는 방법
 - 데이터를 기반해 사용자의 행동을 예측하는 방법

05 ^강빅데이터의 이해와 활용

텍스트 빅데이터





학습목차

- 1 텍스트 처리와 자연어 처리
- 2 자연어처리 기술의 활용
- 3 텍스트 전처리
- 4 단어의 표현 방법
- 5 언어모형

빅데이터의
이해와 활용

1 텍스트 처리와 자연어 처리



1. 텍스트 처리와 자연어 처리

● 텍스트(Text)

- 숫자와 더불어 가장 대표적인 정보의 저장 단위
- 소셜네트워크 서비스의 성장 -> 중요성이 점차 커짐
- 기본적으로는 명목 데이터(nominal data)
- 비명목 데이터로 활용하기 위해서는 텍스트 프로세싱(text processing)과 같은 작업이 선행되어야 함

1. 텍스트 처리와 자연어 처리

● 텍스트 프로세싱(Text Processing)

- 텍스트에서 의미 있는 정보를 찾아내는 과정
- 자연어처리(Natural Language Processing)와는 차이가 있음
- 자연어처리는 텍스트는 물론, 음성기반의 대화(speech), 이미지(image), 사인(sign) 등 많은 것들을 대상으로 함
- 빅데이터 분석에서 텍스트 처리는 대부분 자연어처리에 초점을 맞추고 진행 -> 자연어처리와 텍스트 분석을 나누는 것이 의미가 없어짐

빅데이터의
이해와 활용

2 자연어처리 기술의 활용



2. 자연어처리 기술의 활용

● 자연어(Natural Language)

- 우리가 일상적으로 쓰는 언어
- 소셜데이터를 활용한 빅데이터 분석의 대상은 자연어인 경우가 대부분
 - 텍스트 요약과 분류
 - 감성 분석
 - 의미연결망 분석
 - 기계번역
 - 질의응답과 챗봇
 - 음성 인식



2. 자연어처리 기술의 활용

● 텍스트 분류(Text Classification)

- 텍스트가 어떤 범주에 속하는지 판단하는 작업
- 일반적으로 텍스트로부터 특징(feature)을 추출하고 이를 바탕으로 학습된 모델을 구축
 - 이메일의 스팸 분류
 - 텍스트의 감정 분석



2. 자연어처리 기술의 활용

● 감성 분석(Sentiment Analysis)

- 텍스트에 포함된 의견이나 감정 등을 분석
- 영화평 분석, 고객의 의견 분석, 유권자 메시지 분석 등

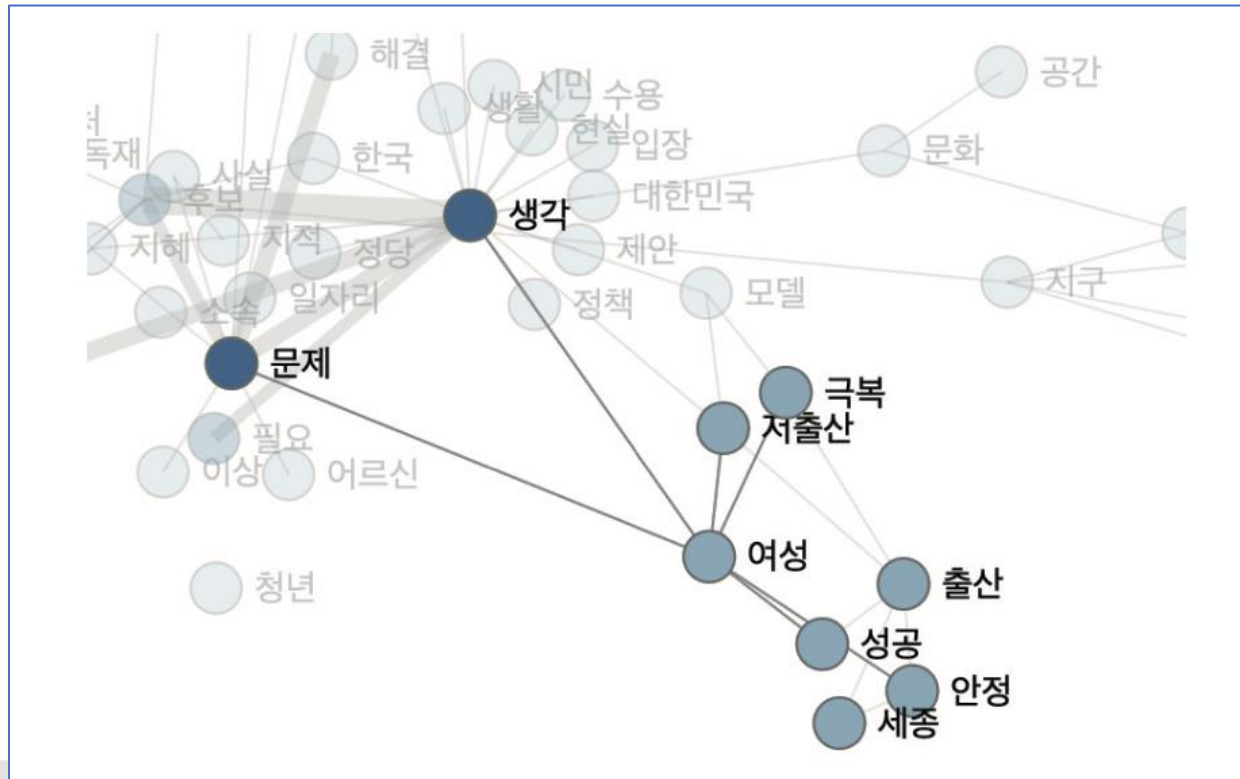
영화 평	분류결과
올해 영화관에서 본 영화중 최고였다매혹적이고 아름답고 슬프기도하다	Positive
진지하고 여운이 남는 영화예요 추천합니다	Positive
재미 드럽게 없음..포와로역 외모도 안어울리고, 지루하고, 긴장감도 없고... 실망입니다. 졸려요..	Negative
평소에 잠을 푹 못 잤는데 심야로 보다가 아주 잘 자고갑니다 너무너무고마 운영화 불면증치료에 좋은 힐링	Negative

네이버 영화평의 감성 분석 사례

2. 자연어처리 기술의 활용

● 의미연결망 분석(Sentiment Network Analysis)

- 단어의 네트워크를 구성하여 단어 간의 관계성을 파악
- 특정 키워드가 내포하는 의미를 확장하여 살펴보는데 유용



단어의 의미연결망 분석 사례

2. 자연어처리 기술의 활용

● 질의응답(Question Answering)

- 주어진 질의에 대한 답을 찾아 제시하는 연구
- 기계독해 이해력 테스트를 위한 데이터셋 SQuAD(Stanford Question Answering Dataset)가 공개되며 관련 연구 활발히 진행

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, *Il milione* (or, *The Million*, known in English as the *Travels of Marco Polo*), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

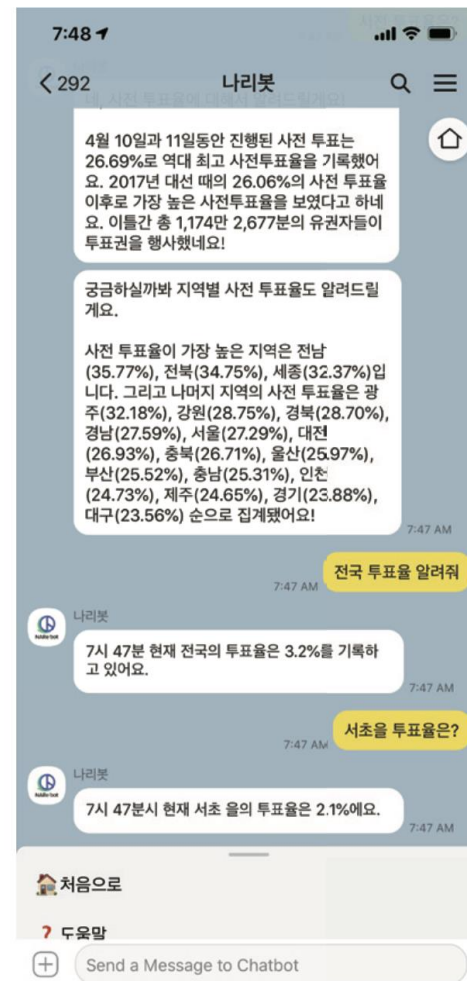
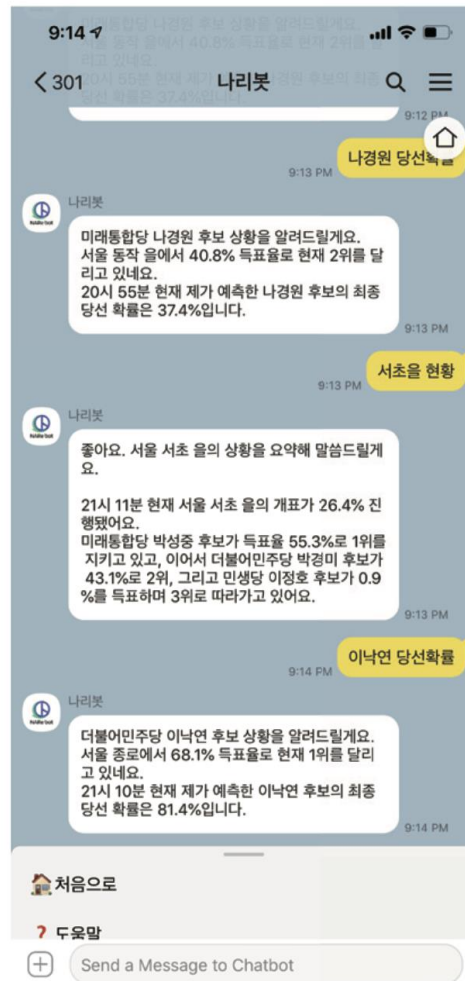
SQuAD 사례

2. 자연어처리 기술의 활용

● 챗봇(Chatbot)

- 사용자의 발화에서 숨은 의도(intent)를 찾아내고 자연스러운 인간의 언어로 답변을 해주는 서비스 -> 대화형 에이전트 (Conversational Agent)
- 챗봇의 주요 기술은 텍스트 분류, 질의응답 등 앞서 소개한 기술들이 활용 됨

21대 국회의원선거에 사용된 나리봇



빅데이터의
이해와 활용

3 텍스트 전처리



3. 텍스트 전처리

- 텍스트 전처리(Text Preprocessing)

- 텍스트 분석의 궁극적 목표는 텍스트가 가진 함의의 이해
- 그러나 컴퓨터는 텍스트를 바로 연산하여 분석할 수 없음
- 따라서 분석 전에 토큰화 혹은 정규화 등의 텍스트 전처리 작업이 필요

- 예) helped, helps -> help

3. 텍스트 전처리

● 토큰화(Tokenization)

- 문장을 가장 작은 단위로 나누는 작업
- 전통적으로는 토큰으로 단어를 사용
- 단어 토큰화를 하는 가장 쉬운 방법은 띄어쓰기(whitespace)를 중심으로 토큰을 만드는 것

예 띄어쓰기 기준으로 한 단어 토큰화

문장: 꿈을 이루고자 하는 용기만 있다면 모든 꿈을 이룰 수 있다

토큰화: ['꿈을', '이루고자', '하는', '용기만', '있다면', '모든', '꿈을', '이룰', '수', '있다']

3. 텍스트 전처리

● 정규화(Normalization)

- 같은 의미지만 표기가 다른 단어들을 통합하는 방법
 - United States, US, USA. -> 같은 단어이지만 자연어처리 과정에서는 서로 다른 단어로 인식
 - text, Text, TEXT 는 서로 다른 단어로 인식
- 그러나 '우리'를 뜻하는 'us'와 미국을 뜻하는 'US'는 구분해야 하기 때문에 정규화 과정에서 주의를 기울여야 함
- 영어의 관사, 전치사, 우리말의 조사 등은 분석에 필요하지 않은 단어 (stopwords)라 삭제하여야 함

3. 텍스트 전처리

● 어간 추출과 형태소 분석

- 같은 의미의 단어를 통합하는 과정에서 주로 많이 사용하는 방법은 단어의 원형을 추출하는 것
- 영어의 경우 어간 추출(stemming) 혹은 표제어 추출 (lemmatization)을 통해 원형 추출
 - helps, helped, helping -> help

3. 텍스트 전처리

어간 추출과 형태소 분석

- 한글의 경우 형태소 분석기를 이용하여 단어의 기본형 추출

예 형태소 분석기를 이용한 품사태깅

문장: 꿈을 이루고자 하는 용기만 있다면 모든 꿈을 이룰 수 있다

형태소 분석 결과: [('꿈', 'NNG'), ('을', 'JKO'), ('이루', 'VV'), ('고자', 'EC'), ('하', 'VV'), ('는', 'ETM'), ('용기', 'NNG'), ('만', 'JX'), ('있', 'VV'), ('다면', 'EC'), ('모든', 'MM'), ('꿈', 'NNG'), ('을', 'JKO'), ('이루', 'VV'), ('ㄷ', 'ETM'), ('수', 'NNB'), ('있', 'VV'), ('다', 'EC')]

- 형태소 분석기는 단어의 기본형을 추출하고 품사를 태깅

3. 텍스트 전처리

- 원-핫 인코딩(One-Hot Encoding)

- 컴퓨터는 연산과정에서 숫자를 사용하기 때문에 자연어 처리 과정에서 문자를 숫자로 변환하여 연산을 함
- 원-핫 인코딩은 출현한 모든 단어 사전 크기의 벡터를 만들고 특정 단어의 위치를 숫자로 표시한 것

3. 텍스트 전처리

● 원-핫 인코딩(One-Hot Encoding)

예 단어 사전 만들기 예시

문장 1: 원 핫 인코딩으로 단어에 인덱스 부여하기

단어사전: ['원': 0, '핫': 1, '인코딩': 2, '으로': 3, '단어': 4, '에': 5, '인덱스': 6, '부여': 7, '하': 8, '기': 9]



예 단어사전의 활용한 원-핫 인코딩

문장 2: 단어 인코딩

원-핫 인코딩 행렬: $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

예 단어사전의 활용한 정수인코딩

문장 2: 단어 인코딩

정수인코딩 행렬: [4, 2]

빅데이터의
이해와 활용

4 단어의 표현방법



4. 단어의 표현방법

● 단어의 표현방법

- 컴퓨터는 사람과 같이 문장과 단어를 이해할 수 없기 때문에 앞서 살펴본 원-핫 인코딩 방법처럼 단어를 숫자로 치환해서 표현하는 방법이 사용됨
- 단어를 숫자로 바꾸어 주면 텍스트를 통계적으로 처리할 수 있어 단어의 빈도수 등을 계산하거나 특정 단어의 중요도를 파악할 수 있음

4. 단어의 표현방법

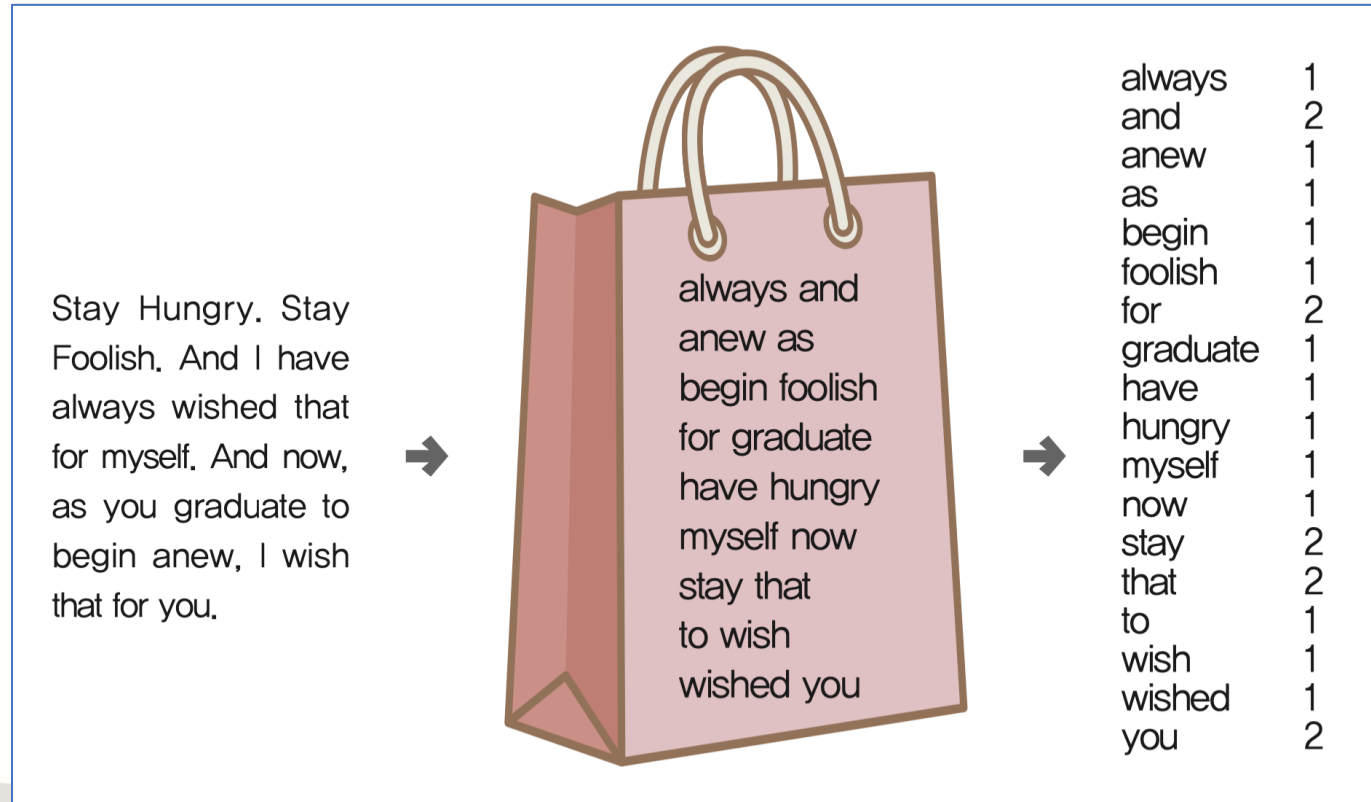
● 단어가방모형(Bag of Words, BoW)

- 단어가방모형은 매우 오래된 텍스트 모형으로 단순한 통계적 언어모형(Statistical Language Model)의 하나임
- 이 모형은 단어의 순서는 고려하지 않고, 각 단어의 출현 빈도만을 계산함
- 단어가방모형을 만드는 방법은 문서 전체에 출현한 단어의 리스트를 만들고 개별단어가 문서에 몇 번 등장하는지를 세는 것

4. 단어의 표현방법

• 단어가방모형(Bag of Words, BoW)

- 스티브 잡스의 스탠포드 대학교 졸업 축사의 일부 문장에 단어가방모형을 적용한 사례



4. 단어의 표현방법

● 문서-단어 행렬(Document-Term Matrix)

- 문서-단어 행렬(Document-Term Matrix)은 단어가방모형을 이용하여 문서에 출현하는 단어들의 빈도를 행렬로 표현한 것

문서 1: tom plays soccer

문서 2: tom loves soccer and baseball

문서 3: baseball is his hobby and his job

- 위의 세 문서로부터 추출할 수 있는 단어의 목록(BoW)은 다음과 같다

BoW: ['tom', 'plays', 'soccer', 'loves', 'and', 'baseball', 'is', 'his', 'hobby',
'job']



4. 단어의 표현방법

- 문서-단어 행렬(Document-Term Matrix)

- 이를 바탕으로 각 문서에 출현한 각 단어의 빈도수를 계산한 표

문서 1: tom plays soccer

문서 2: tom loves soccer and baseball

문서 3: baseball is his hobby and his job



	tom	plays	soccer	loves	and	baseball	is	his	hobby	job
문서 1	1	1	1	0	0	0	0	0	0	0
문서 2	1	0	1	1	1	1	0	0	0	0
문서 3	0	0	0	0	1	1	1	2	1	1

4. 단어의 표현방법

● 문서-단어 행렬(Document-Term Matrix)

- 각 문서의 행렬은 다음과 같이 표현할 수 있음

문서 1: tom plays soccer

문서 2: tom loves soccer and baseball

문서 3: baseball is his hobby and his job

	tom	plays	soccer	loves	and	baseball	is	his	hobby	job
문서 1	1	1	1	0	0	0	0	0	0	0
문서 2	1	0	1	1	1	1	0	0	0	0
문서 3	0	0	0	0	1	1	1	2	1	1



문서1 = [1, 1, 1, 0, 0, 0, 0, 0, 0, 0]

문서2 = [1, 0, 1, 1, 1, 1, 0, 0, 0, 0]

문서3 = [0, 0, 0, 0, 1, 1, 1, 2, 1, 1]

4. 단어의 표현방법

- 문서-단어 행렬(Document-Term Matrix)
 - 문서-단어 행렬의 장점은 특정 단어가 포함된 문서를 찾거나 특정 단어가 어떤 문서에서 얼마나 중요도를 가지는지 파악하는데 도움이 됨
 - 모든 단어가 숫자로 표현되기 때문에 행렬을 이용한 다양한 연산이 가능
 - 그러나 실제로는 위의 예와는 달리 문서에 출현한 단어가 방대하여 행렬의 차원이 수천에서 수만 이상이 될 가능성이 있어 방대한 메모리 공간과 컴퓨팅 자원을 요구하게 됨

4. 단어의 표현방법

- **TF-IDF**(Term Frequency Inverse Document Frequency)
 - 단어 빈도를 뜻하는 TF(Term Frequency)는 특정한 단어가 문서 (혹은 문장) 내에서 얼마나 자주 등장하는지를 나타냄
 - 일반적으로 특정 단어의 TF값이 높으면 문서에서 중요한 단어라고 판단
 - 그러나 어떤 단어가 하나의 문서에서만 자주 등장하는 것이 아니라 여러 문서에서 공통적으로 자주 등장한다면 그 단어는 문서를 대표하는 단어로서의 중요도는 상대적으로 낮아짐
-> 예) a, the, of, is 등

4. 단어의 표현방법

- TF-IDF(Term Frequency Inverse Document Frequency)

- 따라서 각 문서를 대표하는 중요한 단어를 찾아야 하는데, 이때 TF-IDF(Term Frequency Inverse Document Frequency)를 사용
- DF(Document Frequency)는 전체 문서들 중 몇 개의 문서에 해당 단어가 출현했는지를 표현한 값

$$DF = \text{해당 단어가 나타난 문서 수} / \text{전체 문서 수}$$

- IDF(Inverse Document Frequency)는 DF의 역수로 가장 DF가 큰 값을 1로 만들기 위해 사용

$$IDF = \log(\text{전체 문서 수} / \text{해당 단어가 나타난 문서 수})$$

4. 단어의 표현방법

• TF-IDF(Term Frequency Inverse Document Frequency)

- 표를 살펴보면 문서 1에서는 'plays', 문서 2에서는 'loves', 문서 3에서는 'his'가 가장 중요한 단어로 계산됨
- TF-IDF는 문서의 검색 등에 주로 활용됨
- 예를 들어 'loves'가 포함된 문서를 검색하면 문서 2가 검색됨
- TF-IDF는 오래된 문서의 단어 표현 모형이지만 현재도 종종 사용하는 기법

	tom	plays	soccer	loves	and	baseball	is	his	hobby	job
문서 1	0.18	0.48	0.18	0	0	0	0	0	0	0
문서 2	0.18	0	0.18	0.48	0.18	0.18	0	0	0	0
문서 3	0	0	0	0	0.18	0.18	0.48	0.96	0.48	0.48

빅데이터의
이해와 활용

5 언어모형



5. 언어모형

● 언어모형(Language Model)

- 단어 시퀀스에 대한 확률 분포(probability distribution)를 구해 언어를 처리하는 모형
- 특정 단어나 문장이 있을 때 다음에 나타날 단어나 문장에 대한 확률적 분포를 구함
- 과거에는 통계적 모형이 주로 이용되었지만 최근에는 인공지능망을 이용한 모형이 주로 사용됨
- 오타의 수정, 다음 단어 예측을 통한 문장 생성 등 여러 분야에 언어모델이 활용됨

5. 언어모형

언어모형을 이용한 오타의 수정

$$P(\text{'내일 아점에 연락 줘'}) < P(\text{'내일 아침에 연락 줘'})$$

- 언어모형을 이용하여 위의 두 문장의 확률을 구해보면 '내일 아침에 연락 줘'라는 문장이 '내일 아점에 연락 줘'라는 문장보다 높게 나타남
- 일반적으로 '아점에 연락'이라는 단어의 시퀀스보다는 '아침에 연락'이라는 단어의 시퀀스의 확률이 더 높기 때문

5. 언어모형

언어모형을 이용한 다음 단어 예측

‘어제 밤에 라면을 먹고 잤더니 얼굴이 ____’

- 언어모형에서 얼굴이 다음에 올 확률이 가장 높은 단어는 ‘부었다’ 일 것. 우리의 언어습관에서 ‘얼굴이’에 이어서 가장 많이 사용하는 단어이기 때문
- 따라서 일상 언어를 바탕으로 학습된 언어모형에서도 ‘부었다’, ‘붓는다’ 등의 표현이 등장할 확률이 가장 높을 것
- 언어모형을 구축하기 위해서는 텍스트 빅데이터의 학습이 필요한데 전통적으로는 N-gram 언어모형이 많이 사용됨

5. 언어모형

● N-gram 언어모형

- N-gram 언어모형은 단어의 출현횟수에 기반하여 통계적 모델을 구축함
- 그러나 시퀀스 예측을 위해 이전에 등장한 단어를 모두 예측하지 않고 일부 단어만을 사용

예 trigrams의 예

“어제 자기 전에 라면을 먹었더니 얼굴이 부었다”의 trigrams은 아래와 같다.

trigrams: [(‘어제’, ‘자기’, ‘전에’), (‘자기’, ‘전에’, ‘라면을’), (‘전에’, ‘라면을’, ‘먹었더니’), (‘라면을’, ‘먹었더니’, ‘얼굴이’), (‘먹었더니’, ‘얼굴이’, ‘부었다’), (‘얼굴이’, ‘부었다’, ‘ ‘)]

정리하기

- 텍스트를 적절하게 전처리를 해주지 않으면 분석과정에서 잘못된 결과를 만들어낼 수 있다. 텍스트 전처리로는 토큰화, 정규화, 형태소분석 등이 있다.
- 단어의 표현방법으로는 단어의 출현 빈도를 계산하는 단어가방모형(Bag of Words) 과 각 문서를 대표하는 중요한 단어를 찾는 TF-IDF(Term Frequency Inverse Document Frequency) 등이 있다.

06

강

다음시간 안내

빅데이터 시각화

수고하셨습니다!

