

텍스트 수집 및 전처리

한경수
성결대학교 컴퓨터공학과

Introduction



- ◆ 대량의 텍스트를 어떻게 수집할 수 있을까?
 - Web 문서 수집
- ◆ 수집한 텍스트를 그대로 분석하면 될까?
 - 텍스트 전처리

한경수

2

학습 목표



- ◆ 문제정의 및 모델링 역량
 - 웹 문서 수집 방법 및 텍스트 전처리 과정을 이해할 수 있다.
- ◆ 공학기술 및 도구 활용 역량
 - 웹 문서를 수집하고 텍스트 전처리 과정을 수행하는 프로그램을 작성할 수 있다.

한경수

3

웹 문서 수집

한경수

4

웹 문서 수집기

- ◆ Web Crawler
 - 웹 페이지를 자동으로 찾아서 다운로드

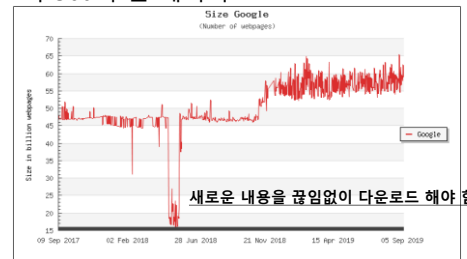


한경수

5

웹 문서 특징 1: 규모

- ◆ 급격하고 지속적으로 증가
 - 약 560억 웹 페이지



출처: worldwidewebsize.com

한경수

6

웹 문서 특징 2: 제어

- ◆ 웹 문서는 문서 수집자나 수집 프로그램의 제어를 받지 않음
 - www.sungkyul.ac.kr의 웹 문서를 다운로드하자. 그런데 문서가 몇 개나 있지?
 - 복사 허용?
 - 접근 차단?
 - 폼(form) 입력 후 접근 가능한 페이지는?

한경수

7

웹 페이지 수집

- ◆ 모든 페이지는 URL을 가짐
 - Uniform resource locator
 - 스킴(scheme) + 호스트명 + 자원명
- ◆ 웹 페이지는 웹 서버에 저장됨
 - HTTP 프로토콜을 사용하여 클라이언트 SW와 정보 교환

http://www.cs.umass.edu/csinfo/people.html

http www.cs.umass.edu /csinfo/people.html
scheme hostname resource

한경수

8

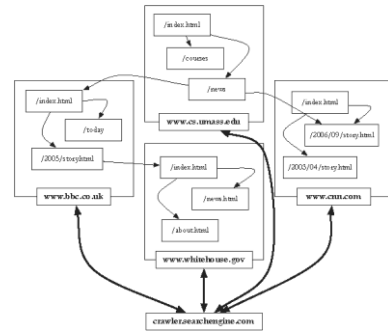
웹 페이지 수집 절차

- ◆ Seed URL 리스트를 가지고 시작
- ◆ Seed는 요청 큐(request queue)에 추가됨
- ◆ 요청 큐에 있는 페이지를 가져오기 시작
- ◆ 다운로드된 페이지를 파싱하여 링크 태그를 찾음
- ◆ 새로운 URL을 요청 큐에 추가
- ◆ 새로운 URL이 없거나 저장공간이 부족할 때까지 계속

한경수

9

웹 페이지 수집 절차



한경수

10

웹 페이지 수집

- ◆ 요청에 대한 응답을 기다리는데 많은 시간을 허비
 - 멀티 쓰레드를 사용하여 수백 개의 페이지를 동시에 수집
- ◆ 한 웹 서버에 수집기가 빈번하게 수집 요청을 한다면?
 - 수집기 요구 > 사용자 요구



한경수

11

Politeness Policy

- ◆ 특정 웹 서버에서 한번에 하나의 페이지만 가져옴
- ◆ 동일한 웹 서버로는 몇 초 혹은 몇 분 간 대기한 후 요청함



한경수

12

수집 제어

- ◆ 웹 사이트 관리자가 수집을 불허하고 싶다면?
- ◆ 로봇 배제 표준
 - robots.txt 파일로 제어

```
User-agent: *           적용 대상 수집기
Disallow: /private/     수집 거부 자원들
Disallow: /confidential/
Disallow: /other/       수집 허용 자원들
Allow: /other/public/


User-agent: FavoredCrawler
Disallow:
```

참고: <https://support.google.com/webmasters/answer/6062608?hl=ko>

한경수

13

13

- ◆ 영화 전문 검색엔진의 수집기는? 
- 웹 문서 전체를 수집해야 할까?
- 영화 관련 문서만 수집?
- 어떻게?

한경수

14

14

집중 수집(focused crawling)

- ◆ 특정 주제에 관한 페이지들만 수집
- ◆ Vertical search에 이용
- ◆ 특정 주제에 대해 인기 있는 페이지들이 시드로 사용됨
 - 한 주제에 관한 페이지는 동일한 주제에 관한 다른 페이지로의 링크를 가지고 있을 것이다.
- ◆ 문서 분류기를 사용하여 페이지가 특정 주제에 관한 것인지 판단

한경수

15

15

심층 웹(Deep Web)

- ◆ 수집기가 찾기 어려운 사이트들
 - Deep web >> indexed web
- ◆ 비공개 사이트
 - 진입 링크 부재, 로그인 요구
- ◆ 폼 입력 결과(form result)
 - 비행 시간표
- ◆ 스크립트 페이지
 - JavaScript, Flash, ...

심층 웹은 색인 웹보다 수백 배 더 클 것으로 추정됨

한경수

16

16



실습1

한경수

17

17

Python 웹 문서 수집

- ◆ Requests 모듈
 - <https://requests.readthedocs.io/>
 - `get(url, params=None, **kwargs):`
웹 문서 수집(GET 요청)
 - url: 수집할 웹 문서 URL 문자열
 - params: 딕셔너리 형식의 파라미터
 - Response 객체 리턴
 - `.status_code`, `.encoding`, `.text`
 - `post(url, data=None, json=None, **kwargs):`
POST 요청
- ◆ Selenium
 - <https://www.selenium.dev/>

한경수

18

18

HTML Parsing

- BeautifulSoup
 - <https://www.crummy.com/software/BeautifulSoup/>
- 4가지 객체
 - Tag
 - 이름(name)
 - 속성
 - 속성의 이름,값은 딕셔너리(attrs) 형태로 접근
예: tag['속성이름']
 - NavigableString
 - 태그로 둘러싸인 텍스트(string)
 - BeautifulSoup
 - 파싱된 문서 전체
 - '[document]'라는 이름이 가진 특별한 Tag 객체로 간주
 - Comment
 - 주석; NavigableString의 한 종류로 취급

한경수

19

19

Beautiful Soup: 파싱 결과 Navigation

- 태그 이름으로 탐색
 - 해당 이름을 가진 첫번째 태그 리턴
 - 해당 이름을 가진 모든 태그를 찾으려면: find_all('태그명')
- 자식 탐색
 - .contents 리스트: 어떤 태그의 (바로 아래)자식들이 담긴 리스트
 - .children 제너레이터를 이용해서 각 자식을 반복적으로 접근할 수도 있음
 - .descendants: 자손들을 재귀적으로 접근
- 부모 탐색: .parent, .parents
- 형제 탐색:
 - .next_sibling, .previous_sibling, .next_siblings, .previous_siblings
- 엘리먼트 단위 탐색; 문서에 등장하는 순서에 따라
 - .next_element, .previous_element, .next_elements, .previous_elements

한경수

20

20

Beautiful Soup: 파싱 결과 탐색

- 필터
 - 문자열 find_all('b')
 - 정규 표현식 find_all(re.compile("^b"))
 - 리스트 find_all(['a', 'b'])
- find_all(name, attrs, recursive, string, limit, **kwargs)
 - 태그의 자손들 중 필터와 매칭되는 것들 탐색
 - CSS 클래스로 탐색하려면:
 - 키워드 인자 class_ 이용 find_all('a', class_='sister')
 - CSS Selector로 탐색하려면: select(), select_one() 이용
- find(name, attrs, recursive, string, **kwargs)
 - 태그의 자손들 중 필터와 매칭된 첫번째 결과
- find_parent(), find_next_sibling(), find_previous_sibling(), find_all_next(), find_all_previous()

한경수

21

21

Beautiful Soup: CSS Selector

- #id 속성 ID가 "id"인 엘리먼트
- .class 클래스 이름이 "class"인 엘리먼트
- [attr] 속성 이름이 "attr"인 속성을 가진 엘리먼트
- [attr=val] 이름이 "attr"이고 값이 "val"인 속성을 가진 엘리먼트
- [attr="val"] 상동
- [attr^=valPrefix] 이름이 "attr"이고 값이 "valPrefix"로 시작하는 속성을 가진 엘리먼트
- [attr\$=valSuffix] 이름이 "attr"이고 값이 "valSuffix"로 끝나는 속성을 가진 엘리먼트
- [attr*=valCont] 이름이 "attr"이고 값이 "valCont"를 포함하는 속성을 가진 엘리먼트

한경수

22

22

Beautiful Soup: CSS Selector

- E#abc ID가 "abc"인 E 엘리먼트
- E.xyz 클래스가 "xyz"인 E 엘리먼트
- E[happy] 속성 이름이 "happy"인 속성을 가진 E 엘리먼트
- E F E의 자손인 F 엘리먼트
- E > F E의 자식(바로 밑)인 F 엘리먼트
- E + F 형제인 E 바로 다음에 나오는 F 엘리먼트
- E ~ F 형제인 E 뒤에 나오는 F 엘리먼트
- E, F, GE, F, G 중 어느 하나라도 매칭되는 모든 엘리먼트들

상세 사항 참고: <https://facelessuser.github.io/soapsieve/selectors/>

한경수

23

23

실습: 네이버 뉴스 검색 결과 파싱

- 사용자로부터 검색어를 입력 받아, 네이버 뉴스를 검색한 후, 검색 결과 중 상위 5개의 뉴스에 대해 (뉴스 제목, URL, 출처)를 추출하여 출력하시오.
- 참고: <https://search.naver.com/search.naver?query=검색어&where=news>

한경수

24

24

텍스트 전처리

본격적인 텍스트 분석을 위해 사전에 처리해야 하는 작업들

- 토큰 분리
- 불용어 제거
- 스테밍
- n-그램
- 형태소 분석 및 품사 태깅

한경수

25

25

토큰 분리(Tokenizing)

한경수

26

26

토큰 분리(Tokenizing)

- ◆ 문자열 → 토큰열; 단어 형성 과정
- ◆ 토큰 분리의 단순 규칙 예:
 - 알파벳이나 숫자로 구성된 3개 이상의 문자를 단어로 간주
 - 공백이나 다른 특수 문자 기준으로 분리
 - 대문자는 소문자로 변환
 - 예:
 - Bigcorp's 2007 bi-annual report showed profits rose 10%. →
 - bigcorp 2007 annual report showed profits rose
 - *너무 단순, 너무 많은 정보 손실 발생*

한경수

27

27

토큰 분리 이슈

- ◆ 짧은 단어도 중요할 수 있음
 - xp, pm, lg, world war II
- ◆ 하이픈(-)
 - 많은 단어 들이 하이픈 있든 없든 동일한 의미임
 - 불필요한 경우:
 - e-bay, wal-mart, active-x, cd-rom, t-shirts
 - 하이픈을 단어의 일부나 구분자로 간주해야 하는 경우:
 - winston-salem, mazda rx-7, pre-diabetes, t-mobile, spanish-speaking

한경수

28

28

토큰 분리 이슈

- ◆ 특수 문자는 태그, URL, 코드 등에서 중요한 부분
- ◆ 대문자로 시작하는 단어는 소문자 단어와 서로 다른 의미를 가질 수 있음
 - Bush, Apple
- ◆ 생략부호(apostrophe)
 - 단어 일부, 소유격 표시, 단순 실수
 - rosie o'donnell, can't, don't, 80's, 1890's, men's straw hats, master's degree, england's ten largest cities, shriner's

한경수

29

29

토큰 분리 이슈

- ◆ 숫자가 중요한 경우도 있음
 - galaxy note 7, nokia 3250, top 10 courses, united 93, quicktime 6.5 pro, 92.3 the beat, 288358
- ◆ 마침표(period): 숫자, 약어, URL, 문장의 끝 등에 출현
 - 6.25, I.B.M., Ph.D., cs.umass.edu, F.E.A.R.

한경수

30

30

토큰 분리

- ◆ 토큰 분리 과정은 단순(simple)하고 유연(flexible)해야 함
 - 2단계로 구성
 - 단계1: 문서의 구조 파악(마크업, 태그)
 - 단계2: 적절한 부분에 대해서만 토큰 분리 적용
 - 어렵고 복잡한 이슈들은 다른 처리 과정들에서 다루도록 함

한경수

31

토큰 분리 규칙: 예

- ◆ 공백문자, 마침표, 하이픈을 단어 구분자로 간주하여 토큰 분리
- ◆ 모두 소문자로 변환
- ◆ 생략부호(apostrophe)는 무시
 - O'Connor → oconnor Bob's → bobs
- ◆ 약어에 사용된 마침표 무시
 - I.B.M. → ibm

한경수

32

빛과 어둠으로 이뤄진 우주의 다양한 현상을 보여주는 새로운 이미지를 허블 우주망원경이 포착했다. 미국항공우주국(NASA, 나사)은 31일(현지시간) 유럽우주기구(ESA)와 함께 공동으로 운영하는 허블 우주망원경이 빛과 어둠에 둘러싸인 젊은 별을 관측한 이미지를 공개했다. 사진 속 중심 아래 부분에는 ...

불용어 제거

텍스트의 모든 단어가 동등하게 중요할까?

한경수

33

불용어 제거(Stopping)

- ◆ 관사, 전치사 등 기능어(function word)는 단어 자체에 의미가 거의 없음
 - the, a, an, that, those, over, under, above
- ◆ 기능어의 특징
 - 매우 자주 사용됨 → 고빈도
 - 기능어 자체로는 의미가 거의 없어 텍스트 분석에 거의 영향을 주지 못함
- ◆ 기능어들은 불용어(stopword)로 간주되므로 제거
- ◆ 여러 단어가 같이 사용됐을 때는 중요할 수도 있음: 예) "to be or not to be"

한경수

34

불용어 제거

- ◆ 불용어 리스트(stopword list)는 컬렉션에서의 고빈도 단어나 표준 리스트를 사용해서 생성될 수 있음

한경수

35

- Q: 성결대 컴퓨터공학

- D1: 성결대 컴퓨터공학부는 취업률이 매우 높다
- D2: 성결대에도 컴퓨터공학부가 있다
- D3: 성결대에서 컴퓨터공학을 공부하려면 매우 열심히

- Q: Michael Phelps swimming

- D1: ... Michael Phelps swam ...
- D2: ... swim ... Michael Phelps ...
- D3: ... Michael Phelps ... He swims ...

스테밍

한경수

36

스테밍(Stemming)

- ◆ 단어에 대해 다양한 형태론적 변형이 존재함
 - 굴절(inflexion) : 복수형, 시제
 - 파생(derivation) : 동사의 명사형(-ation)
- ◆ 대부분 이 변형들은 동일하거나 매우 유사한 의미를 가짐
- ◆ 스테머(stemmer)는 형태론적 변형을 동일한 어근으로 축소시킴: 대개 suffix 제거

한경수

37

Porter 스테머

- ◆ 1970년대부터 정보검색 실험에 사용된 알고리즘 기반 스테머
- ◆ 여러 단계를 거쳐 suffix를 제거함
- ◆ 각 단계는 suffix를 제거하기 위한 규칙들로 구성됨
- ◆ 각 단계에서 적용 가능한 최장길이 규칙이 실행됨
- ◆ 단어(word)가 아닌 스템(stem)을 만들어냄
- ◆ 오류가 많고 수정이 어려움
 - <https://tartarus.org/martin/PorterStemmer/>

한경수

38

Porter 스테머

◆ 예

Step 1a:

- Replace *sses* by *ss* (e.g., *stresses* → *stress*).
- Delete *s* if the preceding word part contains a vowel not immediately before the *s* (e.g., *gaps* → *gap* but *gas* → *gas*).
- Replace *ied* or *ies* by *i* if preceded by more than one letter, otherwise by *ie* (e.g., *ties* → *tie*, *cries* → *cri*).
- If suffix is *us* or *ss* do nothing (e.g., *stress* → *stress*).

Step 1b:

- Replace *eed*, *eedly* by *ee* if it is in the part of the word after the first non-vowel following a vowel (e.g., *agree* → *agree*, *feed* → *feed*).
- Delete *ed*, *edly*, *ing*, *ingly* if the preceding word part contains a vowel, and then if the word ends in *at*, *bl*, or *iz* add *e* (e.g., *fished* → *fish*, *pirating* → *pirate*), or if the word ends with a double letter that is not *ll*, *ss* or *zz*, remove the last letter (e.g., *falling* → *fall*, *dripping* → *drip*), or if the word is short, add *e* (e.g., *hoping* → *hope*).
- Whew!

한경수

39

Porter 스테머

오탐지

False positives

organization/organ
generalization/generic
numerical/numerous
policy/police
university/universe
addition/additive
negligible/negligent
execute/executive
past/paste
ignore/ignorant
special/specialized
head/healing

미탐지

False negatives

european/europe
cylinder/cylindrical
matrices/matrix
urgency/urgent
create/creation
analysis/analyses
useful/usefully
noise/noisy
decompose/decomposition
sparse/sparsity
resolve/resolution
triangle/triangular

시스템 출력	정답	O	X
O	T	T	F
X	F	F	T

False Negative
미탐지

False Positive
오탐지

한경수

40

구(Phrase)와 n-gram

엔진 vs. 검색 엔진

한경수

41

구(Phrase)

- ◆ 단일 단어에 비해 구를 사용하면
 - 더 정밀하게 주제를 묘사함
 - fish vs. tropical fish
 - 모호함이 감소됨
 - apple vs. rotten apple
- ◆ 구를 어떻게 인식할 수 있을까?

한경수

42

Syntactic Phrase

- 문장의 구문구조(syntactic structure)를 사용하여 문법적인 구 인식
- 명사구
 - 명사 명사 or 형용사 명사
- 품사부착기(part-of-speech(POS) tagger) 이용

한경수

43

품사부착기 출력 예

Original text:

Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

Brill tagger:

Document/NN will/MD describe/VB marketing/NN strategies/NNS carried/VBD out/IN by/IN U.S./NNP companies/NNS for/IN their/PRP agricultural/JJ chemicals/NNS /, report/NN predictions/NNS for/IN market/NN share/NN of/IN such/JJ chemicals/NNS /, or/CC report/NN market/NN statistics/NNS for/IN agrochemicals/NNS /, pesticide/NN /, herbicide/NN /, fungicide/NN /, insecticide/NN /, fertilizer/NN /, predicted/VBN sales/NNS /, market/NN share/NN /, stimulate/VB demand/NN /, price/NN cut/NN /, volume/NN of/IN sales/NNS /.

한경수

44

명사구 추출 예 - 고빈도 명사구

TREC data	Phrase	Patent data	Phrase
65824	united states	375362	present invention
61327	article type	191625	u.s. pat
33864	los angeles	147352	preferred embodiment
18062	hong kong	95097	carbon atoms
17788	north korea	87903	group-consisting
17308	new york	81809	room temperature
15513	san diego	78458	seq id
15009	orange county	75850	brief description
12369	prime minister	66407	prior art
12799	first time	59828	perspective view
12067	soviet union	58724	first embodiment
10811	ruian federation	56715	reaction mixture
9912	united nations	54619	detailed description
8127	southern california	54117	ethyl acetate
7640	south korea	52195	example 1
7620	end recording	52003	block diagram
7524	europcan union	46299	second embodiment
7436	south africa	41694	accompanying drawings
7362	san francisco	40554	output signal
7086	news conference	37911	first end
6792	city council	35827	second end
6348	middle east	34881	appended claims
6157	peace process	33947	data end
5955	human rights	32338	cross-sectional view
5837	white house	30193	outer surface

한경수

45

구

- 품사 부착기는 대규모 컬렉션에 사용하기에는 너무 느림
- 더 간단하고 빠르면서도 효과적인 방법이 없을까?



한경수

46

n-그램(n-gram)

- 구의 정의를 단순화해 구를 인식
- 구 = 연속된 n개의 단어
 - 바이그램(bigram; 2-그램): 두 개의 단어열
 - 트라이그램(trigram; 3-그램): 세 개의 단어열
 - 유니그램(unigram; 1-그램): 단일 단어
- 단어열에서 일부를 중복해서 n-그램을 생성
 - In application with large collections
 - {In application, application with, with large, large collections}

한경수

47

n-그램

- 더 자주 출현한 n-그램일수록 의미 있는 구일 가능성이 높음
- n-그램은 Zipf 법칙을 만족할까?
- 어떤 n-그램을 이용해야 할까?
 - 많은 저장 공간이 필요함
 - 예: 1,000 단어로 구성된 문서
 - $2 \leq n \leq 5$ 인 경우, n-그램 3,990개
 - 특정 길이까지 모든 n-그램을 색인
 - 품사 부착보다 더 빠름
- 많은 웹 검색엔진이 n-그램을 이용함



한경수

48

형태소 분석 및 품사 태깅

한경수

49

형태소 분석

- ◆ 표층 어절이 어떤 형태소들로 구성된 것인지 분해하는 작업
- ◆ 예:
 - 입력: 나는
 - 출력:
 - 나/대명사 + 는/조사
 - 나/동사 + 는/어미
 - 날/동사 + 는/어미

한경수

50

품사 태깅

- ◆ 문맥(context)에 따라 모호성을 해소함으로써, 어절의 형태소 및 품사의 조합을 결정하는 작업
- ◆ 예:
 - 입력: 나는 누워 있었다.
 - 출력:
 - 나/대명사 + 는/보조사
 - 눕/동사 + 어/연결어미
 - 있/보조동사 + 었/선어말어미 + 다/종결어미

한경수

51

정리



- ◆ 웹 문서 수집
- ◆ 텍스트 전처리
 - 토큰 분리
 - 불용어 제거
 - 스테밍
 - n-그램
 - 형태소 분석 및 품사 태깅

한경수

52



실습2

한경수

53

NLTK

- ◆ Python 자연어처리 툴킷
 - <https://www.nltk.org/>
- ◆ 설치


```
activate tmclass
conda install nltk
```
- ◆ 필요 리소스 다운로드


```
nltk.download()
```
- ◆ 토큰 분리, 불용어 제거, 스테밍, n-그램 추출, 품사 태깅 등

한경수

54

KoNLPy

- ♦ Python 한국어 형태소 분석기
 - <https://konlpy.org/ko/latest/>
- ♦ 설치
 - JDK 설치
 - <https://www.oracle.com/kr/java/technologies/javase-downloads.html>
 - <https://jdk.java.net/17/>
 - JAVA_HOME 환경변수 설정
 - JAVA_HOME: JDK 설치 위치
 - Path: %JAVA_HOME%\bin을 Path 변수에 추가
 - Jpype1 설치
 - Java와 Python을 연결하는 역할
 - KoNLPy 설치

```
activate tmclass
conda install jpype1
pip install konlpy
```

한경수

55

KoNLPy

- ♦ 실행 시 다음과 같은 오류 발생 시

SystemError: java.nio.file.InvalidPathException: Illegal char <> at index 68: C:\Users\shan\anaconda3\envs\tmclass\lib\site-packages\konlpy\java*

- Jpype1의 버전을 변경하여 재설치

```
activate tmclass
conda remove jpype1
conda install jpype1=1.1.2
```

- Jupyter Notebook 종료 후 다시 실행

한경수

56