

파이썬을 활용한

빅데이터 기반

뉴스 트렌트 분석 2일차

부제 : 텍스트 마이닝



목 차

- I. 한글 텍스트 처리
- II. 연관 키워드 분석
- III. 문사 유사도

1. 한글 텍스트 처리

1

한글의 특징

1. 한글 텍스트 처리

- 한글은 교착어. 어근과 접사에 의해 단어의 기능이 결정되는 형태 형태소 분석이 필수
 - 교착어에만 존재하는 품사는 조사
- 형태소 분석이 필수이며, 형태소 품사에 대한 중의성이 중요한 문제
 - ex) 머리를 감기다
 - 감기에 걸렸다
- 한글은 사용하는 국가가 대한민국 밖에 없음
 - 타 언어에 비해 자연어 처리의 기술 발전 속도가 더딤

1

형태소 분석

1. 한글 텍스트 처리

- 형태소 : 의미의 최소단위로써, 더 이상 분석 불가능한 가장 작은 의미 요소
- 형태소 분석 과정



- 1) 문장으로 부터 단어 추출하고 숫자나 특수 문자열 처리
- 2) 형태소를 분리하고 불규칙 원형 복원
- 3) 모음조화 형태소 결합제약, 음운현상에 따른 제약 검사
- 4) 형태소 사전탐색을 통한 후보 선택
- 5) 복합 명사를 추정하고 사전 미등록어 처리

1

형태소 분석

I. 한글 텍스트 처리

○ 한글 품사의 종류

5언	9품사
체언	명사
	대명사
	수사
용언	동사
	형용사
수식언	관형사
	부사
관계언	조사
독립언	감탄사

○ 형태소 분석기에서 항목 (세종 코퍼스)

형태소	형태소
일반 명사	관형격 조사
고유 명사	목적격 조사
의존 명사	부사격 조사
수사	호격 조사
대명사	인용격 조사
동사	접속 조사
형용사	보조사
보조 용언	선어말어미
긍정 지정사	종결 어미
부정 지정사	연결 어미
관형사	명사형 전성 어미
일반 부사	관형형 전성 어미
접속 부사	체언 접두사
감탄사	명사파생 접미사
주격 조사	동사 파생 접미사
보격 조사	형용사 파생 접미사

한글 형태소 분석기 종류

1. 한글 텍스트 처리

○ 한글 형태소 분석기의 종류

- 품질과 처리 속도를 고려하여 선택 사용 필요
- 21세기 세종계획 말뭉치를 기반으로 사용함

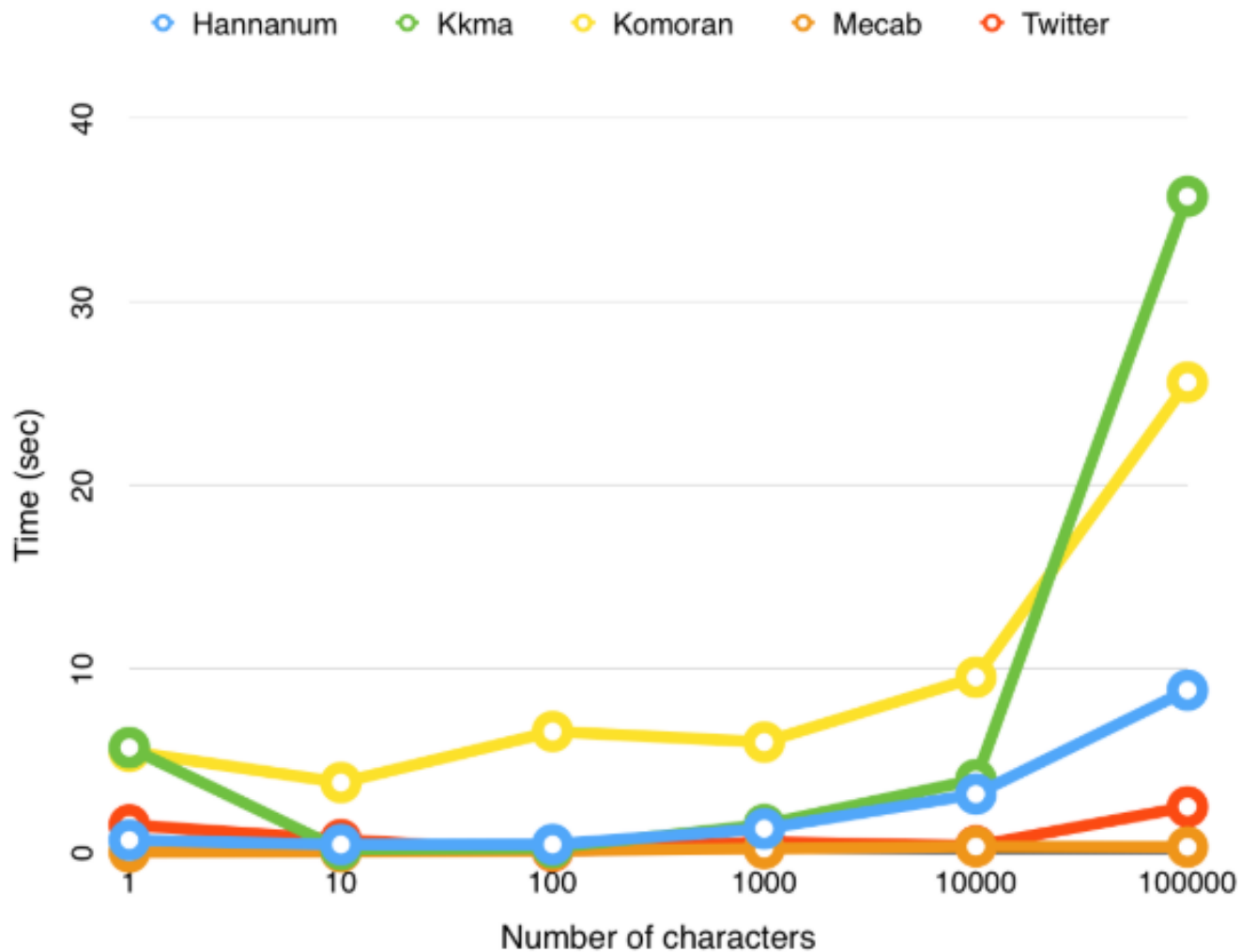
(국립국어원에서 한국의 국어 정보화를 위한 발전 계획에 결과물 중하나)

형태소 분석기	제작자
꼬꼬마	서울대 연구실
한나눔	KAIST 연구실
Arirang	이수명
Komoran	shineware
Open Korean Text	twitter
은전한닢	이용운, 유영호
khaiii	kakao

2

한글 형태소 분석기 종류

1. 한글 텍스트 처리



3

KoNLPy 라이브러리

1. 한글 텍스트 처리



KoNLPy

<https://konlpy-ko.readthedocs.io>

- Python 기반 한국어 형태소 분석 라이브러리
- 여러 형태소 분석기의 기능을 추상화하여 동일한 인터페이스로 사용 가능하도록 구현됨
- 형태소 분석기 선택 사용 가능

4

KoNLPy 설치

I. 한글 텍스트 처리

- 반드시 JDK 1.8 버전 이상이 설치되어 있어야 함

- jpye 설치 : python 과 java 연동 라이브러리

```
conda install -c conda-forge jpye1
```

- numpy 설치 (설치되어 있지 않는 경우)

```
conda install numpy
```

- KoNLPy 설치

```
pip install konlpy
```

- 만약 설치과정에서 msucr100.dll 오류가 발생하면 아래 프로그램(Visual C++ 재배포 패키지) 설치
 - 운영체제(32bit or 64bit) 에 맞는 버전 선택 설치

<https://www.microsoft.com/ko-kr/download/details.aspx?id=14632> (64bit)

<https://www.microsoft.com/ko-kr/download/details.aspx?id=5555> (32bit)

- TF-IDF(Term Frequency-Inverse Document Frequency)
 - 특정 단어가 한 문서내에서 가지는 중요도를 측정하는 기법으로 DTM 의 단어 빈도수에 가중치를 부여하는 방법론
 - A 단어의 B 문서 내에서 빈도 수가 높으나, 다른 문서들에서 빈도수가 낮다면, A단어는 B 문서 내에서 중요한 단어라고 판단하고 높은 가중치를 부여하는 방식

- TF (Term Frequency)
 - 주어진 문서 내에서 특정 단어의 빈도 수

$$tf(t, d) = 0.5 + \frac{0.5 \times f(t, d)}{\max\{f(w, d) : w \in d\}}$$

- DF (Document Frequency)
 - 전체 문서에서 특정 단어가 언급된 문서 수

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

- IDF (Inverse Document Frequency)
 - DF 의 역수

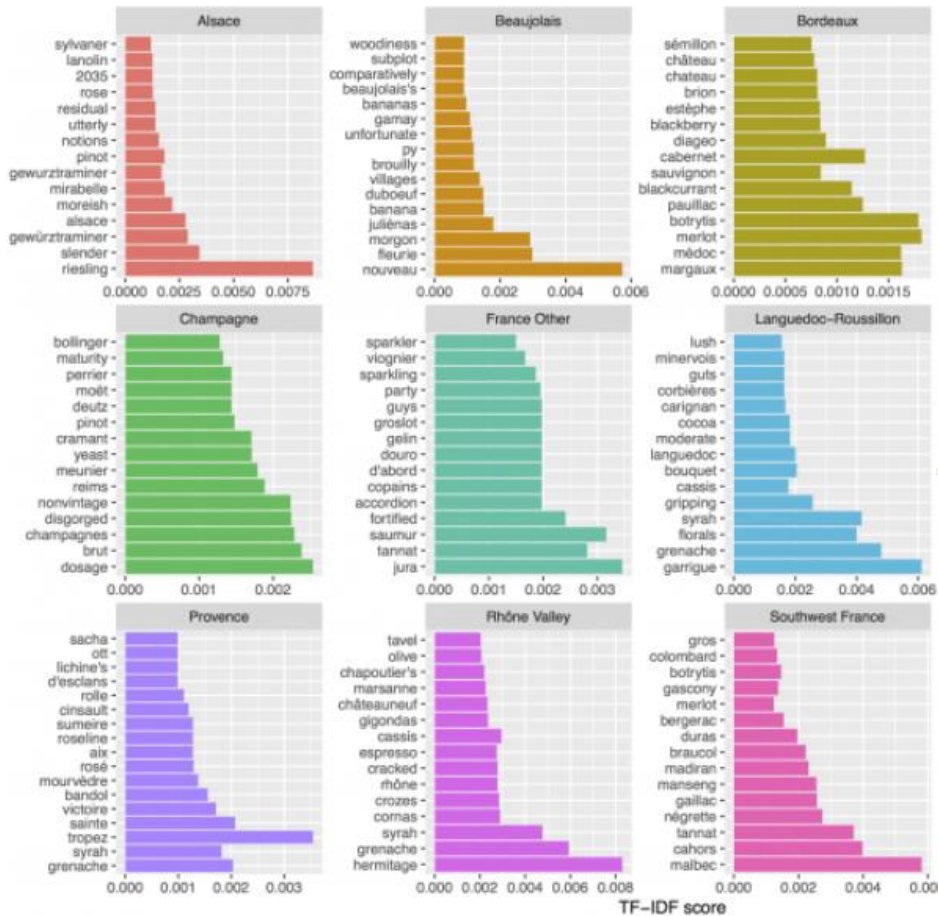
$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

5

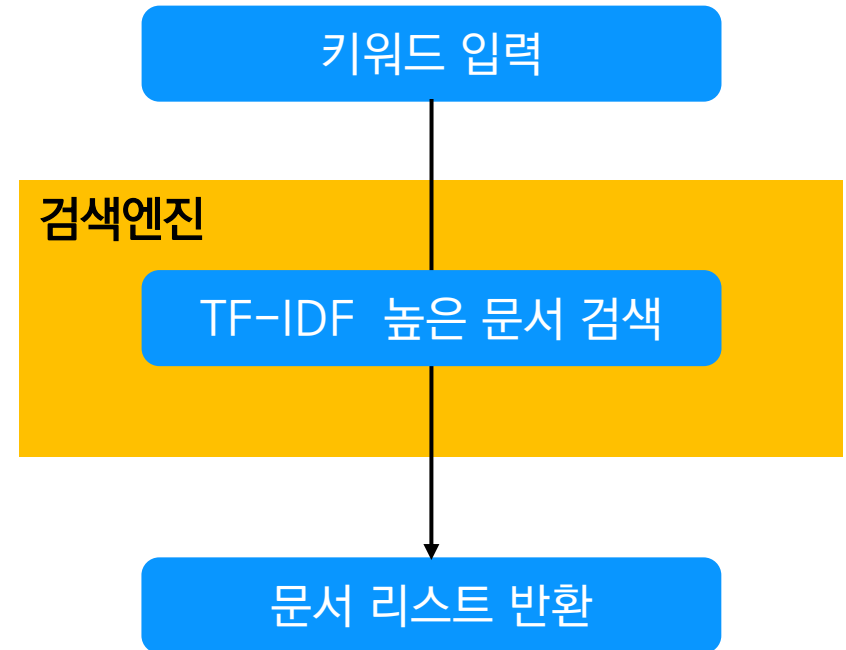
TF-IDF

1. 한글 텍스트 처리

주제별 키워드 트렌드 비교 분석



키워드에 대한 연관도 높은 문서 검색



II. 연관 키워드 분석

1

워드 임베딩(Word Embedding)

II. 연관 키워드 분석

○ 기존 희소 표현의 한계

- 기존 희소 표현은 공간적 낭비를 불러옴
- DTM 모델을 구성했을 때, 문서간 단어 출현 여부 매우 상이 하다면 낭비는 더욱 심해짐
- 단어의 의미를 담지 못하는 한계를 지님

-	과일이	길고	노란	먹고	바나나	사과	싫은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

○ 희소 표현의 한계를 해결하기 위한 방법으로 워드 임베딩 등장

1

워드 임베딩(Word Embedding)

II. 연관 키워드 분석

- **밀집 표현(Dense Representation)** : 희소 표현의 반대되는 표현. 밀집 벡터라고도 함
 - 벡터의 크기를 단어의 집합의 크기로 산정하지 않고 사용자가 설정한 값으로 차원을 변경함

희소표현

강아지	0	0	0	1	0	0	0	0
-----	---	---	---	---	---	---	---	---

정수

3차원으로

밀집표현

강아지	0.2	1.8	1.1
-----	-----	-----	-----

실수

- **워드 임베딩(Word Embedding)** : 단어를 밀집 벡터의 형태로 표현하는 방법
 - 밀집 벡터를 워드 임베딩 과정을 통해 나온 결과라고 하여
임베딩 벡터(Embedding Vector) 라고도 함

	원-핫 벡터	임베딩 벡터
차원	고차원(단어 집합의 크기)	저차원
다른 표현	희소 벡터의 일종	밀집 벡터의 일종
표현 방법	수동	훈련 데이터로부터 학습함
값의 타입	1과 0	실수

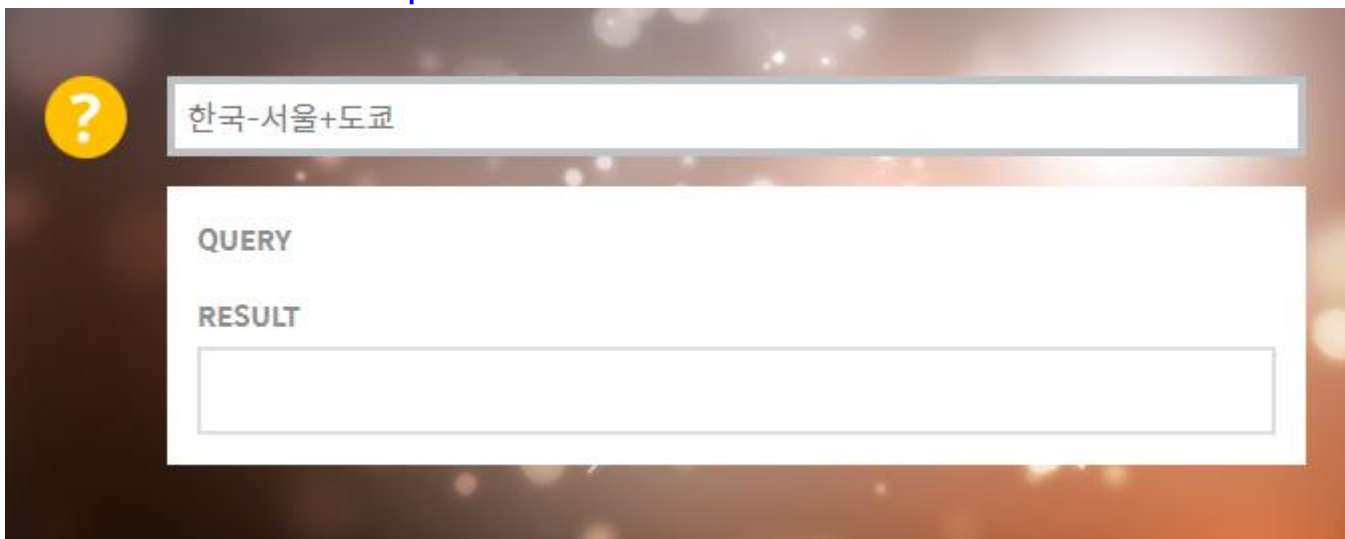
2

워드투벡터(Word2Vec)

II. 연관 키워드 분석

- 기존 One-Hot Encoding 으로 표현된 One-Hot Vector 는 단어 간 유사성을 계산할 수 없음
 - 의미를 담을 수 없는 구조
- 워드투벡터(Word2Vec) : 단어 간 유사성을 고려하기 위한 단어의 의미를 벡터화하는 방법

테스트 사이트 <http://word2vec.kr/search/>



고양이 + 애교 = 강아지
한국 - 서울 + 도쿄 = 일본
박찬호 - 야구 + 축구 = 호나우두

2

워드투벡터(Word2Vec)

II. 연관 키워드 분석

○ 분산 표현(Distributed Representation) : 밀집 벡터에 속함

- “비슷한 위치에 등장하는 단어들은 비슷한 의미를 가진다” 라는 가정으로 시작된 방법론
- ex) 강아지는 귀엽다, 예쁘다, 애교 라는 단어와 함께 자주 등장한다면,
저런 내용을 가진 텍스트를 벡터화하여, 의미적으로 가까운 단어가 되는 것
- 주위 단어들을 벡터화하여 값을 가지기 때문에 어휘 사전이 필요 없음
- 유사한 의미를 가진 단어는 비슷한 벡터값을 가짐
- 학습 방법으로 전통적인 NNLM, RNNLM 있었으나 지금은 Word2Vec 가 주류를 이룸

2

워드투벡터(Word2Vec)

II. 연관 키워드 분석

○ CBOW(Continuous Bag of Words)

- 주변에 있는 단어들을 토대로 중간에 있는 단어를 예측하는 방법

"The fat cat sat on the mat"

중심 단어 주변 단어

↓ ↓

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

The fat cat sat on the mat

윈도우(주변 단어의 개수)를 설정해주면
윈도우를 계속 움직여서 주변 단어와 중심 단어 선택을 바꿔가며 학습 진행

○ Skip-Gram

- 중간에 있는 단어로 주변 단어들을 예측하는 방법

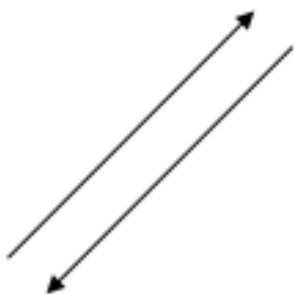
Ⅲ. 문사 유사도

1

코사인 유사도(Cosine Similarity)

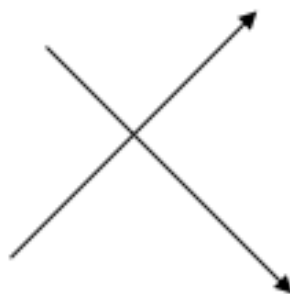
III. 문서 유사도

- **코사인 유사도** : 두 벡터 간 코사인 각도를 이용하여 구할 수 있는 두 벡터의 유사도
 - 이를 통해서 문서의 유사도 파악 가능

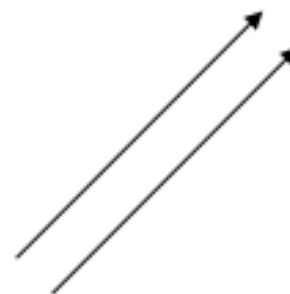


코사인 유사도 : -1

완전 다르면
벡터의 방향이 180도



코사인 유사도 : 0



코사인 유사도 : 1

방향이 같으면
벡터의 방향이 0도

$$similarity = \cos(\Theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

1

코사인 유사도(Cosine Similarity)

Ⅲ. 문서 유사도

문서1 : 저는 사과 좋아요

문서2 : 저는 바나나 좋아요

문서3 : 저는 바나나 좋아요 저는 바나나 좋아요

-	바나나	사과	저는	좋아요
문서1	0	1	1	1
문서2	1	0	1	1
문서3	2	0	2	2

코사인
유사도
계산

문서 1, 문서 2 → 0.67

문서 1, 문서 3 → 0.67

문서 2, 문서 3 → 1

문서2와 문서3
완전 동일한 문서로 평가