캡스톤디자인(2)

Web Crawling, Wordnet, NLP 활용 기반 기사 통합 요약 시스템

목차

1. 개요 및 목적

1.1. 프로젝트 개발 동기

현재 우리가 살아가고 있는 지식 정보화 시대는 사회 일반 산업 자원으로 활용될 수 있도록 가공된 정보와 지식이 사회 구조나 관습 및 인간의 가치관에 큰 영향을 미치는 시대이다. 이러한 정보의 습득은 주로 검색을 통해 이루어진다. 정보의 양이 거대해짐에 따라 기대하는 정보의 선택적인 습득이 어려워지고 이를 검색하기 위한 키워드를 설정하는 것 역시 사람들에게 또다른 과제로 주어진다.

현재 원하는 키워드의 뉴스를 검색하여 정보를 얻는 과정은 다음과 같다. 사용자는 통합 웹 브라우저를 통해 원하는 키워드를 검색하고 웹 브라우저는 이 키워드를 포함하는 여러 뉴스 매체의 기사들을 사용자에게 제공한다. 사용자는 검색의 목적을 달성하기 위하여 각 기사를 개별적으로 선택함으로써 상세 내용을 파악하고 습득한다. 선택되는 기사들은 중복되는 내용을 포함할 가능성이 존재한다. 이러한 중복으로 인하여 사용자는 습득한 기사 내용과 상이한 정보를 파악하기 위하여 해당 정보를 포함하는 기사를 찾을 때까지 선택과 습득을 반복해야 한다. 사용자 관점에서 반복적인 과정은 소요 시간 측면에서 굉장히 비효율적이다.

이 프로젝트에서는 합리적인 시간동안 사용자의 효율적인 기사 정보 습득 보조를 목적으로 한다.

1.2. 프로젝트 개요

해당 프로젝트는 Web Crawling으로 뉴스 기사 데이터를 가져오고 주로 데이터 시각화에 많이 사용되는 Word Cloud를 사용하여 기사 내용을 요약한다. 이 과정에서 Word2Vec을 사용하여 Word Cloud의 성능을 향상시킨다. 해당 프로그램의 작동 과정을 살펴보면 다음과 같다.

Word Cloud는 핵심 단어 시각화 표현 기법으로 문서의 문구와 단어를 분석하여 중요도와 사용 빈도를 기반으로 문서의 직관적인 파악을 보조한다. KorLex는 단어의 동의어 집합, 상위어, 그리고 하위어에 대한 정보를 바탕으로 하는 한국어 Wordnet이다. 이러한 Wordnet을 활용하면 기존에 보유하고 있던 정보 또는 단어의 양을 확장할 수 있다. Word2Vec은 단어를 벡터로 변환하는 도구로 단어의 의미와 관련성을 수학적으로 표현하는 기법이다. 이는 신경망 기반의 언어 모델로 주변 단어의 문맥을 이용하여 단어를 벡터로 표현한다. 이를 통해 단어 간 유사성을 계산하고 단어 간 관련성을 파악할 수 있다. Word2Vec을 이용하면 주어진 단어와 Wordnet으로 확장한 단어들 간의 벡터 유사성을 기반으로 키워드의 확장이 가능하다. 또한 이를 Word Cloud 생성 전처리 과정에 사용하면 비슷한 단어의 반복을 방지할 수 있고 관련 있는 단어들을 묶을 수 있다.

이 프로젝트에서는 언급한 기술들을 기반으로 사용자가 뉴스 검색 시 키워드를 입력하면 주간 뉴스 정보를 Word Cloud로 제공해주는 서비스를 개발한다. 사용자는 해당 키워드 단독, 또는 확장된 키워드에 대한 검색의 여부를 선택할 수 있다. 제공된 Word Cloud는 키워드에 해당하는 뉴스들에 대한 내용을 사용자가 한 눈에 요약하여 파악할 수 있게 돕는다. 이를 통하여 원하는 뉴스의 내용을 개별적으로 읽고 정리하기 힘들거나 주간 뉴스 내용을 포괄적으로 간단히 파악하고 싶은 바쁜 현대인들에게 높은 효율의 정보 습득 제공을 기대한다.

2. 배경

2.1. Word Cloud

워드 클라우드(Word Cloud)는 텍스트 데이터 시각화 도구이다. 워드 클라우드는 텍스트 데이터를 빈도수 기반 순위를 바탕으로 단어들을 시각적으로 강조하여 표현한다. 이는 텍스트 상대적 빈도의 즉각적인 시각적 이해를 보조한다. 워드 클라우드는 다음과 같은 과정으로 생성한다.

1. 텍스트 데이터 수집: 워드 클라우드 생성을 위한 데이터셋은 텍스트 데이터이다. 해당 데이터셋은 웹 문서, 뉴스 기사와 같은 다양한 자원에서 추출할 수 있다.
2. 전처리(Preprocessing): 데이터셋을 활용하기 위하여 전처리 작업이 선행된다. 불필요한 문자, 구두점, 불용어(Stopword; “a”, “the”, “is”와 같이 빈번하게 등장하는 단어)를 제거하거나 형태소 분석을 통하여 문장을 단어의 기본 형태로 변환하는 작업을 포함한다.

3. 단어 빈도 계산: 전처리한 데이터셋에서 단어의 빈도 수를 계산한다. 빈도수를 기반으로 단어의 상대적 중요도를 파악할 수 있다.
4. Word Cloud 생성: 이전 단계가 완료되면 워드 클라우드를 생성한다. 빈도수가 높은 단어는 크고 강조되게 표현되고 빈도수가 낮은 단어는 작고 적게 강조되게 표현된다. 일반적으로 워드 클라우드는 단어의 크기, 색상, 배치 등을 이용하여 시각적인 효과를 부여한다.

워드 클라우드는 주제 분석, 텍스트 요약, 시각적인 표현을 통한 커뮤니케이션과 같은 다양한 분야에 활용한다. 이 프로젝트에서는 워드 클라우드의 범용성을 활용한다.

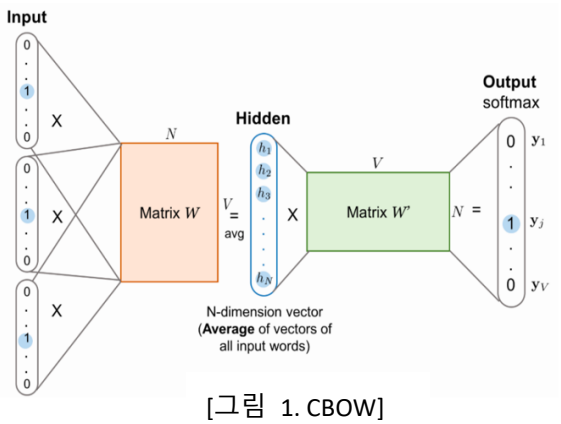
2.2. Word2Vec

Word2Vec은 단어를 벡터로 변환하는 딥러닝 기반 모델로 단어의 의미와 관련성을 수학적으로 표현한다. 단어 간 유사성을 계산하고 계산된 벡터를 기반으로 단어 간 관련성을 파악할 수 있다.

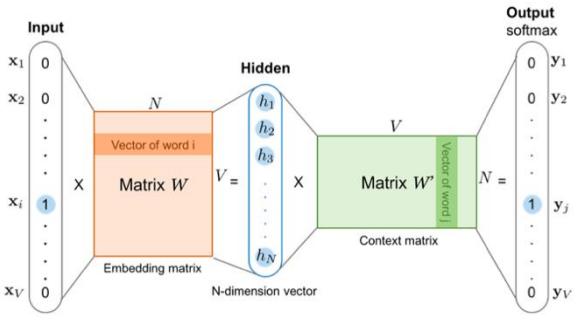
Word2Vec을 대표하는 두 모델을 소개한다.

1. CBOW(Continuous Bag of Words): CBOW는 Context 내 주변 단어들을 입력으로 사용하여 중심 단어를 예측하는 모델이다. [그림 1]과 같이 CBOW는 입력으로 사용된 단어들의 평균을 기반으로 중심 단어를 예측하여 모델을 학습한다. 주로 작은 데이터셋에서 높은 성능을 보인다.

2. Skip-gram: Skip-gram은 CBOW의 반대 개념 기반 학습 모델이다. [그림 2]에서 볼 수 있듯이 중심 단어를 입력으로 사용하여 주변 단어를 예측하여 모델을 학습한다. Skip-gram은 중심 단어와 주변 단어 쌍으로 입력으로 사용한다. 주로 큰 데이터셋에서 높은 성능을 보인다.



[그림 1. CBOW]



[그림 2. Skip-gram]

Word2Vec에서 활용하는 단어 유사성은 벡터 간의 거리나 코사인 유사도(Cosine Similarity)로 측정한다. Word2Vec 모델을 사용 시 "King"과 "Queen" 간의 관련성이 "Man"과 "Woman" 간의 관련성과 유사한가와 같은 판단이 가능하다.

Word2Vec은 자연어 처리(NLP; Natural Language Processing), 문서 분류, 문서 군집화, 정보 검색 등 다양한 응용 분야에 활용한다. 단어 간 의미적 유사성, 단어의 특징을 활용하여 다양한 텍스트 기반 작업을 수행한다. 이 프로젝트에서는 Word2Vec 모델을 단어 유사성 판단에 활용한다.

3. 설계 및 구현

3.1. 프로젝트 일정

3월	프로젝트 개요 도출, 유사 서비스 검토, 기능 확정
4월	소요 기술 및 소요 자원 검토, 소요 기술 학습
5월	뉴스 매체 페이지 분석 및 전처리
여름 방학	불용어 처리 개선
9월	전처리 Word2Vec 도입
10월	전처리 Word2Vec 도입
11월	KorLex 도입, 프로그램 UI 구현
12월	최종 보고서 작성

3.2. 개발 환경

구분	상세 내용
OS	Docker (Linux)

IDE	Jupyter notebook
개발 언어	Python
버전 관리	Docker

3.3. 개발 요구 명세: Requirement Specification

3.3.1. 기능적 요구사항

① 사용자는 검색어를 입력하고 단어 확정 여부를 선택한다.
- 단어 단독 또는 상위어/하위어/동의어 포함 선택이 가능하다.

② 웹 크롤링을 통한 데이터셋 준비
- 검색어가 주어졌을 때 해당 뉴스 매체들의 웹 사이트를 통하여 해당 검색에 대한 기사 검색 결과를 크롤링하여 제공하여야 한다.

③ 효율적인 Word Cloud를 위한 데이터셋의 1차 가공
- 데이터셋에 대한 형태소 분석이 필요하다.
- 형태소 분석이 완료된 데이터셋에 대하여 불용어, 중요도가 낮은 단어를 제외해야 한다.

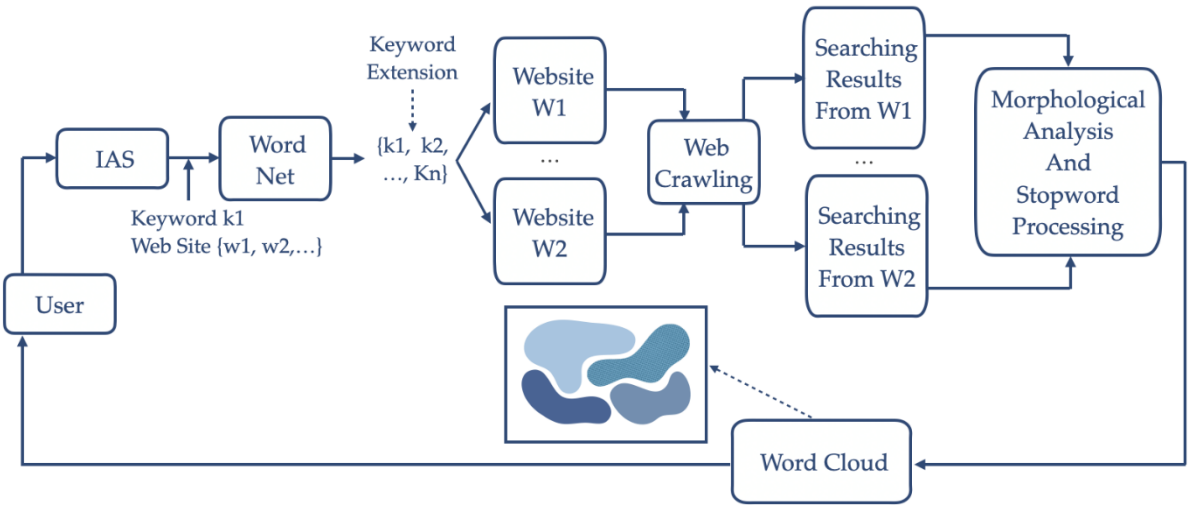
④ 효율적인 Word Cloud를 위한 데이터셋의 2차 가공
- 형태소 분석과 불용어 처리가 완료된 데이터셋을 기반으로 유사한 의미를 가진 단어들 간의 그룹화가 필요하다.
- Word2Vec을 이용하여 단어 벡터간 유사성을 활용하여 그룹화한다.

3.3.2. 비기능적 요구사항

① 사용자가 검색을 했을 때 응답 시간은 3초 이내여야 한다.
- 웹 크롤링 시 응답 시간을 최소화 해야 한다.

② 사용자가 프로그램의 인터페이스를 부가적인 설명 없이도 이해하여 사용할 수 있도록 제공해야 한다.
- 이해하기 쉬운 인터페이스를 개발 해야 한다.

3.4. 개발 설계: Software Architecture



3.5. 실행 결과

3.5.1. Test 환경

3.5.2. Test 결과

3.5.2.1. Preprocessing: Dataset

데이터셋은 BeautifulSoup4을 통하여 검색어를 뉴스매체에서 검색한 결과를 가져와 준비한다. Article body 하나를 살펴보면 다음과 같다.

정우영의 물오른 골 감각이 또 다시 승리를 안겼다. 득점 선두 행진도 탄력을 붙였다.황선홍 감독이 이끄는 24살 이하 남자축구 대표팀은 4일 중국 항저우 황룽 스포츠센터 스타디움에서 열린 2022 항저우아시안게임 준결승전 우즈베키스탄과 경기에서 정우영의 멀티골로 2-1로 이겼다. 한국은 홍콩을 4-0으로 제압하고 결승에 오른 일본과 7일 밤 우승컵을 놓고 다툰다. 아시안게임 두 대회 연속 결승 한일전 성사다.황선홍 감독은 이날 선발 공격진에 이강인(파리 생제르맹)과 정우영(슈투트가르트), 엄원상(울산)을 배치했다. 최전방의 조영욱(서울)을 빼고는 전방 공격수들이 8강 중국전 때와는 달라졌다. 이영표 해설위원은 “최강의 멤버로 선발진을 구성했다”고 분석했다.황선홍 감독은 중원에는 중국전 때와 마찬가지로 주장 백승호(전북)와 홍현석(헨트)에게 공·수의 연결과 공 배급 등을 맡겼다. 수비진에는 설영우(울산), 박진섭(전북), 이한범(미트윌란), 황재원(대구)이 늘어섰고, 골문은 이광연(강원)이 지켰다.시작부터 속도와 패스 플레이로 날카롭게 파고든 한국의 첫골 결실은 전반 4분 나왔다. 오른쪽 측면의 엄원상이 상대 진영 배후를 파고든 뒤 낮고 강하게 공을 올렸고, 침투하던 정우영이 가볍게 발로 터치하면서 골망을 흔들었다. 정우영의 대회 6번째 골.홍현석과 엄원상이 만들어내는 침투와 이강인의 좌우 횡단 패스, 정우영의 발 빠른 침투는 반복적으로 우즈베키스탄의 골문을 위협했다. 앞선의 조영욱도 중거리포로 공세의 파고를 높였다.하지만 우즈베키스탄도 만만치 않았다. 강대강으로 맞선 우즈베키스탄은 전반 25분 벌칙구역 앞 부근에서 프리킥 반칙을 얻어냈고, 주장 자로리디노프가 때린 공은 수비수를 맞고 굴절되면서 몸을 날린 이광연 골키퍼의 손을 맞고 들어갔다. 경기장을 가득 메운 중국 관중들은 수세에 몰렸던 우즈베키스탄을 응원했고, 골이 터지자 환호했다.이후 치고받는 팽팽한 경기는 정우영의 멀티골로 다시 한국으로 기울었다. 정우영은 전반 38분 상대 골지역 왼쪽을 파고들며 상대 수비수가 미처 걷어내지 못한 공을 주워 먹듯 골망 안으로 넣으며 흐름을 가져왔다. 정우영의 대회 7호골.후반 초반은 우즈베키스탄의 공세 파고가 높았다. 한국은 후반 8분께 아크 부근에서 프리킥을 내줬고, 상대 키커의 날카로운 슈팅을 이광연 골키퍼가 정면에서 막아내면서 가슴을 쓸어내렸다.황 감독은 후반 15분께 송민규(전북)와 정호연(광주)을 투입하고 이강인과 정우영을 빼주면서 다시 팀 동력을 높였다. 상대의 거친 플레이도 이어져, 후반 17분께는 엄원상이 돌파하다 위험하게 넘어지기도 했다. 결국 전반 경고를 받았던 우즈베키스탄의 부리프가 후반 28분 조영욱을 막다가 경고누적으로 퇴장당하면서 한국은 수적 우위를 누리게 됐다.이후 한국의 조영욱과 안재준(부천) 등이 개인 능력을 발휘해 골문 앞에서 좋은 기회를 만들었지만 추가골은 터지지 않았다. 오히려 10명이 싸운 우즈베키스탄이 투혼의 경기를 펼치면서 황선홍 감독은 종료 휘슬이 울릴 때까지 안심할 수 없었다.항저우/김창금 선임기자

3.5.2.2. 전처리: Kkma

태깅을 통해 품사를 구분하고 동사와 명사만 추출하였다. Stopword Set을 지정하여 추가적인 불용어를 처리하였다. 이 부분은 Okt, hannanum, komoran도 동일하다.

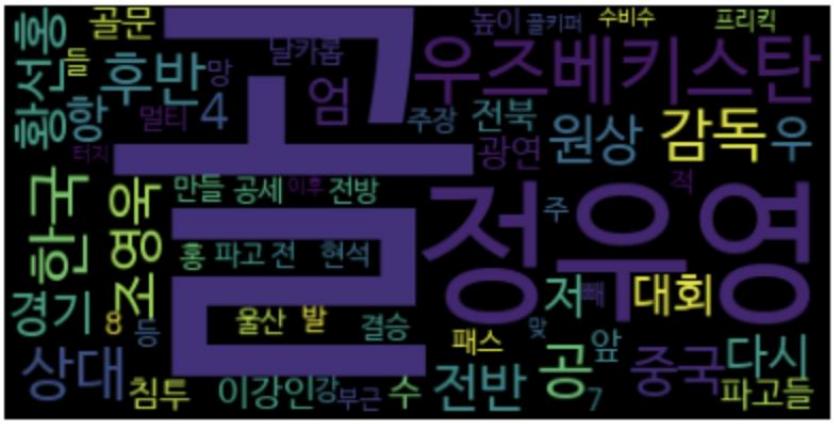
```
['정우',
 '영',
 '물오르',
 '골',
 '감각',
 '다시',
 '승리',
 '안기',
 '득점',
 '선두',
 '행진',
 '탄력',
 '붙이',
 '황',
 '선',
 '홍',
 '감독',
 '이끌',
 '24',
```

전처리한 결과 시각화를 위해 Word Cloud를 사용하였다.

3.5.2.3. 전처리: Okt

3.5.2.4. 전처리: Hannanum

3.5.2.5. 전처리: Komoran



3.5.2.6. 전처리: Word2Vec

3.5.2.7. 전처리 성능 비교

3.5.2.8. Word Cloud 최종 결과

3.5.3. 프로젝트 결과

4. 결론 및 향후 계획

4.1.1. 결론

4.1.2. 추후 발전 방향

5. 참고 문헌

[1] 'Natural Language Processing with Transformers; Building Language Applications with Hugging Face',

Lewis Tunstall, Leandro von Werra & Thomas Wolf, O'Reilly

[2] 'Word Cloud Explorer: Text Analytics Based on Word Clouds', Heimerl, Florian. 2014 47th Hawaii international Conference on System Sciences ISBN