BERTopic: Fine-tune Parameters

DamenC · Following
5 min read · Mar 26

Photo by Maarten van den Heuvel on Unsplash

In general, BERTopic works fine with the out of box model. However, when you have millions of data to process, it may take some time to process your

data with the basic model. In this post, I will show you how to fine-tune some parameters in BERTopic and compare their results. Let's dive in.

Base Model

We first check what parameters are there in the class BERTopic. For a detailed examination, please check the document here: BERTopic. In the official document, there is explanation on each parameter and their default value. Here I want to pick up a few parameters to mention because those parameters play a key role in representing topics from our documents.

```
class BERTopic:
    def __init__(self,
                 language: str = "english",
                 top_n_words: int = 10,
                 n_gram_range: Tuple[int, int] = (1, 1),
                 min_topic_size: int = 10,
                 nr_topics: Union[int, str] = None,
                 low_memory: bool = False,
                 calculate_probabilities: bool = False,
                 seed_topic_list: List[List[str]] = None,
                 embedding_model=None,
                 umap_model: UMAP = None,
                 hdbscan_model: hdbscan.HDBSCAN = None,
                 vectorizer_model: CountVectorizer = None,
                 ctfidf_model: TfidfTransformer = None,
                 representation_model: BaseRepresentation = None,
                 verbose: bool = False,
                 )
    self.XXX
    self.XXX
    ...
    ...
```

- n_gram_range: the default is (1,1), that is it will produce the topic words such as "New" and "York" separately. If you want " New York" to appear,

you can sent this parameter to (1,2).

- umap_model: UMAP (Uniform Manifold Approximation and Projection) is a dimensionality reduction algorithm that is often used for visualization of high-dimensional data. It works by finding a low-dimensional representation of the data that preserves the structure of the original high-dimensional space.
- hdbscan_model: HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that can identify clusters of arbitrary shape and size in a dataset. It works by finding regions of high density in the data and expanding them into clusters, while also identifying noise points that do not belong to any cluster.

Fine-tune the Parameters

We have learned what the parameters are and what they actually do. Now let's fine-tune them and compare the result with the out of box model. Again, we will use the Qatar World Cup data that we prepared previously. If you have not downloaded umap and hbdscan, please pip install.

```
# Base Model

import pandas as pd
import pickle
with open('world_cup_tweets.pkl', 'rb') as f:
    data = pickle.load(f)

data = data.Tweet_processed.to_list()

from bertopic import BERTopic
```

```
model_B = BERTopic(language="english", calculate_probabilities=True, verbose=Tru
topics_B, probs_B = topic_model.fit_transform(data)
```

```
# Fine-tuned Model

import pandas as pd
import pickle
with open('world_cup_tweets.pkl', 'rb') as f:
    data = pickle.load(f)

data = data.Tweet_processed.to_list()

from umap import UMAP
from hdbscan import HDBSCAN

umap_model = UMAP(n_neighbors=3, n_components=3, min_dist=0.05)
hdbscan_model = HDBSCAN(min_cluster_size=80, min_samples=40,
                        gen_min_span_tree=True,
                        prediction_data=True)
from bertopic import BERTopic

model_A = BERTopic(
    umap_model=umap_model,
    hdbscan_model=hdbscan_model,
    top_n_words=10,
    language='english',
    calculate_probabilities=True,
    verbose=True,
    n_gram_range=(1, 2)
)
topics_A, probs_A = model.fit_transform(data)
```

UMAP:

- n_neighbors=3: This parameter determines the number of nearest neighbors used by UMAP to approximate the local structure of the data. In this case, UMAP will look at the three nearest neighbors to each data point when constructing the embedding.

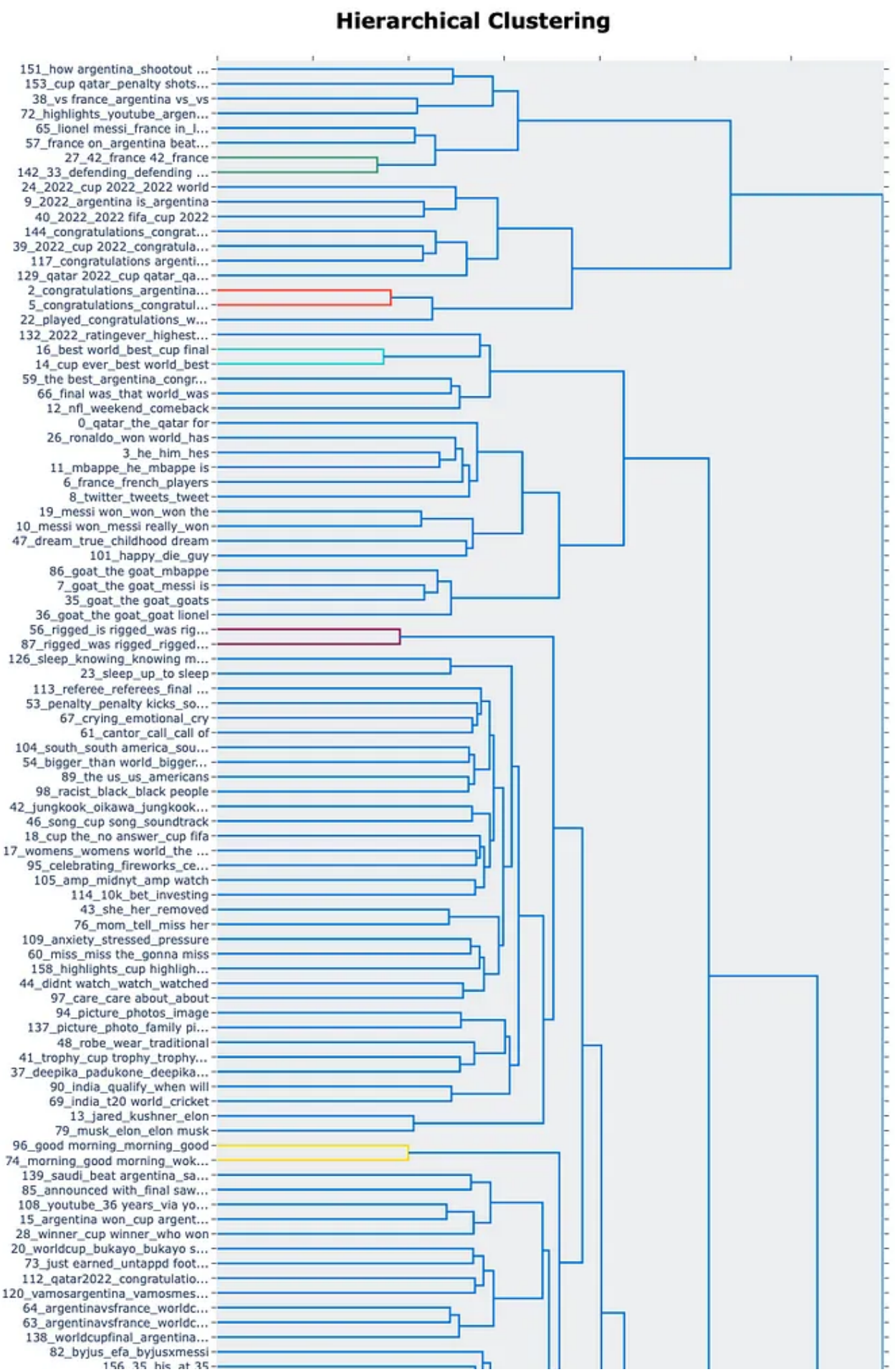
- n_components=3: This parameter specifies the number of dimensions in the embedded space. By default, UMAP will reduce the dimensionality of the data to 2 dimensions, but in this case, it will reduce it to 3.
- min_dist=0.05: This parameter controls the minimum distance between points in the embedded space. A higher value of min_dist will result in more space between points, which can improve the separation of clusters.

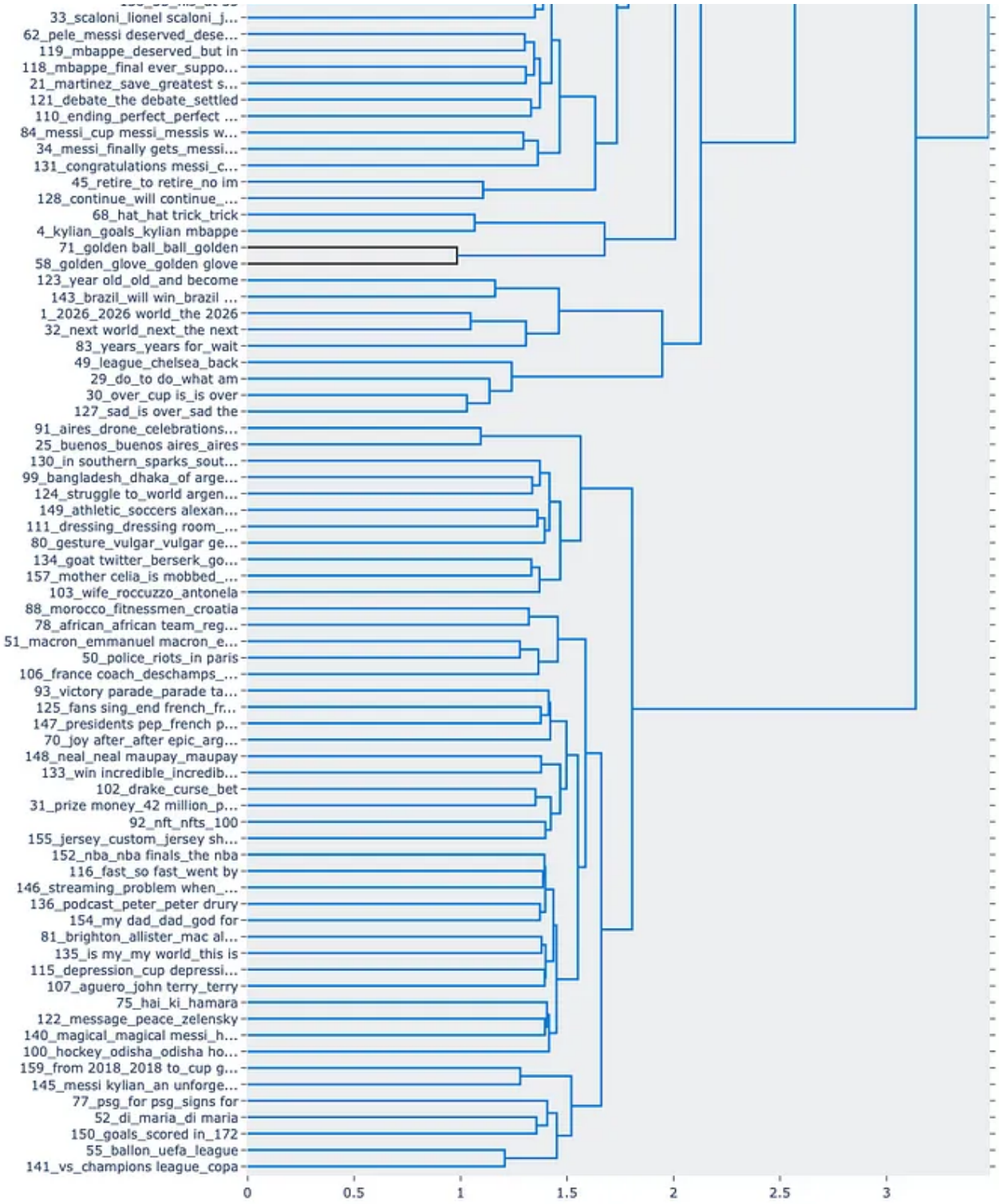
HDBSCAN:

- min_cluster_size=80: This parameter specifies the minimum number of points required for a cluster to be formed. Clusters with fewer points than this threshold will be labeled as noise.
- min_samples=40: This parameter determines the number of samples in a neighborhood required for a point to be considered a core point. Core points are used to build clusters, and points that are not core points are classified as noise.
- gen_min_span_tree=True: This parameter tells HDBSCAN to construct a minimum spanning tree of the input data before clustering. This can help to identify clusters that are connected by only a few points, which might be missed by other clustering algorithms.
- prediction_data=True: This parameter tells HDBSCAN to store additional information about the data, such as the membership probabilities of each point in each cluster. This information can be useful for downstream analysis and visualization.

Compare the Results

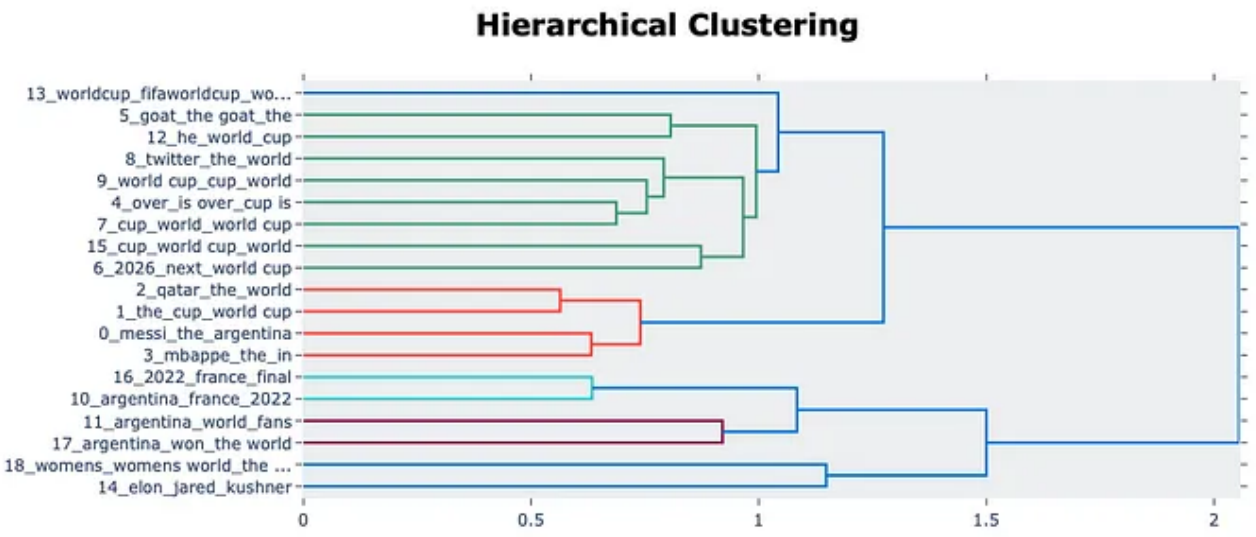
Base Model:


Hierarchical Clustering
151_how argentina_shootout ...
153_cup qatar_penalty shots...
38_vs france_argentina vs_vs
72_highlights_youtube_argen...
65_lionel messi_france in_l...
57_france on_argentina beat...
27_42_france 42_france
142_33_defending_defending ...
24_2022_cup 2022_2022 world
9_2022_argentina is_argentina
40_2022_2022 fifa_cup 2022
144_congratulations_congrat...
39_2022_cup 2022_congratula...
117_congratulations argenti...
129_qatar 2022_cup qatar_qa...
2_congratulations_argentina...
5_congratulations_congratul...
22_played_congratulations_w...
132_2022_ratingever_highest...
16_best world_best_cup final
14_cup ever_best world_best
59_the best_argentina_congr...
66_final was_that world_was
12_nfl_weekend_comeback
0_qatar_the_qatar for
26_ronaldo_won world_has
3_he_him_hes
11_mbappe_he_mbappe is
6_france_french_players
8_twitter_tweets_tweet
19_messi won_won_won the
10_messi won_messi really_won
47_dream_true_childhood dream
101_happy_die_guy
86_goat_the goat_mbappe
7_goat_the goat_messi is
35_goat_the goat_goats
36_goat_the goat_goat lionel
56_rigged_is rigged_was rig...
87_rigged_was rigged_rigged...
126_sleep_knowing_knowing m...
23_sleep_up_to sleep
113_referee_referees_final ...
53_penalty_penalty kicks_so...
67_crying_emotional_cry
61_cantor_call_call of
104_south_south america_sou...
54_bigger_than world_bigger...
89_the us_us_americans
98_racist_black_black people
42_jungkook_oikawa_jungkook...
46_song_cup song_soundtrack
18_cup the_no answer_cup fifa
17_womens_womens world_the ...
95_celebrating_fireworks_ce...
105_amp_midnyt_amp watch
114_10k_bet_investing
43_she_her_removed
76_mom_tell_miss her
109_anxiety_stressed_pressure
60_miss_miss the_gonna miss
158_highlights_cup highligh...
44_didnt watch_watch_watched
97_care_care about_about
94_picture_photos_image
137_picture_photo_family pi...
48_robe_wear_traditional
41_trophy_cup trophy_trophy...
37_deepika_padukone_deepika...
90_india_qualify_when will
69_india_t20 world_cricket
13_jared_kushner_elon
79_musk_elon_elon musk
96_good morning_morning_good
74_morning_good morning_wok...
139_saudi_beat argentina_sa...
85_announced with_final saw...
108_youtube_36 years_via yo...
15_argentina won_cup argent...
28_winner_cup winner_who won
20_worldcup_bukayo_bukayo s...
73_just earned_untappd foot...
112_qatar2022_congratulatio...
120_vamosargentina_vamosmes...
64_argentinavsfrance_worldc...
63_argentinavsfrance_worldc...
138_worldcupfinal_argentina...
82_byjus_efa_byjusxmessi



Base Model Created by Author

Fine-tuned Model:



Fine-tuned Model Created by Author

Clearly, more topics were generated in the base model, which explains the fact that it takes a long time to process texts with a significant amount. Meanwhile, in the fine-tuned model, fewer topics were created given the settings in the parameters.

For those who are interested in how the results will vary with different combinations of parameter settings. I will put the sample code here and you can change the parameters to check different results.

```
from bertopic import BERTopic
from umap import UMAP
from hdbscan import HDBSCAN

# Define a list of parameters to try for UMAP
umap_params = [
    {'n_neighbors': 15, 'n_components': 2, 'min_dist': 0.1},
    {'n_neighbors': 10, 'n_components': 2, 'min_dist': 0.01},
    {'n_neighbors': 3, 'n_components': 2, 'min_dist': 0.001}
]

# Define a list of parameters to try for HDBSCAN
```

```
hdbscan_params = [
    {'min_cluster_size': 100, 'min_samples': 100},
    {'min_cluster_size': 50, 'min_samples': 70},
    {'min_cluster_size': 5, 'min_samples': 50}
]

# Loop over the parameter combinations and fit BERTopic models
for umap_param in umap_params:
    for hdbscan_param in hdbscan_params:
        # Create UMAP and HDBSCAN models with the current parameter combination
        umap_model = UMAP(**umap_param)
        hdbscan_model = HDBSCAN(**hdbscan_param, gen_min_span_tree=True, predict

        # Fit a BERTopic model with the current parameter combination
        model = BERTopic(
            umap_model=umap_model,
            hdbscan_model=hdbscan_model,
            top_n_words=10,
            language='english',
            calculate_probabilities=True,
            verbose=True,
            n_gram_range=(1, 2)
        )
        topics, probs = model.fit_transform(data)

        # Visualize the hierarchy and save the figure to an HTML file
        fig = model.visualize_hierarchy()
        fig.write_html(f'model_umap_{umap_param}_hdbscan_{hdbscan_param}.html')
```

Bert | Topic Modeling | Deep Learning | Data Science | Python

More from the list: "NLP"

Curated by Himanshu Birla

Jon Gi... in Towards Data ...

Characteristics of Word Embeddings

· 11 min read · Sep 4, 2021

Jon Gi... in Towards Data ...

The Word2vec Hyperparameters

· 6 min read · Sep 3, 2021

Jon Gi... in

The Word2ve

· 15 min rea

View list

Written by DamenC

Following

38 Followers

MS in Informatics; MA in International Area Studies; Interested in machine learning, culture and everything in between.

More from DamenC

DamenC

Text Classification with BERT (3)

DamenC

In text classification with BERT (2), we discussed what is PyTorch and how to...

5 min read · Mar 22

10

Topic Based Sentiment Analysis: Linking Topic Modeling with...

Hello friends! Now we learned how to do topic modeling and sentiment analysis on twitter...

6 min read · Mar 15

57

DamenC

Text Classification with BERT (1)

This is part one of a series of text classification demonstrations using the BER...

4 min read · Mar 4

10

DamenC

Using BERTopic to Analyze Qatar World Cup Twitter Data: Part 1

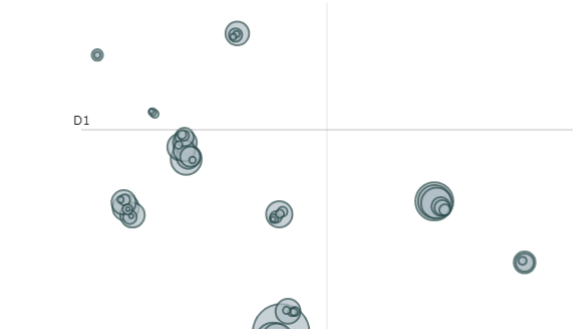
Qatar World Cup was full of surprises! From Saudi Arabia shocking the world by upsettin...

5 min read · Mar 11

12 1

See all from DamenC

Recommended from Medium



Jawwad Shadman Siddique

Topic Modeling Using BERTopic on Newsgroup Dataset: Python...

We go step by step from creating a google collab workspace to visualizing the cluster o...

7 min read · Jul 5

8

Alidu Abubakari in AI Science

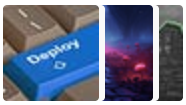
Taking Sentiment Analysis to the Next Level with Huggingface's...

Introduction

17 min read · May 31

104 1

Lists



Predictive Modeling w/ Python

20 stories · 452 saves

Coding & Development

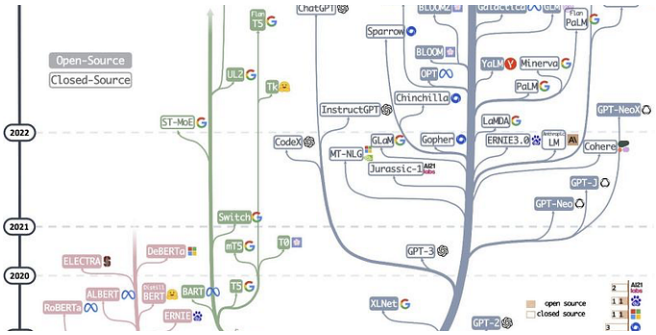
11 stories · 200 saves

Practical Guides to Machine Learning

10 stories · 519 saves

New_Reading_List

174 stories · 133 saves



Kalyanchilakamarri

Integrating ChatGPT and Topic Modeling

Hi Medium readers hope you all are doing well, what's the hot topic in the town? It's...

2 min read · May 22

4

Haifeng Li

A Tutorial on LLM

Generative artificial intelligence (GenAI), especially ChatGPT, captures everyone's...

15 min read · Sep 14

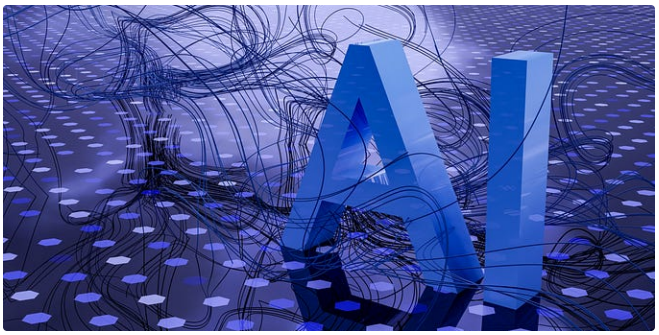
372

Ahmet Taşdemir

Fine-Tuning DistilBERT for Emotion Classification

In this post, we will walk through the process of fine-tuning the DistilBERT model for...

8 min read · Jun 14



The Python Lab

How to Perform Sentiment Analysis using BERT in Python

Sentiment analysis, also known as opinion mining, is a field within natural language...

· 5 min read · May 23

26

See more recommendations