BERTopic

1. BERTopic

BERT embeddings과 클래스 기반(class-based) TF-IDF를 활용하여 주제 설명에서 중요한 단어를 유지하면서도 쉽게 해석할 수 있는 조밀한 클러스터를 만드는 토픽 모델링 기술

주요 3단계

1) 텍스트 데이터 임베딩(SBERT)

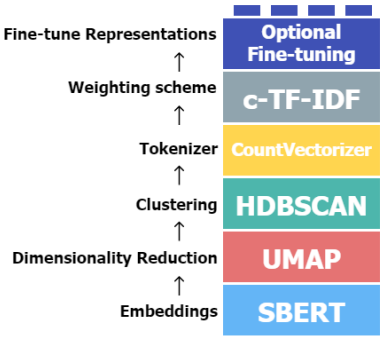
- 문자를 숫자로 변환

2) 문서를 군집화(UMAP → HDBSCAN)

- UMAP을 사용하여 임베딩의 차원을 줄이고 HDBSCAN 기술을 사용
- 차원 축소된 임베딩을 클러스터링하여 의미적으로 유사한 문서 클러스터를 생성함.

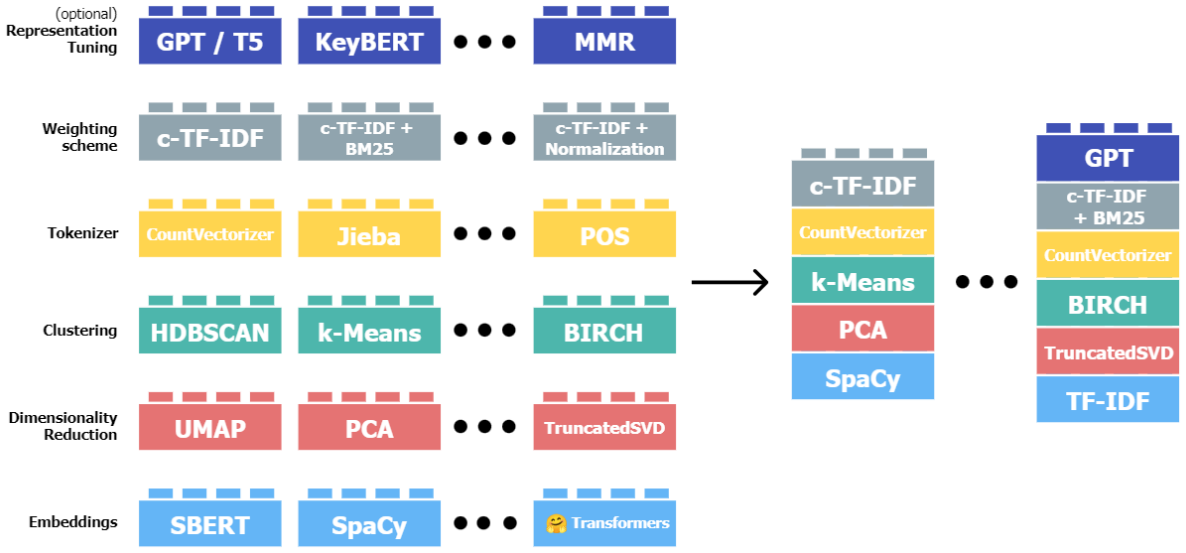
3) 토픽 표현을 생성(CountVectorizer → c-TFIDF)

- 클래스 기반 TF-IDF로 토픽을 추출



BERTopic의 모듈성

- BERTopic 내 프로세스의 각 단계는 독립적으로 구성됨. 예를 들어 토큰화 단계는 문서 변환 시 사용되는 임베딩 모델에 직접적인 영향을 받지 않음. 단계별로 창의성을 발휘하여 자신만의 토픽모델을 만들 수 있음.



- 알고리즘의 각 주요 단계를 명시적으로 정의하여 직관적으로 만들 수 있음.

```
from umap import UMAP
from hdbscan import HDBSCAN
from sentence_transformers import SentenceTransformer
from sklearn.feature_extraction.text import CountVectorizer

from bertopic import BERTopic
from bertopic.representation import KeyBERTInspired
from bertopic.vectorizers import ClassTfidfTransformer


# Step 1 - Extract embeddings
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")

# Step 2 - Reduce dimensionality
umap_model = UMAP(n_neighbors=15, n_components=5, min_dist=0.

# Step 3 - Cluster reduced embeddings
hdbscan_model = HDBSCAN(min_cluster_size=15, metric='euclidea

# Step 4 - Tokenize topics
vectorizer_model = CountVectorizer(stop_words="english")

# Step 5 - Create topic representation
ctfidf_model = ClassTfidfTransformer()
```

```
# Step 6 - (Optional) Fine-tune topic representations with
# a `bertopic.representation` model
representation_model = KeyBERTInspired()

# All steps together
topic_model = BERTopic(
  embedding_model=embedding_model,          # Step 1 - Extrac
  umap_model=umap_model,                    # Step 2 - Reduce
  hdbscan_model=hdbscan_model,              # Step 3 - Cluste
  vectorizer_model=vectorizer_model,        # Step 4 - Tokeni
  ctfidf_model=ctfidf_model,                # Step 5 - Extrac
  representation_model=representation_model # Step 6 - (Optio
)
```

1. Embed documents

- 문서를 숫자로 변환
- BERTopic에서 기본적으로 sentence-transformers를 사용함. 의미적 유사성에 최적화 되어있어 클러스터링 작업에 맞으며 문서 또는 문장 임베딩을 생성하는데 유용함.
- sentence-transformers 내 여러 모델이 있지만 기본값은 다음과 같음.
  - "paraphrase-MiniLM-L6-v2" : 영어 데이터로 학습된 SBERT
  - "paraphrase-multilingual-MiniLM-L12-v2" : 50개 이상의 언어로 학습된 다국어 SBERT

2. Dimensionality reduction

- 클러스터링은 고차원의 데이터를 처리하는데 어려움이 있기 때문에 차원의 개수를 줄여야 함.
- 차원 축소는 PCA도 있지만 기본적으로 UMAP을 사용함. UMAP은 속도가 빠르고 데이터의 global 구조를 더 잘 보존함.
  - UMAP
    - PCA와 t-SNE과 같은 차원 축소 알고리즘
    - 방법: 높은 차원에서 데이터를 그래프로 만들고 낮은 차원으로 그래프를 projection함.

        - 파라미터 : n_neighbors, min_dist
- 데이터의 local, global 구조를 보존하는데 이는 다음 단계에서 의미적으로 유사한 문서들을 클러스터로 모을 때 필요한 정보임.

3. Cluster Documents

- 임베딩 차원을 줄인 후 클러스터링 수행
- HDBSCAN은 다양한 모양의 클러스터를 찾을 수 있고 이상치를 식별하는 기능이 있음.

4. Bag-of-words

- 문서나 텍스트 데이터를 숫자로 변환하는데 위 임베딩과 다른 점은 중요한 단어를 찾기 위한 것.
- 단어 추출하여 개수는 세는 방법인데 클러스터 구조에 대한 가정을 하지 않음.

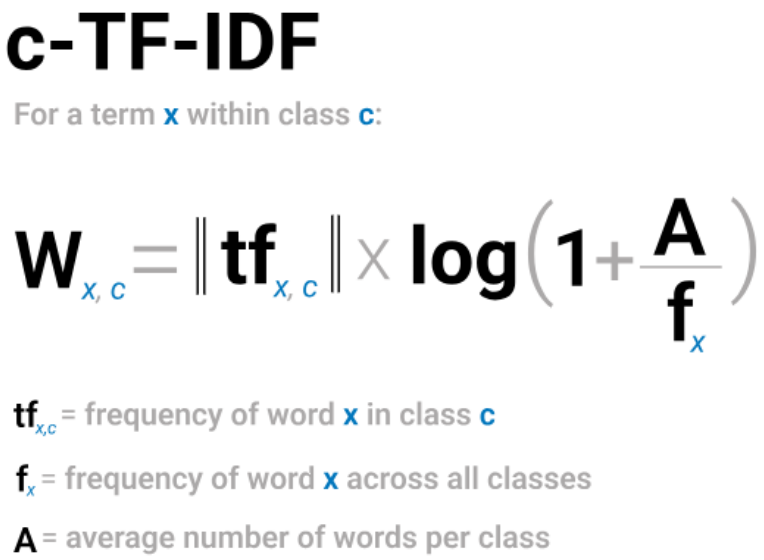
  1)클러스터의 모든 문서를 단일 문서로 결합

  2) 클러스터 내에 단어별 빈도를 계산

  3) 클러스터가 서로 다른 크기를 갖기 때문에 L1 정규화 수행

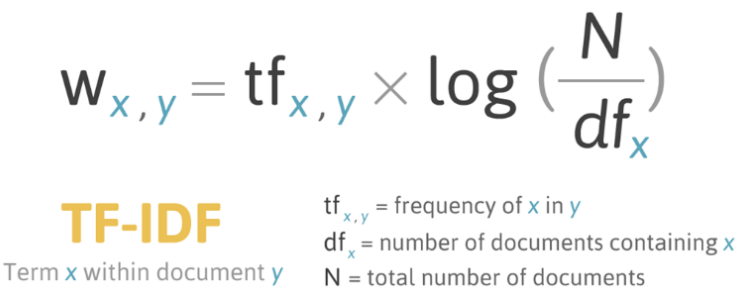
5. Topic representation

- 위에서의 bag-of-words representation 에서 클러스터별 차이를 알고 싶음. 이 클러스터가 다른 것과 다른 점에 대해 알고 싶음.
    - C-TF-IDF (Class-based TF-IDF)
        - 클러스터 내의 문서를 단일 문서로 처리하여 TF-IDF를 적용
        - 클러스터 내에 중요한 단어가 많을수록 해당 주제를 더 잘 나타냄.



- 문서 모음을 설명하는 단어 집합이 만들어짐.

  참고) TF-IDF



6. (Optional) Fine-tune Topic representation

- 토픽 표현에 대해 c-tfidf 상위 값을 사용해도 되지만 미세조정하여 더 나은 결과를 얻을 수 있음.
  - KeyBERT : KeyBERT-inspired model를 이용하여 c-tf-idf로 생성된 키워드들을 미세조정
    1) c-TF-IDF를 활용하여 각 주제(topic)별로 대표 문서를 생성하고, 이를 주제의 새로운 임베딩으로 사용함.
    2) 그 후 후보 키워드와 토픽 임베딩 사이에 유사도를 계산한
  - MMR : Maximal Marginal Relevance를 이용하여 c-tf-idf로 생성된 키워드에서 다양하게 추출
    - 이전 단계에서 주제에 대해 유사한 키워드가 있는지 고려하지 않는데 이러한 중복성 줄이고 다양성을 개선
    - 토픽와 키워드의 거리는 가깝게 / 키워드간 거리는 멀게
  - spaCy: spaCy의 POS 모듈을 통해 특정 품사의 단어만 추출

- Transformers or OpenAI or Cohere: 텍스트 생성 모델을 활용함으로써 가독성 있는 토픽 표현 생성
- LangChain: 랭체인을 이용해 LLM의 지식정보를 확장하거나 연결

BERTopic의 장단점

- **Strengths**
  - 언어모델에 관계없이 성능이 좋음. 최첨단 언어모델 활용시 성능 향상 가능
  - 문서 임베딩 프로세스와 토픽 표현을 분리하여 파인튜닝에 상당히 유연함.
    - 임베딩과 토픽 표현 프로세스에 서로 다른 전처리 가능
  - c-tf-idf를 활용하여 단어의 분포로 토픽을 표현함. BERTopic의 핵심 알고리즘을 변경하지 않고 동적으로 시간에 따라 토픽이 어떻게 변화하는지 분석할 수 있음.
- **Weaknesses**
  - 각 문서가 오직 하나의 토픽을 포함한다고 가정
    - 문서를 문장이나 패러그래프로 분할할 수 있지만 이상적인 표현이 아님.
    - HDBSCAN은 soft clustering 기법이라 확률 행렬을 토픽 분포에 대리값으로 사용하지만 여러 주제가 포함될 수 있다는 것을 고려하지 않음.
  - 트랜스포머 기반 언어모델을 사용하여 문맥 정보를 반영해 임베딩을 하지만 토픽 표현에 있어서는 문맥을 직접적으로 반영하지 않음. 문서 내 단어 출현 빈도를 기반으로 한 bag-of-words 방식으로 주제 표현이 생성되기 때문임.
    - 토픽 표현으로 추출된 단어들이 서로 유사할 수 있고 주제 해석에 있어서 중복 및 불필요한 정보를 포함할 수 있음.

    → 이론적으로는 MMR 기법을 사용하여 해결할 수 있는데, 논문에서는 미적용

출처

https://arxiv.org/abs/2203.05794

(BERTopic: Neural topic modeling with a class-based TF-IDF procedure)

https://maartengr.github.io/BERTopic