

빅데이터 최신기술

문장 유사도 계산

소프트웨어학부

20163162

차운성

1. 과제 수행 내용 설명

1) 2개 문장에 대한 유사도 계산(two-sentence-similarity.py)

- 프로그램 개요

키보드로부터 두 개의 문장을 입력받아, 두 문장의 유사도를 검사하는 프로그램이다. 입력받은 문장을 벡터화 시 음절 trigram을 기준으로 벡터화 하였고, 유사도 검사 방법은 코사인 유사도를 사용하였다.

키보드로부터 문장을 입력받으면, 작성한 getTrigramSet 함수를 이용해 각 문장의 trigram을 추출한 후, BOW 형태의 집합이 반환되도록 하였다. 이후 각 집합의 값들을 python 내장 자료구조인 딕셔너리에 넣어주면서 각 trigram별로 아이디 값을 부여하였다. 딕셔너리 내의 모든 trigram을 검사하면서 첫 번째 BOW에 해당 trigram이 있는 경우 첫 번째 벡터 해당 인덱스에 trigram의 빈도를 넣어주었고, 없는 경우에는 0을 넣어주었다. 두 번째 문장의 경우에도 같은 방식으로 벡터화를 진행하였다. 상기 과정을 통해 얻어진 두 개의 벡터를 이용해 코사인 유사도를 계산하였다.

2) KCC150 말뭉치의 각 문장에 대해 가장 유사한 문장의 쌍 출력(sentence-similarity.py)

- 데이터 전처리 과정

이미 전처리 된 데이터 trigram-sejong을 이용하는 방법도 있지만, KCC150의 말뭉치를 이용해 전처리된 데이터를 이용하고 싶어서 다른 데이터셋을 사용하였다. 먼저 get-ngram을 이용해 KCC150 말뭉치의 trigram을 뽑아낸 후 split와 wordcount를 이용해 각 trigram의 빈도수를 뽑아냈다. 이후 preprocessing.py를 이용해 feature 수를 뽑아냈는데, 빈도수가 50 이상인 trigram과 trigram 중 한 음절이라도 한글이 있는 trigram을 제외한 나머지 trigram을 제외시켰다. 이렇게 했을 때 약 40만개의 feature가 생성되는데, 빈도수의 기준을 낮추면 전처리 된 데이터와 KCC150 말뭉치 한 프로그램에 같이 적재할 수 없을 것 같다는 판단 하에 빈도수의 기준을 50으로 잡았다.

위 과정을 거치면 triVec.dat 라는 이름의 전처리된 데이터가 생성된다. 데이터의 전처리와 이후 이뤄지는 문장 유사도 검사는 모두 EUC-KR로 인코딩 된 KCC150 말뭉치를 기준으로 진행되었다.

- 프로그램 개요

먼저 전처리된 triVec.dat의 trigram과 각 trigram의 id값을 프로그램 내부 딕셔너리에 저장하며, 이 때, 딕셔너리의 키 값은 trigram이 된다. 이후 KCC150 말뭉치를 open해 한 줄 씩 읽으면서 유사도를 검사한다. 프로그램 구동 시 시간 관계 상 비교 시 기준 문장을 KCC150 말뭉치의 상위 20개의 문장으로 간추렸고, 기준 문장과 비교할 비교 대상 문장들은 KCC150 말뭉치의 전체 문장 중 10%로 설정하였다. 이 때에도 한 문장 당 비교해야 할 문장의 수가 총 119만 문장이기 때문에 한 문장의 유사도를 검사할 때 평균적으로 1~2분씩 소요되었다.

먼저 기준 문장에서 trigram을 추출 해 벡터화를 진행하였고, 비교 대상 문장에 대해서도 각각 trigram 추출 및 벡터화를 진행하였다. 해당 과정이 진행되어 두 문장에 대한 벡터를 얻은 후 simDoc.py를 참고하여 작성된 코사인 유사도 함수를 사용해 두 문장의 코사인 유사도를 검사하였고, 이 중 유사도 값이 높은 2개의 문장을 저장하였다. 한 문장에 대해 유사도 검사가 끝나면 문장과 유사도값, 문장의 벡터를 출력한다.

- 세부 구조

전처리된 데이터는 "id \t trigram"의 형태로 저장되어 있는데, load 시 딕셔너리 형태로 프로그램 내부에 저장하도록 하였으며 이 때 딕셔너리의 키 값은 trigram이다.

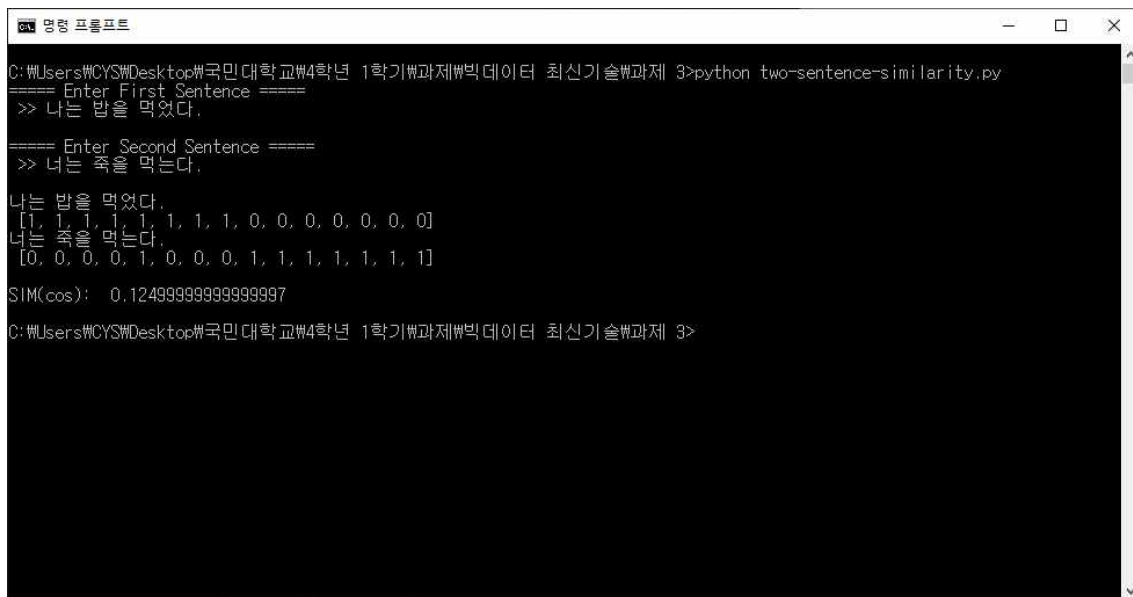
유사도 값이 높은 상위 2개의 문장을 저장하기 위해 2차원 리스트를 이용하였고, 첫 번째 인덱스에는 문장이 들어가고, 두 번째 인덱스로는 유사도 값이 저장된다. 이 후 분석된 유사도 값이 현재 저장되어있는 유사도 값보다 높은 경우를 처리하기 위해 항상 2차원 리스트의 상태를 유사도 값 기준 오름차순으로 정렬된 상태로 유지하였다.

한 문장이 벡터화 된 구조는 [[id, freq], [id, freq], ...]와 같은 형태이며, 이 또한 2차원 리스트로 구현하였다. 이 때 문장의 벡터를 출력 시에 id값을 기준으로 오름차순 정렬된 형태로 출력된다. 문장을 벡터화 할 때에는 먼저 문장의 trigram을 추출하는데, 문장 길이 - 2 까지 for문을 수행하면서 trigram을 추출하고, 해당 trigram이 전처리된 데이터를 저장한 딕셔너리에 있으며 해당 trigram의 id값을 가져오도록 하였고, trigram이 딕셔너리에 없는 경우는 아무런 작업도 수행하지 않고 다음 trigram을 검사하도록 하였다.

벡터화된 두 문장의 유사도는 simDoc.py의 dot과 norm함수를 사용하였는데, 이 때 norm의 경우 강의자료에 기술된 것과 달리 각 차원 값의 제곱의 합에 루트를 씌워두지 않아, 파이썬 내장 라이브러리인 math를 활용해 루트를 씌워준 후 연산하도록 하였다.

2. 실행화면 스크린샷

1) 2개 문장에 대한 유사도 계산



```
C:\Users\CYS\Desktop\국민대학교\4학년 1학기\과제\빅데이터 최신기술\과제 3>python two-sentence-similarity.py
==== Enter First Sentence ====
>> 나는 밥을 먹었다.

==== Enter Second Sentence ====
>> 너는 죽을 먹는다.

나는 밥을 먹었다.
[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
너는 죽을 먹는다.
[0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1]

SIM(cos): 0.12499999999999997
C:\Users\CYS\Desktop\국민대학교\4학년 1학기\과제\빅데이터 최신기술\과제 3>
```

- 이 때 여러 trigram 중 두 문장에서 겹치는 trigram은 “을_먹”이 된다.

2) KCC150 말뭉치의 각 문장에 대해 가장 유사한 문장의 쌍 출력

```
명령 프롬프트 - python sentence-similarity.py KCC150_Korean_sentences_EUCKR.txt triVec.dat
C:\Users\mcys\Desktop\source\Homework\빅데이터 최신기술\과제 3>python sentence-similarity.py KCC150_Korean_sentences_EUCKR.txt triVec.dat
-- Input Sentence
>> 통합보건의료교육은 이 대학만의 특화된 프로그램이다.
[[25652, 1], [35526, 1], [53784, 1], [62219, 1], [63537, 1], [77181, 1], [93797, 1], [101180, 1], [133628, 1], [142668, 1], [154653, 1], [161712, 1], [174906, 1], [202427, 1], [291815, 1], [293458, 1], [296888, 1], [298322, 1], [374722, 1], [377181, 1], [385699, 1], [403349, 1]]

-- Similar Sentence
>> 1. 붓 프로그램이란 자동화된 작업을 반복 수행하는 일종의 복제 프로그램이다.
[[41193, 1], [42860, 1], [47234, 1], [54439, 1], [54850, 1], [55030, 1], [63537, 2], [101180, 2], [123867, 1], [141648, 1], [142554, 1], [152923, 1], [154653, 1], [161712, 2], [193007, 1], [204206, 1], [204597, 1], [239194, 1], [266030, 1], [294477, 1], [296490, 1], [299903, 1], [307290, 1], [310618, 1], [312675, 1], [326889, 1], [332557, 1], [385699, 2], [387778, 1], [396889, 1], [403349, 1]]
similarity:0.3143473067309658

>> 2. 이는 올해 처음 신설된 프로그램이다.
[[23049, 1], [48137, 1], [51859, 1], [59068, 1], [63537, 1], [101180, 1], [123817, 1], [142668, 1], [154653, 1], [161712, 1], [228183, 1], [249064, 1], [277424, 1], [295275, 1], [299516, 1], [351468, 1], [385699, 1], [394112, 1]]
similarity:0.2870189239408639

-- Input Sentence
>> 이에 따라 전달된 후원금은 저소득층과 사회복지시설에 2억원 상당을, 중구 푸드뱅크 설립 지원에 6000만원, 저소득 긴급지원과 시설 등의 지원에 5000만원이 전해진다.
[[613, 1], [641, 1], [1107, 2], [1682, 2], [4783, 1], [23153, 1], [27131, 1], [32083, 1], [36974, 1], [37088, 1], [44688, 1], [45079, 1], [45941, 1], [47862, 1], [55692, 2], [55874, 1], [56032, 1], [57509, 1], [57883, 2], [63319, 1], [66101, 1], [90104, 1], [90155, 1], [94553, 1], [103561, 1], [104454, 1], [107987, 1], [129206, 1], [142705, 1], [145413, 1], [145935, 1], [148094, 1], [148901, 1], [151395, 1], [171705, 1], [174892, 1], [197702, 1], [204654, 1], [216927, 1], [2210182, 1], [227953, 1], [228231, 1], [228395, 1], [233180, 1], [233197, 1], [245892, 1], [245916, 1], [264425, 1], [26675]
```

- 기준 문장인 “통합보건의료교육은 이 대학만의 특화된 프로그램이다.”에서 가장 유사한 2개의 문장이 “붓 프로그램이란 자동화된 작업을 반복 수행하는 일종의 복제 프로그램이다.”와 “이는 올해 처음 신설된 프로그램이다.”이다.

385699 프로그램 **161712 로그램**

- 이 중 첫 번째 문장의 경우 “프로그램이다”와 “프로그램”이 두 문장에서 동일하게 나타나는 음절임을 확인할 수 있으며, 실제로 triVec.dat의 id를 확인해 보면 “프로그”에 해당하는 값인 385699와 “로그램”에 해당하는 id값인 161712가 기준 문장과 비교 대상 문장의 벡터에 나타나는 것을 볼 수 있다.
- 유사도가 높은 두 번째 문장이 벡터화된 경우에 여러 벡터 중 id 385699에 해당하는 trigram이 1번, 161712에 해당하는 trigram이 1번 나타난 것을 확인할 수 있다.

```
선택 명령 프롬프트 - python sentence-similarity.py KCC150_Korean_sentences_EUCKR.txt triVec.dat
0, 1], [152069, 1], [154366, 1], [162716, 1], [241560, 1], [257845, 1], [265973, 1], [267456, 1], [267524, 1], [268467, 1], [275233, 1], [293458, 1], [299820, 1], [300944, 1], [312194, 1], [325417, 1], [339806, 1], [356268, 1], [375446, 1], [377778, 1], [388425, 1], [388611, 1], [391839, 1]]
similarity:0.24738534799764675

-- Input Sentence
>> 그러나 이 건물은 겉에서 보는 것과는 달리 지어진 지가 꽤 오래되었는지 방이 약간 낡았다는 느낌을 주었다.
[[16218, 1], [29206, 1], [29372, 1], [29407, 1], [32646, 1], [33252, 1], [34523, 1], [35039, 1], [41610, 1], [42661, 1], [49878, 1], [51542, 1], [53784, 1], [57273, 1], [57741, 1], [57864, 1], [67136, 1], [77320, 1], [78459, 1], [90550, 1], [101206, 1], [111006, 1], [111924, 1], [112405, 1], [123091, 1], [123264, 1], [123401, 1], [123416, 1], [124269, 1], [126915, 1], [129318, 1], [142186, 1], [154051, 1], [155595, 1], [168534, 1], [187438, 1], [195945, 1], [202715, 1], [224059, 1], [258621, 1], [260556, 1], [263960, 1], [266627, 1], [267791, 1], [275786, 1], [292907, 1], [294772, 1], [298151, 1], [298678, 1], [334646, 1], [339498, 1], [339889, 1], [341285, 1], [343586, 1]]

-- Similar Sentence
>> 1. 그러나 건물은 이미 반파된 상태다.
[[16218, 1], [29206, 1], [41265, 1], [45186, 1], [53918, 1], [77320, 1], [101206, 1], [111989, 1], [142449, 1], [155595, 1], [187438, 1], [187911, 1], [221355, 1], [293458, 1], [300280, 1], [378557, 1]]
similarity:0.19622098205031857

>> 2. 그러나 이 역시 오래가지 못했다.
[[16218, 1], [39837, 1], [50867, 1], [51542, 1], [53784, 1], [69624, 1], [101206, 1], [112405, 1], [153976, 1], [155595, 1], [183640, 1], [244756, 1], [270629, 1], [275778, 1], [298702, 1], [339471, 1]]
similarity:0.19622098205031857

-- Input Sentence
>> 그런데 황희찬이 다시 1부로 올라갈 수도 있다.
[[3272, 1], [16220, 1], [26972, 1], [34771, 1], [47045, 1], [51826, 1], [54752, 1], [65907, 1], [71654, 1], [101242, 1], [127328, 1], [136024, 1], [137002, 1], [151578, 1], [156492, 1], [161300, 1], [206809, 1], [237119, 1], [244412, 1], [27259, 1], [298308, 1], [348779, 1], [405969, 1], [409919, 1]]
```

- 해당 케이스의 “그러나”와 “건물은”과 관련된 trigram이 모든 문장에 있음을 볼 수 있다.

```

명령 프롬프트 - python sentence-similarity.py KCC150_Korean_sentences_EUCKR.txt triVec.dat
>> 2. 높은 나무 꼭대기까지 올라갈 수도 있죠.
[[17090, 1], [32538, 1], [32969, 1], [47045, 1], [51826, 1], [54775, 1], [71654, 1], [105412, 1], [109661, 1], [110766, 1], [113072, 1], [121961, 1], [131804, 1], [137002, 1], [151578, 1], [183838, 1], [237119, 1], [277259, 1], [292995, 1], [339645, 1]]
similarity:0.3055050463303894

----- Input Sentence
>> 응답자 중 가장 많은 의견이었다.
[[22971, 1], [28278, 1], [38505, 1], [53704, 1], [57495, 1], [69383, 1], [80739, 1], [130146, 1], [175175, 1], [293457, 1], [295904, 1], [297031, 1], [301259, 1], [310087, 1], [313797, 1], [336393, 1]]

----- Similar Sentence
>> 1. 각 구단 중 가장 많은 숫자다.
[[15185, 1], [28278, 1], [30838, 1], [38505, 1], [47426, 1], [57495, 1], [69383, 1], [69969, 1], [94782, 1], [128054, 1], [175175, 1], [240605, 1], [293318, 1], [313797, 1], [336393, 1]]
similarity:0.4244373438195827

>> 2. 군 중 가장 많은 포상을 받았다.
[[16090, 1], [28278, 1], [38505, 1], [41298, 1], [57495, 1], [63124, 1], [69383, 1], [98147, 1], [175175, 1], [193531, 1], [220855, 1], [293670, 1], [294478, 1], [313797, 1], [336393, 1], [382712, 1]]
similarity:0.4117647058823529

----- Input Sentence
>> P씨는 23일 오전 8시30분쯤 벤츠 승용차를 몰고 서울 마장동 내부순환로를 달리다 커브길에서 좌우 방호벽을 차례로 들이받았다.
[[1784, 1], [3956, 1], [5151, 1], [6086, 1], [10312, 1], [13794, 1], [15090, 1], [27767, 1], [33574, 1], [35039, 1], [36896, 1], [38277, 1], [39722, 1], [41658, 1], [42358, 1], [45619, 1], [47741, 1], [51658, 1], [57082, 1], [58618, 1], [60664, 1], [84875, 1], [108257, 1], [117072, 1], [123193, 1], [126698, 1], [129324, 1], [139761, 1], [147038, 1], [160615, 1], [160975, 1], [161902, 1], [167115, 1], [167235, 1], [168954, 1], [173039, 1], [183148, 1], [193531, 1], [196300, 1], [200502, 1], [200762, 1], [207230, 1], [209863, 1], [210876, 1], [224323, 1], [225184, 1], [239950, 1], [244035, 1], [244035, 1]]

```

- 해당 케이스에서는 “중 가장 많은”의 경우가 모든 문장에서 반복적으로 출현함을 볼 수 있다.

57495 _중_ 336393 중_가 28278 _가장
69383 _가장_ 175175 많은_ 313797 장_많

- 위의 6개의 trigram이 3개의 문장에서 공통적으로 출현한 trigram이다.
- trigram의 id값을 보면, id값은 trigram을 오름차순으로 정렬한 순서대로 부여되었음을 확인할 수 있다.