[방법론 세미나]

Document Clustering

- BERTopic -

2024-01-11

Seoul National University
Technology Intelligence Lab

박상현

목차

1. **Introduction**
2. **Framework**
3. **Experiment**

INTRODUCTION

Background

- To uncover common themes and the underlying narrative in text, topic models have proven to be a powerful unsupervised tool.

  - Conventional models, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and Non-Negative Matrix Factorization (NMF) (Févotte and Idier, 2011), describe a document as a bag-of-words and model each document as a mixture of latent topics.

- One limitation of these models is that through **bag-of-words representations**, they **disregard semantic relationships among words.**

- As an answer to this issue**, text embedding techniques** have rapidly **become popular** in the natural language processing field.

  - Although embedding techniques have been used for a variety of tasks, ranging from classification to neural search engines, researchers have started to adopt these powerful contextual representations for topic modeling.

  * Similarly, Top2Vec leverages Doc2Vec’s word- and document representations to learn jointly embedded topic, document, and word vectors (Angelov, 2020; Le and Mikolov, 2014).

INTRODUCTION

Limitations of Previous method

- The aforementioned topic modeling techniques assume that words in close proximity to a cluster's centroid are most representative of that cluster, and thereby a topic.

  - In practice, however, **a cluster will not always lie within a sphere around a cluster centroid.**

- As such, the assumption cannot hold for every cluster of documents, and the representation of those clusters, and thereby the topic might be misleading. Although (Sia et al., 2020) attempts to overcome this issue by re-ranking topic words based on their frequency in a cluster, the initial candidates are still generated from a centroid-based perspective.

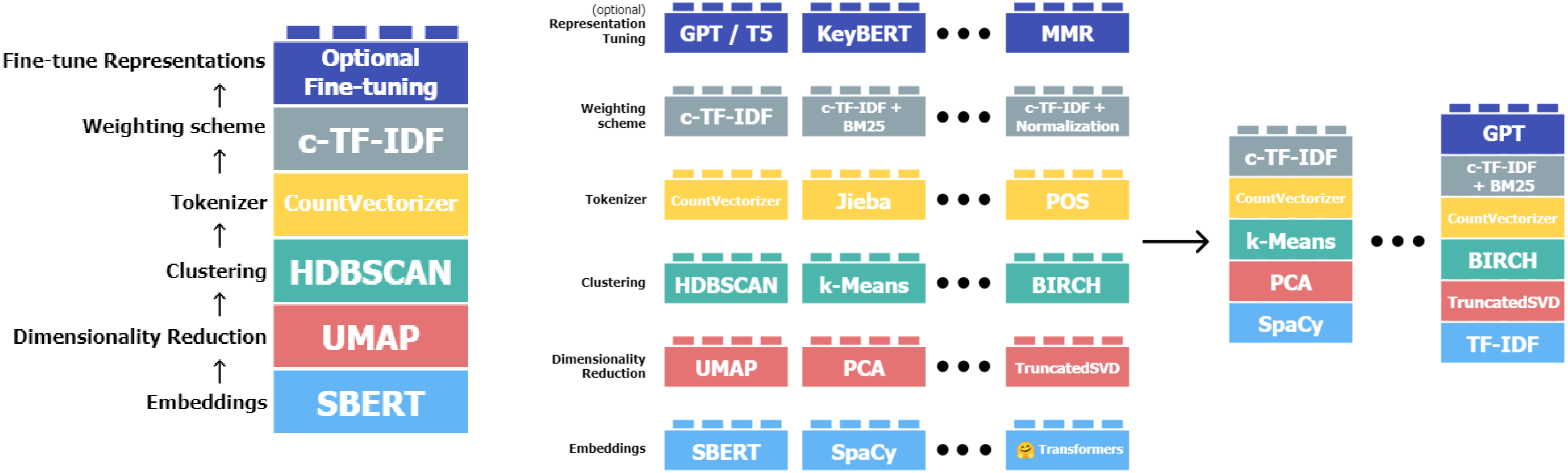
- In this paper, we introduce BERTopic, a topic model that leverages clustering techniques and a class-based variation of TF-IDF to generate coherent topic representations.

FRAMEWORK

Module

- BERTopic can be viewed as a sequence of steps to create its topic representations. There are five steps to this process.

  - Although these steps are the default, there is some modularity to BERTopic.


Fine-tune Representations ↑ Optional Fine-tuning
Weighting scheme ↑ c-TF-IDF
Tokenizer ↑ CountVectorizer
Clustering ↑ HDBSCAN
Dimensionality Reduction ↑ UMAP
Embeddings SBERT

(optional) Representation Tuning	GPT / T5	KeyBERT	● ● ●	MMR
Weighting scheme	c-TF-IDF	c-TF-IDF + BM25	● ● ●	c-TF-IDF + Normalization
Tokenizer	CountVectorizer	Jieba	● ● ●	POS
Clustering	HDBSCAN	k-Means	● ● ●	BIRCH
Dimensionality Reduction	UMAP	PCA	● ● ●	TruncatedSVD
Embeddings	SBERT	SpaCy	● ● ●	🤗 Transformers

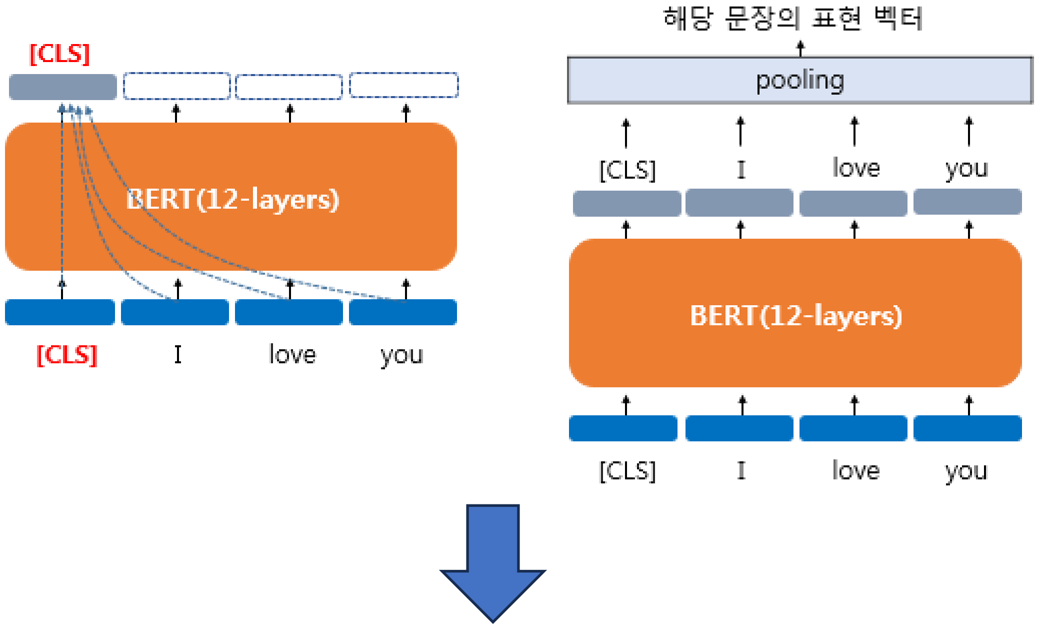
FRAMEWORK

1. Embedding

- BERTopic starts with transforming our input documents into numerical representations.

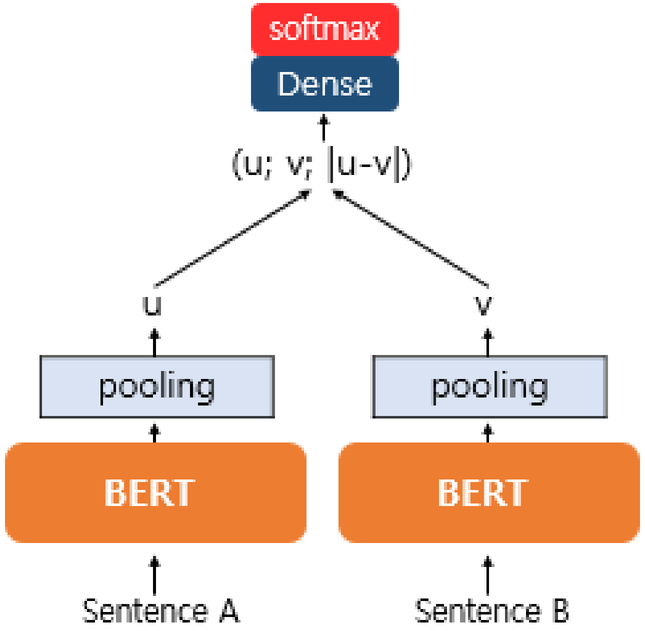
  - Although there are many methods for doing so the default in BERTopic is sentence-transformers.

BERT based Sentence Embedding



BAD

$$h = (u; v; |u - v|)$$



FRAMEWORK

2. Dimensionally Reduction

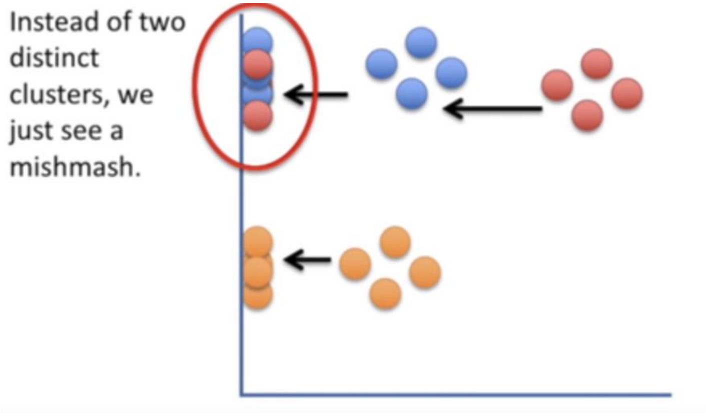
- After having created our numerical representations of the documents we have to reduce the dimensionality of these representations.

  - There are great approaches that can reduce dimensionality, such as PCA, but as a default UMAP is selected in BERTopic.

- UMAP : It is a technique that can keep some of a dataset's local and global structure when reducing its dimensionality.

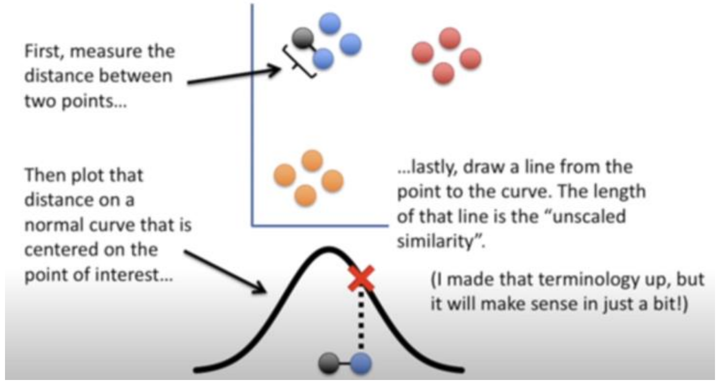
  - This structure is important to keep as it contains the information necessary to create clusters of semantically similar documents.

PCA



Instead of two distinct clusters, we just see a mishmash.

T-SNE



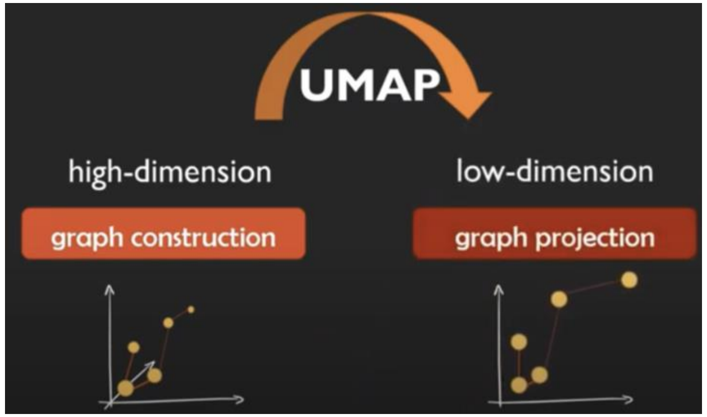
First, measure the distance between two points...

Then plot that distance on a normal curve that is centered on the point of interest...

...lastly, draw a line from the point to the curve. The length of that line is the "unscaled similarity".

(I made that terminology up, but it will make sense in just a bit!)

UMAP



FRAMEWORK

3. Clustering

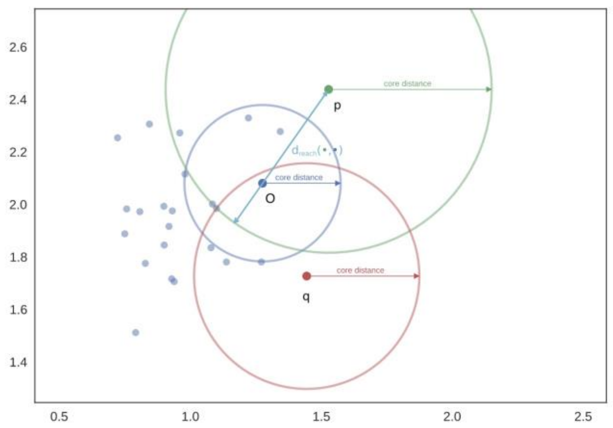
- After reducing the dimensionality of our input embeddings, we need to cluster them into groups of similar embeddings to extract our topics.

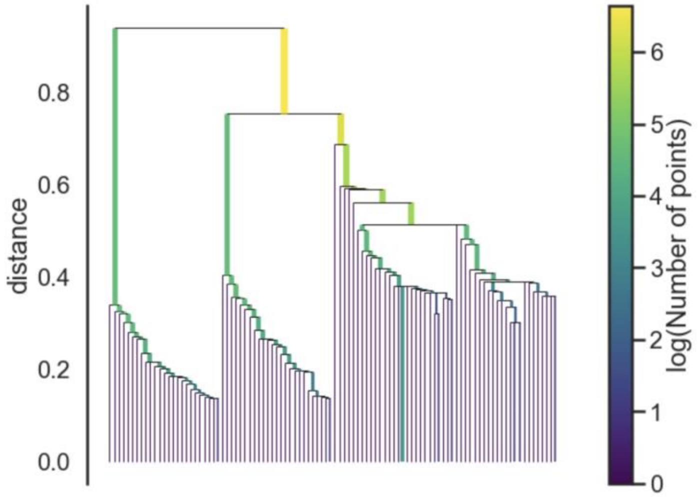
- For that, we leverage a density-based clustering technique, HDBSCAN.

  - It is an extension of DBSCAN that finds clusters of varying densities by converting DBSCAN into a hierarchical clustering algorithm.

  - It can find clusters of different shapes and has the nice feature of identifying outliers where possible.

  - Moreover, (Allaoui et al., 2020) demonstrated that reducing high dimensional embeddings with UMAP can improve the performance of well-known clustering algorithms, such as k-Means and HDBSCAN, both in terms of clustering accuracy and time





FRAMEWORK

4. Vectorizers

- In topic modeling, the quality of the topic representations is key for interpreting the topics, communicating results, and understanding patterns.

  - In practice, there is not one correct way of creating topic representations.

- One often underestimated component of BERTopic is the CountVectorizer and c-TF-IDF calculation.

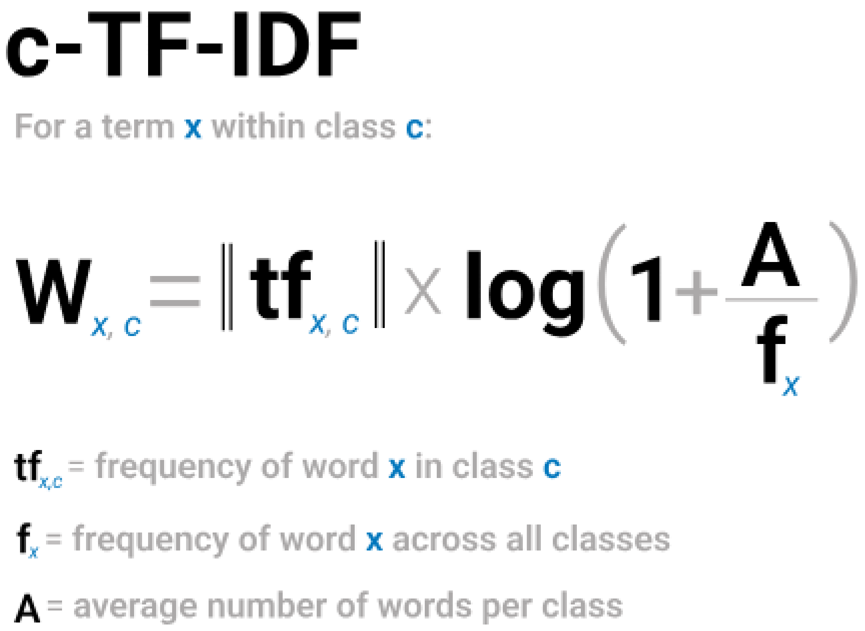
  - Together, they are responsible for creating the topic representations and luckily can be quite flexible in parameter tuning.

  - ngram_range, stop_words, min_df, max_features, tokenizer.

FRAMEWORK

5. c-TF-IDF

- In BERTopic, we can model this behavior by leveraging the c-TF-IDF representations of topics. Here, we assume that the temporal nature of topics should not influence the creation of global topics.



c-TF-IDF

For a term **x** within class **c**:

$$W_{x,c} = \|\mathbf{tf}_{x,c}\| \times \log\left(1 + \frac{A}{f_x}\right)$$

$\mathbf{tf}_{x,c}$ = frequency of word **x** in class **c**

f_x = frequency of word **x** across all classes

A = average number of words per class

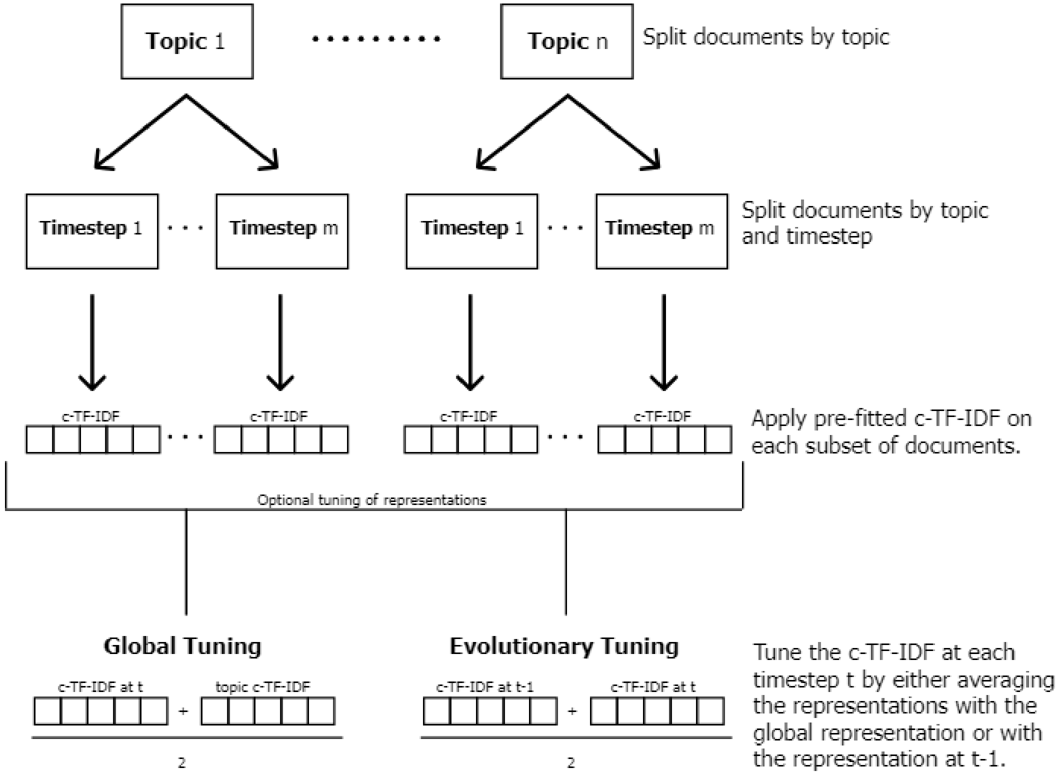
$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

$$c - TF - IDF_i = \frac{t_i}{w_i} \times \log \frac{m}{\sum_j^n t_j}$$

FRAMEWORK

Variations : Dynamic Topic modelling

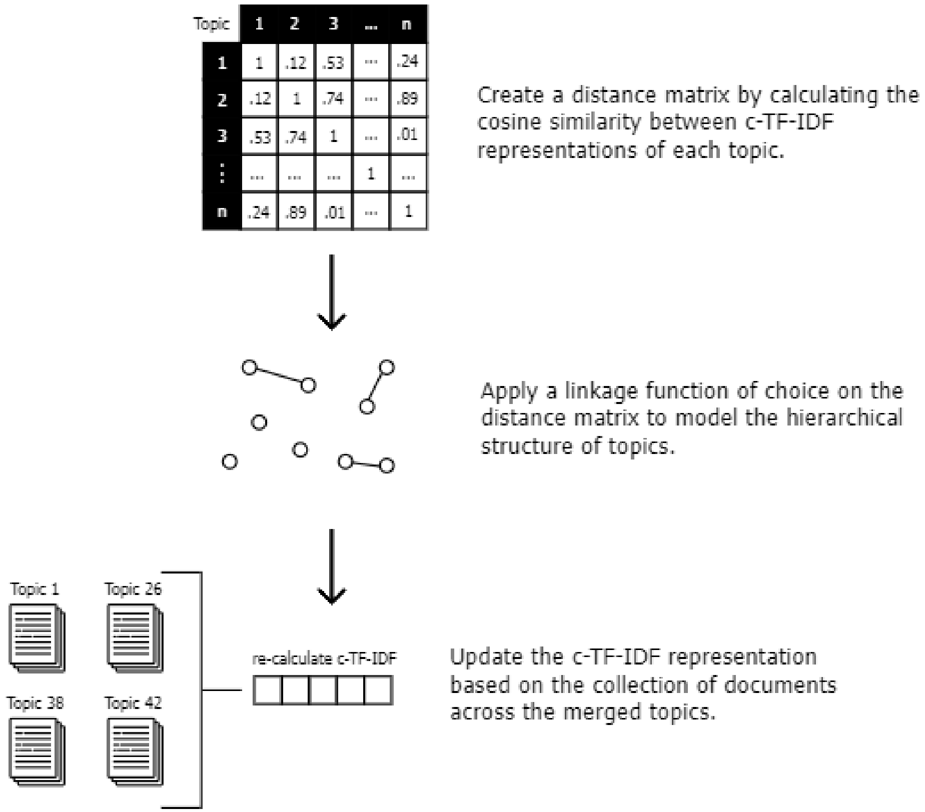
- In BERTopic, we assume that the temporal nature of topics should not influence the creation of global topics.
  - A topic representation at timestep t can be fine-tuned globally by averaging its c-TF-IDF representation with that of the **global representation**.
  - This allows each topic representation to move slightly towards the global representation whilst still keeping some of its specific words.
- A topic representation at timestep t can be fine-tuned **evolutionary** by averaging its c-TF-IDF representation with that of the c-TF-IDF representation at timestep t-1. This is done for each topic representation allowing for the representations to evolve over time.


Topic 1
Topic n
Split documents by topic
Timestep 1
Timestep m
Timestep 1
Timestep m
Split documents by topic and timestep
c-TF-IDF
c-TF-IDF
c-TF-IDF
c-TF-IDF
Apply pre-fitted c-TF-IDF on each subset of documents.
Optional tuning of representations
Global Tuning
c-TF-IDF at t
topic c-TF-IDF
2
Evolutionary Tuning
c-TF-IDF at t-1
c-TF-IDF at t
2
Tune the c-TF-IDF at each timestep t by either averaging the representations with the global representation or with the representation at t-1.

FRAMEWORK

Variations : Hierarchical Topic Modelling

- In BERTopic, we can approximate this potential hierarchy by making use of our topic-term matrix (c-TF-IDF matrix).
  - This matrix contains information about the importance of every word in every topic and makes for a nice numerical representation of our topics.
  - The smaller the distance between two c-TF-IDF representations, the more similar we assume they are

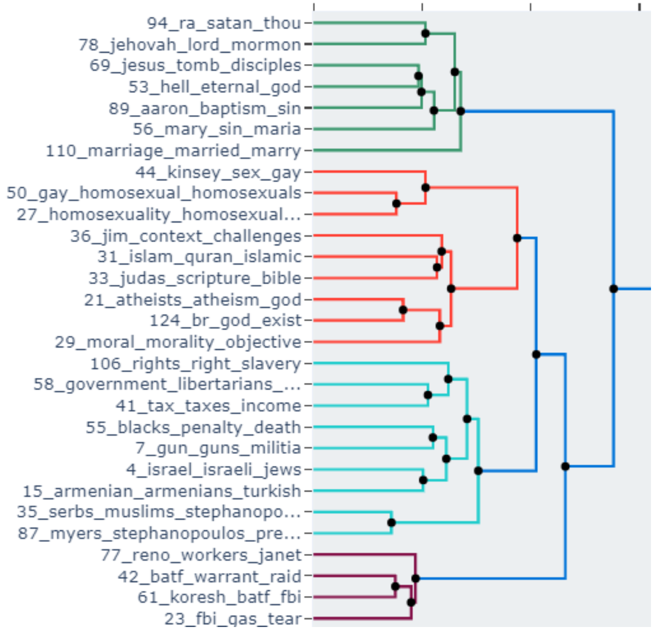


Topic	1	2	3	...	n
1	1	.12	.53	...	.24
2	.12	1	.74	...	.89
3	.53	.74	1	...	.01
⋮	...	...	...	1	...
n	.24	.89	.01	...	1

Create a distance matrix by calculating the cosine similarity between c-TF-IDF representations of each topic.

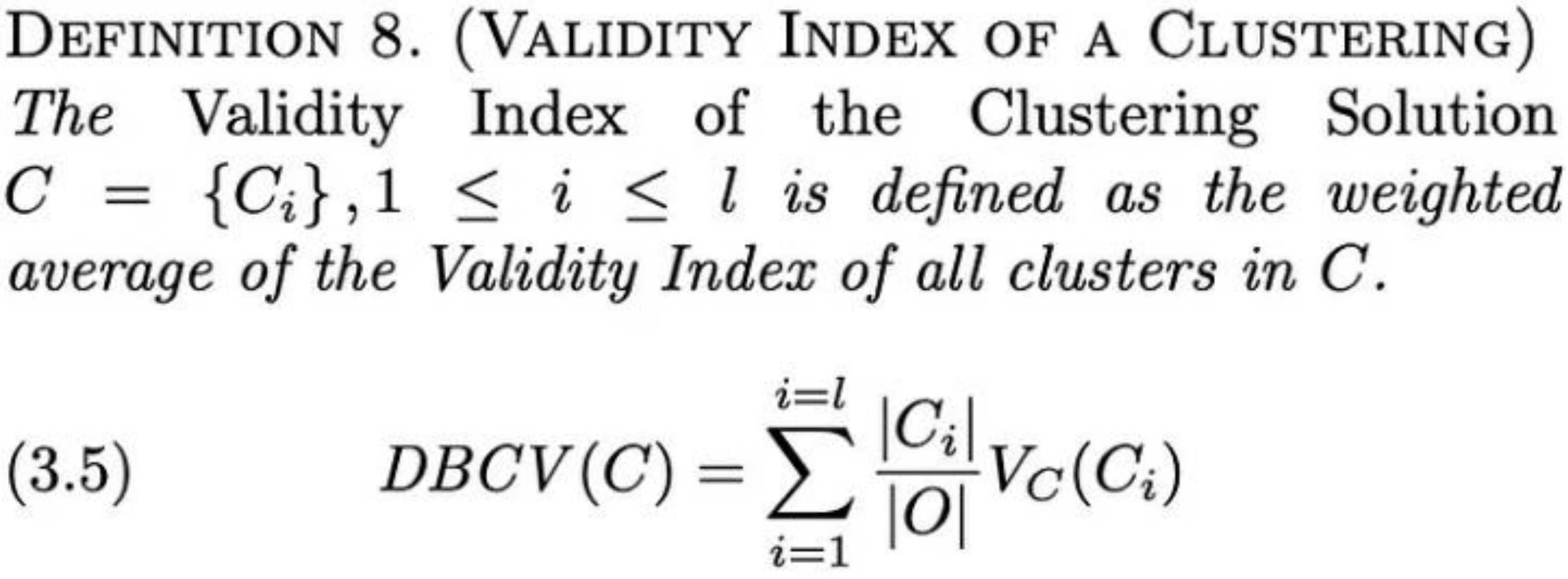
Apply a linkage function of choice on the distance matrix to model the hierarchical structure of topics.

Update the c-TF-IDF representation based on the collection of documents across the merged topics.



EXPERIMENT

HDBSCAN 성능 평가



DEFINITION 8. (VALIDITY INDEX OF A CLUSTERING) *The* Validity Index of the Clustering Solution $C = \{C_i\}, 1 \leq i \leq l$ *is defined as the weighted average of the Validity Index of all clusters in* C.

$$DBCV(C) = \sum_{i=1}^{i=l} \frac{|C_i|}{|O|} V_C(C_i) \tag{3.5}$$

EXPERIMENT

~~HDBSCAN 조정(Clustering method – EOM)~~

Cluster_selection_method

- EOM(Excess of Mass) : 클러스터 트리에서 가장 큰 클러스터들을 선택

- Leaf(Leaf) : 클러스터 트리에서 가장 아래쪽에 위치한 노드들을 선택

▪ 결과

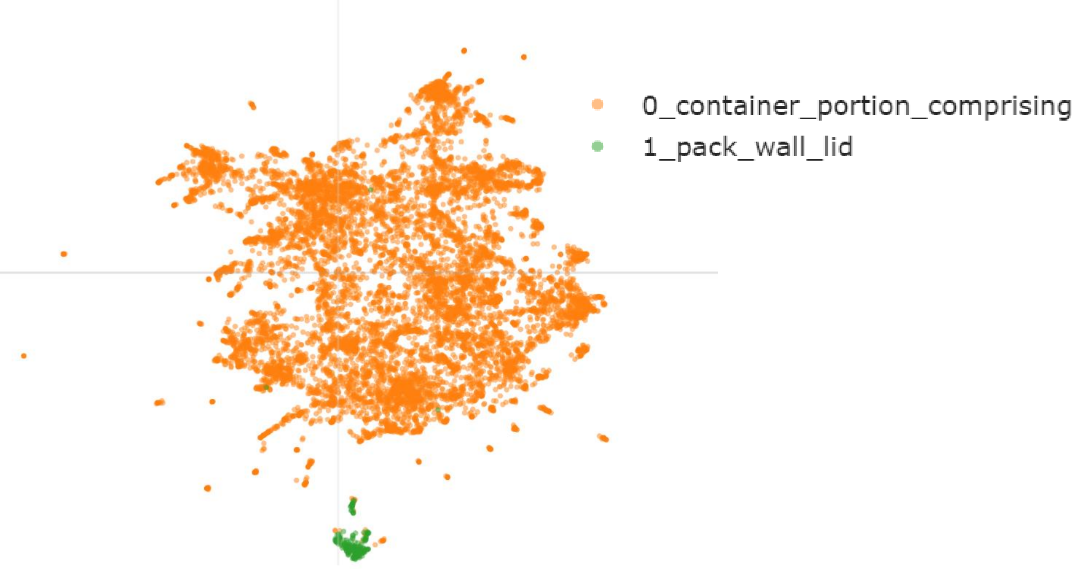
최소 클러스터 사이즈 : 30

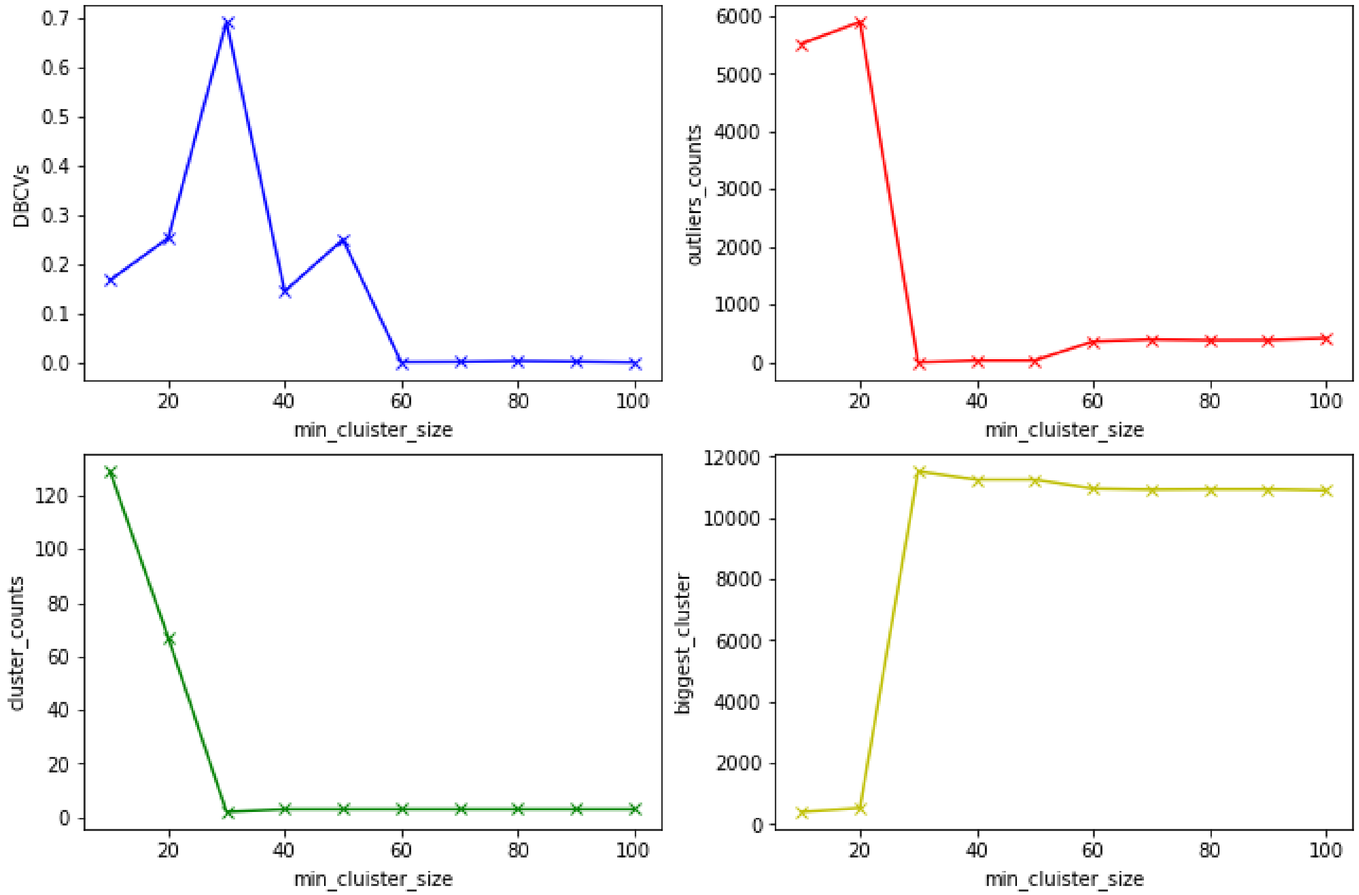
최대 DBCVs : 0.693

아웃라이어 비율 : 0.0

클러스터 수 : 2

최대 클러스터의 전체 비율 : 0.997





EXPERIMENT

HDBSCAN 조정(Clustering method – leaf)

Cluster_selection_method

- EOM(Excess of Mass) : 클러스터 트리에서 가장 큰 클러스터들을 선택

- Leaf(Leaf) : 클러스터 트리에서 가장 아래쪽에 위치한 노드들을 선택

- **결과**

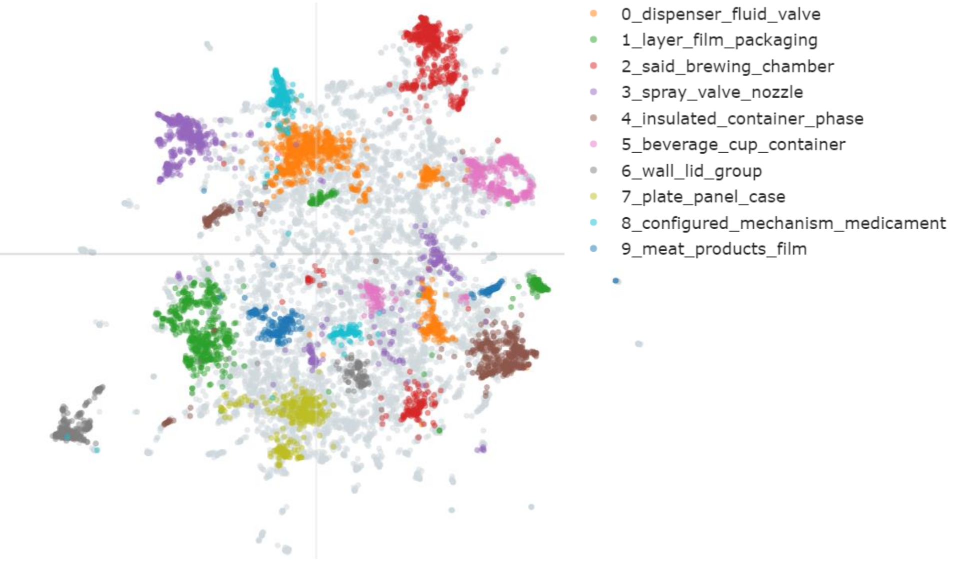
최소 클러스터 사이즈 : 50

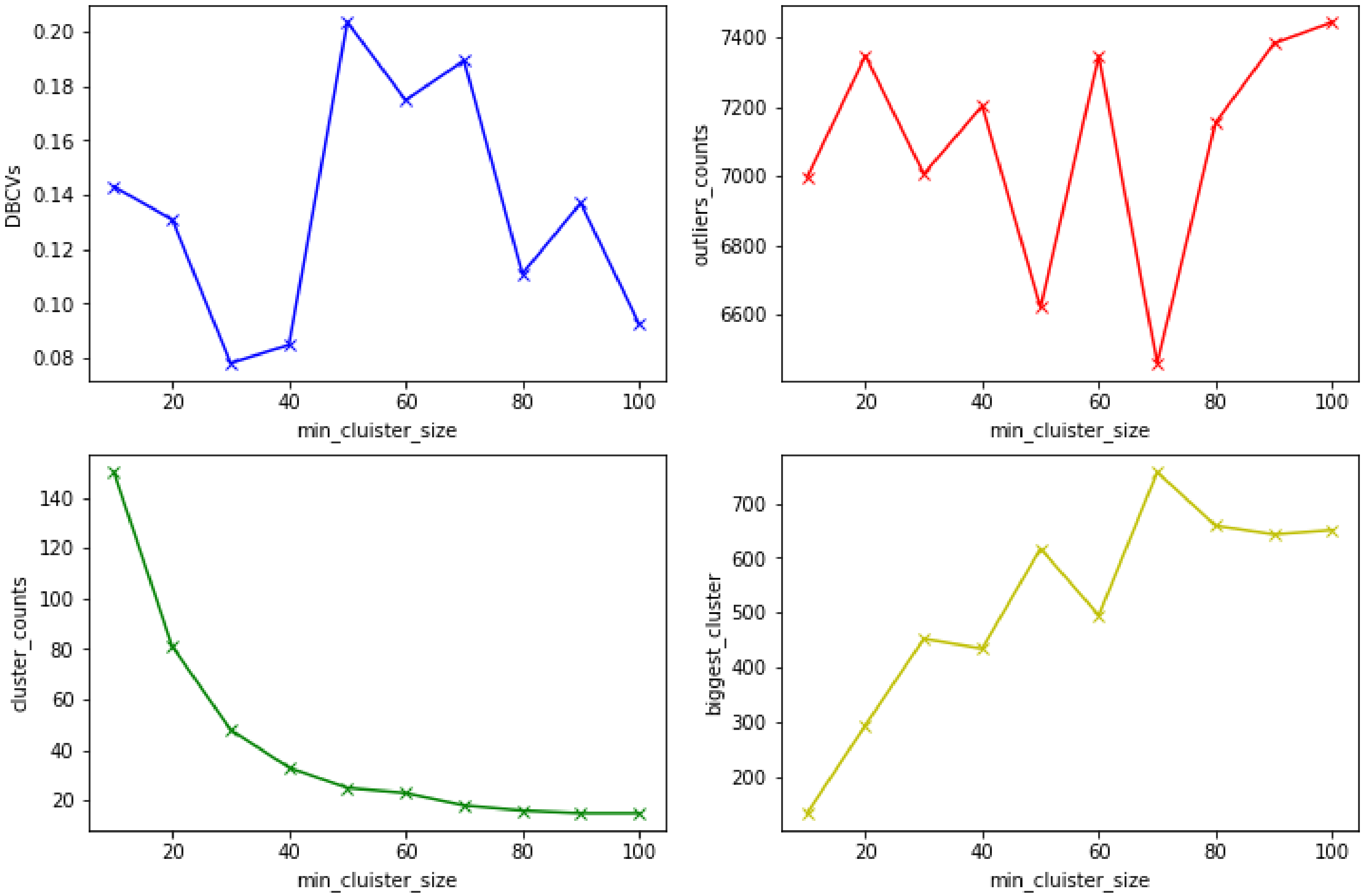
최대 DBCVs : 0.203

아웃라이어 비율 : 0.573

클러스터 수 : 25

최대 클러스터의 전체 비율 : 0.053





EXPERIMENT

~~HDBSCAN 조정(min_sample = 0 → 5)~~

min_sample

핵심 샘플을 결정하기 위한 최소 샘플 수로 클러스터의 밀도를 경정하여 높은 값은 더 높은 밀도의 클러스터를 생성

- **결과**

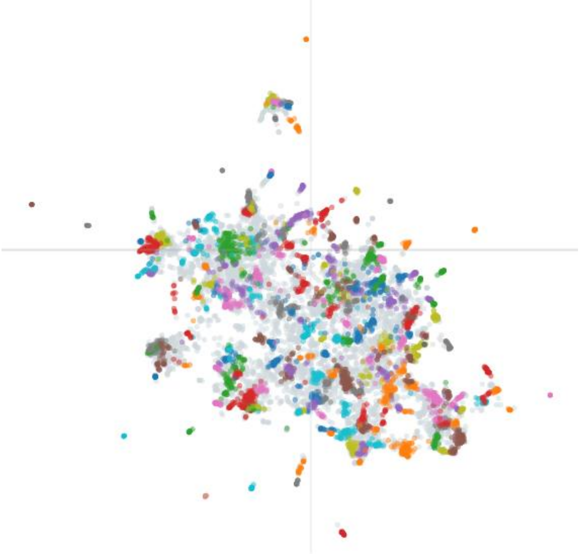
최소 클러스터 사이즈 : 10

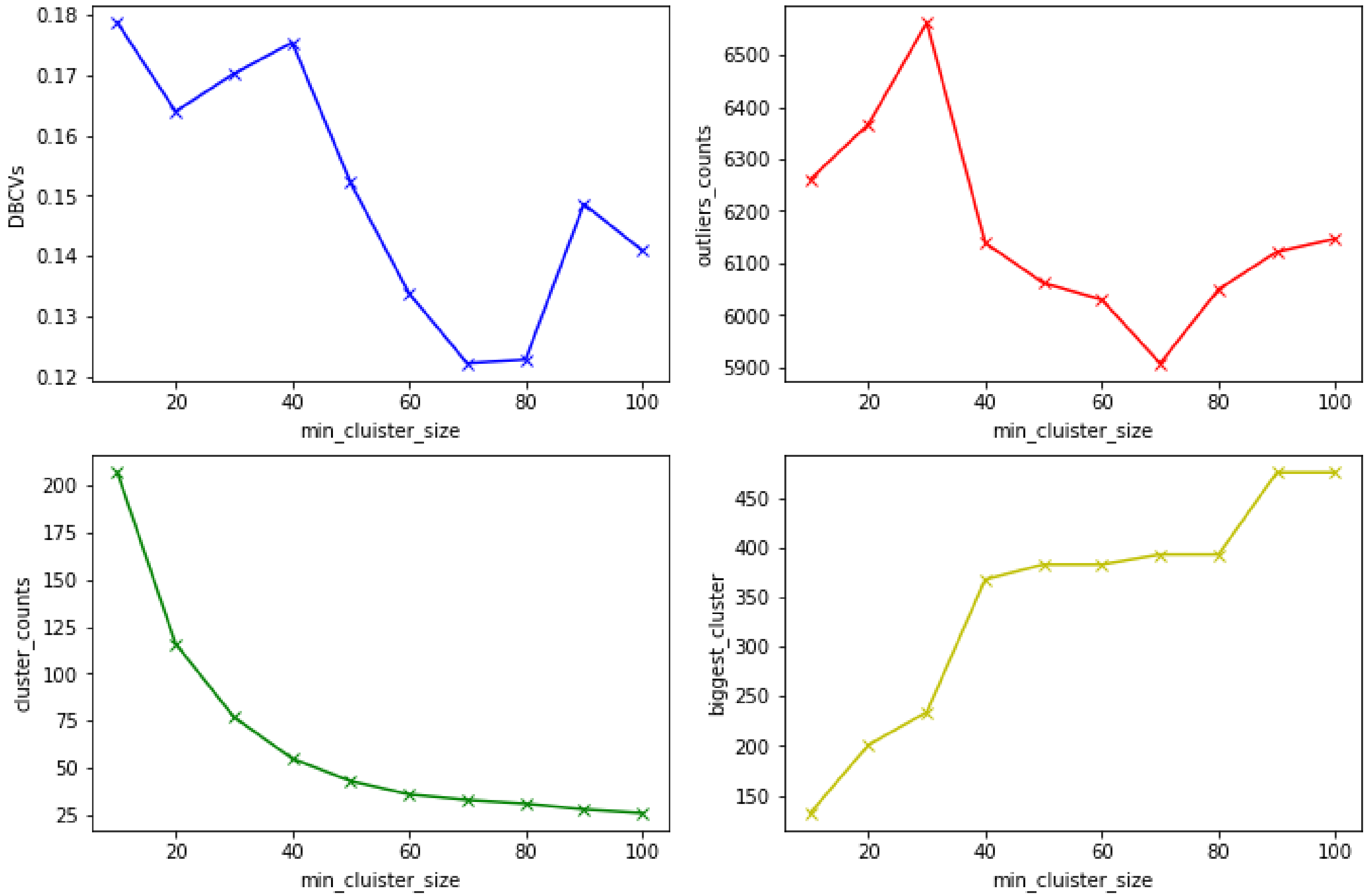
최대 DBCVs : 0.179

아웃라이어 비율 : 0.542

클러스터 수 : 207

최대 클러스터의 전체 비율 : 0.011





EXPERIMENT

HDBSCAN 조정(min_sample = 0 → 10)

min_sample

핵심 샘플을 결정하기 위한 최소 샘플 수로 클러스터의 밀도를 경정하여 높은 값은 더 높은 밀도의 클러스터를 생성

- **결과**

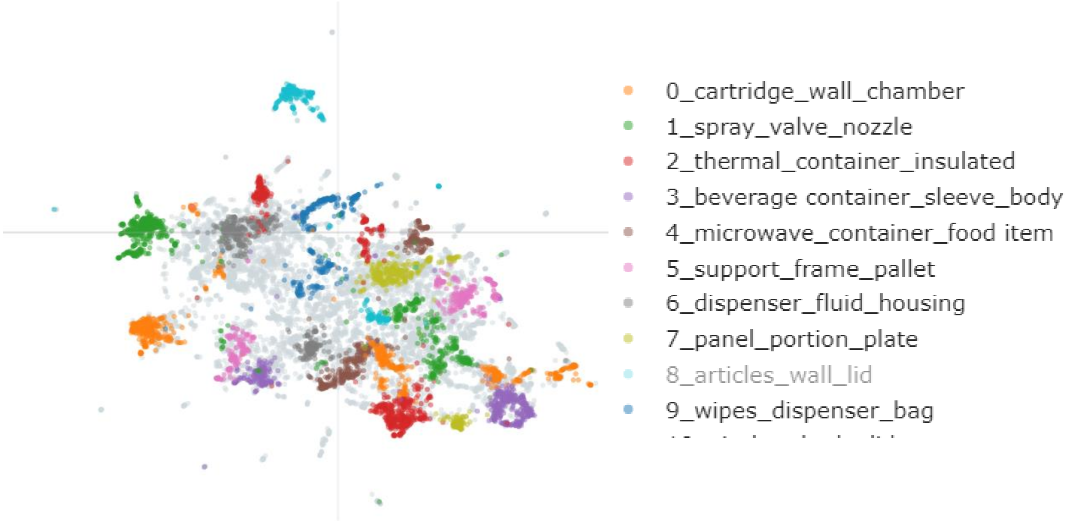
최소 클러스터 사이즈 : 100

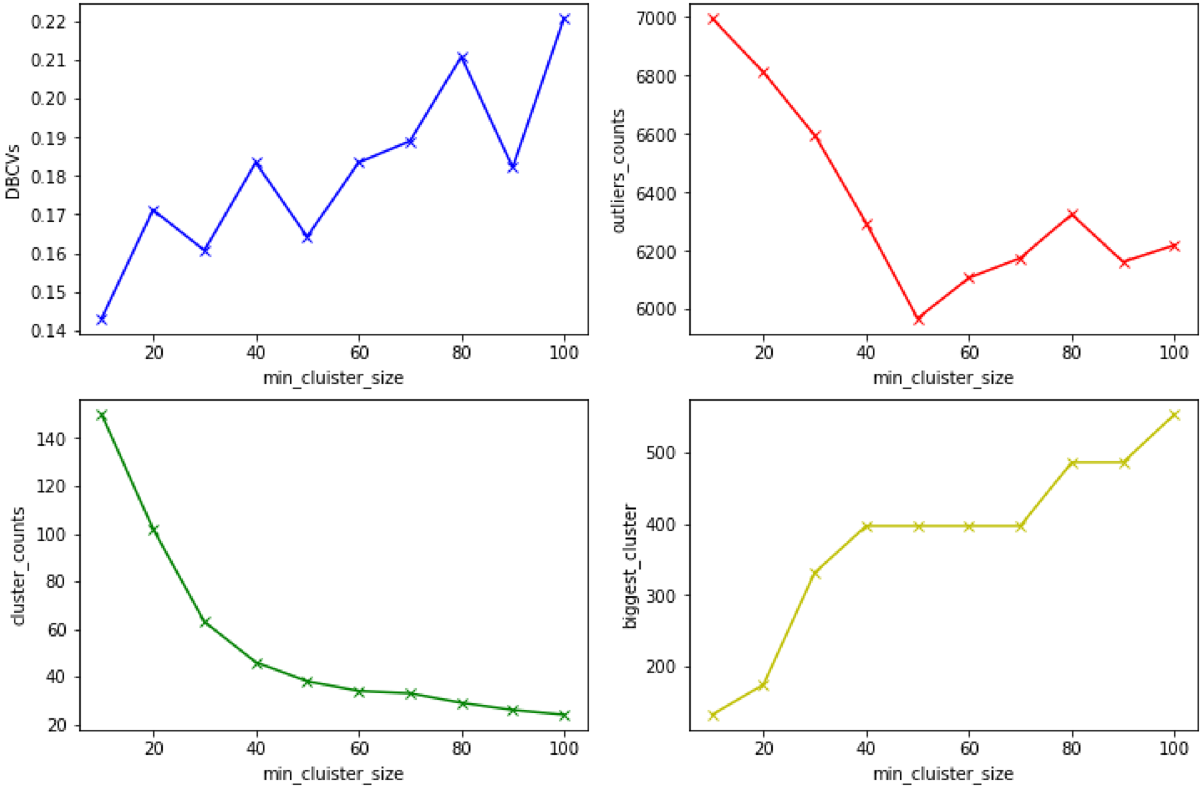
최대 DBCVs : 0.221

아웃라이어 비율 : 0.538

클러스터 수 : 24

최대 클러스터의 전체 비율 : 0.048





EXPERIMENT

~~UMAP 조정(n_neighbor = 15 → 50)~~

n_neighbors

UMAP 알고리듬에서 각 데이터 포인트를 고려할 때 주변을 탐색하는 이웃의 수를 정의하여, 값이 크면 글로벌 한 구조를 포착

▪ 결과

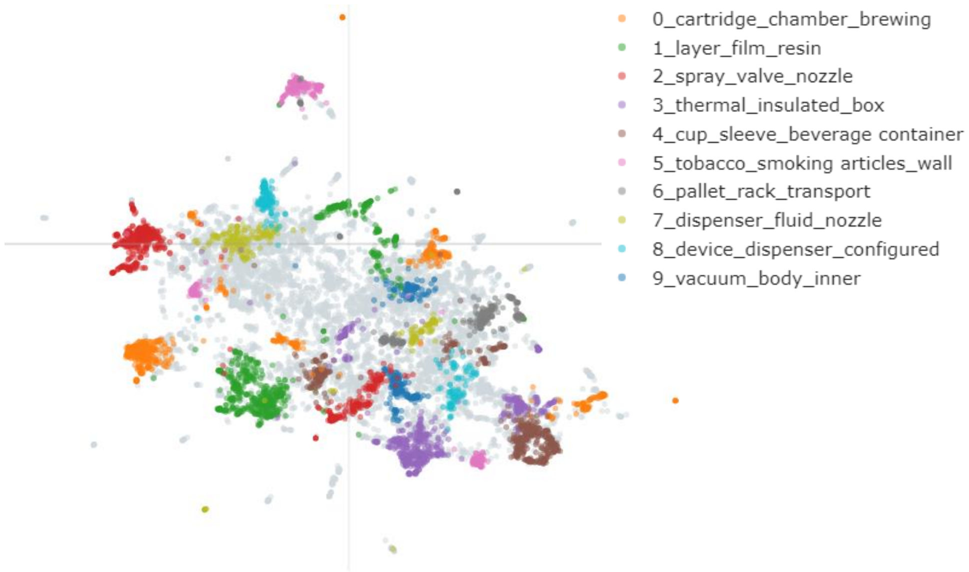
최소 클러스터 사이즈 : 70

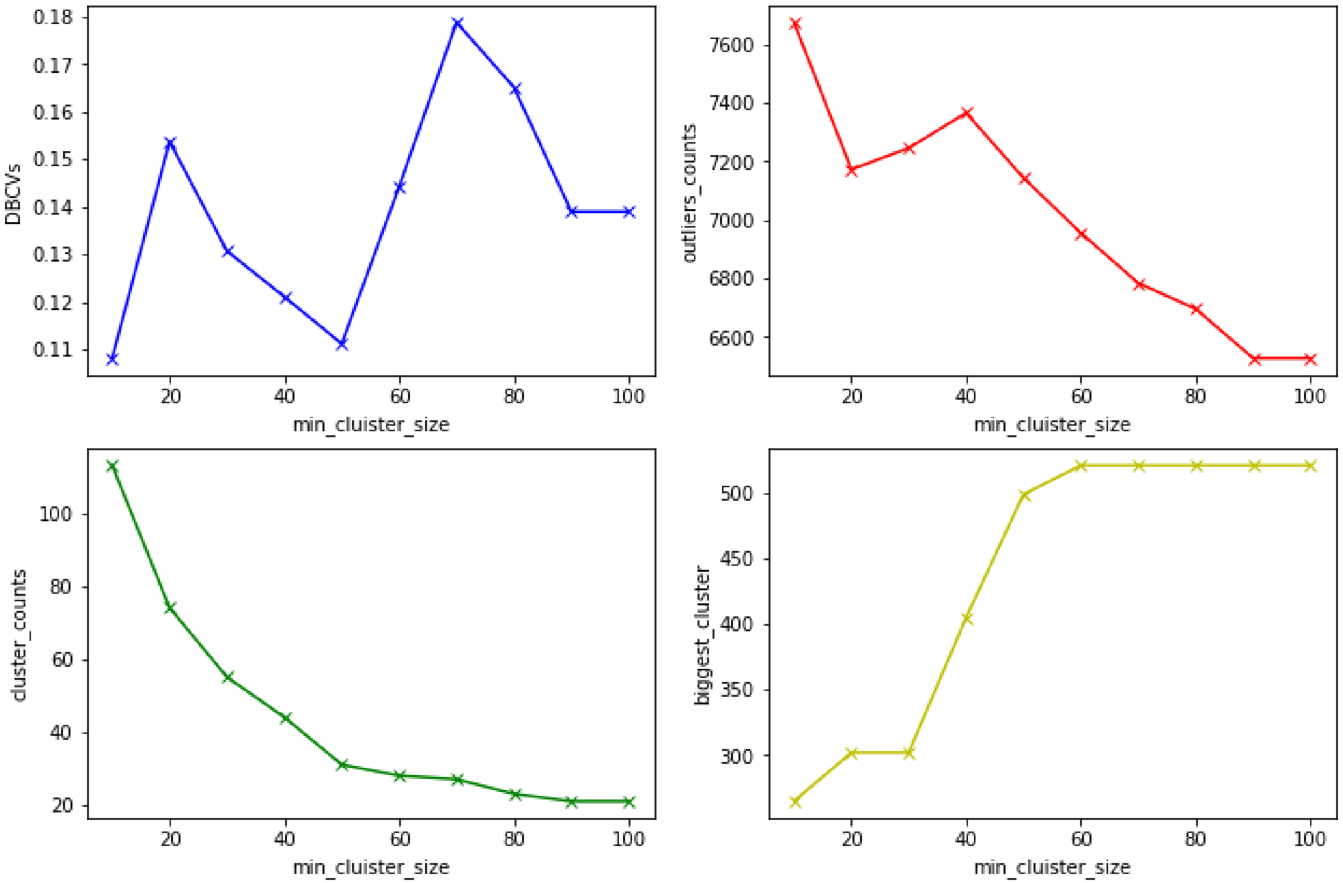
최대 DBCVs : 0.179

아웃라이어 비율 : 0.587

클러스터 수 : 27

최대 클러스터의 전체 비율 : 0.045





EXPERIMENT

~~UMAP 조정(n_components = 5 → 15)~~

- **결과**

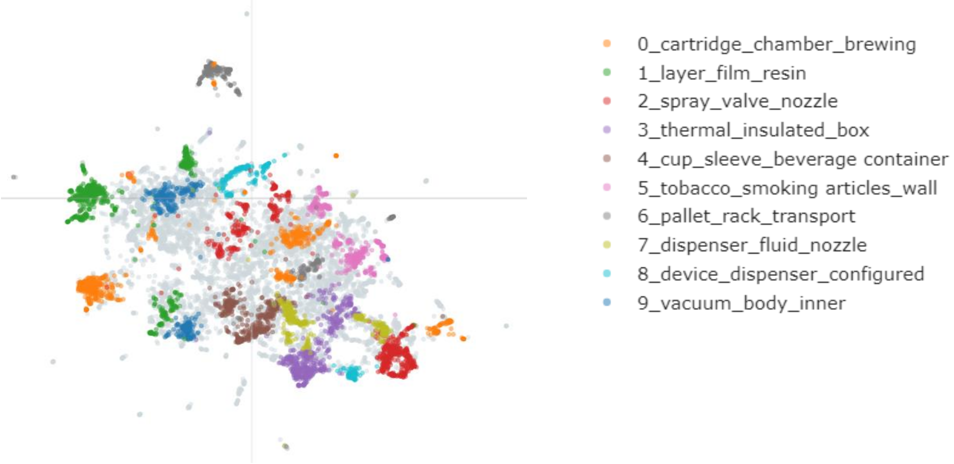
최소 클러스터 사이즈 : 100

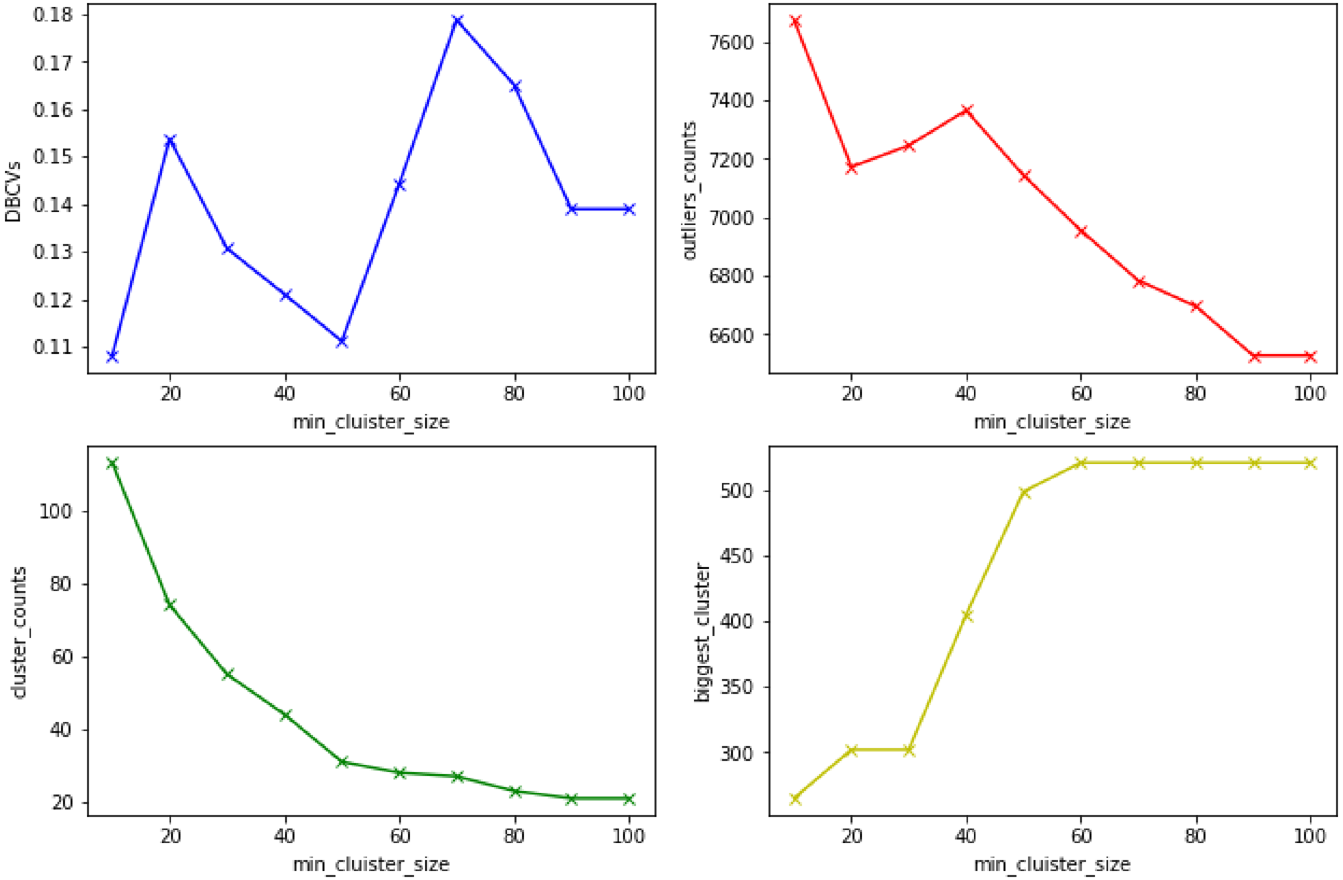
최대 DBCVs : 0.208

아웃라이어 비율 : 0.557

클러스터 수 : 24

최대 클러스터의 전체 비율 : 0.049





EXPERIMENT

~~Embedding model 교체(bge-large-en-v1.5)~~

Embeeding model

- PatentsBERTa : S-BERT를 특허 청구항으로 튜닝한 모델
- bge-large-en-v1.5 : MTEB 리더보드에서 SOTA성능을 보이는 모델

▪ 결과

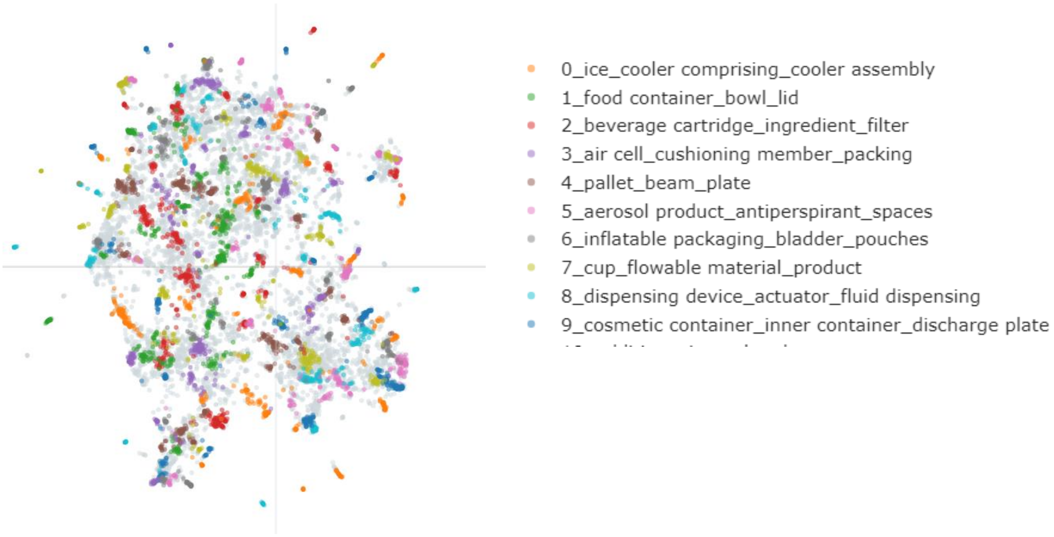
최소 클러스터 사이즈 : 10

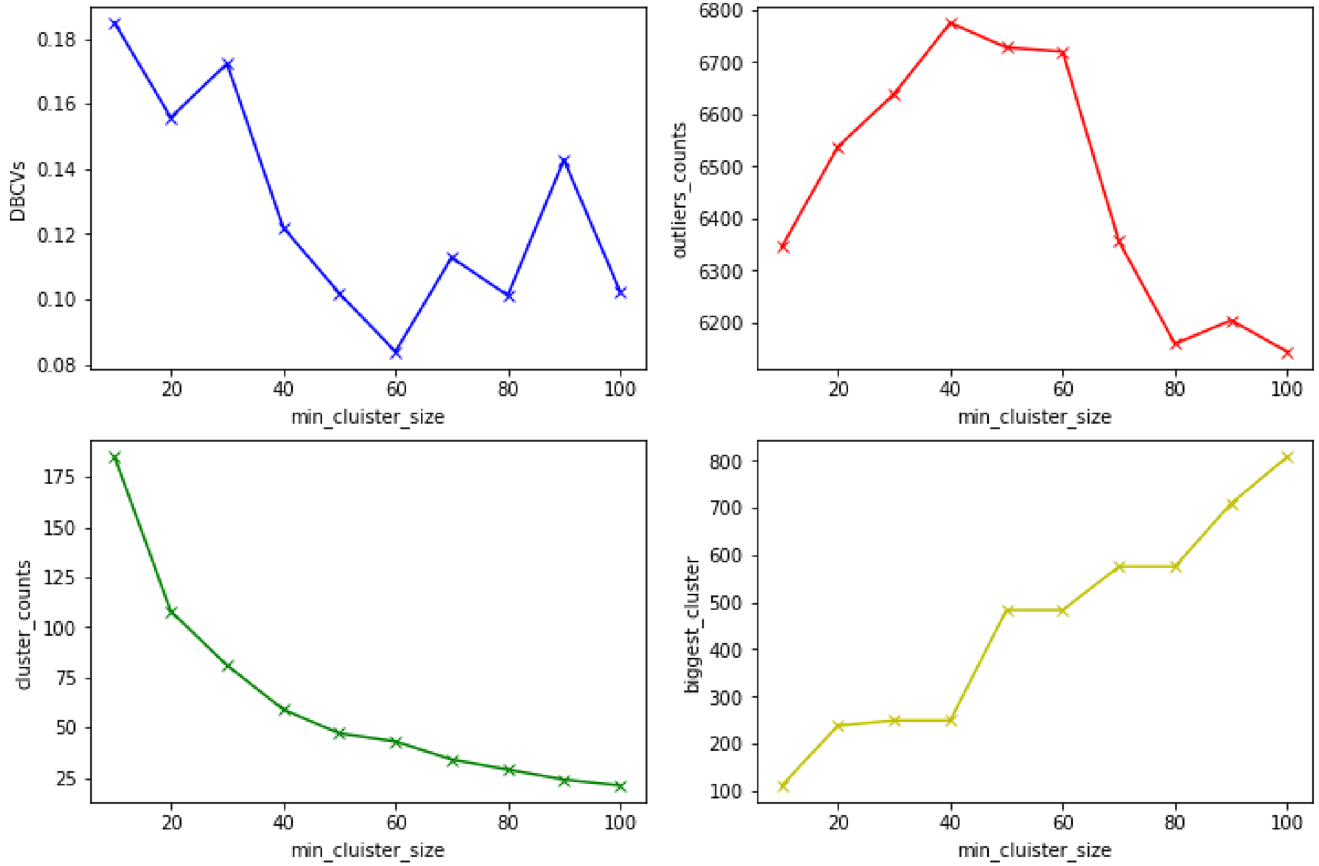
최대 DBCVs : 0.185

아웃라이어 비율 : 0.549

클러스터 수 : 185

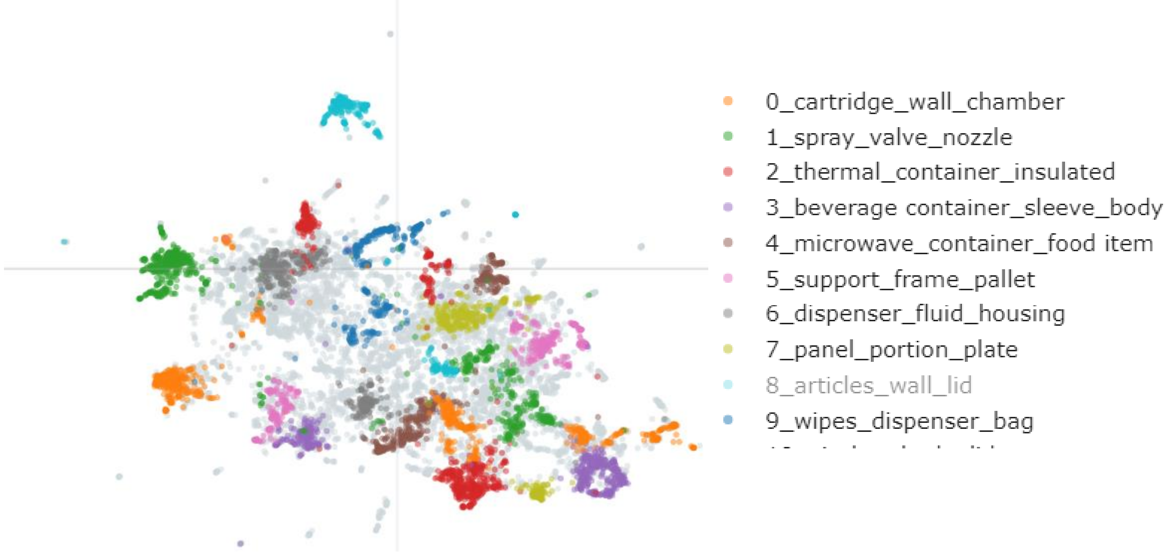
최대 클러스터의 전체 비율 : 0.01

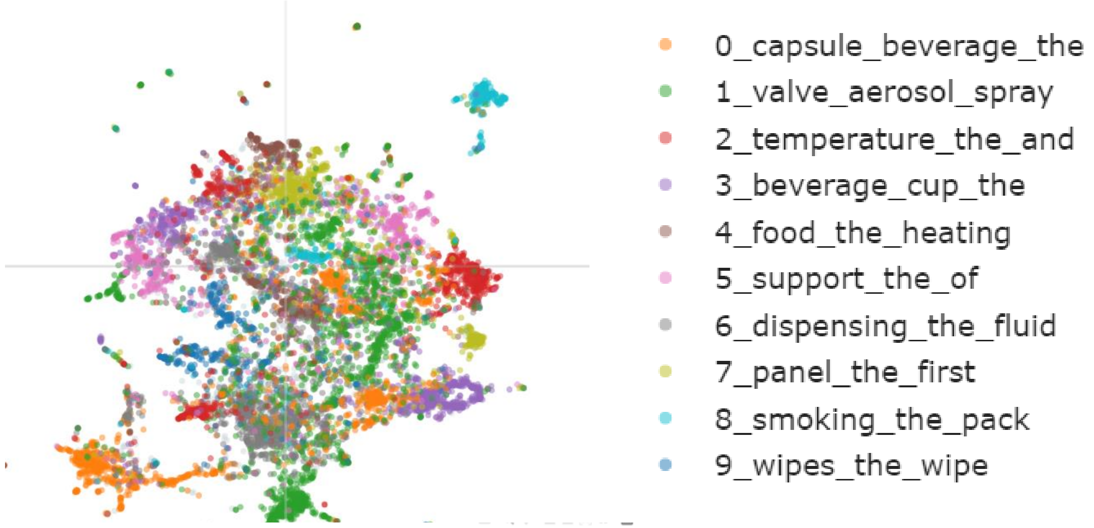




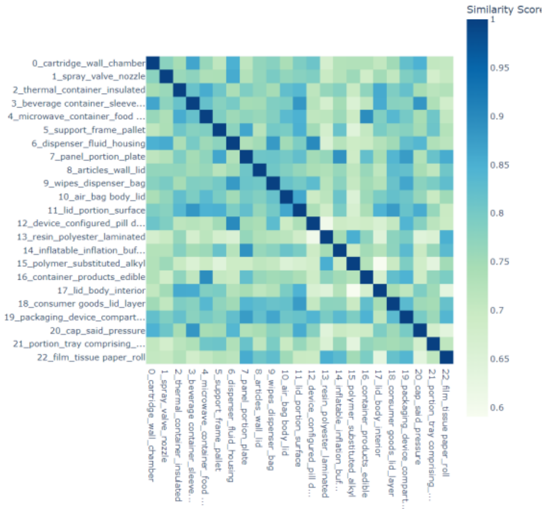
EXPERIMENT

~~Outlier reduction 적용~~

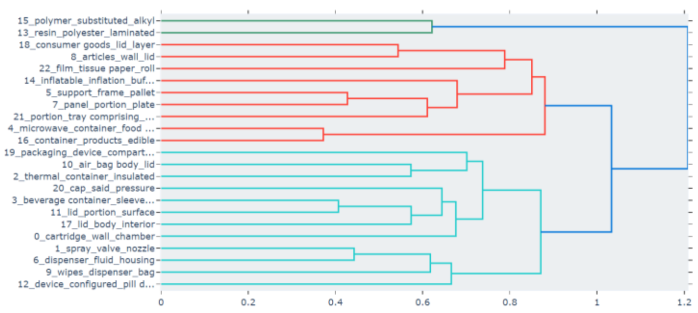




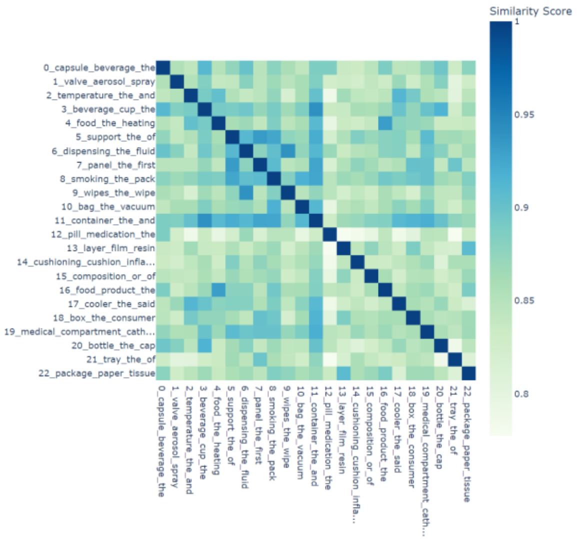
Similarity Matrix



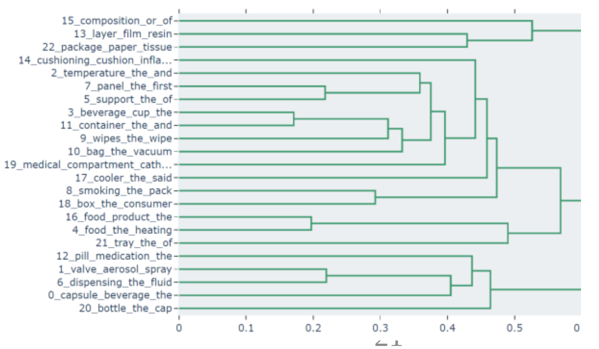
Hierarchical Clustering



Similarity Matrix



Hierarchical Clustering


15_composition_or_of
13_layer_film_resin
22_package_paper_tissue
14_cushioning_cushion_infla...
2_temperature_the_and
7_panel_the_first
5_support_the_of
3_beverage_cup_the
11_container_the_and
9_wipes_the_wipe
10_bag_the_vacuum
19_medical_compartment_cath...
17_cooler_the_said
8_smoking_the_pack
18_box_the_consumer
16_food_product_the
4_food_the_heating
21_tray_the_of
12_pill_medication_the
1_valve_aerosol_spray
6_dispensing_the_fluid
0_capsule_beverage_the
20_bottle_the_cap