

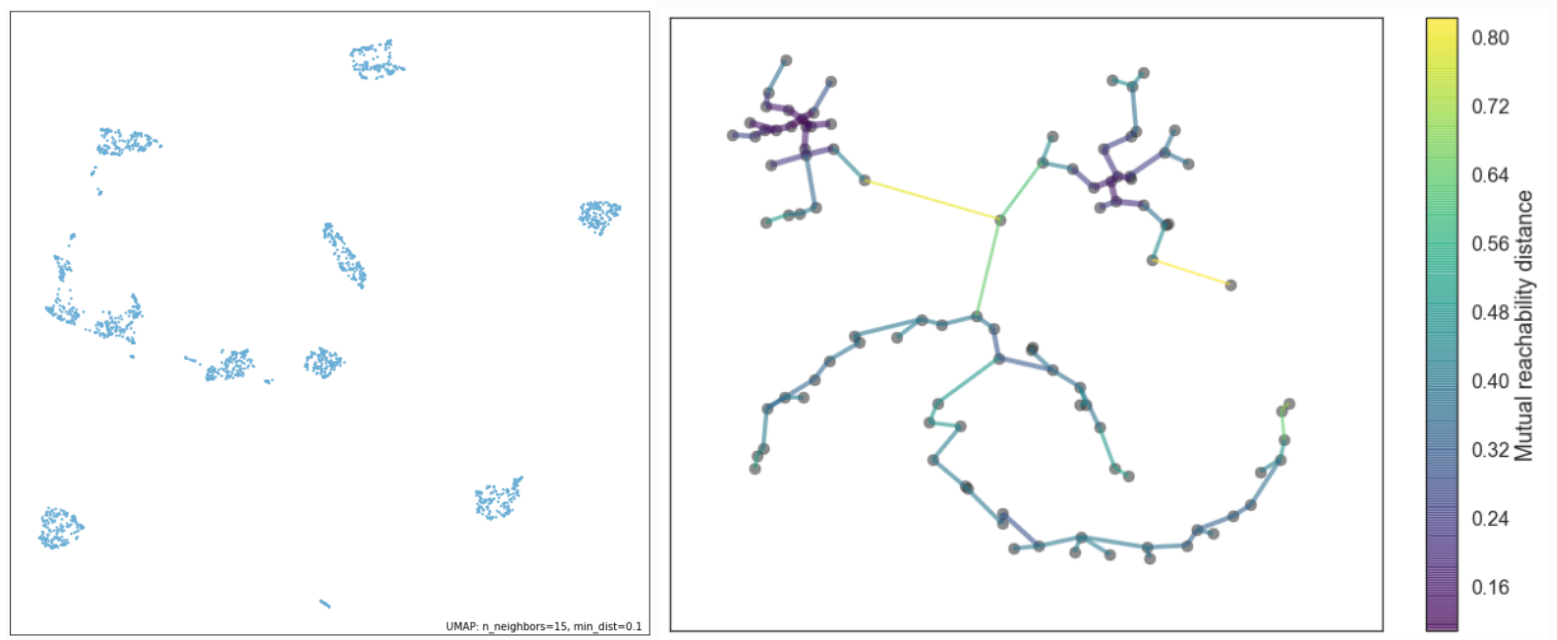
Topeax

An Improved Clustering Topic Model with Density Peak Detection and Lexical-Semantic Term Importance

Márton Kardos

Problem 1

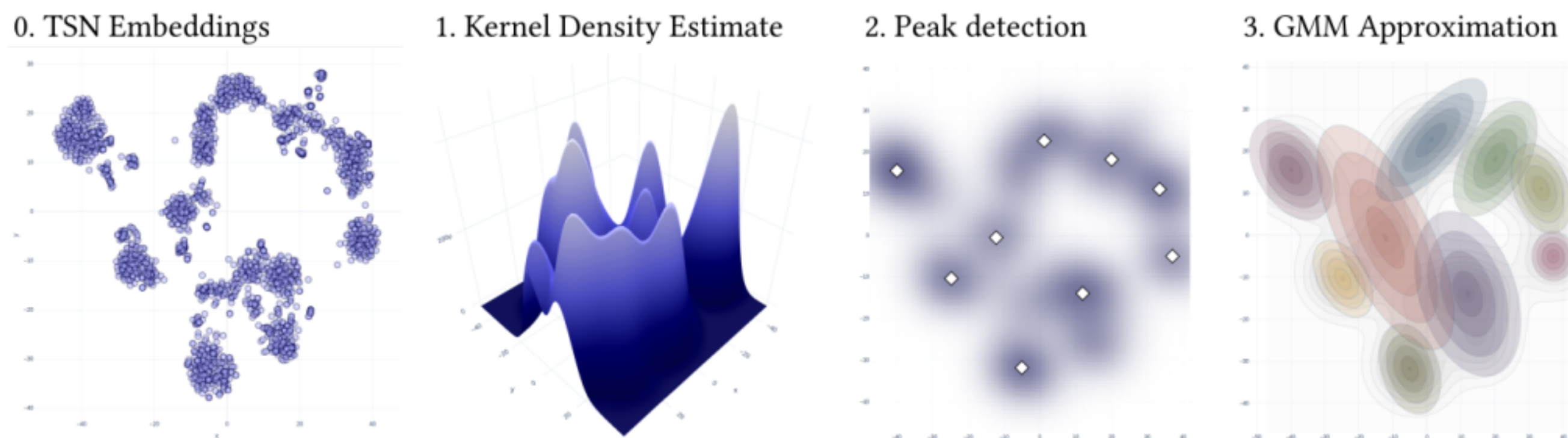
Top2Vec and BERTopic use UMAP and HDBSCAN, which in theory allows them to discover the number of clusters...



...but as we will see, in practice they grossly overestimate it..

Solution 1

I introduce the Peax clustering algorithm, which finds peaks in a kernel density estimate, then estimates components as a mixture of Multivariate Gaussians. (also UMAP → TSNE)



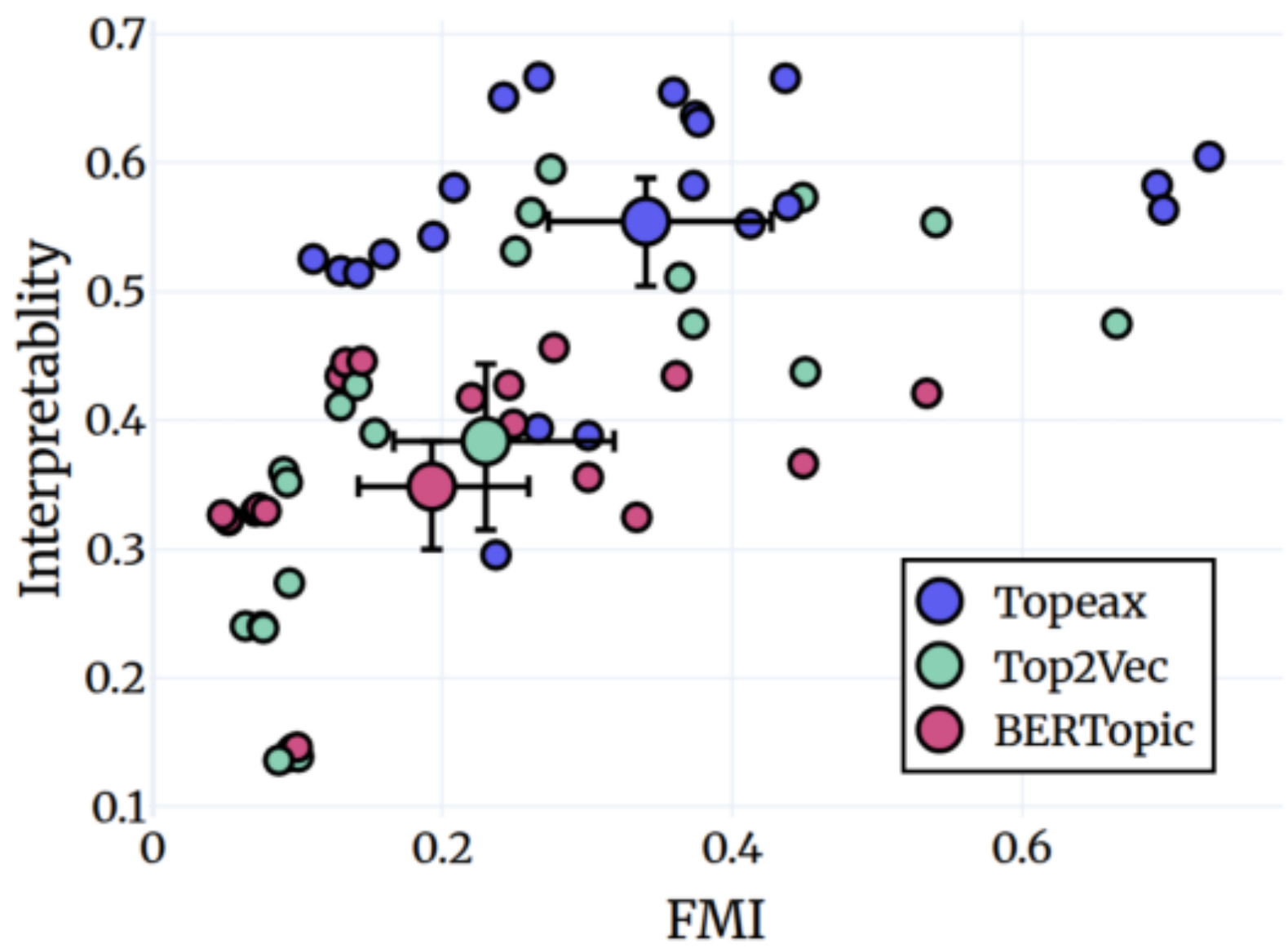
Experimental Method

I evaluate Topeax, BERTopic and Top2Vec on MTEB(eng, v2) clustering tasks + BBC, and Tweets. Topic quality (C_{in} , C_{ex} , diversity, interpretability) and cluster quality (FMI) are monitored. I ran robustness checks to subsampling and hyperparameters.

Topic and Cluster Quality

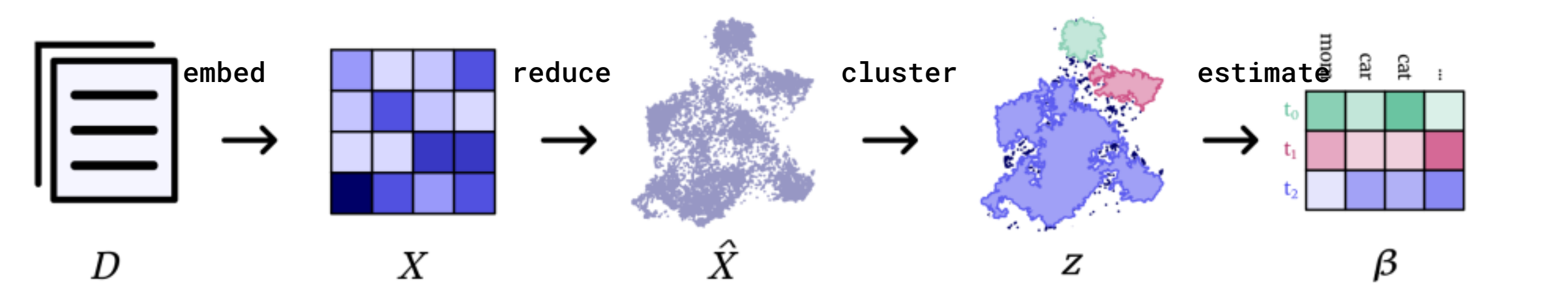
Topeax outperforms baselines in metrics of topic quality, except for extrinsic coherence. It also has significantly higher agreement with human-assigned clusters than other methods.

Model	C_{in}	C_{ex}	d	I
Topeax	0.35±0.15	0.32±0.09	0.96±0.05	0.55±0.10
Top2Vec	0.21±0.11	0.39±0.09	0.57±0.29	0.38±0.15
BERTopic	0.24±0.12	0.17±0.04	0.64±0.17	0.35±0.10



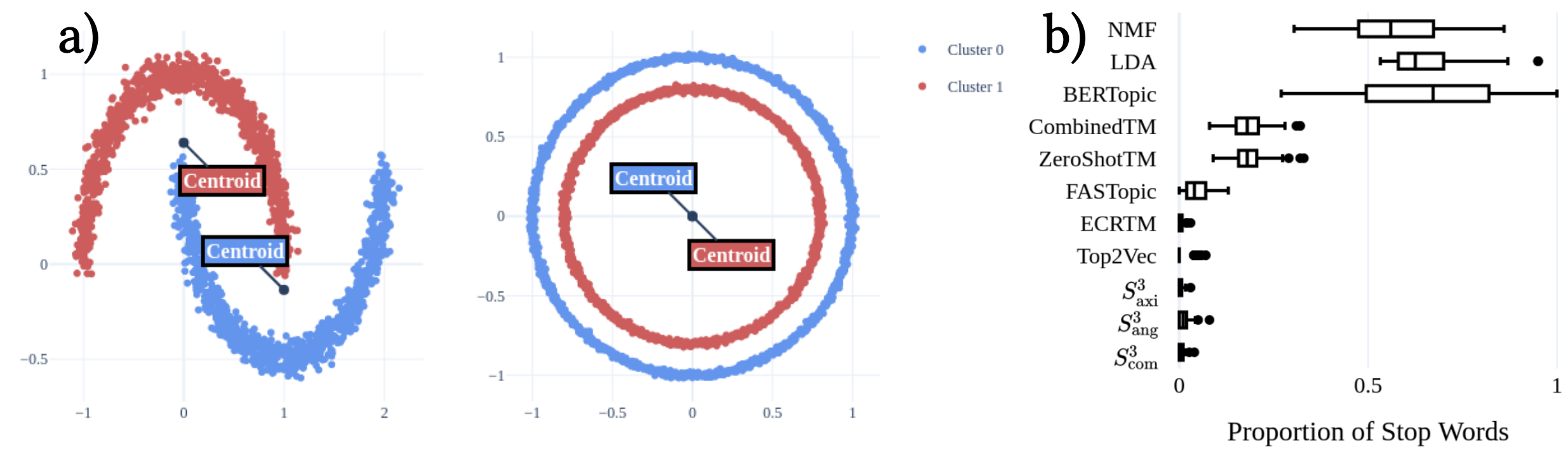
Coefficients	Estimate	p-value	95% CI
Intercept (<i>Topeax</i>)	0.3405	0.000	[0.267, 0.414]
Topeax	-0.1106	0.038	[-0.215, -0.006]
BERTopic	-0.1479	0.006	[-0.252, -0.044]

Clustering transformer embeddings is the most popular modern paradigm for topic modelling.



Problem 2

a) Top2Vec estimates term importance based on distance from centroids, but clusters are based on density, and are not spherical.



b) BERTopic generates junk topics filled with stop words.

Solution 2

I estimate term importance scores based both on **lexical** and **semantic** valence by using a combination of word embedding proximity to weighted average embedding and NPMI.

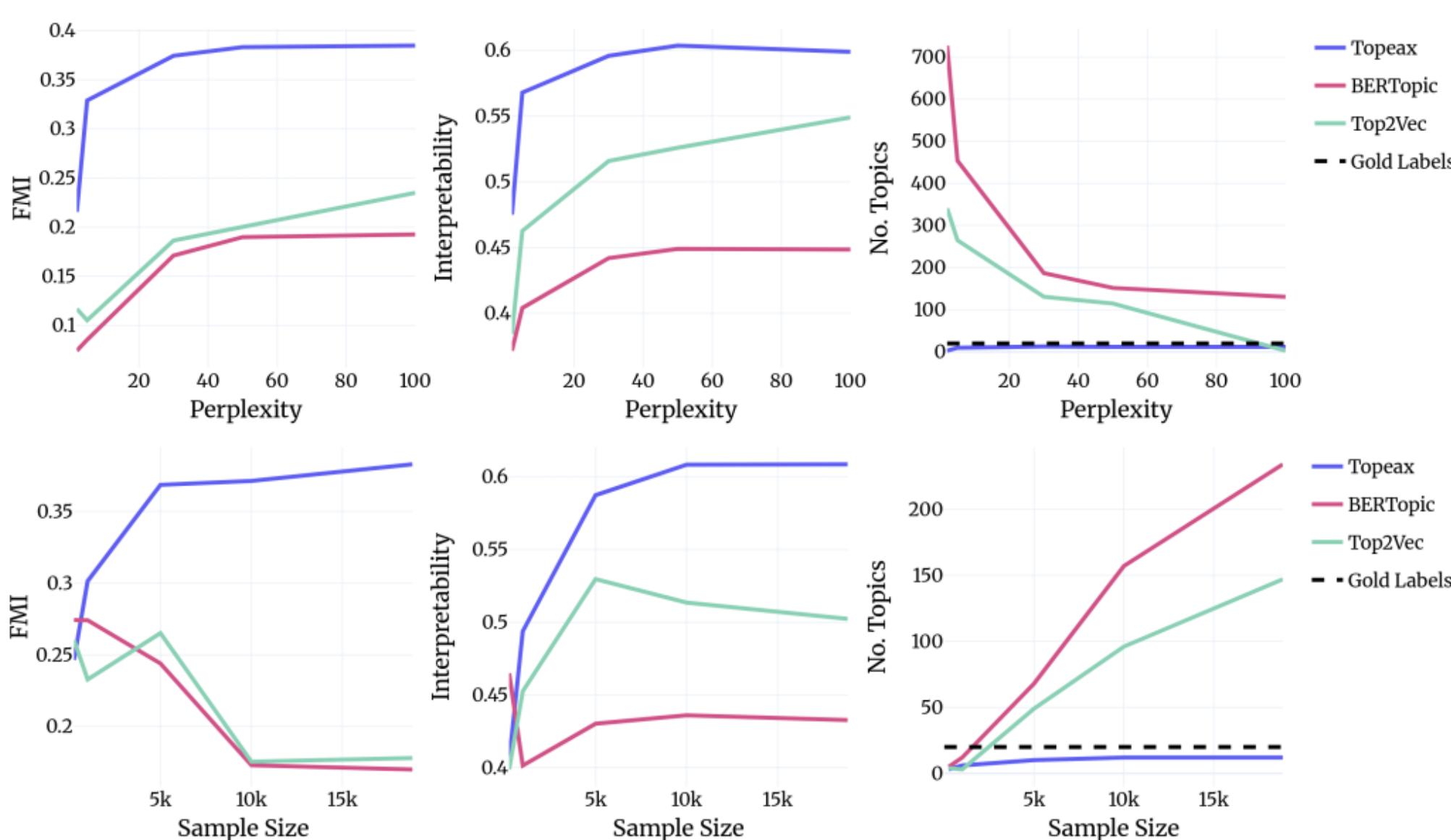
$$t_k = \frac{\sum_d r_{kd} \cdot x_d}{\sum_d r_{kd}}; s_{kj} = \cos(t_k, w_j)$$

$$+ \text{npmi}_{kj} = \frac{\text{pmi}_{kj}}{-\log_2 p(v_j, z_k)}$$

$$= \beta_{kj} = \sqrt{\frac{1 + \text{npmi}_{kj}}{2}} \cdot \frac{1 + s_{kj}}{2}$$

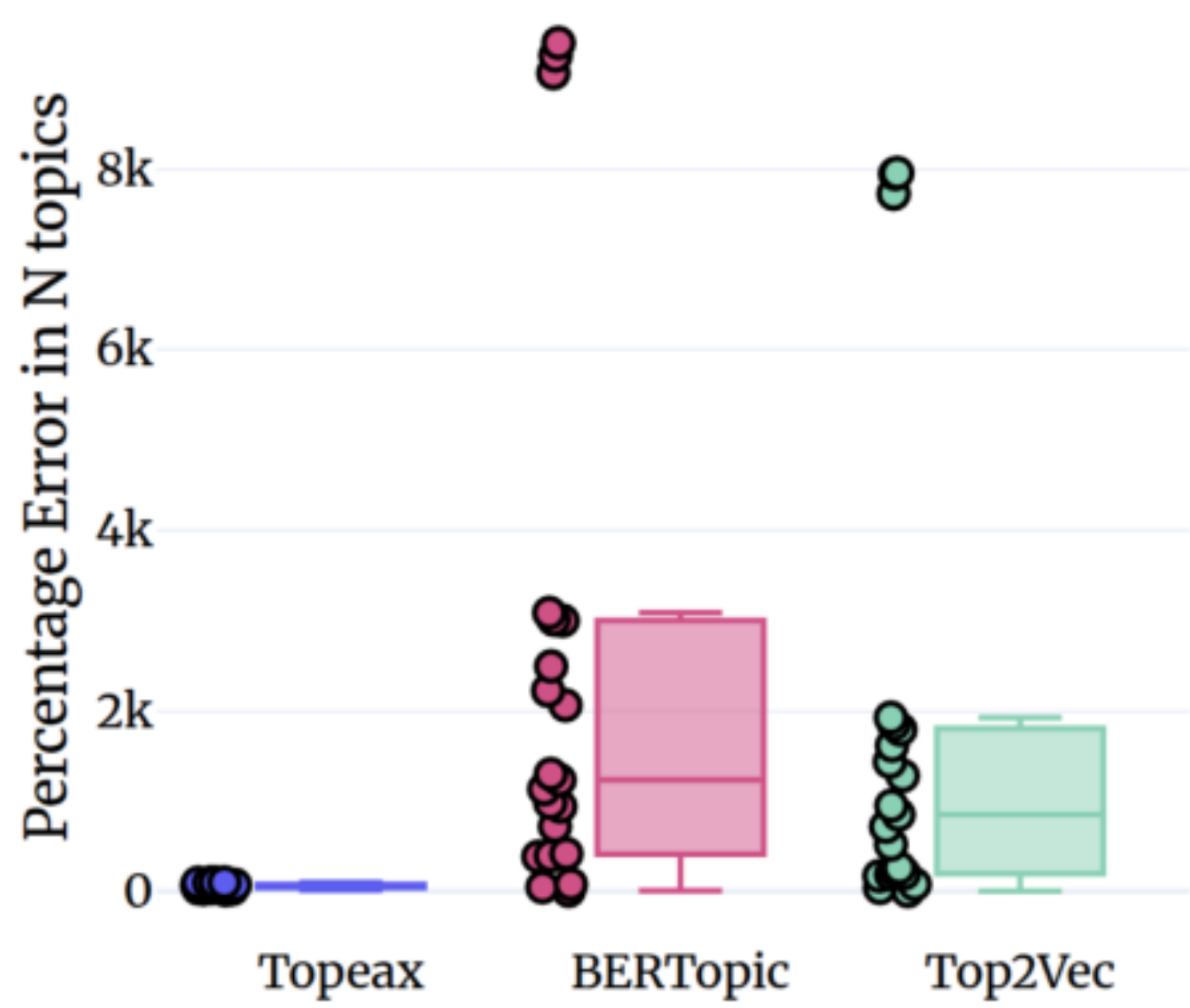
Robustness to Subsampling and Perplexity

Topeax is considerably less volatile when corpora are subsampled, or when the perplexity parameter is adjusted in UMAP or TSNE than other methods, and converges on best performance when the full corpus is available.



Finding Number of Clusters

Topeax is substantially better at identifying the number of clusters in datasets than baselines.



Easy-to-use Implementation

Topeax is implemented in the Turftopic Python package. It is easy to use, and is fully scikit-learn compatible.

```
# pip install turftopic, datasets, plotly
from datasets import load_dataset
from turftopic import Topeax

ds = load_dataset("gopalkalpande/bbc-news-summary", split="train")
topeax = Topeax(random_state=42)
doc_topic = topeax.fit_transform(list(ds["Summaries"]))

topeax.print_topics()
```

ID	Highest Ranking
0	mobile, microsoft, digital, technology, broadband, phones, devices, internet, mobiles, computer
1	economy, growth, economic, deficit, prices, gdp, inflation, currency, rates, exports
2	profits, shareholders, shares, takeover, shareholder, company, profit, merger, investors, financial
3	film, actor, oscar, films, actress, oscars, bafta, movie, awards, actors
4	band, album, song, singer, concert, rock, songs, rapper, rap, grammy
5	tory, blair, labour, ukip, mps, minister, election, tories, mr, ministers
6	olympic, tennis, iaaf, federer, wimbledon, doping, roddick, champion, athletics, olympics
7	rugby, liverpool, england, mourinho, chelsea, premiership, arsenal, gerrard, hodgson, gareth