9조 Final Report

오윤진, 정우성, 진시윤, 황상민

1. 문제 정의

상품 리뷰는 오늘날 소비자의 구매의사결정에 중요한 영향을 미친다. 그러나 대부분의 경우에는, 이러한 리뷰 데이터가 별도의 구분 없이 일괄적으로 보여지는 경우가 많다. 따라서 소비자는 해당 제품이 본인의 구매 기준에 부합하는지 확인하기 위해, 직접 리뷰들을 일일이 확인해야 한다. 가령 화장품을 한 개 구매하더라도 사람마다 고려하는 기준이 다른데, 소비자는 비효율적으로 자신의 구매 기준과 관련 없는 정보들까지 모두 살펴봐야 하는 상황인 것이다.

이러한 문제를 해결하기 위하여 본 프로젝트에서는 (대표적으로) 뷰티 카테고리의 상품 리뷰 데이터셋을 활용하여, 리뷰 텍스트에 담긴의 상품 속성 정보들을 추출하고, 이를 바탕으로 리뷰 토픽(일종의 분류 태그)를 생성하여 리뷰들을 분류해보고자 하였다. 이를 통해 제품 리뷰를 읽는 사용자에게 더 나은 사용자 경험을 제공하는 것이 궁극적인 목표이다.

2. 방법론

중간 보고서까지의 실험을 통해 '문서 클러스터링 이후 클러스터 별 토픽 모델링'을 구현하기 위한 파이프라인을 완성하였다. 파이프라인은 문서 임베딩 - 차원 축소 - 클러스터링 - 클러스터 별 토픽 모델링으로 구성하였으며, 각 단계에서는 SentenceBERT - UMAP - HDBSCAN – (LDA 또는 c-TF-IDF) 알고리즘을 사용하였다.

2.1. SentenceBERT

본 프로젝트에서는 본래 텍스트 마이닝에서 일반적으로 많이 사용되는 TF-IDF 방식을 이용하여 문서 임베딩 작업을 수행하고자 하였다. TF-IDF 란 단어 빈도(Term Frequency)와 역문서 빈도(Inverse Document Frequency) 지표를 결합해 단어의 중요도를 바탕으로 문서를 임베딩하는 기법이다. 그러나, TF-IDF를 임베딩 기법으로 활용하고자 하였더니, 임베딩 후의 데이터 차원이 너무 크다는 문제점이 두드러지게 나타났다. 따라서 기존의 문서 임베딩 기법을 TF-IDF 기반 Sparse embedding에서 SentenceBERT[1] 기반 Dense Embedding으로 변경하였다. TF-IDF를 SentenceBERT로 변경해 사용한 결과, 원데이터의 표현력을 잃지 않고 보다 dense 한 문서 임베딩을 생성해낼 수 있었을 뿐더러, 이후 파이프라인에서 HDBSCAN 알고리즘이 효율적인 시간 내에 실행되게 할 수 있었다.

2.2. UMAP

실험해보았던 (시행착오를 겪었던) 차원 축소 기법은 PCA, SVD, t-SNE, UMAP 총 4가지였고, 데이터의 비선형성과 시간 복잡도를 고려하여 UMAP이 가장 적합하다고 판단하였다. UMAP (Uniform Manifold Approximation and Projection)[2]은 t-SNE의 단점 중 하나였던 속도를 해결하였고, 비선형의 복잡한 데이터 구조를 보존할 수 있다는 장점이 있어 SVD의 한계를 극복할 수 있었다.

HDBSCAN은 UMAP의 차원 축소 결과를 받아 이어지기 때문에, UMAP을 통해 몇 차원으로 축소되었는가가 중요한 요소 중 하나였다. HDBSCAN의 docs[3]에 따르면, HDBSCAN은 medium dimensional 데이터에서 가장 성능이 뛰어나고, 그 차원이 증가할수록 점점 성능이 떨어진다고 한다. 또한 통상적으로, 데이터의 차원이 50차원에서 100차원 사이일 경우에 최적의 성능을 보인다고 기술되어있다. 따라서, 본 팀에서는 전반적인 파이프라인의 성능을 공통적으로 향상시키기 위하여 UMAP의 축소될 차원을 50차원으로 설정하였다.

2.3 HDBSCAN

상기 과정을 거쳐 본 팀은 입력 데이터를 적당한 크기의 차원을 가진 dense 행렬로 표현하였다. 그리고 이 데이터를 효율적인 시간 안에 적은 메모리를 사용하여, label 없이 deterministic하게 클러스터링하기 위하여 HDBSCAN 알고리즘을 채택하였다.

UMAP의 차원 축소 결과와 HDBSCAN의 min_samples(클러스터로 인정되기 위해 필요한 밀집도), min_cluster_size(한 클러스터를 구성하는 데이터의 최소 개수)의 적절한 조합을 찾아 클러스터링의 성능을 높이고자 하였다. HDBSCAN의 경우 모델 자체적으로 DBCV 점수(클러스터 내부의 밀집도와 클러스터 간 밀집도의 조합)를 평가 메트릭으로 제공하고 있어, 이를 바탕으로 해당 하이퍼파라미터들을 조절해보았다. 하지만 DBCV 점수만을 활용하여 클러스터링을 하였을 때, 아래와 같이 점수는 높지만 outlier가 너무 많거나 클러스터의 개수 또는 클러스터 하나의 크기가 너무 작은 경우가 발생하였다.

1	13789
2	7123
3	1007
4	131
5	101
Outlier	7849

Table 1. HDBSCAN 결과 중, DBCV 점수가 가장 높았던 클러스터링 결과

본 팀은 정의한 문제 상황의 해결을 위해서는 한 클러스터의 최소 크기를 전체 샘플의 5% 정도로 outlier의 최대 개수를 전체 샘플의 10% 이내로 정할 필요가 있다고 판단하였다. 이에 DBCV 점수가 0.5 이상이고 지정한 조건에 부합하는 결과들 중 가장 적합해 보이는 조합을 하이퍼파라미터로 결정하였다. 또한 클러스터링 결과 outlier로 label 된 데이터는 이것이 내용 측면에서 다른 리뷰들과 동떨어진 리뷰라 판단하여 이후 토픽 모델링 및 키워드 태깅 과정에서 배제하였다.

2.4.1 LDA

앞선 파이프라인에 적절한 모델을 적용하여 문서를 클러스터링 한 후, 각 클러스터별로 LDA를 적용하여 토픽을 추출하였다. 클러스터링을 통해 유사한 문서들을 묶음으로써 토픽 모델링에서 문서를 대표할만한 토픽들이 더 일관되게 추출될 것으로 기대하였으며, 클러스터링을 한 경우와 그렇지 않은 경우의 결과를 비교해보며 클러스터링 방법론의 효과를 검증해보려고 하였다. 중간 보고서와 마찬가지로 LDA 모델의 하이퍼파라미터(토픽의 개수 등)를 튜닝할 때에는 topic coherence score(각 토픽을 구성하는 단어들 간의 의미적 일관성을 나타낸 지표)를 활용하였으며, 키워드 추출 시에는 Term Saliency(특정 단어가 토픽을 생성하는 데 기여한 정도와 등장 빈도를 복합 고려한 지표)를 사용하였다. 클러스터별로 LDA 모델을 따로따로 적용하기 때문에, 하이퍼파라미터 튜닝과 키워드 추출 또한 클러스터 단위로 수행하였다.

2.4.2 C-TF-IDF

앞선 2.4.1의 방법론은 문서의 일부에 해당하는 하나의 클러스터 내에서만 키워드를 추출하기 때문에, 전체 문서를 고려하지 못한다는 한계점이 존재하였다. 따라서, 이를 극복할 수 있는 새로운 document representation 방법론인 c-TF-IDF 을 키워드 추출에 적용해보았다.

c-TF-IDF[4] 란 기존 TF-IDF 기반의 representation 방법론으로서, 기존의 TF-IDF가 전체 문서 단위에서 각 단어의 빈도수를 측정하였던 것과 달리 이것은 특정 클러스터(카테고리, 클래스, 토픽) 단위에서 단어의 빈도수를 측정함으로써 각 토픽의 representation을 생성(추출)해낸다.

$$W_{t,d} = tf_{t,d} \cdot \log(\frac{N}{df_t}) \qquad W_{t,c} = tf_{t,c} \cdot \log(1 + \frac{A}{tf_t})$$

[Classic TF-IDF] **[cTF-IDF]**

c-TF-IDF 는 각 단어의 클러스터 별 존재 여부를 바탕으로 가중치를 부여해, 한 문서에서만 집중적으로 나타나는 키워드에 대해 높은 점수를 부여하여 키워드로 추출될 가능성을 더욱 높인다. 따라서 문서의 의미론적 관계를 더욱 잘 포착하여 정확한 키워드 추출이 가능하다. 또한 전체 문서 내

각 클러스터를 단위로 토픽을 표현하는데, 이때 각 토픽과 그와 관련된 토픽 단어들의 분포를 생성함으로써 각 클러스터에 내 단어들의 중요성을 포착할 수 있다.

구현은 앞선 파이프라인을 따라 문서에서 클러스터를 추출한 후, BERTopic 라이브러리[5]를 활용하였다. 이를 활용하여 각 클러스터 문서 별 c-TF-IDF representation 을 추출하였다. 이때 필요한 하이퍼파라미터에는 top_n_words(각 토픽별로 추출하고자 하는 단어의 개수), min_topic_size(하나의 토픽이 가져야 하는 최소 구성 단어의 개수) 등이 있었는데, 해당 값들의 튜닝에는 위 LDA 와 마찬가지로 topic coherence score를 활용하였다. 키워드 추출 시에는 c-TF-IDF score(여러 클러스터가 존재할 때, 어떤 단어가 특정 클러스터 내에서 얼마나 중요한 것인지를 나타내는 점수)를 기준으로 상위 5개의 단어를 클러스터별로 추출하였다. 궁극적으로는 이러한 과정을 통해 얻은 키워드 추출 결과를 앞선 LDA 의 결과들과 비교해보고자 하였다.

3. 결과와 해석

3.1 Baseline (LDA without Clustering)

Baseline 모델로써, 클러스터링을 하지 않고 전체 문서에 대해 LDA를 적용한 결과는 아래와 같다. Term saliency 기준 상위 30개의 단어를 추출한 결과이다.

love	face	nail	coat	price	light
buy	scent	color	work	great	skin
hair	soft	conditioner	use	foundation	smell
look	brush	oil	acne	34	dry
cream	polish	like	lotion	shampoo	product

Table 2. 클러스터링 없이, 전체문서 대상 LDA (baseline)에서 추출된 Top-30 words

Topic Coherence
0.4598

전반적으로 nail, color, skin, hair 등 beauty 리뷰와 관련된 키워드들이 잘 추출되었다. 그러나 EDA 결과 확인했던 curly, spray, shower 등 중심 토픽보다는 덜 등장하지만, '헤어 관련 리뷰'라는 세부 분야를 대표할 것으로 추정되는 단어들은 등장하지 않는 모습을 보였다.

3.2 LDA with Clustering

클러스터 번호	데이터 개수
1	18957
2	7550
3	2600
outlier	893

Table 3. HDBSCAN 클러스터링 결과

다음은 클러스터링을 선행한 뒤 각 클러스터 단위로 LDA를 적용한 결과이다. 클러스터링의 경우 위 표와 같이 Outlier를 제외하고 총 3개의 클러스터가 추출되었으며, 3개의 클러스터 각각에서 Term Saliency 기준 상위 5개의 단어를 추출하였다.

클러스터	**토픽 키워드**				
1	smell	skin	color	**soap**	wash
2	color	nail	coat	dry	top
3	color	**iron**	**scalp**	hair	dryer

Table 4. LDA with Clustering에서 추출한 토픽 키워드

클러스터 번호	**Topic Coherence**
1	0.4933
2	0.4285
3	0.4520
평균	**0.4580**

추출된 키워드를 분석해보았을 때 각 클러스터는 순서대로 피부, 네일, 헤어에 관한 리뷰들을 담고 있는 것으로 추정된다. 전체 문서를 대상으로 토픽 모델링을 수행한 결과보다 soap, iron, scalp 등 세부적인 키워드들이 추출된 모습을 보였다. 다만 color, dry(dryer) 등 중복되는 키워드들이 많아 클러스터 간 구분이 명확하게 되지 않는 모습을 보였다. 수행 대상이 되는 클러스터에 등장하는

단어만 고려하고, 다른 클러스터에 등장하는 단어를 고려하지 않는 LDA의 특징 때문인 것으로 보인다.

Topic coherence score는 클러스터링을 하지 않은 경우와 큰 차이를 보이지 않았다. Beauty 리뷰 전체를 고려하느냐 클러스터링 된 일부분을 고려하느냐의 차이만 있을 뿐, 토픽 모델링 방법론이 바뀌지 않았기에 큰 성능 변화가 없었던 것으로 파악된다.

3.3 c-TF-IDF

클러스터	토픽 키워드				
1	skin	use	product	like	get
2	nail	polish	coat	color	use
3	hair	use	product	shampoo	like

Table 5. c-TF-IDF에서 추출된 클러스터별 토픽 키워드

마지막으로 c-TF-IDF 을 활용하여 각 클러스터별 키워드를 추출해본 결과이다. 클러스터링 결과는 Table 1의 결과와 동일한 것을 사용하였다. 다음과 같은 하이퍼파라미터 탐색 과정을 통해, top_n_words=5, min_topic_size=10 으로 고정하고 실험을 진행하였다.

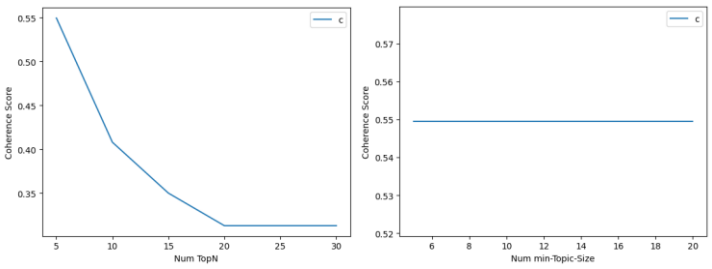

Figure 1. Hyperparameter Searching for c-TF-IDF

실험 결과, C-TF-IDF 기반으로 추출한 토픽은 LDA에 비해 클러스터 간에 좀 더 명확히 구분되는 모습을 보였다(피부 관련 / 네일 관련 / 헤어 관련). 또한, LDA에 비해 세부적인 의미보다는 각 클러스터를 대표할 만한 범용적인 키워드들이 추출되는 모습을 보였다. 그러나 use, product, like 등 클러스터 간에 중복되는 키워드들이 여전히 존재하고 LDA에 비해 단어의 다양성이 떨어지는 모습을 보였다. Table 6은 c-tf-idf, LDA를 사용해 뽑은 실제 클러스터링 결과들과 그로부터 추출된 토픽 키워드들이다.

reviewText	cluster	topic keyword (3.3. c-tf-idf)	topic keyword (3.2. LDA w/ clustering)
I've tried many facial cleansers which I was very disappointed in. This washed my makeup off and didn't cause breakouts or dry **skin**. This cleanser seems to help my acne and it smells wonderful. You only need a small amount of **product** and rub it on your face, it works even better if left on longer than 15 seconds.	0	['skin', 'product']	['smell', 'skin', 'wash']
Obagi is a permanent part of my daily routine. This is an important part to keeping my **skin** glowing and smooth.	0	['skin']	['skin']
OhLaLa!Oddly enough, for a person whose color loves are 'ick' greens and orange, I have become particularly fond of pink **nail polishes** - mostly vibrant pinks.OPI Charged Up Cherry is a luscious, vibrant, roaring, pink. I have worn it by itself which is an "oh! so perfect" look, and I have also used it as an **accent color** and in polka dots! It's a versatile color that has an excellent 2 **coat** formula!A new favorite!	1	['nail', 'polish', 'coat', 'color', 'use']	['color', 'nail', 'coat']
I've tried several withe **color** from different brands and konad white is exactly what i was looking for, love it... best white **nail polish** to do stamping.	1	['nail', 'polish', 'color']	['color', 'nail']
When we haven't had soft water, this **shampoo** and conditioner combo have worked really well to make our hair not feel **like** it's a waxy, greasy mess right after washing it. It's not a miracle **product**, but it makes a huge difference without stripping your hair too much **like** some clarifying **shampoos** too (they can make your hair feel **like** straw).	2	['hair', 'product', 'shampoo', 'like']	['hair']
OK the **shampoo** surprised me with how well it worked and then the conditioner surprised me too. Happily it worked great and made my **hair** very smooth. Glad I found these two **products** for my straight hair.	2	['hair', 'product', 'shampoo']	['hair']

Table 6. Main result(topic modeling)

Topic Modeling Method	Coherence Score
C-TF-IDF	**0.5495**
LDA w/o clustering (baseline)	0.4598
LDA w/ clustering	0.4580

Table 7. Coherence scores for comparison

reviewText	cluster
This is just a 10% benzoyl peroxide **cream**. It's just as good as any other 10% benzoyl peroxide and you can't beat the price.	0
I like this product but it make my **fingers red** when I leave it for long but it's ok really thanks	1
Suave Professionals, **conditioner**, humectant **moisture**, 28ozMy favorite suave! Just as good as the name brand stuff only much cheaper!	2

Table 8. Good cases of clustering without any overlapped words (c-tf-idf)

Table 7은 추출된 토픽들이 얼마나 일관된 주제를 가지는지를 나타내는 coherence score를 나타낸 것인데, 해당 표를 통해 동일한 조건에서 c-TF-IDF 방식을 채택한 경우가 LDA를 활용한 방식들(baseline, LDA with clustering)에 비해서 더 나음을 알 수 있었다. Table 6의 예시들과 핵심 단어들이 겹치지 않았지만 잘 클러스터되어 있는 것을 Table 8에서 확인할 수 있다. 예를 들어, Table 8에서 cluster 1로 분류되어 있는 문장의 경우, Table 6의 cluster 1 예시들처럼 'nail'이나 'color'라는 토픽 단어가 실제로 쓰이지 않았지만 'fingers'나 'red'라는 단어들을 바탕으로 맥락을 추론해 같은 클러스터에 묶였음을 볼 수 있다. 이는 dense embedding의 영향으로 보이며, 단어 간 의미론적 관계를 보다 잘 포착한 것이라 해석 가능하다.

4. 본 결과를 바탕으로 향후 가능한 추가 연구 개발 방향

2.3에 기술하였듯, HDBSCAN 클러스터링 모델의 hyperparameter 결정에 있어 절대적인 메트릭(DBCV)만을 기준 삼는 것은, 그 클러스터링 결과가 본 프로젝트에서 제시한 문제 상황의 해결에 적합하지 않았다. 따라서 DBCV 점수와 더불어 자의적인 판단 기준을 hyperparameter 선택에

고려하였다. 그러나 Table 3과 같이 outlier 로 분류된 리뷰의 수가 많고, 클러스터의 종류도 적을뿐만 아니라 그 클러스터 사이즈가 불균형하다는 문제점이 존재한다.

본 프로젝트에서는 outlier가 너무 많이 나오는 등의 문제가 클러스터링 모델 자체의 문제인지 확인해보기 위하여 다음과 같이 그 성능을 검증해보았다. 일반적인 클러스터링 태스크에 활용되는, 보다 명확히 내용이 구분이 될 수 있는 데이터로서, 뷰티, 호텔, 영화 카테고리의 리뷰가 섞인 데이터를 사용하여 다시 클러스터링을 진행해보았고 결과는 아래와 같았다.

카테고리	데이터 개수
영화	50000
호텔	29995
뷰티	20491

Table 9. 새로 사용한 데이터의 카테고리 별 개수

클러스터 번호	데이터 개수	영화 리뷰 개수	호텔 리뷰 개수	뷰티 리뷰 개수
0	49992	49959	27	6
1	20548	29	39	20480
2	29862	2	29857	3
Outlier	84			

Table 10. 구현한 파이프라인을 이용한 클러스터링 결과

실험 결과, 문서를 구분짓는 명확한 특징이 있다면 본 팀이 구현한 파이프라인이 상당히 우수한 클러스터링 성능을 보여주었다. 이를 통해 파이프라인의 문서 임베딩 과정에서 각 문서를 분류할 수 있는 효과적인 feature를 만들어내지 못하고 있다고 판단하였다. 또한 sentenceBERT의 경우 구어체에 적합하게 학습된 모델이 아니기에 임베딩 과정에서 특징을 명확하게 추출해내지 못했을 가능성도 있다고 판단된다. 따라서 리뷰 문서의 내용만으로 특징적인 feature를 추출해낼 수 있는 임베딩 기법에 대한 추가 연구가 필요하다고 생각한다.

5. Code

5.1 파일 구조

./

├── Final.ipynb ———————————————————— All codes for this report

├── cache ———————————————————————— Directory for caching HDBSCAN

├── result ———————————————————————— Directory to store experiment

└── dataset

  ├── beauty.csv ———————————————— Main dataset with 30,000 beauty reviews

  └── whole.csv ———————————————— 100,000 reviews of movie, beauty, hotel

5.2 실험 환경

- Google Colab Pro Plus, T4 GPU
- 시스템 RAM 12.7 GB

6. Reference

[1] Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[2] McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

[3] https://hdbscan.readthedocs.io/en/latest/faq.html#q-i-am-not-getting-the-claimed-performance-why-not

[4] Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

[5] https://maartengr.github.io/BERTopic/api/ctfidf.html