Introduction to Artificial Intelligence (Fall 2024)

Homework #1

Team #: 5

Student IDs: 2021313549 / 2024319690 / 2020312668

[BERTopic]

BERTopic is a topic modeling technique that uses BERT embeddings and class-based TF-IDF to create dense clusters that are easy to interpret, while preserving important words for topic description.

1. Embedding Text Data

   Documents are embedded using SBERT. BERTopic uses the following default BERT models:
   - "paraphrase-MiniLM-L6-v2": SBERT trained on English data
   - "paraphrase-multilingual-MiniLM-L12-v2": Multilingual SBERT trained on data from over 50 languages

2. Document Clustering

   Using UMAP, the embedding dimensions are reduced, and HDBSCAN is then applied to cluster these dimensionally reduced embeddings to generate clusters of semantically similar documents.

3. Generating Topic Representations

   In the final stage, topics are extracted through class-based TF-IDF.

[Notice]

The assignment consists of two parts: a basic implementation and an advanced implementation.

- Basic Implementation [70/100]

For the basic implementation, you are not required to modify the code. Simply execute the provided code and answer the questions.

- Advanced Implementation [30/100]

For the advanced implementation, you should modify the original code by adjusting hyperparameters or making other modifications. Explain the reasons behind your changes and describe the changed results.

[Submission]

Submit both a report and code for the assignment. The report should be converted to a PDF, and the code should be written in Google Colab (Google Colab Code) and submitted as an .ipynb file.

Name the report as **{Team #}_AI_PA1.pdf** and the code as **{Team #}_AI_PA1.ipynb**. Then, compress both files into a single zip file named **{Team #}_AI_PA1.zip** before submitting.

Points will be deducted if the required format is not followed.

Both Korean and English are allowed.

Any violations of academic integrity such as plagiarism will result in an F grade.

[Inquiry]

Please send all assignment inquiries only to swe3011_g@g.skku.edu

[Basic Implementation]

1. Install libraries and Load dataset

Install Libraries**:** The necessary libraries are installed for the project using pip. These include:

- **datasets** for loading the dataset,
- **sentence_transformers** for generating text embeddings,
- **umap-learn** for dimensionality reduction,
- **hdbscan** for clustering,
- **bertopic** for topic modeling.

Run the codes.

```
!pip install datasets
!pip install sentence_transformers
!pip install umap-learn
!pip install hdbscan
!pip install bertopic
```

```
from datasets import load_dataset

dataset = load_dataset("CShorten/ML-ArXiv-Papers")["train"]

# Extract abstracts to train on and corresponding titles
abstracts = dataset["abstract"][:1000]
titles = dataset["title"][:1000]
```

2. Pre-calculate Embeddings (5pts)

The code snippet below pre-calculates embeddings for a set of abstracts using the Sentence Transformer model "all-MiniLM-L6-v2". Run the code and answer the question.

```
from sentence_transformers import SentenceTransformer

# Pre-calculate embeddings
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")
embeddings = embedding_model.encode(abstracts, show_progress_bar=True)
```

What role do embeddings play in text processing tasks like topic modeling?

Embedding is the process of converting text data into vector representations. In topic modeling, embeddings convert abstract information into vector data, enabling more effective analysis.

The "*all-MiniLM-L6-v2*" model is a powerful tool in natural language processing that maps sentences and paragraphs into a 384-dimensional dense vector space. This capability makes it well-suited for tasks such as clustering and semantic search.

By representing abstracts as vectors, we can mathematically compute their similarity to group related topics into clusters.

3. Reducing dimensional space (5pts)

The code snippet below uses UMAP to reduce the dimensionality of the document embeddings. Run the code and answer the question.

```
from umap import UMAP

umap_model = UMAP(n_neighbors=15, n_components=5, min_dist=0.0, metric='cosine',
random_state=42)
```

Why is it important to reduce the dimensionality of embeddings before clustering?

Improved efficiency: High-dimensional data demands more memory and processing power. By lowering dimensionality using UMAP make clustering algorithms faster and more efficient.
Improved quality of clusters: Reducing dimensionality of embeddings can improve clustering quality by increasing data density and removing unnecessary noise.
Advantageous at visualization: Lower-dimensional data is easier to visualize, particularly in two or three dimensions, making patterns more interpretable.

4. Clustering (10pts)

The code snippet below uses HDBSCAN to cluster the dimensionally reduced embeddings. Run the code and answer the questions.

```
from hdbscan import HDBSCAN

hdbscan_model = HDBSCAN(min_cluster_size=150, metric='euclidean',
cluster_selection_method='eom', prediction_data=True)
```

What is the purpose of setting **min_cluster_size** in HDBSCAN, and how does it influence the clustering results?

The parameter **min_cluster_size** in HDBSCAN determines the minimum number of points needed to form a cluster.

Setting a higher **min_cluster_size** results in fewer, larger, and more stable clusters by enforcing stricter clustering requirements, which helps the model focus on significant clusters and ignore small, noisy ones.

Conversely, a smaller **min_cluster_size** allows for more, smaller clusters, which can lead to overly fine clustering and less meaningful results.

The right balance improves the consistency and interpretability of clustering outcomes.

Briefly explain the characteristics and advantages of Euclidean clustering.

Characteristics of Euclidean Clustering:

- Distance-Based: Uses Euclidean distance as a measure of similarity, meaning points that are closer together in space are considered more similar.
- Geometry-Sensitive: Best suited for spherical or compact clusters where points are evenly distributed around a center.
- Metric Dependency: The shape and quality of clusters depend on data scale and dimensionality, as Euclidean distance can become less effective in high-dimensional spaces.

Advantages of Euclidean Clustering:

- Simplicity: Easy to understand and implement, particularly in low-dimensional spaces.
- Intuitive Results: Provides clear, interpretable results when clusters are compact and well-separated.
- Efficiency: Effective for clustering tasks with smaller, low-dimensional datasets where Euclidean distance is a meaningful similarity measure.
- Compatibility: Works well with algorithms like K-Means and DBSCAN, which rely on distance measures to group data.

5. Vectorize (10pts)

The code snippet below uses CountVectorizer to convert the text data into a vectorized format. Run the code and answer the question.

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer_model = CountVectorizer(stop_words="english", min_df=1, ngram_range=(1,
2))
```

What is the purpose of setting **min_df** in CountVectorizer, and how does adjusting this parameter affect the resulting vocabulary?

The **min_df** parameter in CountVectorizer sets the minimum number of occurrences required for each word to be included in the vocabulary. Words that appear less than this threshold are ignored.

Setting **min_df** to a lower value, such as 1, includes all words, which captures a broader range of vocabulary, including rare terms. This can be beneficial if rare terms carry meaningful information but may also introduce noise and increase computational cost.

Conversely, increasing **min_df** to a higher value (e.g., 5 or 10) reduces vocabulary size by excluding infrequent terms, focusing on more common and potentially more informative words. This adjustment can enhance efficiency, reduce noise, and lower the dimensionality of the vectorized data, making the model more robust to noise.

In the given code, **min_df** is set to 1, so every word that appears in the documents will be included.

6. Representation Models (10pts)

The code snippet below configures multiple representation models to enhance topic representation in BERTopic. Each model has a unique approach to selecting and refining topic keywords. Run the code and answer the question.

```
from bertopic.representation import KeyBERTInspired, MaximalMarginalRelevance,
PartOfSpeech

# KeyBERT
keybert_model = KeyBERTInspired()

# Part-of-Speech
pos_model = PartOfSpeech("en_core_web_sm")

# MMR
mmr_model = MaximalMarginalRelevance(diversity=0.3)

# All representation models
representation_model = {
   "KeyBERT": keybert_model,
   "MMR": mmr_model,
   "POS": pos_model
}
```

Briefly explain the features of the three models: KeyBERT, Part-of-Speech, and MMR.

KeyBERT: Uses BERT embeddings and cosine similarity to extract the most representative keywords from text, focusing on those that capture the main topic of the document. It provides clear, concise keywords that reflect the core themes.

Part-of-Speech (POS): Utilizes POS tagging to filter for nouns and verbs, ensuring keywords are semantically meaningful and relevant to the topic. Ideal for refining keywords in noisy or unstructured text.

Maximal Marginal Relevance (MMR): Balances relevance and diversity by selecting keywords that are both highly relevant and varied, reducing redundancy. This approach covers different aspects of a topic, enhancing interpretability.

7. Training (10pts)

The code snippet below sets up and trains a BERTopic model. The model is configured with multiple pipeline components and hyperparameters. Run the code and answer the question.

```
from bertopic import BERTopic

topic_model = BERTopic(

  # Pipeline models
  embedding_model=embedding_model,
  umap_model=umap_model,
  hdbscan_model=hdbscan_model,
  vectorizer_model=vectorizer_model,
  representation_model=representation_model,

  # Hyperparameters
  top_n_words=5,
  verbose=True
)

# Train model
topics, probs = topic_model.fit_transform(abstracts, embeddings)

# Show topics
topic_model.get_topic_info()
```

How does changing **top_n_words** affect the interpretability of each topic?

The **top_n_words** parameter controls the number of words used to describe a topic. Lower values create a clear, focused summary, while higher values offer more detailed description but risk reducing clarity if too many words are included.

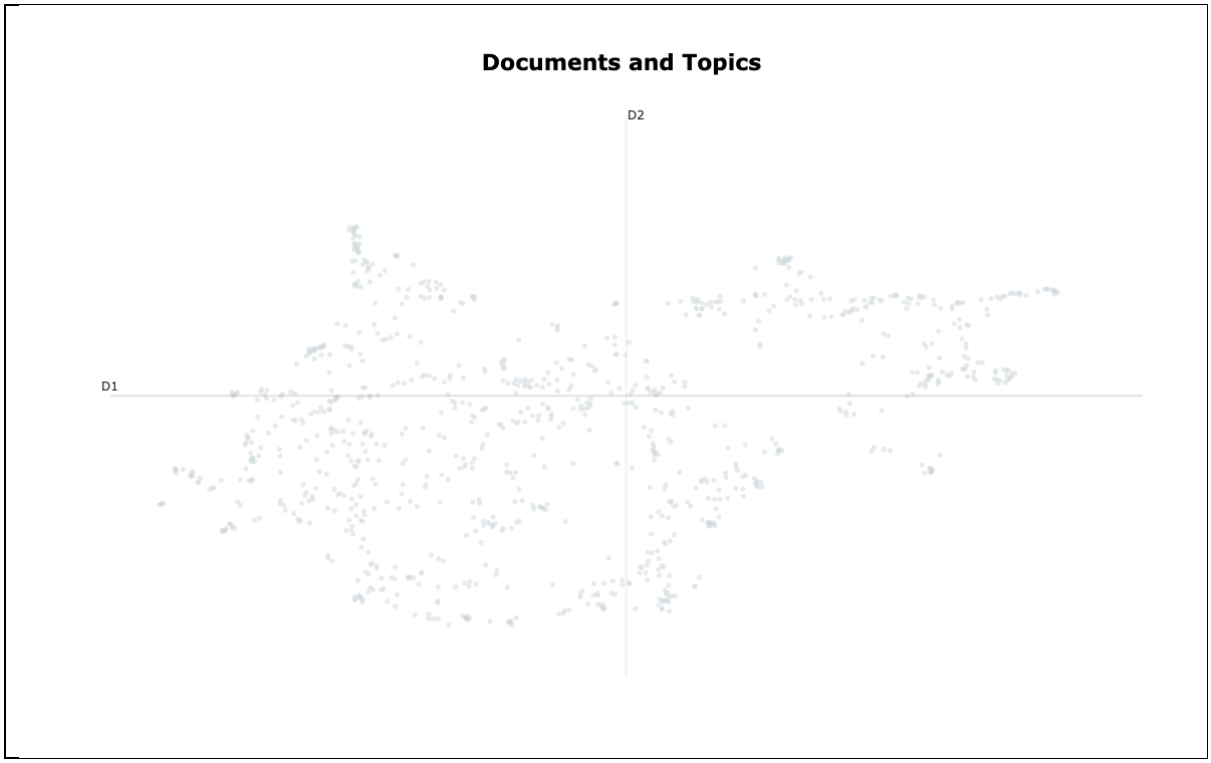
8. Visualize (20pts)

The code snippet below reduces the dimensionality of the document embeddings to two dimensions using UMAP, making it easier to visualize the structure of topics in 2D space. Run the code and answer the questions.

```
# Reduce dimensionality of embeddings, this step is optional but much faster to
perform iteratively:
reduced_embeddings = UMAP(n_neighbors=10, n_components=2, min_dist=0.0,
metric='cosine').fit_transform(embeddings)

# Visualize the documents in 2-dimensional space and show the titles on hover
instead of the abstracts
topic_model.visualize_documents(titles, reduced_embeddings=reduced_embeddings,
custom_labels=True)
```

Attach the resulting visualization image.



Describe the visualized result, explaining any visible clusters, patterns, or interesting observations.

Using UMAP, embeddings' dimensionality is reduced to 2 to visualize the distribution of documents.
Each dot represents a document. The D1 and D2 axes represents the two main dimensions derived from the original high-dimensional embeddings.
There are no distinctly separated clusters visible, but there are areas where points are relatively denser.

High density means that they have semantically similar topics. Conversely, low density indicates that their topics are isolated or unique.
Since the plot is loosely clustered, we can say that documents tend not to share highly specific topics.
However, we can see that there are no significant outliers too.
So, interestingly, we can also say that documents tend to share broad similarities of topics.

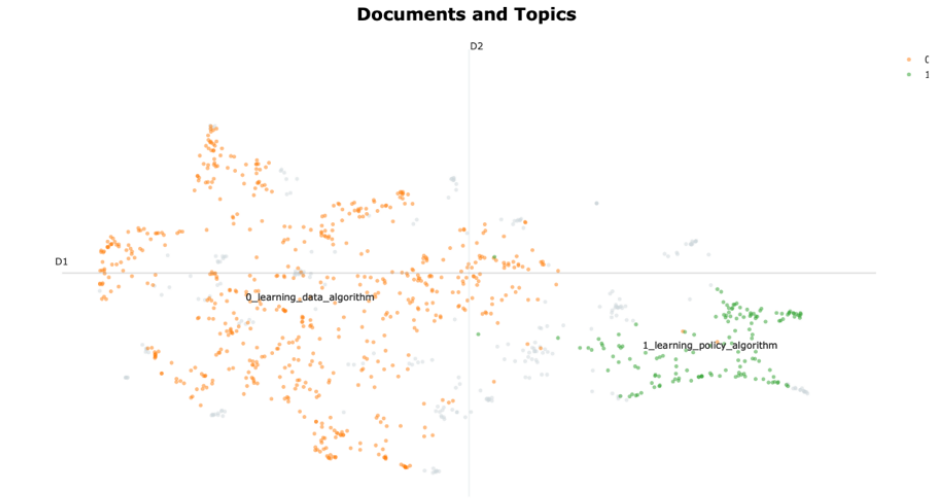
[Advanced Implementation]

The basic implementation includes several adjustable hyperparameters (e.g., n_neighbors, min_cluster_size, min_df, top_n_words). Adjust these hyperparameters to modify the topic modeling results, and explain the purpose of each adjustment and the differences observed. (Requirements: Resulting visualization images, at least three A4 pages, and two or more hyperparameters.) **(30pts)**

Our **initial state** is:
UMAP: n_neighbors = 15, n_component = 5, min_dist = 0.0, metric = 'cosine'
HDBSCAN: min_cluster_size = 150, metric = 'euclidean'
CountVectorizer: min_df = 1
BERTopic: top_n_words = 5

Change Log
Change 1: Lowered min_cluster_size to 50
Result:

	Topic	Count	Name	Representation	KeyBERT	MMR	POS	Representative_Docs
0	-1	240	-1_algorithm_learning_data_algorithms	[algorithm, learning, data, algorithms, proble...	[learning algorithm, learning, privacy, machin...	[algorithm, algorithms, paper, models, optimiz...	[algorithm, learning, data, algorithms, proble...	[This work considers computationally efficie...
1	0	617	0_learning_data_algorithm_problem	[learning, data, algorithm, problem, based, pa...	[sparse, svm, machine learning, learning, clas...	[data, algorithm, paper, methods, clustering, ...	[learning, data, algorithm, problem, paper, mo...	[Conditional random field (CRF) and Structur...
2	1	143	1_learning_policy_algorithm_regret	[learning, policy, algorithm, regret, problem,...	[bandit problem, reinforcement learning, bandi...	[policy, regret, reinforcement, optimal, reinf...	[learning, policy, algorithm, regret, problem,...	[We consider an opportunistic spectrum acces...

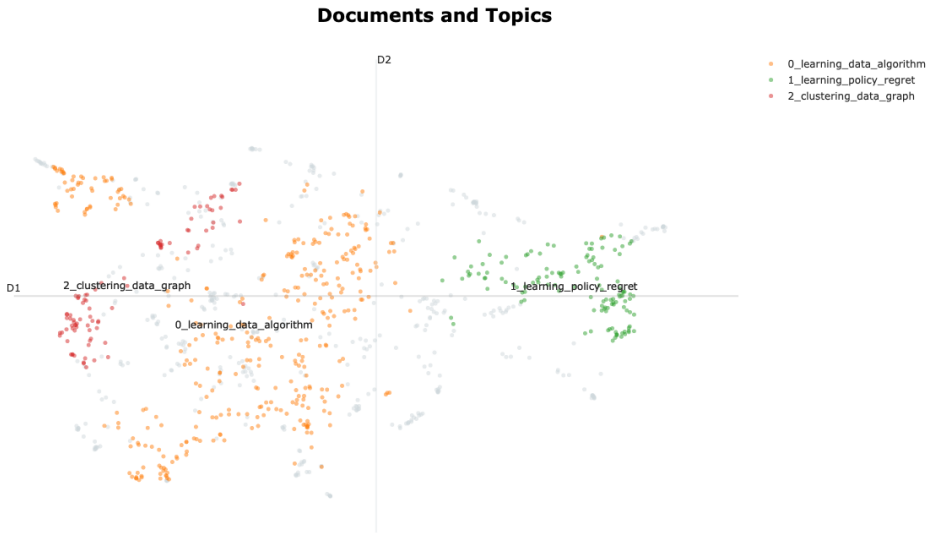


Interpretation: To make clusters more visible and clearer, we lowered minimum cluster size. It successfully made two clusters to visualize. We will keep this one.

Change 2: Increased top_n_words to 10
Result: The result plot was very same from the previous state (change1).
Interpretation: top_n_words does not affect clustering result.

Change 3: Increased min_dist to 0.1
Result:

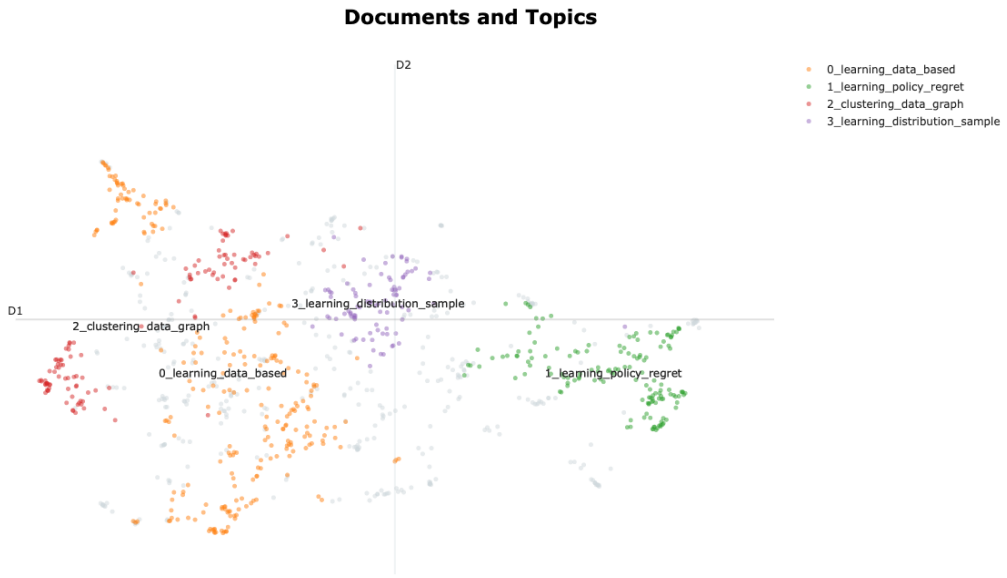
	Topic	Count	Name	Representation	KeyBERT	MMR	POS	Representative_Docs
0	-1	435	-1_algorithm_learning_data_problem	[algorithm, learning, data, problem, algorithm...	[learning, networks, stochastic, algorithms, o...	[algorithm, data, problem, algorithms, method,...	[algorithm, learning, data, problem, algorithm...	[This paper introduces a novel message-passi...
1	0	346	0_learning_data_algorithm_based	[learning, data, algorithm, based, problem, pa...	[classifiers, svm, classifier, classification,...	[learning, algorithm, problem, paper, classifi...	[learning, data, algorithm, problem, paper, cl...	[Two ubiquitous aspects of large-scale data ...
2	1	129	1_learning_policy_regret_algorithm	[learning, policy, regret, algorithm, problem,...	[bandit problem, bandit, reinforcement learnin...	[regret, reinforcement, optimal, reinforcement...	[learning, policy, regret, algorithm, problem,...	[The fundamental problem of multiple seconda...
3	2	90	2_clustering_data_graph_algorithm	[clustering, data, graph, algorithm, problem, ...	[spectral clustering, clustering algorithms, c...	[clustering, data, algorithm, algorithms, grap...	[clustering, data, graph, algorithm, problem, ...	[We formulate weighted graph clustering as a...



Interpretation: min_dist sets a minimum distance between each point. It might lower the density of each cluster, but we expect that it could excavate a new cluster. As we expected, a new cluster came out. So we decided to keep this change.

Change 4: lowered min_cluster_size to 30
Result:

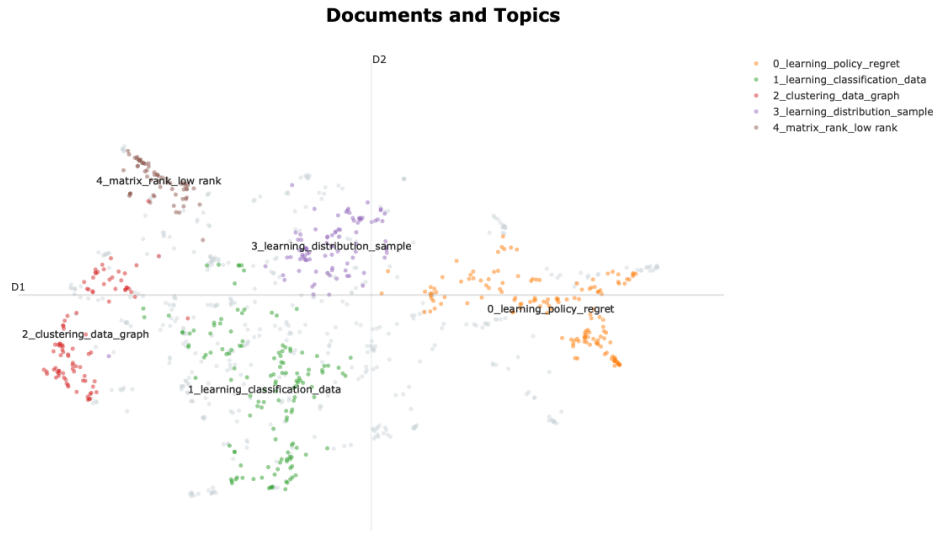
	Topic	Count	Name	Representation	KeyBERT	MMR	POS	Representative_Docs
0	-1	389	-1_learning_data_algorithm_algorithms	[learning, data, algorithm, algorithms, proble...	[learning, machine learning, classification, g...	[data, algorithm, algorithms, problem, perform...	[learning, data, algorithm, algorithms, proble...	[We consider the problem of energy-efficient...
1	0	262	0_learning_data_based_method	[learning, data, based, method, matrix, proble...	[classifiers, svm, lasso, regularization, clas...	[matrix, algorithm, classification, paper, met...	[learning, data, method, matrix, problem, algo...	[Feature selection with specific multivariat...
2	1	146	1_learning_policy_regret_algorithm	[learning, policy, regret, algorithm, problem,...	[bandit problem, reinforcement learning, bandi...	[regret, reinforcement, optimal, reinforcement...	[learning, policy, regret, algorithm, problem,...	[We consider an opportunistic spectrum acces...
3	2	116	2_clustering_data_graph_algorithm	[clustering, data, graph, algorithm, model, pr...	[clustering algorithms, clustering, cluster, s...	[clustering, algorithm, algorithms, clusters, ...	[clustering, data, graph, algorithm, model, pr...	[Recent spectral clustering methods are a pr...
4	3	87	3_learning_distribution_sample_random	[learning, distribution, sample, random, funct...	[learning algorithm, sample complexity, learna...	[learning, information, eps, distributions, es...	[learning, distribution, sample, random, funct...	[A k-modal probability distribution over t...



Interpretation: We lowered minimum cluster size even more in order to make more clusters. Now we have four clusters. But doesn't seem optimal since the clusters are not clearly separated.

Change 5: lowered n_neighbors to 7
Result:

	Topic	Count	Name	Representation	KeyBERT	MMR	POS	Representative_Docs
0	-1	478	-1_learning_data_algorithm_algorithms	[learning, data, algorithm, algorithms, proble...	[learning, lasso, sparse, machine learning, cl...	[data, algorithm, algorithms, problem, paper, ...	[learning, data, algorithm, algorithms, proble...	[Sparse learning has recently received incre...
1	0	147	0_learning_policy_regret_algorithm	[learning, policy, regret, algorithm, problem,...	[bandit problem, bandit, reinforcement learnin...	[policy, regret, optimal, online, reinforcemen...	[learning, policy, regret, algorithm, problem,...	[The fundamental problem of multiple seconda...
2	1	138	1_learning_classification_data_kernel	[learning, classification, data, kernel, metho...	[vector machines, classifiers, svm, feature se...	[classification, kernel, feature, methods, pap...	[learning, classification, data, kernel, metho...	[Feature selection with specific multivariat...
3	2	90	2_clustering_data_graph_algorithm	[clustering, data, graph, algorithm, graphs, m...	[clustering algorithms, clustering, cluster, c...	[clustering, graphs, clusters, algorithms, pap...	[clustering, data, graph, algorithm, graphs, m...	[We formulate weighted graph clustering as a...
4	3	88	3_learning_distribution_sample_function	[learning, distribution, sample, function, alg...	[sample complexity, learnable, learning algori...	[eps, information, distributions, pac, samples...	[learning, distribution, sample, function, com...	[A k-modal probability distribution over t...
5	4	59	4_matrix_rank_low rank_sparse	[matrix, rank, low rank, sparse, low, problem,...	[matrix completion, compressed sensing, matrix...	[sparse, entries, matrices, norm, matrix compl...	[matrix, rank, low rank, sparse, low, problem,...	[Minimizing the rank of a matrix subject to ...

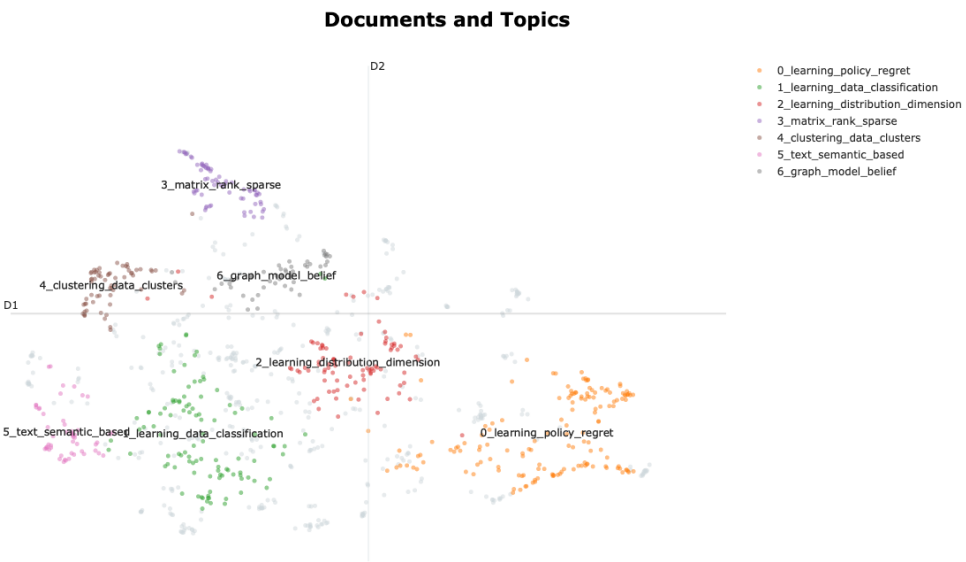


Interpretation: Lowered n_neighbors in order to make clusters to focus more on local structures than overall relationship. As a result, a new cluster came out and now we can see that all clusters are clearly distinguished.

Change 6: increased min_df to 2
Result: The result plot was very same from the previous state (change5).
Interpretation: min_df has no effect on clustering result.

Change 7: rollback min_dist to 0.0
Result:

	Topic	Count	Name	Representation	KeyBERT	MMR	POS	Representative_Docs
0	-1	427	-1_learning_algorithm_data_problem	[learning, algorithm, data, problem, based, pa...	[classifiers, learning, machine learning, clas...	[learning, algorithm, paper, algorithms, metho...	[learning, algorithm, data, problem, paper, al...	[Sparse learning has recently received incre...
1	0	153	0_learning_policy_regret_algorithm	[learning, policy, regret, algorithm, problem,...	[bandit problem, bandit, reinforcement learnin...	[regret, reinforcement, optimal, reinforcement...	[learning, policy, regret, algorithm, problem,...	[The fundamental problem of multiple seconda...
2	1	105	1_learning_data_classification_feature	[learning, data, classification, feature, kern...	[kernel learning, classifiers, feature selecti...	[classification, kernel, features, selection, ...	[learning, data, classification, feature, kern...	[Kernel-based machine learning algorithms ar...
3	2	86	2_learning_distribution_dimension_algorithm	[learning, distribution, dimension, algorithm,...	[sample complexity, learnability, learnable, l...	[learning, dimension, eps, complexity, distrib...	[learning, distribution, dimension, result, fu...	[A k-modal probability distribution over t...
4	3	64	3_matrix_rank_sparse_low rank	[matrix, rank, sparse, low rank, low, problem,...	[matrix completion, compressed sensing, matrix...	[matrix, sparse, low rank, algorithm, norm, co...	[matrix, rank, sparse, low rank, low, problem,...	[Recovering intrinsic data structure from co...
5	4	63	4_clustering_data_clusters_means	[clustering, data, clusters, means, algorithms...	[clustering, clustering algorithms, cluster, c...	[clustering, clusters, algorithms, cluster, cl...	[clustering, data, clusters, means, algorithms...	[Recent spectral clustering methods are a pr...
6	5	52	5_text_semantic_based_learning	[text, semantic, based, learning, words, appro...	[text classification, information retrieval, c...	[text, semantic, knowledge, classification, wo...	[text, semantic, learning, words, approach, kn...	[We participated in three of the protein-pro...
7	6	50	6_graph_model_belief_algorithm	[graph, model, belief, algorithm, models, prob...	[graphical models, belief propagation, graphic...	[algorithm, graphs, belief propagation, propag...	[graph, model, belief, algorithm, models, prob...	[Markov random fields are used to model high...



Interpretation: Since min_dist lowers the density, we tried to rollback the change and see what happens. And surprisingly, new clusters just pop out.

This is our **final state:**
UMAP: n_neighbors = 7, n_component = 5, min_dist = 0.0, metric = ‘cosine’
HDBSCAN: min_cluster_size = 30, metric = ‘euclidean’
CountVectorizer: min_df = 1
BERTopic: top_n_words = 10