Word2vec을 이용한 상품 제목 클러스터링 (가공식품)

요약

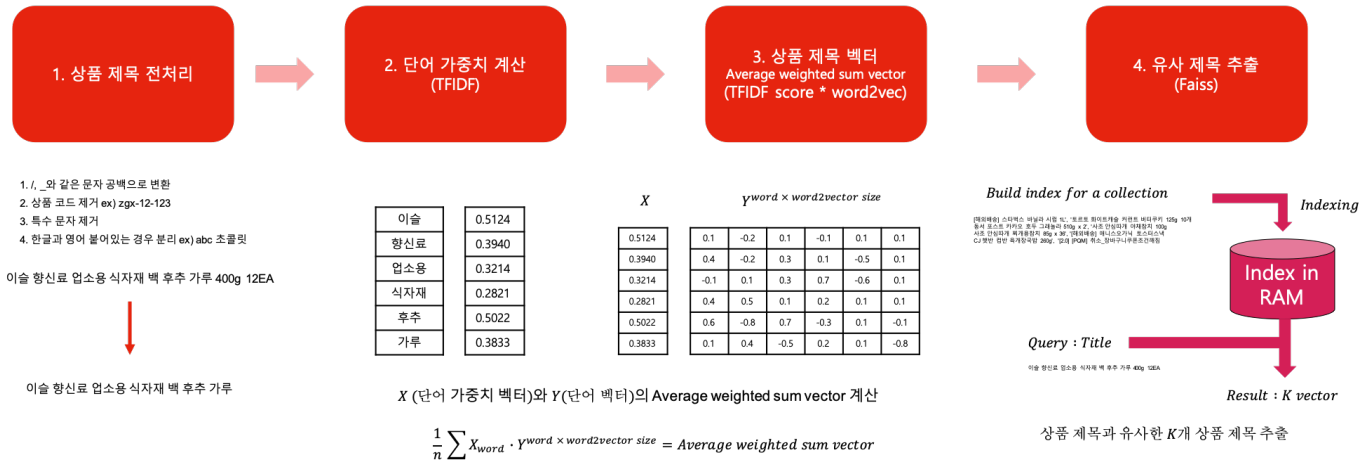
- weighted word vector을 이용해서 카탈로그(동일 상품)을 만들기

방법

- embedding 방법 : weighted word vector(Title), mobilenet image feature vector(Image)
- 유사도 계산 : faiss, cosine similarity

Overview

- 상품 제목 분류 아키텍처



상품 제목 분류 아키텍처 그림

- 작업 진행
  1. 상품 제목 전처리 형태소 분석 or 띄어쓰기 필요 (konlpy.okt, mecab) ex) 다시마전복장1kg [전복10마리600g내외+간장 총 1kg] 분리가 어려움
  2. 전처리된 상품 제목의 단어들의 가중치 계산 (TFIDF)
  3. word2vec을 이용하여 상품 제목의 단어들과 가중치를 곱해서 상품 제목 벡터를 계산
  4. Faiss 알고리즘을 통해 상품 제목의 벡터와 가까운 distance에 있는 유사한 K개 상품 제목 추출

- 1차 성능 평가 (정성적 평가)

아래의 3개의 제목 유형에 대해서 성능 측정 (Raw vector, Normalized vector - size : 50 - 200)

- Type1 : 정상적인 제목이지만 브랜드 동의어가 섞여있는 제목

- Type2 : 핵심어와 공용 검색어가 혼용되어 사용된 제목

- Type3 : 일반적인 제목

- 1차 결과 (정성적 평가) - (https://drive.google.com/open?id=1ZXUYOepYx7OOrg4Hj6-Y9ZdRwPUiVG1vus5vqC2yN8Q)
  1. Minmax scaler를 통한 200차원의 word vector가 Faiss 알고리즘의 distance(threshold)를 정함에 있어서 용이 함. (0.2) - 50차원과 200차원의 성능의 차이는 크게 나지 않지만, threshold를 정의함에 있어서 용이 함
  2. Type2와 같이 공용검색어가 많은 제목의 경우 성능이 크게 좋지 않음
  3. 단어의 수가 2개 또는 3개와 같이 다른 단어의 수를 가진 제목에서 성능이 좋지 않음 ex) 화미 업소용식자재 날 콩가루 봉 vs

날콩가루 화미 - 화미 업소용식자재 날 콩가루 400g 2봉 vs 날콩가루(화미 400g)X2

4. Text 전처리 (띄어쓰기, 모델명, 중량) 필요
5. Windowsize = 5 // 다른 NLP 처리와 다르게 제목의 단어들은 keyword 중심으로 이루어져 있다. 따라서, 단어의 평균 크기인 5를 기준으로한 window size로 해당 제목의 topic/domain information을 보고자 했다.
    a. Larger windows tend to capture more topic/domain information: what other words (of any type) are used in related discussions? Smaller windows tend to capture more about word itself: what other words are functionally similar? (Their own extension, the dependency-based embeddings, seems best at finding most-similar words, synonyms or obvious-alternatives that could drop-in as replacements of the origin word.)

- 1차 결론
    1. 위의 결과를 토대로 핵심 keyword가 되는 단어를 TFIDF weight를 통해 선정(공용 검색어 제거)하여 해당 단어를 포함하고 있는 상품제목을 1차 필터링 후에 중량과 모델 코드 또는 이미지 유사도를 통해서 2차 필터링 제안
    2. 정량적으로 평가할 수 있는 평가 Metrics를 통해 embedding model과 TFIDF 모델의 성능 비교

- 2차 성능 평가 (정량적 평가)
  이미지유사도와 제목 유사도를 고려하여 묶여진 가공식품 27만개의 Dataset을 정답으로 간주하고 성능을 평가함.

  - 평가 metric 제안 https://drive.google.com/open?id=1k82TJoV-qDGO1lepC9HMRfMze8aFYz3vaB1k9iP7RD8
    - Accuracy : 모델이 예측한 상품 제목이 실제 정답 상품 제목에 얼마나 포함되어 있는지 : (전체 정답 수 - 모델이 예측한 정답 수) / 전체 정답 수
    - Precision : 모델이 예측한 상품 제목 중에 실제 정답 상품 제목이 얼마나 포함되어 있는지 : 모델이 예측한 정답 수 / 모델이 예측한 상품 제목의 수

  - 모델 비교

    1. Embedding model1 : Average weighted sum (200) - word2vec (전체) - faiis : 0.2 / window size = 5
        - Cleansing 후 모든 word에 대한 word2vec 학습, faiis distance 0.2 이내, 모든 단어에 대해서 average weighted sum
    2. Embedding model2 : Average weighted sum (200) - word2vec (전체) - faiis : 0.2 (TFIDF TOP 2 단어) / window size = 5
        - Cleansing 후 모든 word에 대한 word2vec 학습, faiis distance 0.2 이내, TFIDF 가중치 TOP 2 단어에 대해서 average weighted sum
    3. Embedding model3 : Average weighted sum (200) - word2vec (전체) - faiis : 0.2 (TFIDF TOP 3 단어) / window size = 5
        - Cleansing 후 모든 word에 대한 word2vec 학습, faiis distance 0.2 이내, TFIDF 가중치 TOP 3 단어에 대해서 average weighted sum
    4. Embedding model4 : Average weighted sum (200) - word2vec (top3) - faiis : 0.2 - (TFIDF TOP 3 단어) / window size = 1
        - Cleansing 후 TFIDF TOP3 word에 대한 word2vec 학습, faiis distance 0.2 이내, TFIDF 가중치 TOP 3 단어에 대해서 average weighted sum
    5. Embedding model5 : Average weighted sum (200) - word2vec (top3) - faiis : 0.2 - (TFIDF TOP 2 단어) / window size = 1
        - Cleansing 후 TFIDF TOP3 word에 대한 word2vec 학습, faiis distance 0.2 이내, TFIDF 가중치 TOP 2 단어에 대해서 average weighted sum
    6. TFIDF model - top2 : TFIDF top2 word
        - Cleansing 후 TFIDF TOP 2 단어를 포함하고 있는 상품 제목 추출

Model 이름	Word2vec 학습 단어	Embedding에 사용된 단어의 수	기타
Em_Model 1	상품 제목 전체	상품 제목 전체	Window size = 5, faiss range = 0.2
Em_Model 2	상품 제목 전체	가중치 TOP-2 단어	Window size = 5, faiss range = 0.2
Em_Model 3	상품 제목 전체	가중치 TOP-3 단어	Window size = 5, faiss range = 0.2
Em_Model 4	가중치 TOP-3 단어	가중치 TOP-3 단어	Window size = 1, faiss range = 0.2
Em_Model 5	가중치 TOP-2 단어	가중치 TOP-2 단어	Window size = 1, faiss range = 0.2
TFIDF model			TOP-2 단어가 포함된 상품 제목 전체

Model 요약

- 2차 결과 (정량적 평가) – (https://drive.google.com/open?id=1ZXUYOepYx7OOrg4Hj6-Y9ZdRwPUiVG1vus5vqC2yN8Q)

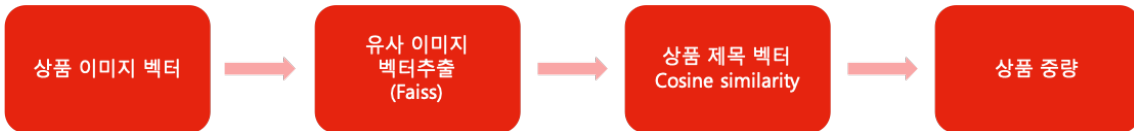
Group count가 2이상인 전체 데이터 (n=5000)			Group count가 10이상인 전체 데이터 (n=5000)	
Model	Acc(%)	Prec(%)	Acc(%)	Prec(%)
embedding model1	86	47	73	60
embedding model2	91	29	84	45
embedding model3	88	39	76	53
embedding model4	89	11	77	22
embedding model5	91	7.9	86	18
TFIDF model – top2	90	50	83	58

- 2차 결론
  1. 상품 제목 cleansing이 잘되어야 함. 단어가 없는 경우 찾아 내지 못함
  2. Word embedding의 경우 단어가 많은 경우에 성능이 더 좋음을 알 수 있음 (평균 단어 미만, 이상으로 비교하였음).
  3. 상품 제목과 같이 단어의 수가 적은경우 (상품코드, 중량 제거된 상품 제목) embedding방법 보다 단순 가중치 높은 단어의 조합한 모델이 더 좋은 성능을 보임.
     a. 상품 제목과 같이 적은 Keyword가 핵심이 된다 (사용자들도 검색할 때 2~3개 단어로 검색하기 때문)
  4. TFIDF model의 경우 2개의 단어가 모두 같을 때만 검색이 되므로 보완이 필요
  5. Word embedding방식과 다르게 TFIDF model은 연산이 N^2만큼 수행하게 됨 → Word embedding을 개선하여 활용해보기

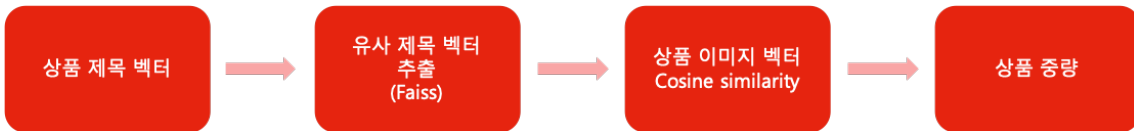
- 추가 진행 (range distance를 더 줄임)

Group count가 2이상인 전체 데이터 (n=5000, range=0.01)			Group count가 2이상인 전체 데이터 (n=5000, range=0.001)	
Model	Acc(%)	Prec(%)	Acc(%)	Prec(%)
embedding model1	84	66	83	75
embedding model2	90	53	90	53
embedding model3	86	68	86	68
embedding model4	86	68	86	68
embedding model5	90	53	90	53
TFIDF model – top2	–	–	–	–

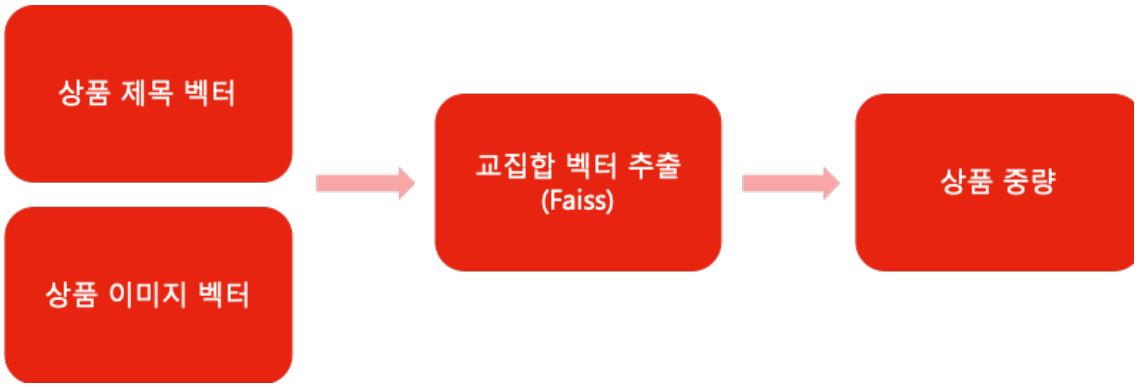
- 3차 작업 진행
  1. 상품 이미지와 상품 제목의 벡터를 이용하여 상품 클러스터링 진행
     a. Image → Title → 중량



     b. Title → Image → 중량



     c. Intersection → 중량



d. Concatenation → 중량

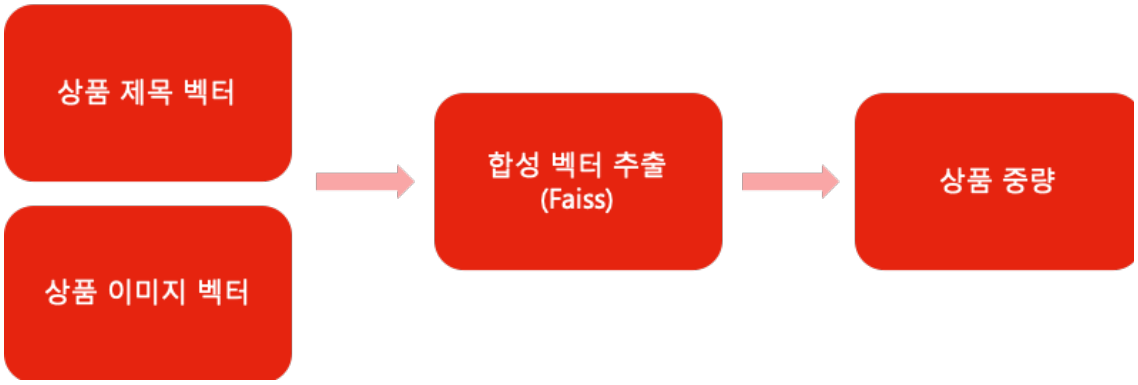


Image -> Title -> 중량				Title -> Image -> 중량				Intersection -> 중량				Concatenation -> 중량		
Image	Title(Cosine)	Acc(%)	Prec(%)	Text	Image(Cosine)	Acc(%)	Prec(%)	Image	Title	Acc(%)	Prec(%)	Total	Acc(%)	Prec(%)
0.3	0.9	91	25	0.1	0.9	74~82	79~90	0.3	0.1	72~79	80~91	0.01	67.2~70.8	91.5~95.3
·	0.95	90	26~28		0.95	74~82	79~90	·	0.2	72~79	80~91	0.05	70~74.9	86.2~93.5
·	0.99	86~88	35~49		0.99	69~75	84~93	·	0.3	72~79	80~91	0.1	75~79	81.6~90.7
·	0.999	77~85	60~82	0.3	0.9	82~89	62~78	0.4	0.1	72~79	80~91	0.15	78.6~83.5	76.1~87.5
·	0.9995	76~85	67~86		0.95	77~82	75~86	·	0.15	72~79	80~91	0.2	82~86	70~84
0.3	0.9999	76~84	76~89		0.99	71~76	82~90	·	0.2	72~79	80~91	0.25	84~89	63~80
0.4	0.9999	76~84	76~89	·	·	·	·	·	·	·	·	·	·	

2. Title과 Image 경향성 확인

Title -> 중량			Image -> 중량		
Title	Acc(%)	Prec(%)	Image	Acc(%)	Prec(%)
0.1	82~92	40~75	0.1	85	56 ·
0.2	84~92	32~67	0.2	89	40 ·
0.3	85~92	27~61	0.3	91	25 ·
0.4	86~93	24~56	0.4	· 91	17 ·

3. 상품 이미지와 상품 제목의 벡터를 이용하여 상품 클러스터링 진행 보완 방안

- 모델 No
- 가격


new_1022_Title_...mbedding_yg.pdf