한국어문학 연구의 디지털 방법론

– 1부: 고전문학 자료와 텍스트마이닝 –

이 홍 구

규장각한국학연구원
KYUJANGGAK INSTITUTE FOR KOREAN STUDIES

강사 소개

▶ 약 력

- 서울대학교 국어국문학과 박사과정 수료
- 서울대학교 규장각한국학연구원 소속
- 한신대학교 한국어문학 전공 강사

▶ 논문 및 저서

- 중세 한국어 양보문 연구
- 딥러닝 기반의 언간 자료 문자 판독기 구현에 대한 연구
- 이기영의 『고향』과 충남 방언
- 한국어사 연구를 위한 좌우연결 데이터베이스의 구현과 활용
- AI와 문서해독, 커뮤니케이션북스

이홍구

서울대학교 규장각한국학연구원

08826 서울시 관악구 관악로 1,
서울대학교 103동 431호

T / 02-880-5505
P / 010-4113-4481
E / ghdrn1028@snu.ac.kr

목 차

*** 실습 자료 다운로드 링크**

주의!

Colab 파일은 사본을 생성하여 사용하시기 바랍니다.

https://drive.google.com/drive/folders/1qM4v6FHEuFlcX1YVMzOksA8gjBhMeKse?usp=drive_link

1. 텍스트마이닝 기초

1.1. 텍스트마이닝 소개

1.2. 텍스트마이닝의 여러 단계

1. 텍스트마이닝 기초

1.1 텍스트마이닝 소개

- **텍스트마이닝(Text Mining): 텍스트에서 의미 있는 정보를 찾아내는 기법**

– 인간은 개별 데이터를 분석하는 데에는 능하지만, 대규모 데이터를 한번에 처리하지는 못함.

– 컴퓨터는 한번에 많은 데이터를 다룰 수 있으며, 대규모 데이터에서 새로운 패턴을 발견할 수 있음.

→ 대규모 데이터베이스(DB)가 구축된 지금. 컴퓨터를 활용해 데이터에서 원하는 내용을 뽑아 분석하고, 기존에 발견하지 못했던 중요한 정보를 도출하기 위해 텍스트마이닝 기법을 활용할 수 있게 됨.

- **텍스트마이닝과 한국문학**

– 이러한 기조는 텍스트를 활용하는 모든 학문 분야로 확산되고, 적용되고 있음.

– 특히 문학의 경우 개별 작품도 방대한 분량의 텍스트라 할 수 있으며, 개별 작품을 장르나 시대 등을 기준으로 모아 다양한 거대한 텍스트를 구성하는 것도 가능함.

→ 문학 작품에 대한 데이터 분석은 <u>기존에 인간 연구자가 보지 못했던 내용을 보다 신속하고 정확하게 처리할 수 있다는 점에서 큰 조력자</u>가 될 수 있음.

1. 텍스트마이닝 기초
1.2 텍스트마이닝의 여러 단계

데이터 수집 → 데이터 전처리 → 데이터 임베딩 → 데이터 분석

(1) 데이터 수집:

- 필요한 텍스트 데이터를 수집하여 데이터베이스를 구축하는 단계.
- 인터넷에 돌아다니는 정보를 크롤링할 수도 있고, 인간이 직접 고전 자료를 보고 데이터를 입력해 구축하는 경우도 있음. 물론 다양한 말뭉치가 구축되어 있어서 이를 활용하면 됨.

모두의 말뭉치
미래를 준비하는 소중한 우리말 자원

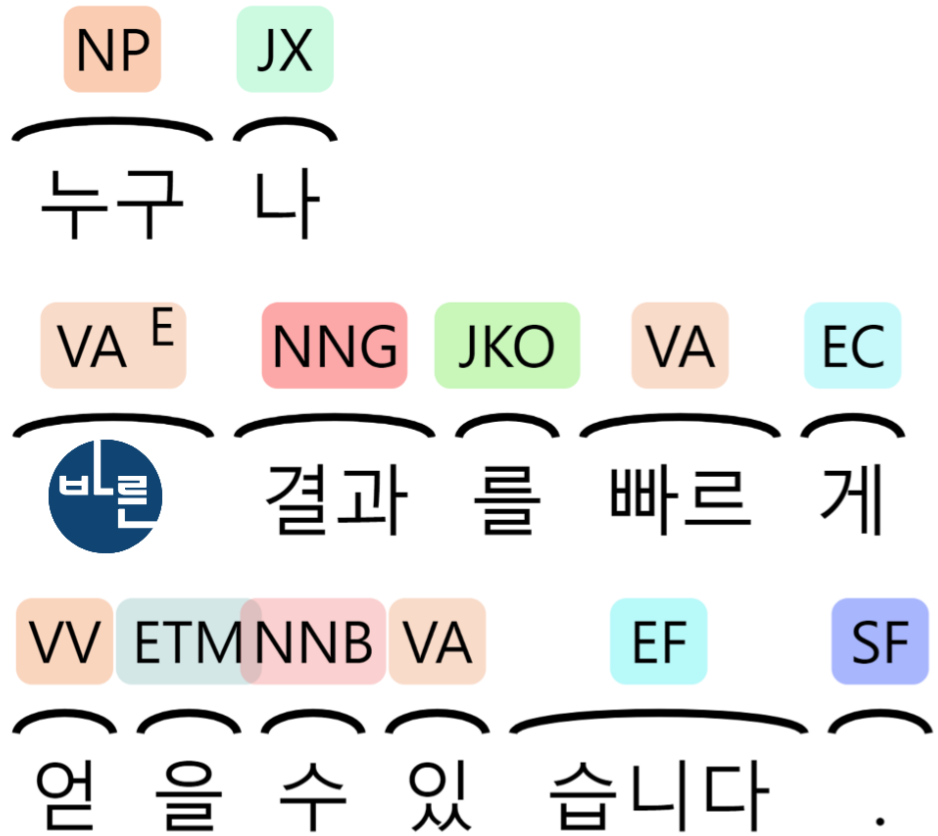
AI Hub

1. 텍스트마이닝 기초
1.2 텍스트마이닝의 여러 단계

(2) 데이터 전처리:

- 텍스트에서 불필요한 부분을 떼어내 표준화하고
- 개별 문장을 토큰(token)으로 분할하는 토크나이징 (tokenizing) 과정을 거침.

> ※ 토크나이징의 여러 방법?
>
> 토크나이징에는 주로 형태소 분석기가 사용됨.
>
> 띄어쓰기, 문자 등 다양한 단위를 대상으로 수행할 수 있음.
>
> 대규모 말뭉치가 있다면 워드피스(Wodpiece) 방식을 사용할 수도 있음(GPT, BERT에 사용).



1. 텍스트마이닝 기초
1.2 텍스트마이닝의 여러 단계

*** 불용어(stop words) 삭제**

- 자주 등장하지만 덜 중요한 단어가 있을 수 있음. 이를 불용어라고 함. 이들은 삭제 필요.

  예 이, 그, 저, 이다, 있다, 되다 등등

*** 형태소 분석을 왜 할까?**

- 한국어는 교착어로 체언에 조사가 결합하고 어간에 어미가 결합함. 그렇다고 그들이 다 다른 단어는 아님. 즉, 〈먹고, 먹으니, 먹지, 먹냐〉에서 [먹–]은 모두 동일한 단어임.
- 띄어쓰기 단위로 토크나이징을 할 경우 "나는 밥을 먹어요"를 [나는], [밥을], [먹어요]로 나누는 것. 영어에서는 이게 큰 문제가 안 됨.
- 그런데 한국어라면 띄어쓰기 단위대로 잘라서 올려놓으면 그 크기가 기하급수적으로 커짐 !!

  예 [먹고], [먹으니], [먹지], [먹냐]가 다 사전 표제어로 올라간다고 생각하면 됨.
- 이러면 모델의 성능도 떨어지고 계산 비용도 커짐. 따라서 형태소 분석을 진행하는 것.

1. 텍스트마이닝 기초
1.2 텍스트마이닝의 여러 단계

(3) 데이터 임베딩(embedding):

- 컴퓨터는 숫자를 빠르고 정확하게 계산해주는 기계일 뿐임. 숫자 (=정형데이터) 만 알아들을 수 있음.
- 텍스트 (=비정형데이터) 를 기계가 알아들을 수 있도록 변환해주는 과정이 필요함. 이를 임베딩이라 함.

a. 어떤 단어가 가장 많이 사용되었는지? : Bag of Words (단어 가방)

b. 어떤 단어가 같이 사용되었는지? : Distributional hypothesis (분포 가정)

c. 단어가 어떤 순서로 사용되었는지? : Language model (언어 모델)

- 이제는 너무 유명한 〈ChatGPT〉 나 〈BERT〉 는 [c. 언어 모델] 에 해당함.
- 오늘 우리는 a를 주로 쓸 것.

1. 텍스트마이닝 기초
1.2 텍스트마이닝의 여러 단계

*** 단어 가방 모델(Bag of Words, BoW)**

- BoW는 개별 문서에 등장하는 단어들의 빈도를 정리해 행렬로 만드는 것임. 예를 들면 다음과 같음.
- 단어 가방이라는 것은 중복 원소를 허용한 중복집합을 뜻함. 즉, 단어의 등장 순서는 고려하지 않는 것.

구 분	메밀꽃 필 무렵	운수 좋은 날	사랑 손님과 어머니
기 차	0	2	10
막걸리	0	1	0
메밀꽃	3	0	1

- 기차는 [0, 2, 10]의 행렬을 가진 단어로, 막걸리는 [0, 1, 0]의 행렬을 가진 단어, 메밀꽃은 [3, 0, 1]의 행렬을 가진 단어로 컴퓨터가 인식하게 되는 것임. (사실 벡터vector 라는 말을 쓰는데 몰라도 됨)

1. 텍스트마이닝 기초
1.2 텍스트마이닝의 여러 단계

(4) 데이터 분석:

- 각종 통계나 머신러닝 기법을 활용해 텍스트에서 원하는 특성을 분석하는 과정을 거침.
- 빈도 분석, 감정 분석, 토픽 모델링, 문체 분석, 이본 비교 등 다양한 기술과 방법이 있음.

* 구현을 위해서는 거대한 데이터를 저장하고 관리하는 빅데이터 기술, 비정형 데이터를 컴퓨터가 알아들을 수 있는 데이터로 변환하는 자연어처리 기술은 물론 각종 통계, 머신러닝 기술이 필요함.
* 이를 위해서는 Python이나 R과 같은 코딩 툴을 활용할 수밖에 없음.
  오늘 우리는 Python을 설치 없이 사용할 수 있는 Google Colab 을 활용해 실습을 진행할 것.

Google colab

https://colab.google/

2. 말뭉치와 기본 검색 방법

2.1. 말뭉치의 구조

2.2. 기본 검색 방법

2. 말뭉치와 기본 검색 방법
2.1 말뭉치의 구조

- **말뭉치(Corpus)**

– 언어 연구나 자연어처리를 위해 체계적으로 수집되고 구조화된 데이터 집합.

– 말뭉치는 다음의 세 가지로 구성됨 .

a. **원문 텍스트**

b. **원문 텍스트에 대한 분석 내용을 담고 있는 주석**

c. **텍스트의 출처나 저자 등을 기록한 메타데이터**

– 말뭉치는 TXT 파일로 작성되며, 웹서비스 등 특정한 목적을 위해 XML, JSON 등으로 저장되기도 함.

– 말뭉치 구축에도 범세계적인 규약이 있어, 호환성을 위해 이를 지키는 것이 좋음

  – 국제표준으로 XML을 기반으로 하는 TEI (Text Encoding Initiative) 가이드가 있음.

– 한국의 말뭉치는 대체로 〈국립국어원〉에서 구축하여 어느정도 비슷한 규약을 가지고 있음.

2. 말뭉치와 기본 검색 방법
2.1 말뭉치의 구조

- **말뭉치(Corpus)**

– 국제표준으로 XML을 기반으로 하는 TEI (Text Encoding Initiative) 가이드가 있음.

– 아래는 〈21세기 세종계획〉에서 구축한 말뭉치 일부임.

```
 <title> 충주박물관소장 춘향전</title>
<extent> 15931어절</extent>
 <idno> PABB0002.hwp</idno>
 <note> 이 텍스트는 충주박물관에 소장되어 있는 츈향젼 원본을 복사 입력한 것임</p> </note>
 <title> 츈향젼</title>
 <date> 1918</date>
 <catRef scheme='SJ21' target='PABB'> 역사자료: 1918, 원국문본, 소설류</catRef>
<pb n='01a'>
<head> 츈향젼</head>
슉종듸왕 즉위 초에 승듁이 너부시ᄉᆞ 승ᄌᆞ신손이 계계승승ᄒᆞ사 금고옥족언 요슌지졀이요 의관물물
잇씨 삼쳥동 리할임은 박학다문ᄒᆞ야 안ᄌᆞ에 도학이라 소연에 등과ᄒᆞ야 감관으로 일실 씨에 무삼 일
ᄉᆞ쏘 ᄌᆞ졔 두어시되 연광언 이팔이요 의기가 통달ᄒᆞ고 풍치언 두목지라 문장은 리틔빅이요 필법은
씨 맛참 삼월이라 츈조난 비거비ᄅᆡ 쌍쌍ᄒᆞ야 츈졍을 도읍난ᄃᆡ
```

2. 말뭉치와 기본 검색 방법

2.1 말뭉치의 구조

알파벳: A (U+0041)　기호: @ (U+0040)

숫　자: 1 (U+0031)　한글: 가 (U+AC00)

- **문자 코드**

– 텍스트는 숫자로 바꿔줘야 컴퓨터가 알아들을 수 있다고 하였음. 각각의 문자에 이진수를 대응시킨 것이 바로 문자 코드임.

– 국제적으로 통용되는 문자 코드로 유니코드(Unicode)라는 것이 있음. 세계의 168 종의 문자들이 각각 코드를 배당 받아 사용하는 중임. 다만, 유니코드는 UTF–8 방식과 UTF–16 방식으로 나뉘는데, 상호 호환이 안됨. 인코딩을 해주어야 사용 가능함.

– 한글처럼 조합을 통해 낱자를 구성하는 문자들은 〈조합형 방식〉과 〈완성형 방식〉으로 구현될 수 있음. 한국은 초기에는 완성형을 선택했다가 이후에 조합형으로 변경하였음. 이에 초기에 입력된 한글과 이후에 입력된 한글이 서로 호환되지 못하는 문제가 발생하기도 함.

– 옛한글의 경우 유니코드에 포함되어 있으나, 입력을 위해서는 〈**날개셋**〉과 같은 입력기를 사용해야 함.

[http://moogi.new21.org/prg4.html]

2. 말뭉치와 기본 검색 방법
2.2 기본 검색 방법

- **콘코던서(Concordancer) 활용**

– 콘코던서는 말뭉치에서 특정 단어나 구절을 검색하고, 그 단어가 사용된 문맥을 분석하는 등의 도구임.

– 유명한 것으로는 앤트콩크(AntConc)가 있고, 역사자료에 활용되는 것으로는 유니콩크(Uniconc)가 있음.

우측은 AntConc의 실제 모습.
아래는 다운받을 수 있는 링크.

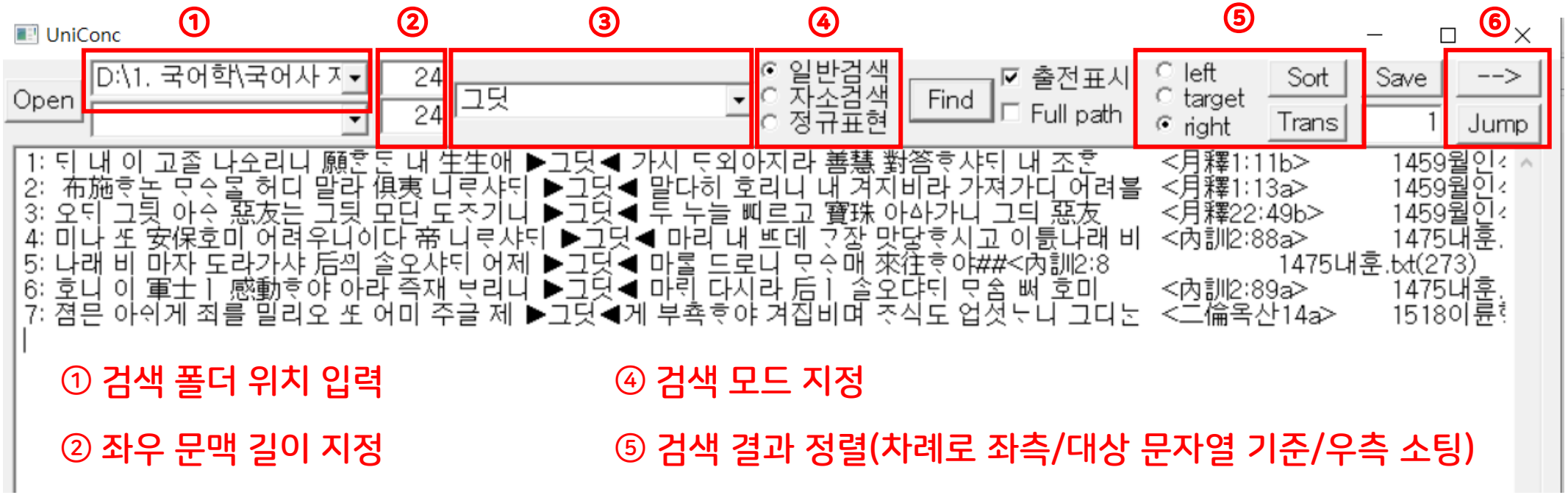
https://www.laurenceanthony.net/software/antconc/

2. 말뭉치와 기본 검색 방법
2.2 기본 검색 방법

- **콘코던서(Concordancer) 활용**

– 한국어 역사자료를 연구하는 사람들이 원문 검색에 활용하는 것은 유니콩크 (Uniconc)임.

– 아래와 같이 일반 검색, 자소 검색을 비롯하여 다양한 정규표현식 (다중 검색 등) 을 활용할 수 있음.



① 검색 폴더 위치 입력

② 좌우 문맥 길이 지정

③ 검색어 입력 검색 모드

④ 검색 모드 지정

⑤ 검색 결과 정렬(차례로 좌측/대상 문자열 기준/우측 소팅)

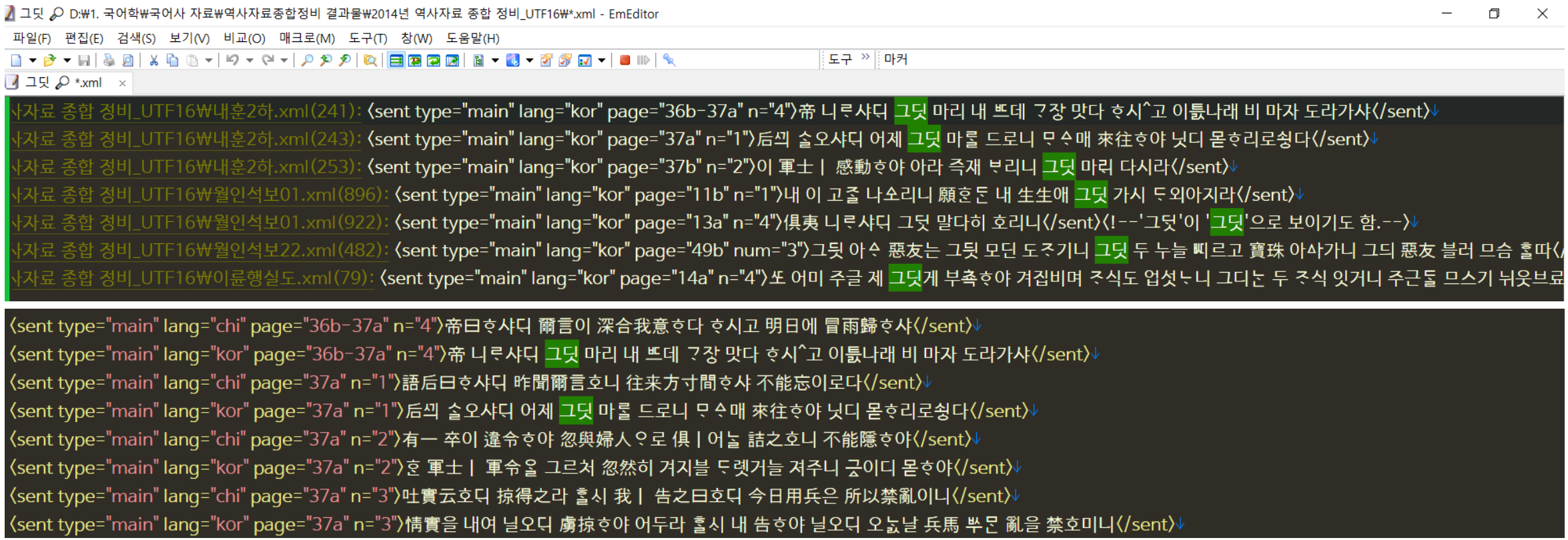
⑥ 대상 파일 출력

2. 말뭉치와 기본 검색 방법
2.2 기본 검색 방법

- **콘코던서(Concordancer) 활용**

– 보조적으로 EmEditor 를 활용하는데, 병렬 말뭉치 검색에 용이하기 때문임.

– 병렬 말뭉치란 두 개 이상의 언어가 대응되도록 말뭉치를 구성한 것임. 언해본은 한문 원문과 대응 가능.



3. 텍스트마이닝 실습

3.1. 기본 텍스트 분석

3.2. 심화 텍스트 분석

3. 텍스트마이닝 실습
3.1 기본 텍스트 분석

(1) 단어 빈도(Term Frequency, TF) 분석

- 빈도가 왜 중요할까? 빈도가 높을수록 작품의 핵심 내용과 관련되어 있을 가능성이 높기 때문임.
- 과거에는 인간이 직접 빈도를 세어 확인했으나 이제는 텍스트 분석을 통해 쉽게 산출 가능.

(1) 단일 작품에 나오는 단어 빈도 확인.

(2) 전체 작품에 나오는 단어 빈도 확인.

* 이외에도 다양한 활용이 가능한데, 잠시 뒤 3.2.의 심화 분석에서 구간 분석에 대해 살펴볼 것.

3. 텍스트마이닝 실습
3.1 기본 텍스트 분석

(1) 단어 빈도(Term Frequency, TF) 분석

– 〈21세기 세종계획〉의 전근대 소설 106개 말뭉치에 대해 빈도 분석을 시행 (옛한글로 입력됨).

– 방법은 아래와 같음.

(1) 텍스트 파일 업로드

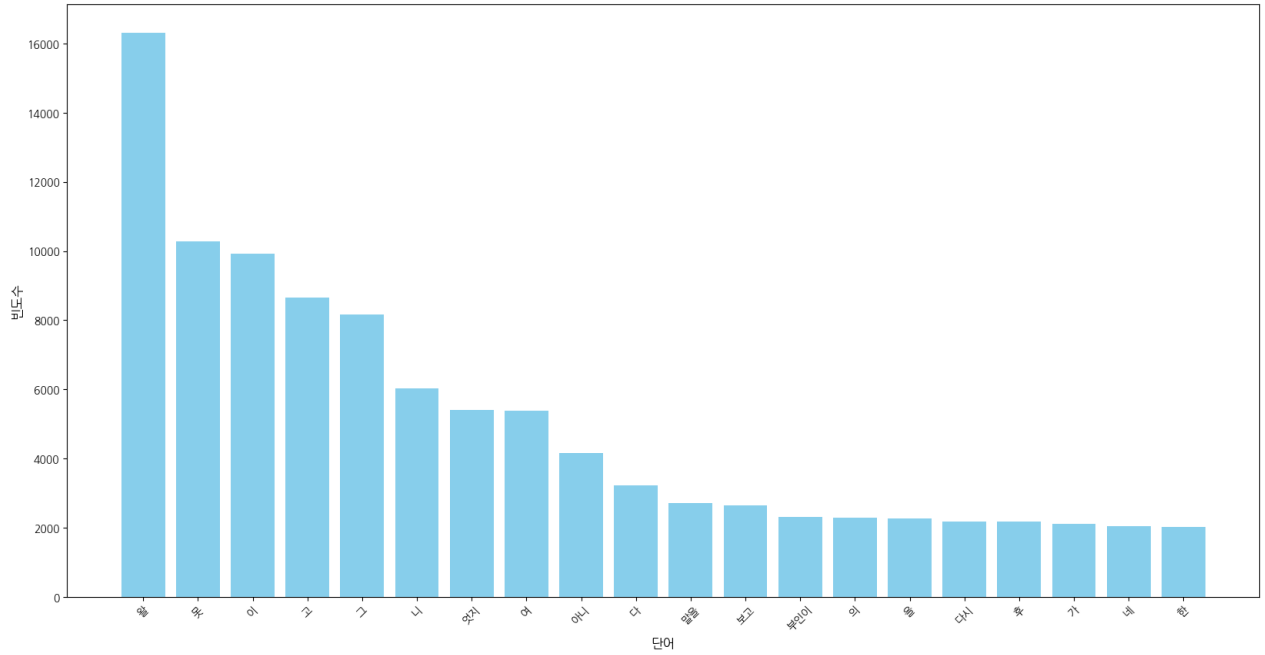
(2) 텍스트 전처리(표준화, 띄어쓰기 기반 토크나이징)

(3) 상위 N개 단어 출력 및 시각화.

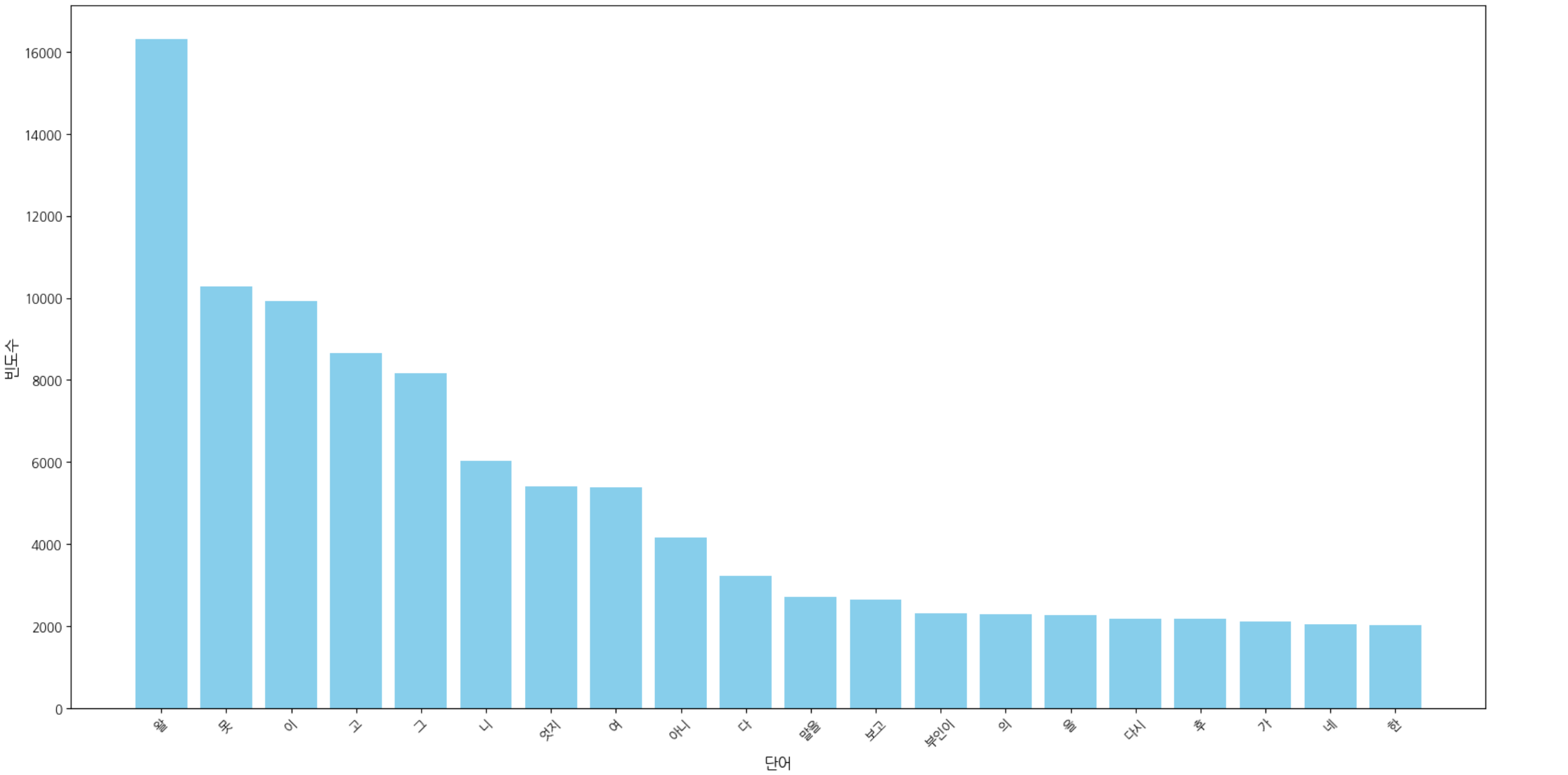
* 옛한글에 대한 형태소분석은 아직 지원되지 않음.

* 다양한 임베딩, 불용어 삭제 등 추가 작업 가능.

〈 상위 20개 단어 빈도 분석 〉



〈 상위 20개 단어 빈도 분석 〉

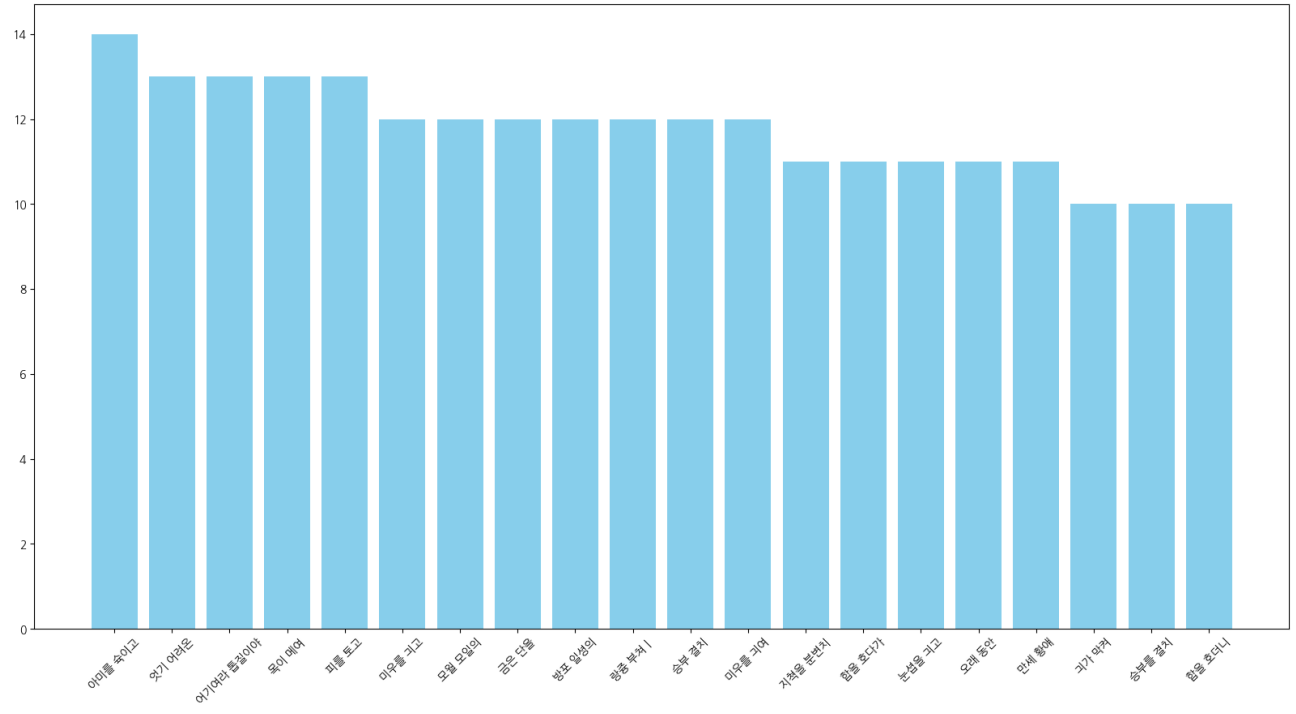


3. 텍스트마이닝 실습
3.1 기본 텍스트 분석

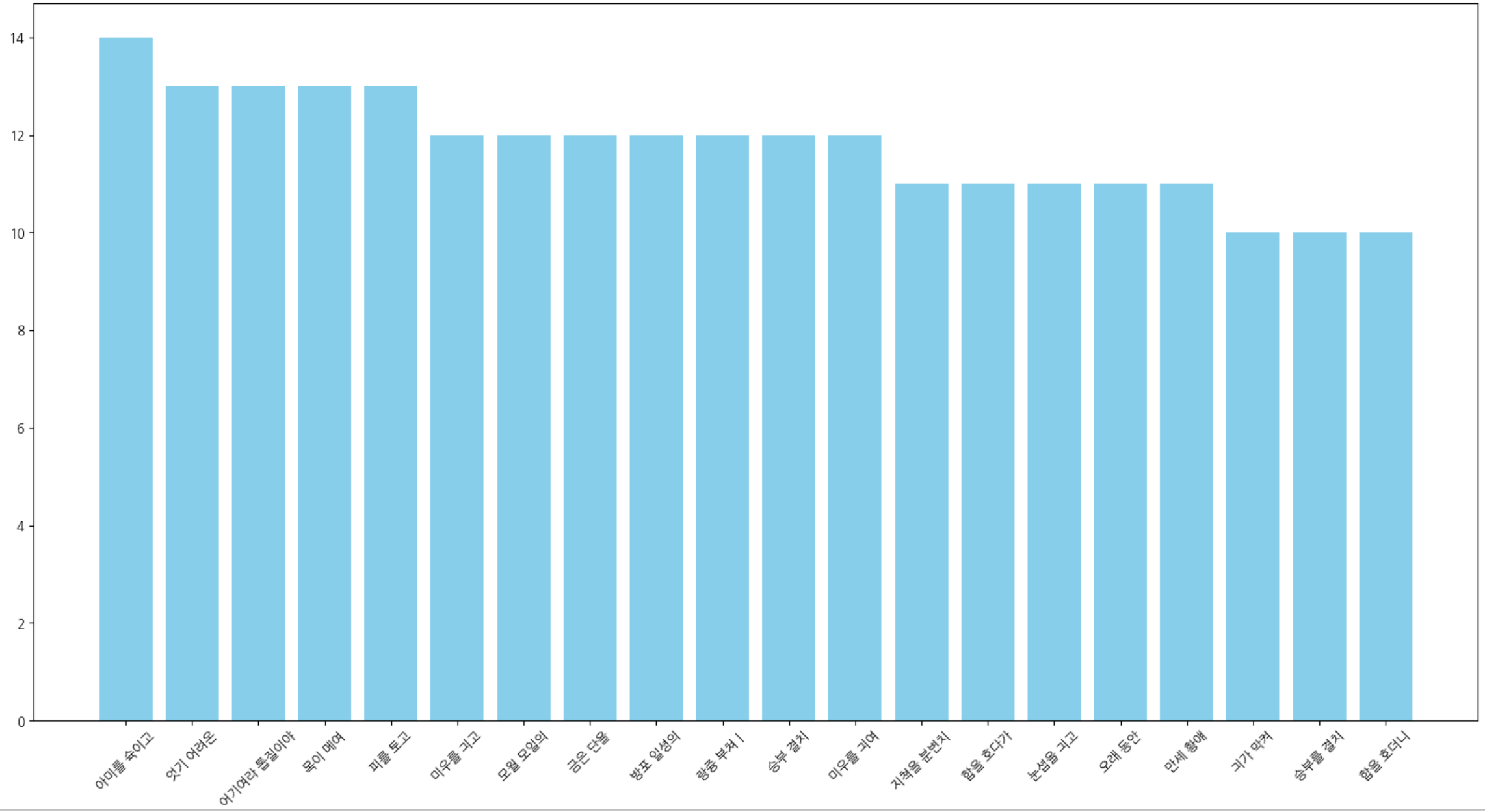
(2) N – gram 분석

- 연속된 N개의 단어 단위(2–gram, 3–gram)를 분석하여 단어 간의 연관성을 파악하는 방법.
- 단순 빈도 분석은 단어의 출현 순서를 알려주지 못한다는 단점이 있는데, 이를 어느정도 극복 가능하다는 장점이 있음.
- 언어학적으로는 문장에서 자주 사용되는 특정 구문을 발견하는 데 용이함.

〈 2–gram 상위 20개 빈도 분석 〉



〈 2-gram 상위 20개 빈도 분석 〉



3. 텍스트마이닝 실습
3.2 심화 텍스트 분석

*** 주의 사항**

- 기본 분석은 형태소분석과 임베딩을 하지 않고 단순히 띄어쓰기 단위 토큰화를 해도 가능은 함.
- 심화 분석은 이 과정을 적용해야 효용이 높음. 특히, 감정분석은 감정사전이 구축되어 있어야 더 잘 됨.
- 중세 한국어로 작성된 말뭉치는 형태소 분석기도 적용이 안 되고, 중세 기준의 감정 사전도 없음.

▶ 이러한 까닭에 중세 문헌에 심화 텍스트 분석을 할 경우 결과물을 장담하기 어려움.

- [중세 문헌] → [현대 번역] → [영어 번역] 후 분석을 수행하는 경우도 있는데, 궁여지책임.
- 이외에도 복잡한 방법이 있기는 하나 짧은 시간 내에 보이기 어려우므로, 현대 말뭉치를 가져와서 진행하겠음.

3. 텍스트마이닝 실습
3.2 심화 텍스트 분석

(1) 구간별 빈도 분석

- 앞서 단어 빈도 분석을 수행한 바 있음.
- 이를 빈도 분석을 시계열에 따라서 적용하거나, 구간별로 나누어서 분석하는 것이 가능함.
- 특히 문학의 경우, 등장하는 주요 단어가 작품의 흐름에 따라 어떻게 변화하는지 분석하는 데 사용 가능.
- 인물의 정보를 알고 있다면 구간별 등장 빈도를 확인하는 것도 가능함, 이는 장소 등 다양한 적용이 가능.

(1) 텍스트 수집

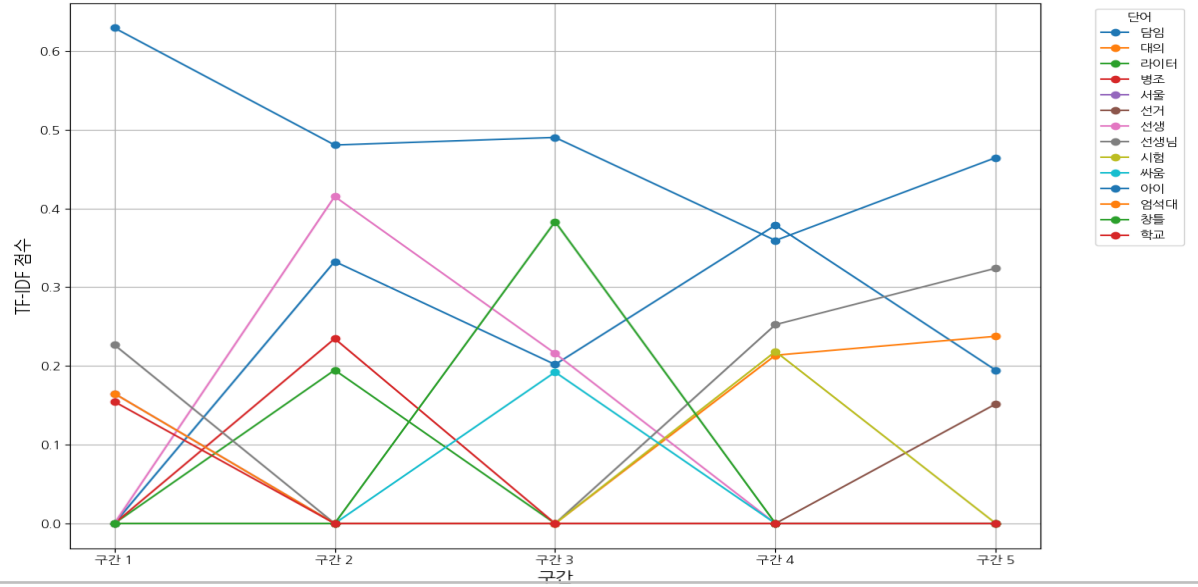
(2) 텍스트 전처리(표준화, 형태소 분석기 적용)

(3) 텍스트 구간 분절(5구간으로 설정)

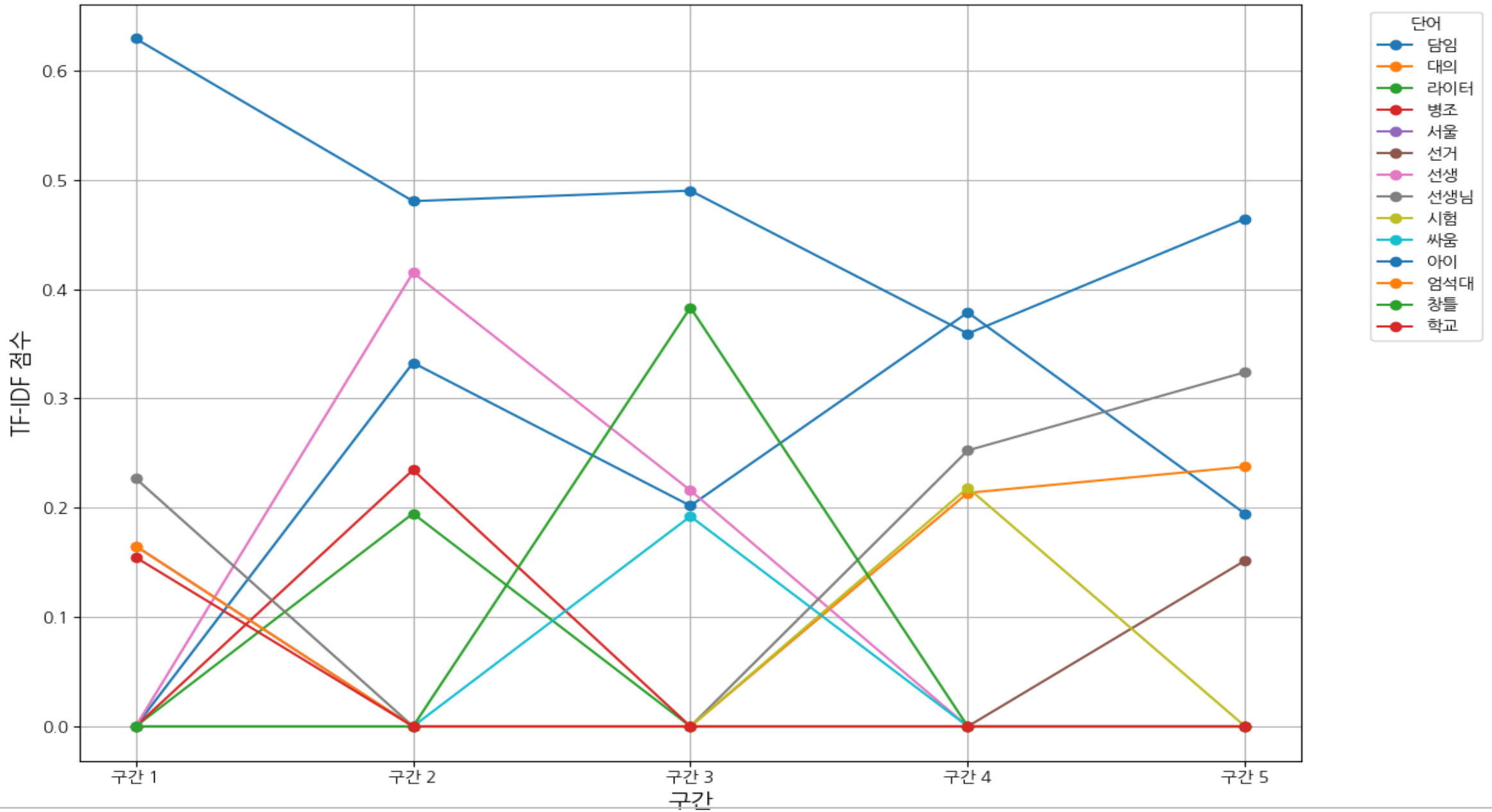
(4) 텍스트 구간별 핵심어 추출(임베딩: TF-IDF)

(5) 정리 및 시각화

〈 우리들의 일그러진 영웅 구간별 핵심어 빈도 변화〉



〈 우리들의 일그러진 영웅 구간별 핵심어 빈도 변화〉



3. 텍스트마이닝 실습
3.2 심화 텍스트 분석

* TF-IDF(Term Frequency-Inverse Document)

- 앞서 단순 빈도를 활용해 단어-문서를 임베딩하는 방식을 BoW라고 하였음.
- 단순 빈도만을 취할 경우 〈을/를〉, 〈이/가〉, 〈여기〉, 〈저기〉, 〈거기〉 등과 같이 고빈도 등장 어휘 중 큰 의미가 없는 것들로 인해 해당 문서의 주제를 알기 어려울 수 있음.
- 이를 보완하는 것이 TF-IDF 기법으로 중요한 단어에 가중치를 부여하는 방식을 취함.
- 가중치 부여는 모든 문서에 나타나는 단어는 가중치를 낮게 부여하고 특정 문서에만 나타나는 단어는 가중치를 높게 부여하는 식임. 즉, 〈이/가〉 와 같은 주격조사는 어느 문서든 나올 확률이 높으므로 이런 단어들의 중요성은 낮추는 것임.

3. 텍스트마이닝 실습
3.2 심화 텍스트 분석

(2) 토픽 모델링(Topic Modeling)

- 주어진 문서 집합에서 숨겨진 주제(Topic)를 뽑아내는 데 사용됨.
- 가장 많이 사용되는 것으로 잠재 디리클레 할당(Latent Dirichlet Allocation, LDA)이 있음.
- 방법은 다음과 같음.

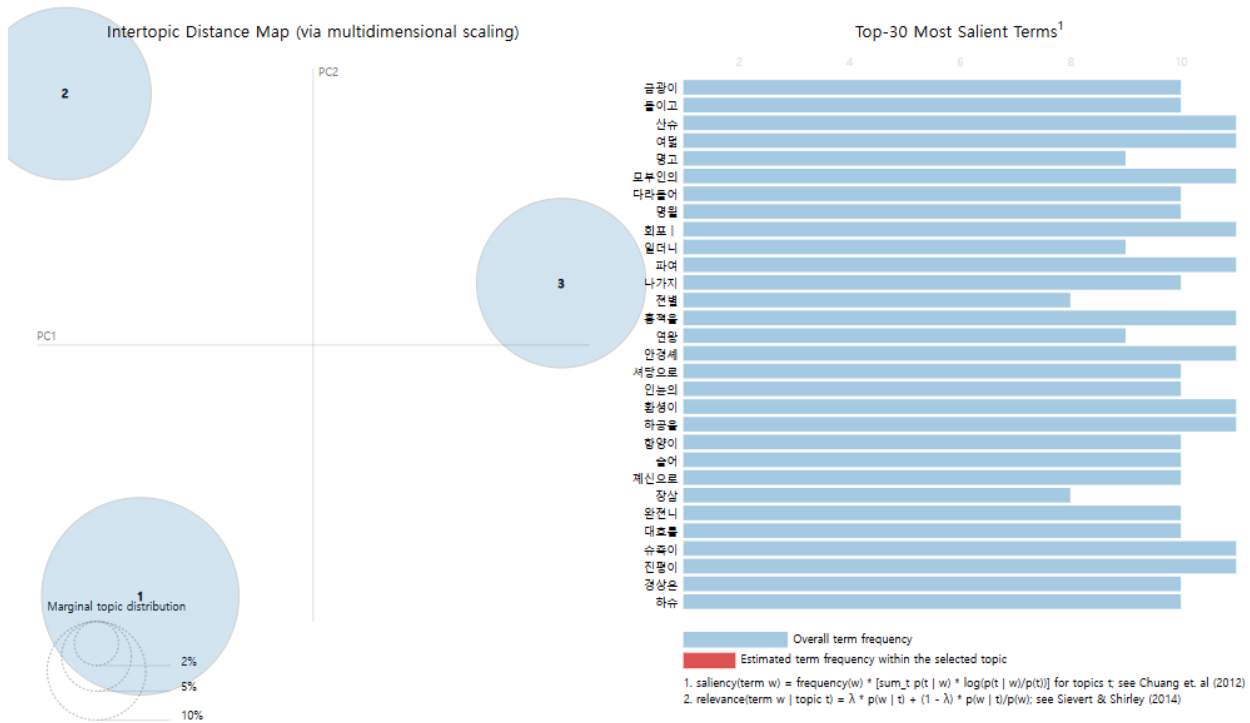
(1) 텍스트 수집

(2) 텍스트 전처리(표준화, 형태소분석기 활용)

(3) 텍스트 임베딩(TF-IDF 활용)

(4) LDA 모델 실행

(5) 시각화 및 정리



3. 텍스트마이닝 실습
3.2 심화 텍스트 분석

* 잠재 디리클레 할당(Latent Dirichlet Allocation, LDA)

- 문서 데이터에서 숨겨진 주제을 발견하기 위해 사용하는 토픽 모델링 기법임.
- 각 단어들을 특정 토픽에 할당하고, 각 문서의 토픽 비율을 측정하는 방식을 취함.
- 디리클레 분포는 확률분포의 하나로, 이러한 분포가 사용되기 때문에 그러한 이름이 붙은 것임.
- 주제별 분포 결과물의 예를 들면 다음과 같음.

구분	토픽 비율
문서 1	스포츠 90%, 요리 10%
문서 2	스포츠 5%, 요리 95%
문서 3	스포츠 85%, 요리 15%

3. 텍스트마이닝 실습
3.2. 심화 텍스트 분석

(3) 감정 분석(Sentiment Analysis)

- 텍스트 데이터에 담긴 감정을 자동으로 식별하고 분류하는 것을 말함.
- 기본적으로는 긍정 – 중립 – 부정 을 식별하며, 기쁨, 슬픔, 분노 등 다양한 감정도 식별할 수 있음.
- 문학 작품의 경우, 스토리 전개에 따른 감정 변화를 시각화 해볼 수 있음.
- 대사나 행동에 나타난 감정을 분석하여 인물의 성격이나 인물들의 관계성을 확인하는 것도 가능함.

(1) 텍스트 수집

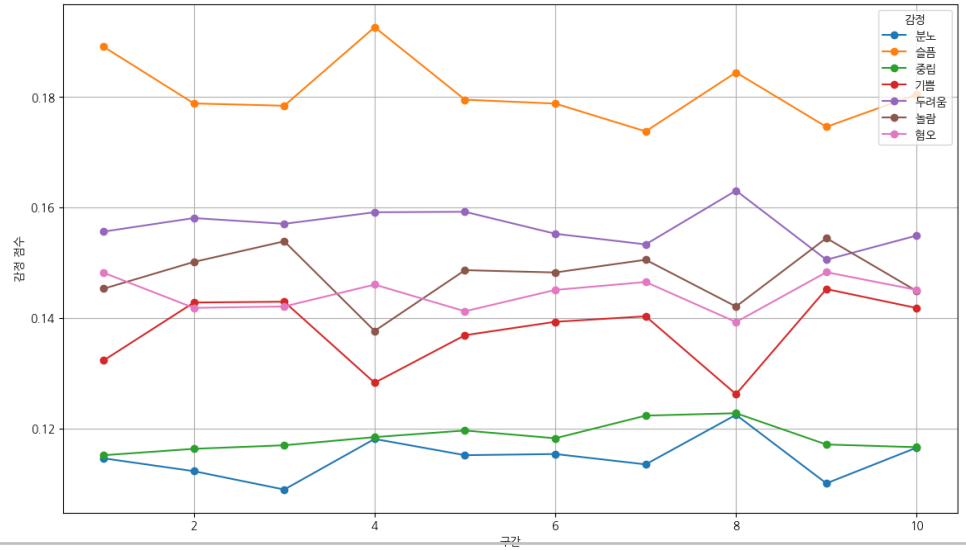
(2) 텍스트 전처리(표준화, 구간 분절)

(3) 사전 학습된 모델에 적용(KoBERT)

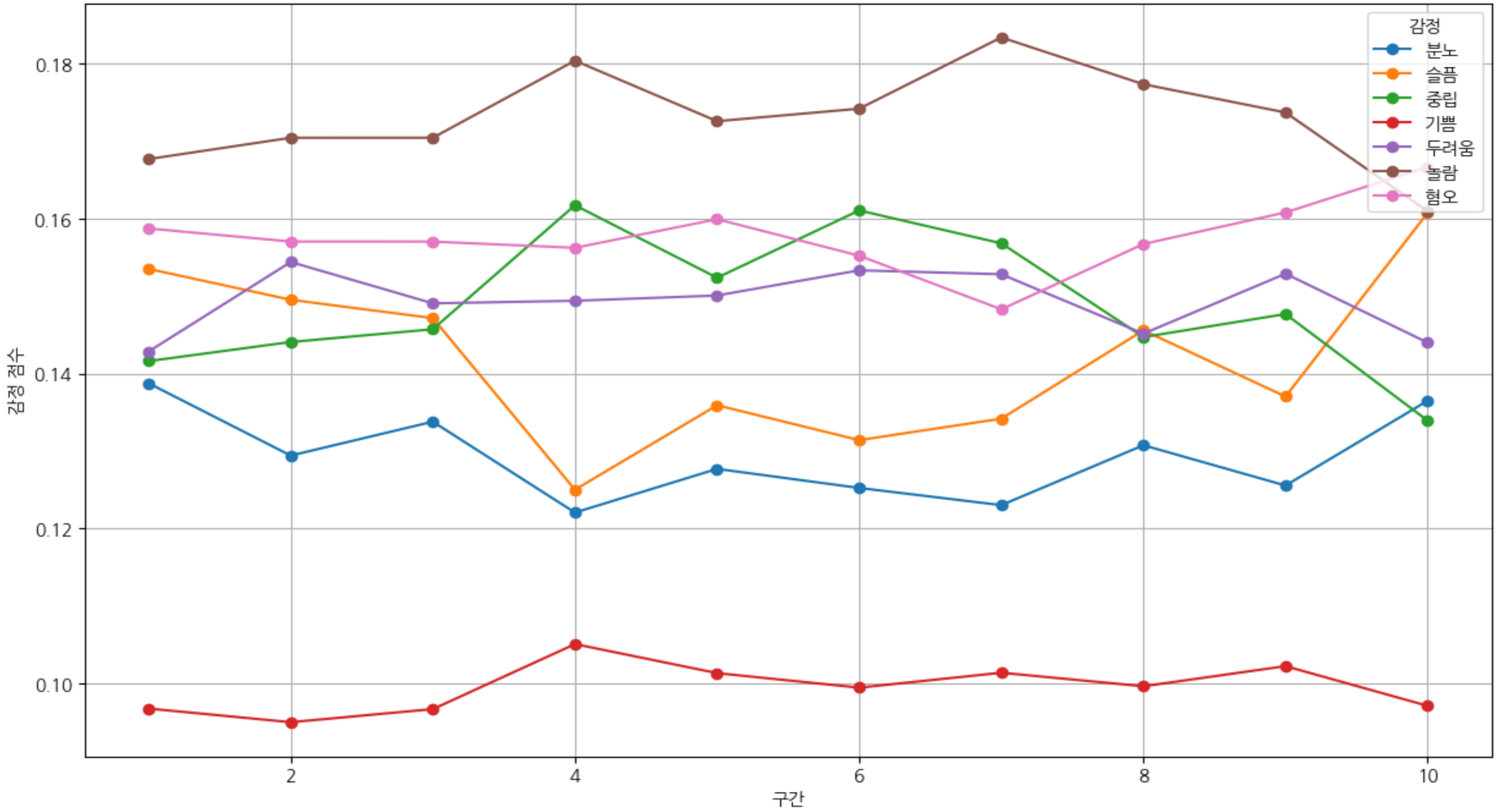
(4) 시각화 및 정리

* 언어 모델은 알아서 토크나이징, 임베딩, 분석을 한번에 처리할 수 있음.

〈 우리들의 일그러진 영웅 구간별 감정 변화 〉



〈 우리들의 일그러진 영웅 구간별 감정 변화 〉



3. 텍스트마이닝 실습
3.2. 심화 텍스트 분석

*** BERT (Bidirectional Encoder Representations from Transformers)**

- 2018년 Google에서 개발한 자연어 처리 모델로, GPT와 같이 Transformer 아키텍처를 기반으로 함. 마스크 언어 모델을 이용하여 양방향 문맥을 학습할 수 있다는 특징을 지님. 전이학습(Fine-tuning)을 통해 문서 분류, 감정 분석 등 다양한 과제에 활용될 수 있음.
- BERT와 같은 언어 모델은 사용자가 단순히 원문 텍스트를 입력하기만 하면, 모델이 텍스트를 토큰화하고, 벡터로 변환하며, 최종적으로 원하는 과제의 결과를 출력할 수 있어 매우 편리함.

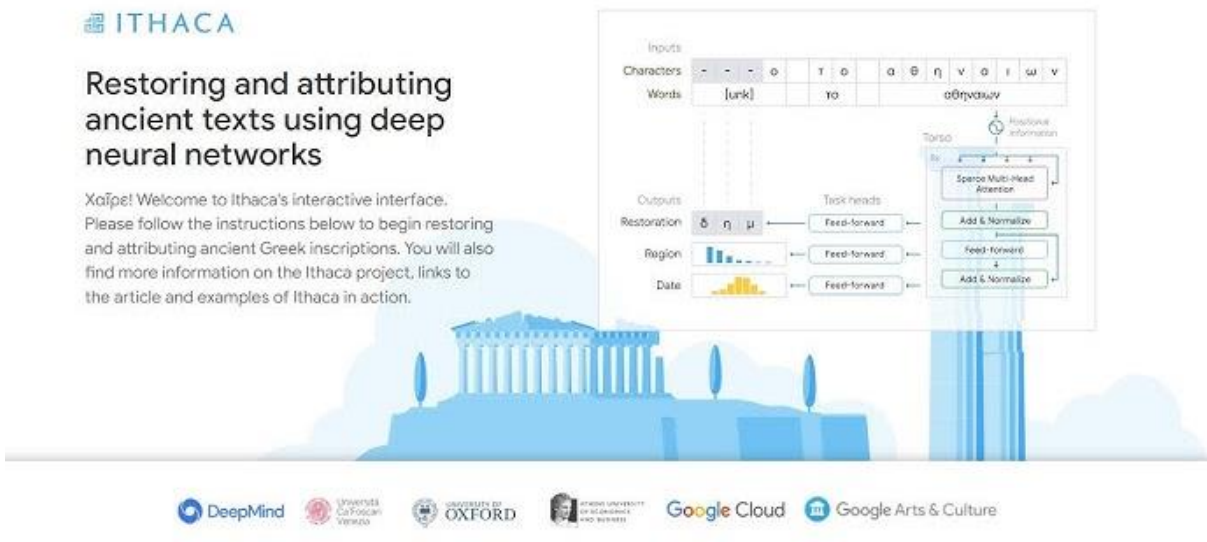
*** KoBERT**

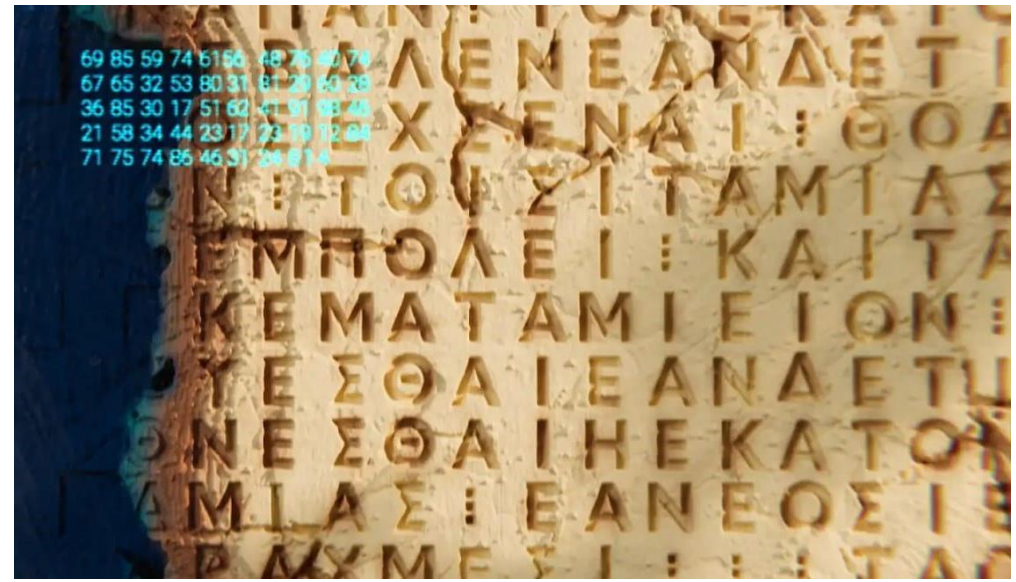
- KoBERT는 SK텔레콤에서 개발한 한국어에 특화된 기반 자연어 처리 모델임. 한국어의 문법을 더 잘 이해하기 위해 한국어 코퍼스를 사용해 학습된 모델이라고 할 수 있음.
- KoBERT 감정 분석 모델은 KoBERT를 기반으로, 감정 레이블이 부착된 대규모 한국어 데이터를 활용해 전이학습(Fine-tuning)한 것임. 이에 한국어 말뭉치를 적용한 감정 분석에서 높은 정확도를 제공함.

3. 텍스트마이닝 실습
3.2. 심화 텍스트 분석

(4) 기 타

- 텍스트 유사도 분석 및 데이터 비교에 사용되는 여러 도구 (Beyond Compare)를 활용하여 고전 문학 작품의 이본 분석을 하는 것도 가능함.
- 장르 분류, 저자 식별, 문체 분석 등 텍스트 마이닝은 다양한 분야에 사용되고 있는데, 심지어 고대 언어를 스스로 식별하고 작성 시기 및 위치까지 추정해주는 모델도 운용 중임(아래는 구글의 ITHACA).





Q&A

– 2부: 고전문학 자료와 텍스트마이닝 –