상품 Description 유사도를 이용한 상품 Matching 시스템

Product matching system using Product description similarity

요 약

소비자의 상품 선택에 있어 베스트셀러 상위에 위치한 상품의 판매량이 높다. 상위에 위치한 상품과 하위에 위치한 상품의 유사도를 추출하여 상위 상품과 하위 상품을 연결해 주는 서비스를 제안하였다. 이를 위해 본 논문은 상품설명(description)을 TF-IDF로 대표 단어를 추출하고 Word2vec를 이용하여 상위 상품과 하위 상품의 유사도를 계산, 상위 상품과 하위 상품에 유사한 상품이 있다는 유사도를 보였다.

1. 서 론

인터넷에서 도서 상품에 대한 정보량이 폭증하면서 고객이 도서 선택에 어려움을 겪는 상황이 발생하고 있다. 이 문제를 해결하기 위해 아마존(Amazon)을 비롯한 많은 국내외 인터넷 서점들이 추천시스템을 통해 고객에게 적합한 도서를 제공하고 구매를 유도하고 있다. 상품의 순위는 소비자가 상품을 선택하는데 있어서 영향을 미친다. 예를 들어 책이 판매될 때 베스트셀러 상위에 오르게 되면 하루에 5권 나가던 책이 500권 이상 나가게 된다. 이 이유는 소비자가 다른 소비자를 따라 하려는 심리 때문이다.

본 논문은 하위에 랭크된 책의 판매량을 향상시키기 위한 연구로 도서에 대한 주제어의 유사도를 판단하여 상위에 있는 책들과 유사한 책의 추출을 제안한다.

주제어 추출을 위해 아마존(amazon)의 책에 대한 데이터(meta_Books.json)의 description의 단어를 추출하여 단어의 빈도 값을 표현하는 TF-IDF가중치 모델과 유사도를 표현하는 word2vec를 이용하여 유사도를 계산하는 방법을 제안한다. TF-IDF(Term Frequency - Inverse Document Frequency)는 정보 검색과 텍스트 마이닝에서 이용하는 가중치로, 여러 문서로 이루어진 문서군이 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내는 통계적 수치이다. 문서의 핵심어를 추출하거나, 검색 엔진에서 검색 결과의 순위를 결정하거나, 문서들 사이의 비슷한 정도를 구하는 등의 용도로 사용할 수 있다.[1] Word2vec는 단어의 의미를 벡터형태로 표현하는 계량기법으로, 각 단어를 300차원 정도의 공간에서 벡터로 표현하고 있다. 두 벡터값을 계산하여 0 ~ 1 사이의 값이 나타나고 1에 가까울수록 유사도가 높다.

본 논문의 구성은 다음과 같다. 본론에서는 Hadoop을 이용하여 TF-IDF와 Word2vec를 계산하고, 이를 이용하여 두 상품의 벡터값을 추출하고, 두 값을 이용한 두 상품의 유사도 판별 프로세스를 제안한다. 결론으로 본 논문에서 제시한 프로세스를 이용하여 인터넷 서점에서의 서비스 방법을 제안한다.

2. 본 론

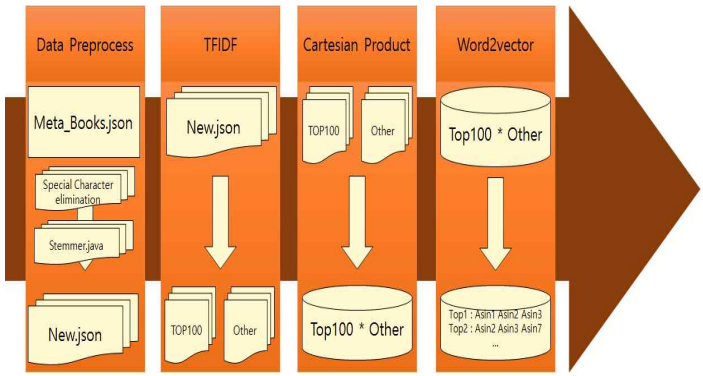


그림 1 유사도 판별 hadoop 프로세스 과정

1. TF-IDF와 Word2vec를 이용한 두 상품의 유사도 판별

1.1 data preProcess

TF-IDF와 Word2vec의 전처리 과정으로서 상품설명(description)의 단어를 어간추출(stemming)하여 새로운 json파일로 만들었다.

이는 TF-IDF와 Word2vec를 사용하기 위해 중요한 과정이며 이 과정에서 'love'와 'lovely', 'loves' 등 같은 단어 이지만 품사만 다른 경우를 같은 단어로 바꾸어 준다.[2]

1.2 TF-IDF

TF(단어 빈도, term frequency)는 특정한 단어가 문서 내에 얼마나 자주 등장하는지를 나타내는 값으로, 이 값이 높을수록 문서에서 중요하다고 생각할 수 있다. 하지만 단어 자체가 문서군 내에서 자주 사용되는 경우, 이것은 그 단어가 흔하게 등장한다는 것을 의미한다. 따라서, 일반적으로 특정 문서 d_i에서 단어 t_i의 중요도는 다음 식(1)과 같이 표준화 된다.

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \quad \text{.....(1)}$$

위 논문에서 문서대신 상품설명(descriptoin)을 사용한다.

이것을 DF(문서 빈도, document frequency)라고 하며, 이 값의 역수를 IDF(역문서 빈도, inverse document frequency)라고 한다. IDF는 해당 단어의 일반적인 중요도를 나타내는 수치이다. 전체 문서의 수를 해당 단어가 포함된 문서들의 수로 나눈 값에 로그를 취하며 다음 식(2)과 같이 표준화 된다.

$$idf_i = \log \frac{|D|}{|d_j : t_i \in d_j|} \quad \text{.....(2)}$$

위 논문에서 $|D|$는 상품의 수이며,

$$|d_j : t_i \in d_j|$$

은 단어 t_i가 등장하는 상품의 수이다.

TF-IDF는 TF와 IDF를 곱한 값이다. 이를 다음 식(3)과 같이 나타낸다.[3]

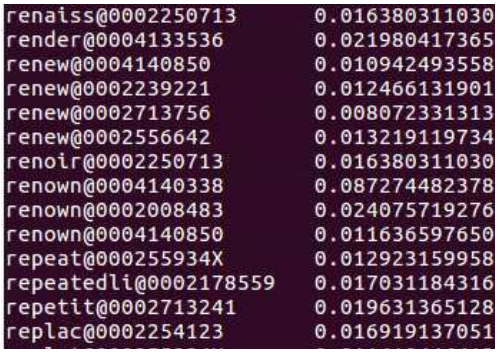
$$tfidf_{ij} = tf_{ij} \cdot idf_i \quad \text{.....(3)}$$

본 논문은 Hadoop을 이용하여 TF-IDF를 계산하는 과정은 그림(2)과 같이 3가지의 Job으로 이루어져 있다.

첫째, map에서 meta_Books.json을 파싱하고, 그 결과로 ((word@product), 1)를 reduce로 보내고 reduce에서 각 key별로 개수를 구해 ((word@product), n)으로 출력한다. (n은 단어의 출력 개수)

둘째, map에서 ((word@product), n)를 입력받아 ((product), word=n)형식으로 바꿔 reduce로 출력한다. reduce에서 ((word@product), (n/N))으로 reduce 해준다. (N은 word@product의 개수)

셋째, map에서 ((term@product), (n/N))를 입력받아 ((term), (product = n/N))의 형식으로 바꿔 reduce로 출력한다. reduce에서는 받은 값을 TF-IDF를 계산하여 최종적으로 각 단어들의 TF-IDF를 얻는다. 각 단어들의 TF-IDF의 값은 그림(3)과 같다. 마지막으로, 각 상품을 대표하는 단어 5개를 추출하여 Word2vec의 입력 값으로 사용한다.



```
renaiss@0002250713      0.016380311030
render@0004133536       0.021980417365
renew@0004140850        0.010942493558
renew@0002239221        0.012466131901
renew@0002713756        0.008072331313
renew@0002556642        0.013219119734
renoir@0002250713       0.016380311030
renown@0004140338       0.087274482378
renown@0002008483       0.024075719276
renown@0004140850       0.011636597650
repeat@000255934X       0.012923159958
repeatedli@0002178559   0.017031184316
repetit@0002713241      0.019631365128
replac@0002254123       0.016919137051
```

그림 3 TF-IDF 결과 값 예시

1.3 Cartesian Product

집합론에서, 두 집합의 곱집합(product set) 또는 데카르트 곱(Cartesian product)은 두 집합에서 하나씩 고른 두 원소의 순서쌍들의 집합이다.

Word2vec를 사용하기 위해 상위 상품100개의 상품설명과 그 외 상품의 상품설명에서 TF-IDF를 이용하여 각각의 상품설명의 주제어 5개를 추출하였고, 각각의 주제어를 Cartesian Product를 하였다.

1.4 Word2vec

Word2vec는 텍스트를 처리하는 2계층 신경망이다. 입력은 말뭉치(text corpus)이고 출력은 벡터집합이다. 즉, 해당 corpus는 단어에 대한 특징벡터 이다. Word2vec는 deep neural network는 아니지만 deep net가 이해할 수 있는 숫자 형식으로 텍스트를 변환한다. 각 단어의 벡터별 내적을 계산하면 0 ~ 1 사이의 값이 나오고 1에 가까운 값일수록 두 단어의 유사도는 높다.

본 논문에서 Word2vec를 이용하여 상위 상품의 상품설명과 그 외 상품의 상품설명의 유사도를 추출하였다. 이를 이용하여 최종적으로 상위에 상품과 유사한 상품을 3개 추출한다.

본 논문은 Hadoop을 이용하여 Word2vec를 계산하는 과정은 2개의 Job으로 이루어져 있다.

첫째, map에서 key를 (‘상위 상품코드(t)’ , ‘상위 상품설명의 단어(tw)’ , ‘하위 상품코드(o)’)로 이루어져 있고, value로 상위 상품설명 단어와 상위 상품설명 단어(tw)의 유사도로 이루어져 있다. 유사도는

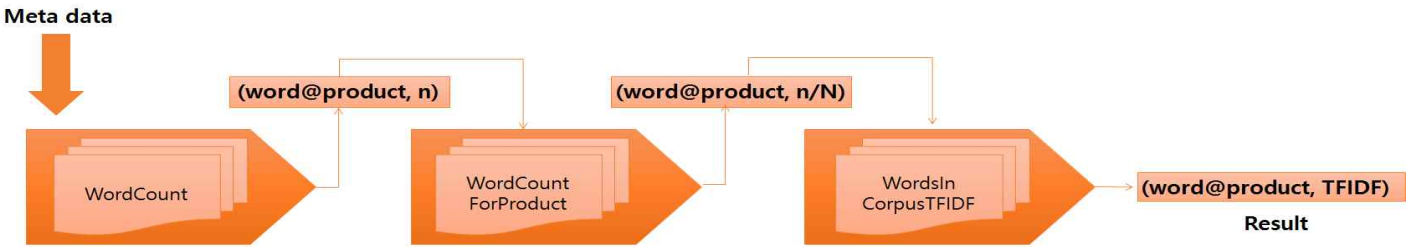


그림 2 Hadoop을 이용한 TF-IDF처리과정

단어를 내적 하여 유사도(s)를 추출한다. reduce에서는 key는 그대로 사용하며, map에서 받은 value의 max값(m)을 추출하여 value로 사용한다. 예시는 그림(4)과 같다.

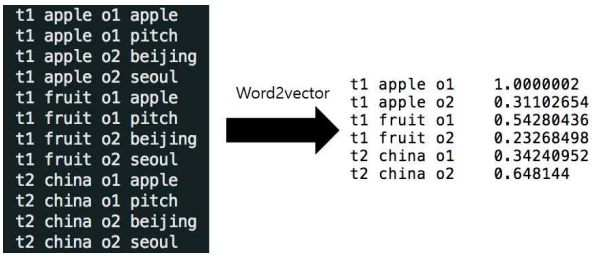


그림 4 step1의 결과

둘째, map에서 key로 (‘상위 상품 코드(t)’ ‘하위 상품 코드(o)’)로 이루어져 있으며, value는 단어들의 유사도의 max값(m)이 들어간다. reduce에서는 key는 그래도 사용하며, map에서 받은 value의 유사도 평균값(average)을 추출하여 value로 사용한다.

셋째, map에서 key를 (‘상위 상품 코드(t)’)로 바꿔주며 value로 (‘하위 상품 코드,(o)’ , ‘유사도평균값(average)’)으로 출력한다. reduce에서 유사도의 평균값이 큰 10개의 상품만 선택해 최종적으로 (‘상위 상품 코드(t)’ , ‘하위 상품 코드(o) 유사도 평균값(average)’)의 형태로 상위 상품 코드별로 각 10개의 상품과 유사도 값이 출력된다. 결과는 그림(5)과 같다.

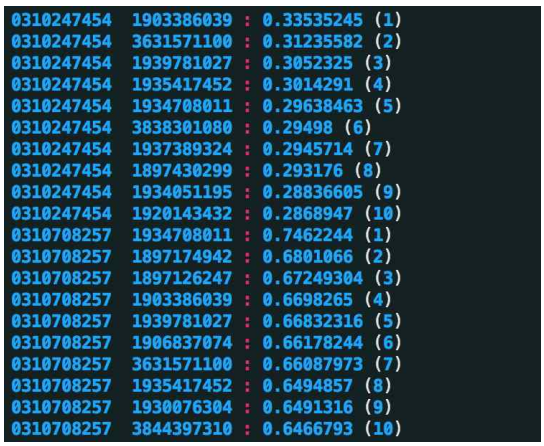


그림 5 Word2vec결과

(상위 상품 코드, 하위 상품 코드 : 유사도)

3. 결론 및 서비스 방향, 향후 연구

상품의 판매량, 평점 등을 이용하여 인터넷 서점에서 단지 랭크되는 기존 서비스의 도서 판매량의 향상을 위한 선행연구로 본 연구는 TF-IDF와 Word2vec를 이용하여 상위 상품과 유사한 하위 상품을 매칭해주는 서비스 방법을 제안하였다.

상품설명(description)을 전 처리 과정으로 ‘love’와 ‘lovely’, ‘loves’ 등 같은 단어 이지만 품사만 다른 경우를 같은 단어로 바꾸어 준다. 이 결과 값을 이용하여 TF-IDF를 계산하고 Word2vec를 계산하여 각 상품별 유사도를 판단한다. 최종 결과는 각 상위 상품의 유사한 하위 상품 10개를 선택하여 보여준다. 본 연구를 통해 상위 랭크에 위치한 상품과 하위 랭크에 위치한 상품의 유사도를 판단하였다. 실험 결과 상위 상품과 하위 상품의 유사도가 높은 결과를 얻을 수 있었다.

본 논문의 최종 결과를 이용하여 상위 상품과 유사하지만 광고부족이나, 잘 알려지지 않아 하위 랭크에 있는 상품의 판매량을 향상시키기 위해 상위 상품과 연결하여 판매하도록 도와주는 서비스를 판매자에게 제공할 예정이다.

향후 상품의 정확성을 위해 상품의 평점을 추가하여 상품의 질을 향상시키고, 상품설명(description)뿐만 아니라 상품의 본문내용을 추가하여 더 정확한 유사도를 추출하는 연구를 진행 할 예정이다.

참고문헌

[1] S. G. Lee, H.-J. Kim, “Keyword Extraction from News Corpus using Modified TF-IDF”, The Journal of Society for e-Business Studies, Vol.14, No.4, pp.59-73, 2009

[2] 이성직, "TF-IDF의 변형을 이용한 전자뉴스에서의 키워드 추출 기법", 한국전자거래학회지 14 (4) : 59 ~ 73 , 2009

[3] 유은순, “TF-IDF와 소설 텍스트의 구조를 이용한 주제어 추출 연구”, 한국컴퓨터정보학회논문지 20 (2) : 121-129, 2015