캡스톤 디자인(2)

Web Crawling, NLP 기반 Semantic Cloud 기사 통합 요약 시스템

목차

1 개요 및 목적 ---------- 3

1.1 프로젝트 개발 동기

1.2 프로젝트 개요

2 배경 ---------- 4

2.1 KoNLPy

2.2 Word2Vec

2.3 t-SNE

2.4 K-medoids

2.4.1 Silhouette method

2.5 Word Cloud

2.5.1 Semantic Cloud

2.6 기존 연구 분석

3 설계 및 구현 ---------- 8

3.1 개발 요구 명세: Requirement Specification

3.1.1 기능적 요구사항

3.1.2 비기능적 요구사항

3.2 개발 설계: Software Architecture

3.3 데이터셋: 웹 크롤링 결과

3.4 데이터셋: 전처리

3.4.1 1차 전처리: 형태소 분석 및 불용어 처리

3.4.2 2차 전처리: Word2Vec

3.4.3 3차 전처리: t-SNE 기반 차원 축소 및 빈도수 포함

3.4.4 4차 전처리: K-medoids

3.5 Semantic Cloud: 클러스터링 기반 색상 도입 및 재배치

4 실행 결과 ---------- 14

4.1 실행 환경

4.2 프로젝트 최종 실행 결과

5 결론 및 향후 확장 가능성 ---------- 15

5.1 결론

5.2 향후 확장 가능성

6 참고 문헌 ---------- 17

1. 개요 및 목적

1.1. 프로젝트 개발 동기

지식 정보화 시대에서 사람들의 정보 습득은 주로 검색을 통해 이루어진다. 정보의 양이 증가함에 따라 검색 결과를 정제하기 위한 시간이 필요하고 이에 따라 검색에서 습득까지 불필요한 시간과 에너지가 소비된다.

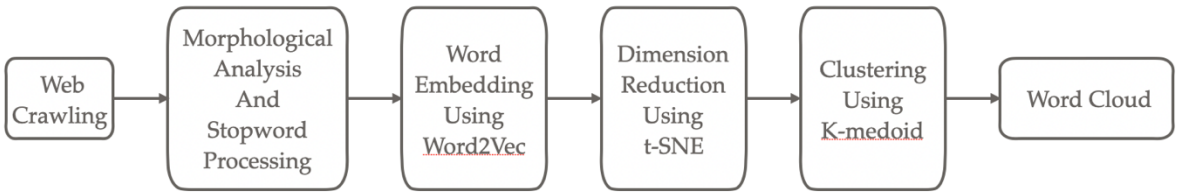
기존 사용자가 인터넷 뉴스를 통하여 정보를 얻기까지 과정을 알아보자. 먼저 원하는 키워드를 검색한다. 결과로 키워드를 포함하는 다양한 뉴스 매체의 기사들이 제공된다. 뉴스의 내용을 파악하기 위해 사용자는 개별적으로 뉴스 내용을 읽어야한다. 제공되는 뉴스 정보는 중복되는 내용을 포함한다. 키워드에 해당하는 여러 정보를 습득하기 위해서는 반복적인 작업이 필요하다. 이 과정은 걸리는 시간과 노력 측면에서 비효율적이다.

이러한 이유로 인하여 사용자가 키워드 검색 시 최근 뉴스의 전체적인 정보를 우선적으로 파악 가능한 뉴스 통합 검색 요약 시스템 프로젝트를 진행하였다.

1.2. 프로젝트 개요

프로젝트는 웹 크롤링(Web Crawling)으로 키워드를 포함하는 뉴스 기사 데이터셋을 구축한다. 이후 다양한 자연어처리 기술을 전처리 과정에 활용하여 문맥 파악이 가능한 워드 클라우드(Word Cloud)인 Semantic Cloud를 사용자에게 제공한다. 프로그램의 작동 과정을 간단히 살펴보자.

사용자는 검색어를 입력한다. 검색어를 포함하는 기사들을 Web Crawling을 통하여 데이터셋으로 구축한다. 데이터셋에 Word2Vec 모델을 적용하여 문맥 정보를 벡터화하고 t-SNE를 통하여 2차원으로 벡터 차원을 축소한다. 축소한 벡터 데이터셋에 K-medoiod를 적용하여 벡터 유사성을 기반 군집 분석(Clustering)을 진행한다. 군집(Cluster) 기반으로 단어 집합에 색상을 지정하고 빈도수 순으로 Word Cloud를 출력한다. 최종 Semantic Cloud를 사용자에게 제공한다.



[그림1] 프로그램 작동 과정

2.1. KoNLPy

한국어는 전세계에서 13번째로 많이 사용되는 언어이다. 한국어에는 영어에는 존재하지 않는 '조사'라는 개념으로 인하여 문장 안에서 단어의 순서가 변경되어도 뜻이 변하지 않는 특징이 존재한다. 따라서 영어 기반의 형태소 분석과 한국어 문장의 형태소 분석은 다른 접근 방식이 필요하다. KoNLPy는 오픈 소스 소프트웨어이며 한국어 형태소 분석을 위한 패키지를 제공한다. KoNLPy는 KAIST Semantic Web Research Center이 개발한 Hannanum, 서울대학교 IDS 연구실이 개발한 Kkma, Shineware에서 개발한 Komoran, 과거 트위터 형태소 분석에 사용된 오픈 소스 한국어 분석기 Open Korean Text(Okt)와 같은 다양한 형태소 분석 태깅 라이브러리를 파이썬 환경에서 사용할 수 있도록 포함한다. 형태소 분석 태깅 라이브러리를 기반으로 모든 품사의 형태소 분석이 가능하다.

2.2. Word2Vec

Word2Vec는 신경망(Neural Network) 기반의 언어 모델로 주변 단어의 문맥을 이용하여 단어를 벡터로 표현한다. 단어 임베딩(Word Embedding)을 통하여 단어 간 유사성을 계산하고 단어 간 관련성을 파악할 수 있다.

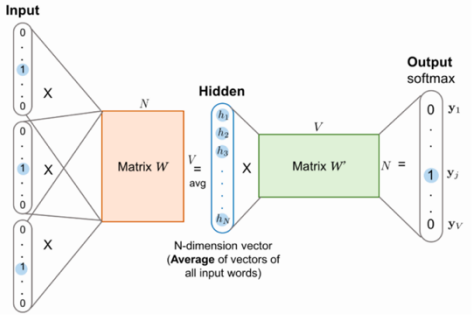
Word2Vec은 주로 두가지 방법으로 구현된다: CBOW(Continuous Bag of Words)와 Skip-gram이다.

1. CBOW: CBOW는 context 내 주변 단어들을 입력으로 사용하여 중심 단어를 예측한다. 주변 단어들의 평균을 이용하여 중심 단어를 예측한다. 주로 데이터셋의 크기가 작을 경우에 사용한다.

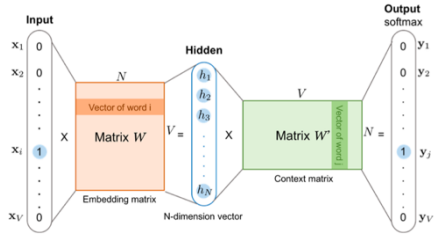
2. Skip-gram: Skip-gram은 CBOW의 반대 개념으로 중심 단어를 입력으로 사용하여 주변 단어를 한다. 중심 단어와 주변 단어를 한 쌍으로 사용하여 모델을 학습한다. 주로 데이터셋의 크기가 클 경우 사용하며 일반적으로 CBOW에 비하여 성능이 좋다.

Word2vec에서 벡터 유사성은 벡터 간의 거리 또는 코사인 유사도(Cosine similarity)로 측정한다. 모델 학습 후 "king"과 "queen" 간의 관련성이 "man"과 "woman" 사이의 관련성과 유사함을 파악할 수 있다.

Word2vec는 자연어 처리(Natural Language Processing), 문서 분류, 문서 군집화, 정보 검색 등 다양한 응용 분야에서 활용한다. 단어 간 의미적 유사성, 단어의 특성을 활용하여 다양한 텍스트 기반 작업을 수행할 수 있다.



[그림2] CBOW



[그림3] Skip-gram

2.3. t-SNE

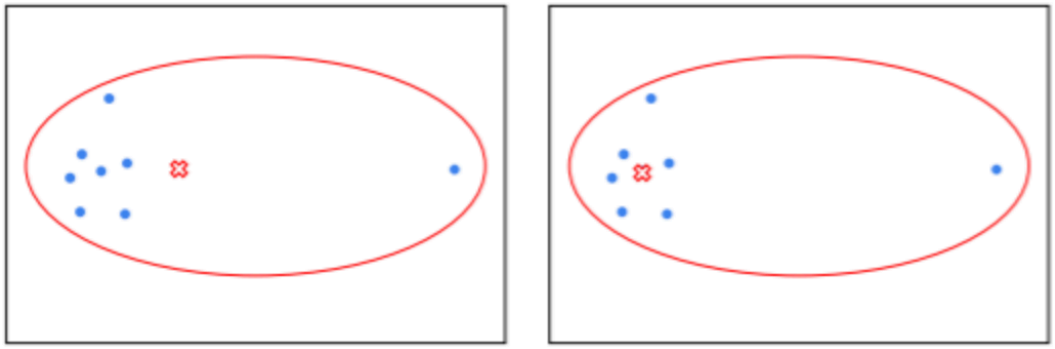
t-SNE는 비선형적인 방법의 차원 축소 방법이다. 고차원의 데이터셋 시각화 시 좋은 성능을 보인다. 고차원 공간에서의 점들의 유사성과 그에 해당하는 저차원 공간에서 점의 유사성을 계산한다. 점 A, B가 존재할 때 A 중심 정규분포에서 확률 밀도에 비례하여 이웃 선택 시 점 A가 점 B를 이웃으로 선택한다는 조건부 확률로 계산된다. 고차원 및 저차원 공간에서의 조건부 확률(또는 유사점) 간 차이를 최소화하는 방향으로 진행한다. 조건부 확률 차이의 합을 최소화하기 위해 경사 하강 알고리즘(Gradient descent) 방법을 적용하여 전체 데이터 포인트의 KL-divergence 합계를 최소화한다. KL-divergence는 한 확률 분포가 두번째 예상 확률 분포와 다른 정도를 측정하는 척도이다.

t-SNE는 고차원에서 쌍으로 이루어진 유사성을 측정하는 분포와 저차원에서 쌍으로 유사성을 측정하는 두가지 분포의 KL-divergence를 최소화한다. 결론적으로 t-SNE는 고차원의 데이터를 낮은 차원의 공간으로 매핑하고 다수의 특징을 포함하는 데이터 포인트의 유사성을 기반으로 점들의 클러스터를 식별함으로써 데이터에서 패턴을 발견한다.

[그림4] t-SNE 예시

2.4. K-medoids

K-means Clustering이 초기 중심값에 민감한 반응을 보이고 노이즈(Noise)와 이상값(Outlier)에 민감하다는 단점을 보완하여 변형한 비지도 학습(Unsupervised learning) 클러스터링이다. 클러스터의 무게 중심을 구하기 위하여 데이터의 평균이 아닌 중간점(Medoids)을 사용한다. 실제 클러스터에 존재하는 점을 무게 중심으로 활용하기 때문에 데이터의 outlier에 대하여 K-means Clustering에 비하여 좋은 성능을 보인다. 최적의 클러스터 개수 K를 구하기 위하여 K-means와 동일하게 Elbow method와 Silhouette method를 이용한다.



[그림5] K-means, K-Medoids 중심점 선택 예시(좌측 K-means, 우측 K-medoids)

2.4.1. Silhouette method

Silhouette method는 객체와 그 객체가 속한 클러스터 데이터들의 비유사성(Dissimilarity)을 계산한다. a(i)는 객체 i와 객체가 속하는 클러스터 데이터들의 비유사성을 의미한다. g(x_i)는 x_i가 속한 클러스터이다.

$$a(i) = \frac{1}{|g(x_i)| - 1} \sum_{j \in g(x_i)} d(x_i, x_j) \quad \text{---------- 식 1}$$

b(i)는 객체가 속하지 않은 다른 클러스터 데이터들의 비유사성을 계산한다. gk는 xi가 속하지 않은 다른 클러스터 k 이다.

$$b(i) = min_k \left(\frac{1}{|g_k|} \sum_{j \in g_k} d(x_i, x_j) \right) \quad \text{---------- 식 2}$$

식1, 2에 따라 실루엣 s(i)는 식3으로 계산된다.

$$s(i) = \frac{b(i) - a(i)}{max[a(i), b(i)]} \quad where -1 \leq s(i) \leq 1 \quad \text{---- 식 3}$$

2.5. Word Cloud

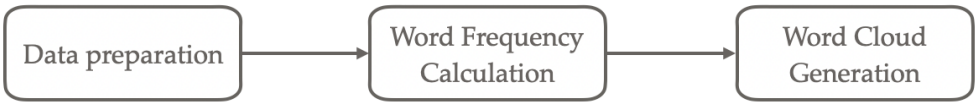
Word Cloud는 텍스트 데이터의 시각화를 위해 사용한다. 특히 텍스트 데이터에서 가장 빈번하게 등장하는 단어들을 시각적으로 강조하여 표현한다. 텍스트의 상대적인 빈도를 시각적으로 이해하기 쉽게 해주어 텍스트 데이터의 특징을 파악할 수 있다. Word Cloud는 다음과 같은 과정으로 생성된다.

1. 텍스트 데이터 수집: Word Cloud를 생성하기 위해서는 데이터셋이 필요하다. 이 데이터는 웹 문서, 뉴스 기사 등 다양한 자원에서 가져올 수 있다.
2. 전처리: 분석 전 전처리 작업을 수행해야 한다. 불필요한 문자, 구두점, 불용어(“a”, “the”, “is”와 같이 빈번하게 등장하는 단어)를 제거하거나 형태소 분석을 통해 단어의 기본 형태로 변환하는 것과 같은 작업을 포

함할 수 있다.

3. 단어 빈도 계산: 전처리한 데이터셋에서 단어의 빈도를 계산한다. 텍스트 내 단어의 상대적 중요도를 파악할 수 있다.
4. Word Cloud 생성: 단어 빈도 계산을 완료한 리스트로 Word Cloud를 생성한다. 빈도가 높은 단어는 크고 강조되게 표현되고 빈도가 낮은 단어는 작고 덜 강조되게 표현된다. 일반적으로 Word Cloud는 단어의 크기, 색상, 배치 등을 이용하여 시각적인 효과를 부여한다.

Word Cloud는 주제 분석, 텍스트 요약, 시각적인 표현을 통한 커뮤니케이션 등 다양한 분야에서 활용한다. 데이터 사이언스, 자연어 처리, 정보 검색, 마케팅 분석 등에서 텍스트 데이터의 시각화와 분석에 널리 사용한다.



[그림6] Word Cloud 형성 과정

2.5.1. Semantic Cloud

Word Cloud는 텍스트 문맥에의 단어 빈도수 정보를 제공한다. Semantic Cloud는 단어의 의미론적 속성을 보존하여 텍스트의 일반적인 주제를 직관적으로 드러내고 요약하여 제공한다. 프로젝트의 Semantic Cloud는 의미론적 속성을 색상을 통하여 제공한다.

2.6. 기존 연구 분석

Florian Heimerl et al.은 기존 Word Cloud의 유용성을 탐구한다. Word Cloud 접근 방식의 실행 가능성과 효과성을 입증하는 사용하기 쉽고 강력한 다양한 구현 방식을 제안한다.

Jin Xu et al.은 의미론적 단어 표현, 단어 유사도 그래프 구축, force-directed 단어 레이아웃, Word Cloud 시각화를 통하여 Semantic Word Cloud를 제안한다. CBOW 모델을 사용하여 단어 임베딩을 형성하고 MDS를 사용하여 고차원의 단어 벡터를 저차원에 투영하여 단어 유사도 그래프를 구축한다. Energy 모델을 사용하여 단어의 위치를 최적화하고 시각화한다.

Weiwei Cui et al.은 시간이 지남에 따라 변화하는 텍스트 내용을 묘사하기 위해 Tag Cloud를 사용하는 맥락 보존 Word Cloud를 제안한다. 내용의 의미론적 일관성과 시각화의 공간적 안정성을 균형 있게 유지하는 시간 기반 Tag Cloud 레이아웃의 Word Cloud를 제안한다.

3. 설계 및 구현

3.1. 개발 요구 명세: Requirement Specification

3.1.1. 기능적 요구사항

① 웹 크롤링을 통한 데이터셋 구축
- 검색어에 해당하는 다양한 뉴스 매체의 뉴스 기사 검색 결과를 크롤링하여 하나의 데이터셋을 구축하여야 한다.

② Semantic Cloud를 위한 데이터셋의 1차 가공
- 데이터셋에 대한 형태소 분석이 필요하다.
- 형태소 분석이 완료된 데이터셋에 대해 불용어, 중요하지 않은 단어를 제외해야 한다.

③ Semantic Cloud를 위한 데이터셋의 2차 가공
- 형태소 분석과 불용어 처리가 완료된 데이터셋을 이용해 유사한 의미를 가진 단어들 간의 유사도 측정이 필요하다.
- Word2Vec을 이용해 단어 벡터를 형성해야 한다.

④ Semantic Cloud를 위한 데이터셋의 3차 가공
- Word2Vec 모델을 적용한 결과 데이터는 고차원의 데이터이므로 차원 축소가 필요하다.
- t-SNE를 이용하여 2차원으로 차원 축소가 필요하다.

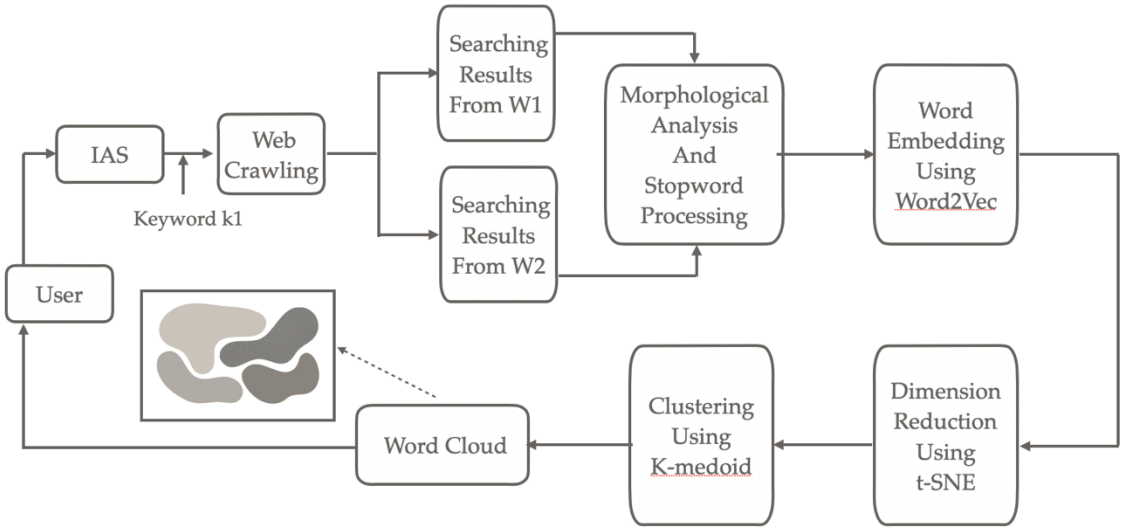
⑤ Semantic Cloud를 위한 데이터셋의 4차 가공
- t-SNE로 차원 축소한 데이터를 활용하기 위하여 군집화가 추가적으로 필요하다.
- K-medoids를 도입하여 벡터 유사도 기반 클러스터링이 추가되어야 한다.

⑥ 최종 Semantic Cloud 제공
- 데이터셋 가공을 통하여 전처리가 완료된 데이터셋을 활용하여 의미론적 특징을 보존한 Word Cloud를 형성하여야 한다.
- 클러스터링을 기반으로 같은 클러스터의 단어는 같은 색으로 출력하여야 한다.
- 빈도수를 기반으로 시각화 되어야 한다.

3.1.2. 비기능적 요구사항

① 사용자가 프로그램의 인터페이스를 부가적인 설명 없이 이해하여 사용할 수 있도록 제공해야 한다.
- 이해하기 쉬운 인터페이스 개발이 필요하다.

3.2. 개발 설계: Software Architecture



[그림7] IAS(프로그램) Software Architecture

3.3. 데이터셋: 웹 크롤링 결과

BeautifulSoup4, requests를 이용하여 사용자가 입력한 키워드에 대한 네이버 뉴스의 기사를 크롤링한다. 크롤링한 기사들의 날짜, URL, 기사 내용을 csv 파일로 저장한다. '축구'에 대한 2023. 11. 28 - 2023.11.29 데이터셋의 상위 5개 데이터를 살펴보자.

	date	link	content
0	2023.11.28.	https://n.news.naver.com/mnews/article/001/001...	[축구 국가대표 황의조[연합뉴스 자료사진](서울=연합뉴스) 안홍석 기자 = 축구 국...
1	2023.11.28.	https://n.news.naver.com/mnews/article/366/000...	[대한민국 축구 국가대표팀 황의조 /뉴스1 축구 국가대표 황의조(노리치...
2	2023.11.28.	https://n.news.naver.com/mnews/article/052/000...	[©연합뉴스불법 촬영 혐의로 경찰 수사를 받는 축구 국가대표 황의조(31·노리치시티...
3	2023.11.28.	https://n.news.naver.com/mnews/article/020/000...	[대한민국 축구 국가대표팀 황의조가 19일 오전 2026 FIFA 북중미 월드컵 아...
4	2023.11.28.	https://sports.news.naver.com/news.nhn?oid=421...	[(서울=뉴스1) 김성진 기자 = 이윤남 대한축구협회 윤리위원장이 28일 오후 서울...

[그림8] IAS(프로그램) 데이터셋 Head

0번째 데이터의 기사내용은 아래와 같다.

```
['[축구 국가대표 황의조[연합뉴스 자료사진](서울=연합뉴스) 안홍석 기자 = 축구 국가대표 황의조(노리치시티)가 성행위 영상 불법 촬영 혐의를 벗을 때까지 태극마크를 달지 못하게 됐다.    대한축구협회는 28일 오후 이윤남 윤리위원장, 마이클 뮐러 전력강화위원장, 정해성 대회위원장, 최영일 부회장 등이 참여한 회의를 열고 황의조에 대한 수사기관의 명확한 결론이 나올 때까지 그를 국가대표로 선발하지 않기로 결정했다고 밝혔다.    이 위원장은 "국가대표 선수가 고도의 도덕성과 책임감을 가지고 국가대표의 명예를 유지해야 할 의무가 있고, 그런 점에서 본인의 사생활 등 여러 부분을 관리해야 한다는 점을 고려했다"고 결정 이유를 밝혔다.    ahs@yna.co.kr]']
```

[그림9] 0번째 데이터 content

3.4. 데이터셋: 전처리

3.4.1. 1차 전처리: 형태소 분석 및 불용어 처리

구축한 데이터셋의 기사 내용은 기사 내용 파악에 불필요한 단어를 포함한다. 불필요하게 반복되는 단어, 특수문자를 제외하고 한 글자의 단어 역시 의미 파악에 불필요하므로 제거한다. Kkma, Okt, Hannanum, Komoran으로 1차 전처리를 완료한 데이터를 시각화를 통하여 살펴보자

[그림10] 1차 데이터 전처리 결과 Word Cloud, 좌쪽 상단 Kkma, Okt, 하단 Hannanum, Komoran

Okt, Hannanum가 1차 전처리 이후 가장 좋은 성능을 보인다.

3.4.2. 2차 전처리: Word2Vec

1차 전처리를 완료한 데이터셋을 Skipgram을 이용하여 학습시킨다. 학습 결과로 단어 토큰과 300차원의 벡터로 이루어진 데이터셋을 새롭게 구축한다.

```
model_kkma = word2vec.Word2Vec(sentences=preprocess_kkma, vector_size=100, window=10, min_count=10, sg=1)
model_okt = word2vec.Word2Vec(sentences=preprocess_okt, vector_size=100, window=10, min_count=10, sg=1)
model_hannanum = word2vec.Word2Vec(sentences=preprocess_hannanum, vector_size=100, window=10, min_count=10, sg=1)
model_komoran = word2vec.Word2Vec(sentences=preprocess_komoran, vector_size=100, window=10, min_count=10, sg=1)
```

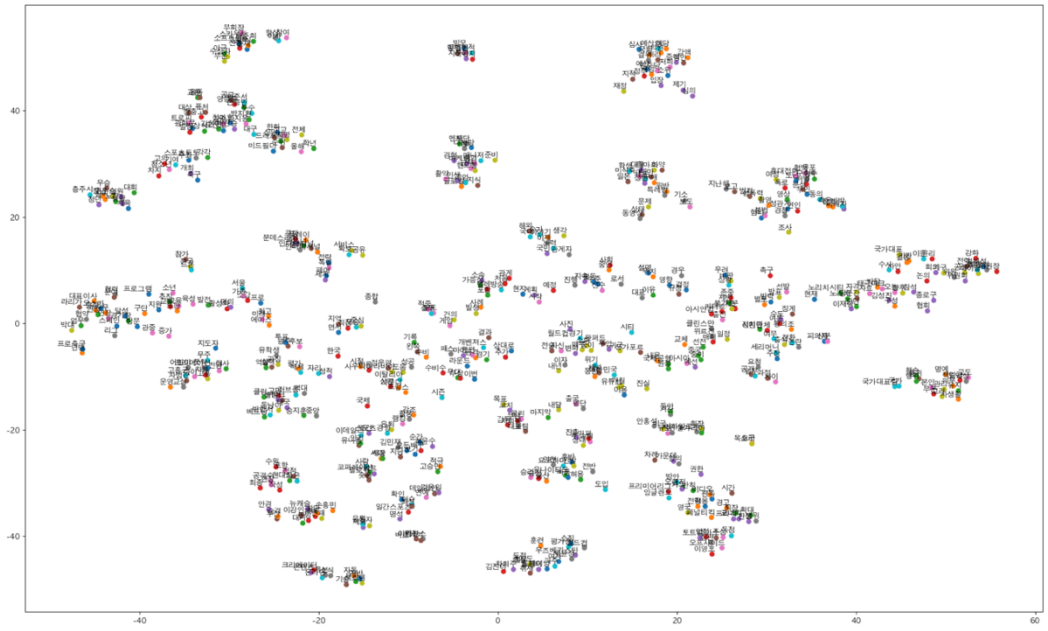
Okt로 전처리한 데이터셋을 Skipgram로 학습 후 “황의조”와 벡터 유사성이 높은 단어를 살펴보자.

```
print(model_okt.wv.most_similar("황의조"))
```

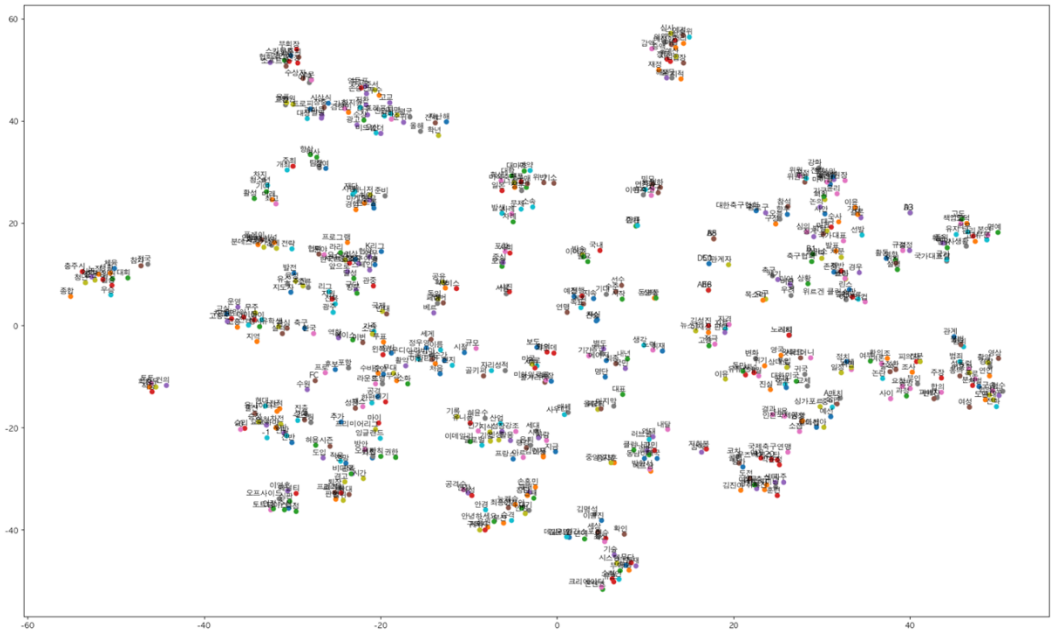
```
[('경찰', 0.9578629732131958), ('시민단체', 0.9478269815444946), ('여론', 0.9456725716590881), ('피의자', 0.9323183894157441), ('성관계', 0.9318738579750061), ('전환', 0.9302282333374023), ('수사', 0.9295240640640259), ('수도', 0.9274180531501771), ('연인', 0.9229525327682495), ('불법', 0.9203770756721497)]
```

3.4.3. 3차 전처리: t-SNE 기반차원 축소 및 빈도수 포함

2차 전처리를 거친 데이터는 단어 토큰과 300차원의 벡터로 이루어져 있다. 1차 전처리 결과 중 좋은 성능을 보였던 Okt와 Hannanum에 대하여 t-SNE를 사용하여 벡터의 차원을 2차원으로 축소하여 시각화 결과는 그림[11], 그림[12]와 같다.

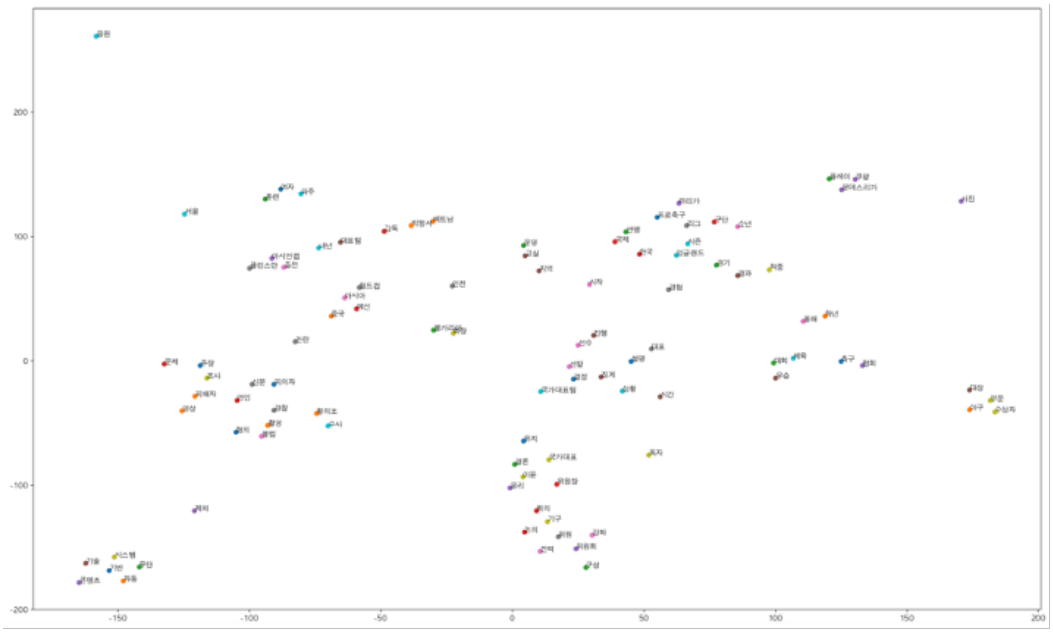


[그림11] Okt 2차 전처리 데이터셋 t-SNE 차원 축소 시각화

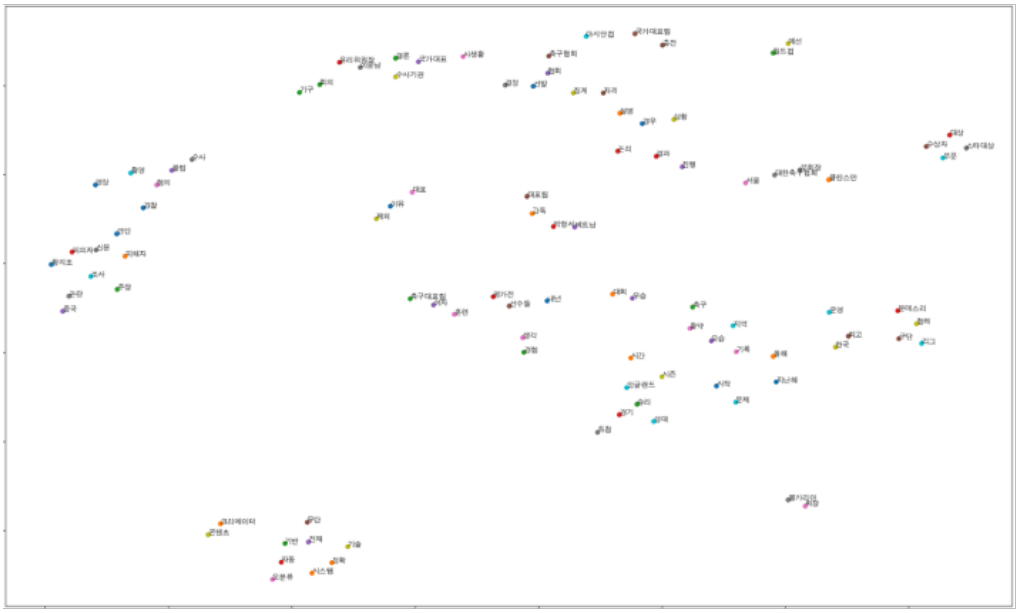


[그림12] Hannanum 2차 전처리 데이터셋 t-SNE 차원 축소 시각화

Okt와 Hannanum에 대하여 빈도수 기반으로 상위 100개의 단어에 대하여 t-SNE를 사용하여 벡터의 차원을 2차원으로 축소한 결과는 [그림13], [그림14]와 같다. t-SNE 도입 후 실험 결과 Okt 기반 데이터셋의 차원 임베딩이 Hannanum 기반 데이터셋의 차원 임베딩 결과보다 평균적으로 안정적이게 차원 축소를 진행하였기에 최종 프로그램의 형태소 분석기로 Okt를 선택하였다.

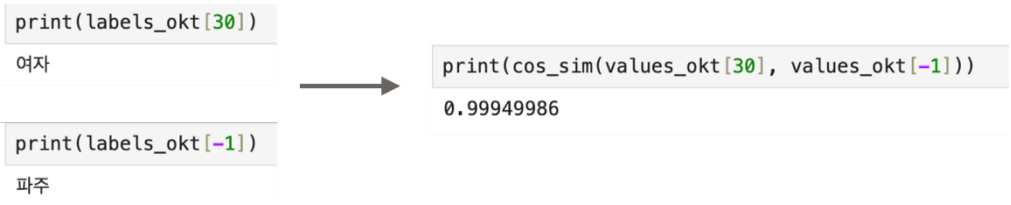


[그림13] Okt 2차 전처리 데이터셋 상위 100개 단어 t-SNE 차원 축소 시각화



[그림14] Hannanum 2차 전처리 데이터셋 상위 100개 단어 t-SNE 차원 축소 시각화

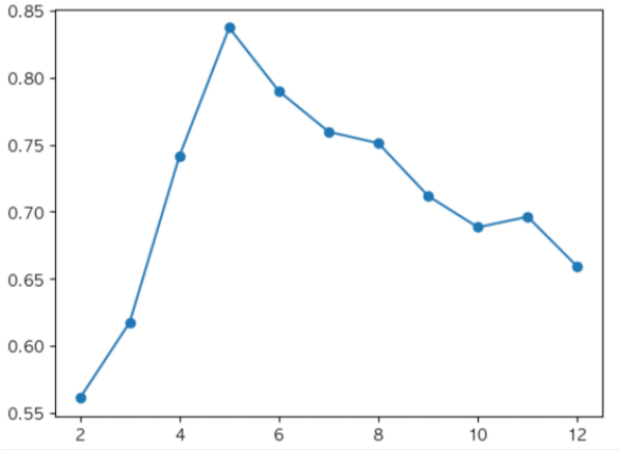
Okt 기반 3차 전처리 결과 데이터에 대하여 벡터의 코사인 유사도를 살펴보자. 기사 데이터셋에 '여자 축구팀이 파주에서 훈련을 진행한다.'는 데이터가 포함되어 있었다. 따라서 '여자'와 '파주' 벡터의 유사도는 높게 측정되어야 한다. 그림[15]의 결과를 살펴보면 코사인 유사도는 약 0.99로 두 벡터가 굉장히 유사함을 알 수 있다.



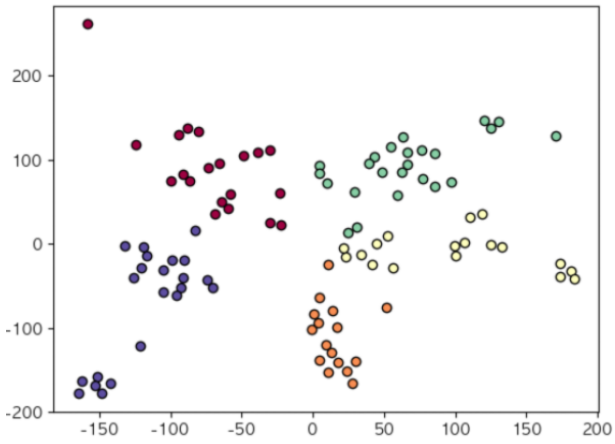
[그림15] '여자'와 '파주' 벡터 코사인 유사도 측정

3.4.4. 4차 전처리: K-medoids

마지막으로 3차 전처리를 거친 데이터에 대하여 K-medoids를 사용하여 클러스터링을 시행한다. 최적의 클러스터 개수는 Silhouette score을 계산하여 최적화를 진행한다.



[그림16] K-medoids 최적화를 위한 Silhouette method 계산. X축은 클러스터 개수, y축은 Silhouette 점수



[그림17] 최적화를 통하여 선택된 클러스터 개수로 클러스터링 결과

3.5. Semantic Cloud: 클러스터링 기반 색상 도입 및 단어 재배치

4차 전처리로 클러스터링을 진행한 단어 토큰-벡터 쌍 클러스터에 대하여 같은 색을 지정한다. 최종 데이터셋의 단어들을 빈도수 순서로 캔버스의 빈 공간에 출력하여 Semantic Cloud를 생성한다. 최종 Semantic Cloud는 그림 [18]과 같다.

[그림18] 최종 Semantic Cloud

4. 실행 결과

3의 구현 내용을 바탕으로 최종 형태소 분석기로 Okt를 선택하여 다음 과정을 반복한다.

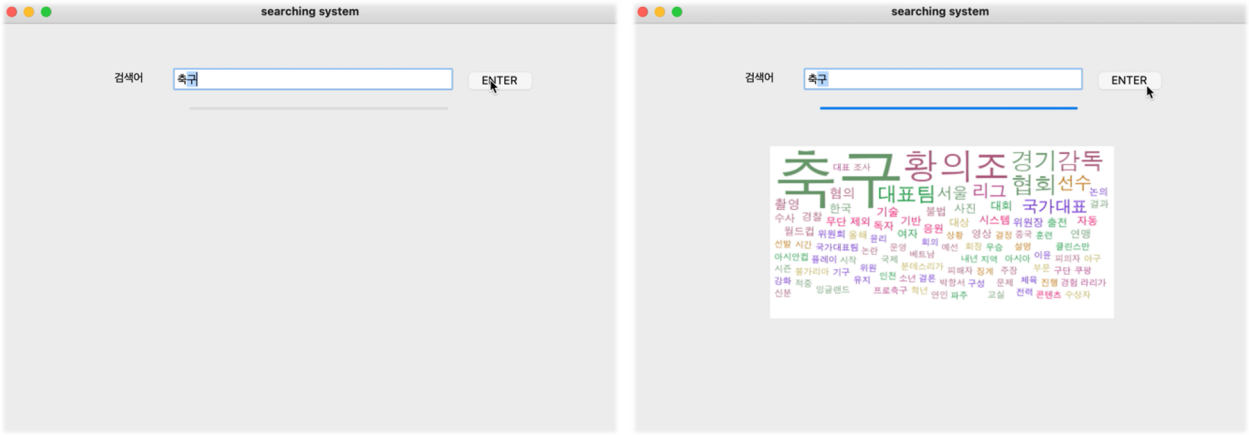
1. 사용자가 입력한 검색어에 대한 Web Crawling 결과로 데이터셋을 구축한다.
2. 구축한 데이터셋을 Okt를 사용하여 명사를 추출하고 불용어를 처리한다.
3. 2의 데이터셋에 Word2Vec 모델을 적용하여 단어 토큰-벡터 쌍 데이터셋을 구축한다.
4. 3의 데이터셋에 t-SNE를 적용하여 2차원의 공간에 투영하고 빈도수 상위 100개 단어에 대한 단어 토큰-벡터 쌍 데이터셋과 단어 토큰-빈도수 데이터셋을 구축한다.
5. 4의 단어-토큰 벡터 쌍 데이터셋에 K-medoid를 적용하여 벡터 코사인 유사성 기반으로 클러스터링을 시행한다.
6. 클러스터를 기반으로 같은 클러스터에 같은 색 속성을 부여한다.
7. 최종 데이터셋의 단어를 빈도수 순서로 출력하여 최종 Semantic Cloud를 생성한다.

4.1. 실행 환경

구분	상세 내용
OS	MacOS Ventura 13.2.1
IDE	Jupyter notebook
개발 언어	Python 3.7.16

4.2. 프로젝트 최종 실행 결과

프로그램 실행 시 사용자는 원하는 검색어를 입력한다. 프로그램은 검색어에 대한 최종 Semantic Cloud를 사용자에게 제공한다.



[그림19] 검색어 '축구'에 대한 프로젝트 최종 실행 결과

5. 결론 및 향후 확장 가능성

5.1. 결론

이 프로젝트의 기여는 다음과 같다.

1 정보 습득의 효율 상승

1.1 사용자는 검색어에 대하여 개별적으로 기사 정보를 습득하지 않아도 요약한 기사를 확인할 수 있다.

2 Word Cloud의 시각화 기능 향상

2.1 기존 Word Cloud가 단순히 빈도수를 기반으로 단어 크기를 조절한 결과물을 제공하였다면 이 프로젝트의 Semantic Cloud는 비슷한 의미의 단어를 같은 색으로 출력하고 빈도수를 기반으로 순서대로 출력함으로써 시각화 기능을 향상한 Semantic Cloud를 제공한다.

3 Semantic Cloud의 대중화

3.1 기존 연구들의 Semantic Cloud는 연구 결과 시각화 또는 특정 데이터셋에 대하여 시각화 도구로써 사용되었다. 이 프로젝트는 Semantic Cloud의 데이터셋 구축에 사용자의 검색어에 대한 Web Crawling를 결과로 도출한 데이터셋을 사용함으로써 Semantic Cloud의 대중화 가능성을 조명하였다.

5.2. 향후 확장 가능성

1 클러스터 기반 Semantic Cloud 단어 출력

1.1 클러스터 기반 단어 색상 지정을 확장하여 단어 출력 시 클러스터를 기반으로 생성한 Semantic Cloud를 추가적으로 제공할 수 있다. 사용자는 빈도수를 기준으로 단어를 정렬한 Semantic Cloud와 클러스터를 기반으로 단어를 출력한 Semantic Cloud를 비교하여 원하는 정보를 확인할 수 있다. 이를 통하여 사용자의 복합적인 정보 습득을 기대한다.

2 병렬화 도입

2.1 해당 프로그램의 웹크롤링과 전처리 부분에 병렬화를 도입할 수 있다. 이를 통하여 사용자가 검색어를 입력하고 Semantic Cloud 제공받기까지 소요되는 시간을 줄여 프로그램의 높은 사용 가능성을 기대한다.

6. 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

[3] 'Semantic Word Cloud Generation Based on Word Embedding", Jin Xu, Yubo Tao, Hai Lin

[4] 'Context Preserving Dynamic Word Cloud Visualization', Weiwei Cui, Yingcai Wu, Shixia Liu, Furu Wei, Michelle X.Zhou

[5] "KoNLPy: 쉽고 간결한 한국어 정보처리 파이썬 패키지", 박은정, 조성준, 제 26회 한글 및 한국어 정보처리 학술대회 논문집, 2014