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

Márton Kardos

Aarhus University

martonkardos@cas.au.dk

Abstract

Text clustering is today the most popular paradigm of topic modelling both in research and industry settings. Despite clustering topic models' apparent success, we identify a number of issues in Top2Vec and BERTopic, which remain largely unsolved. Firstly, these approaches are unreliable at discovering the number of topics in a corpus, due to sensitivity to hyperparameters. Secondly, while BERTopic ignores the semantic distance of keywords to topic vectors, Top2Vec ignores word counts in the corpus. This results in, on the one hand, hardly interpretable topics due to the presence of stop words and junk words, and lack of variety and trust on the other. In this paper, I introduce a new approach, Topeax, which discovers the number of clusters from peaks in density estimates, and combines lexical and semantic term importance to gain high-quality topic keywords.

1. Introduction

2. Model Specification

I introduce Topeax, a novel topic modelling approach based on document clustering. The model differs in a number of aspects from traditional clustering topic models like BERTopic and Top2Vec.
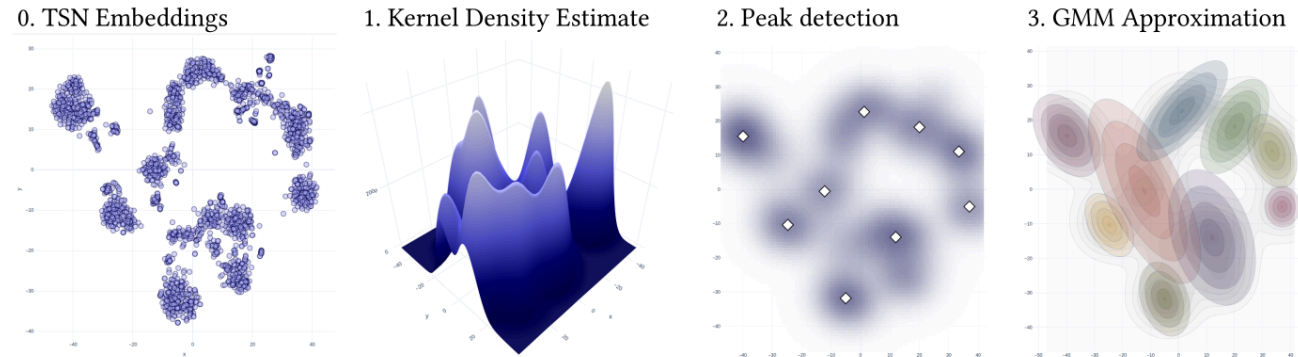

Figure 1: A schematic overview of the Peax clustering algorithm.
*The illustrations were generated from the **political ideologies** dataset.*

2.1. Dimensionality Reduction

Unlike other clustering topic models, Topeax relies on t-Distributed Stochastic Neighbour Embeddings (cite it here) instead of UMAP. I use the the cosine metric to calculate document similarities for TSNE, as it is widely used for model training and downstream applications. The number of dimensions was fixed to 2 in all of our experiments, as this allows us to visualize the reduced embeddings. Additionally, TSNE has fewer hyperparameters than UMAP. While it has been demonstrated that TSNE can be sensitive the chosen value of `perplexity`, we will show that, within a reasonable range, this will not have an effect on the number of topics or topic quality.
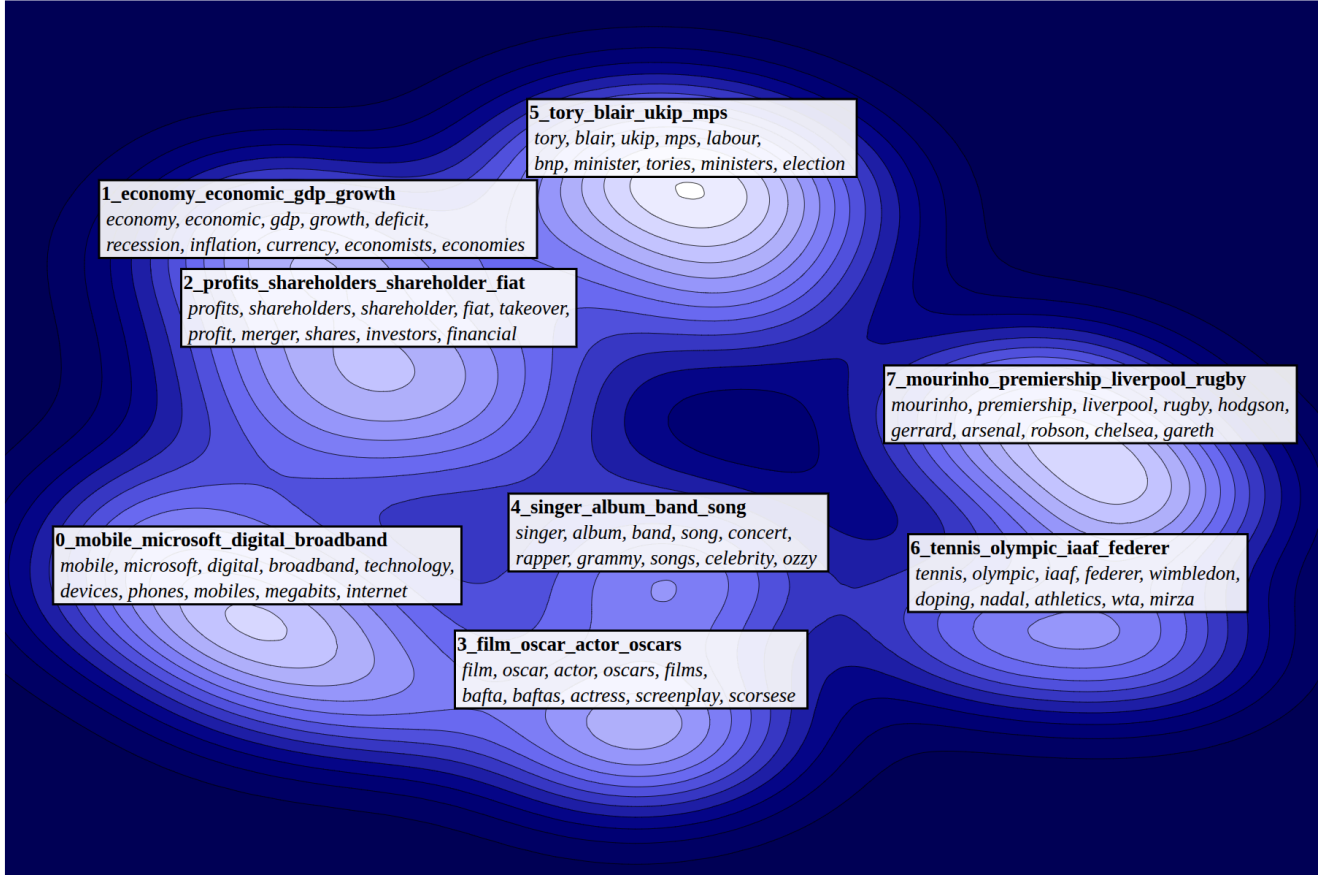
Figure 2: Topeax model illustrated on the BBC News dataset. Topics are identified at density peaks, and keywords get selected based on combined term importance.

2.2. The Peax Clustering Model

While HDBSCAN is the choice of clustering model for both BERTopic and Top2Vec, I introduce a new technique for document clustering, termed **Peax**, which, instead, clusters documents based on density peaks in the reduced document space.

The Peax algorithm consists of the following steps:

1. A Gaussian Kernel Density Estimate (KDE) is obtained over the reduced document embeddings. Bandwidth is determined with the Scott method.
2. The KDE is evaluated on a 100x100 grid over the embedding space. Density peaks are then detected by applying a local-maximum filter to the KDE heatmap. A neighbourhood connectivity of 25 is used, which means, every pixel is included within a 5 unit radius.
3. Cluster centres are assigned to these density peaks. The density structure of each cluster is estimated by fitting a Gaussian mixture model, with its means fixed to the peaks, using the Expectation-Maximization algorithm. Documents are assigned to the component with the highest responsibility:

$$\hat{z}_d = \arg\max_k(r_{\mathrm{kd}}) \text{ , and } r_{\mathrm{kd}} = p(z_k = 1 \mid \hat{x}_d)$$

where $\hat{z}_d$ is the estimated underlying component assigned to document d, $\hat{x}_d$ is the TSN embedding of document d, and r_{kd} is the responsibility of component k for document d.

2.3. Term Importance Estimation

To mitigate the issues experienced with c-TF-IDF and centroid-based term importance estimation in previously proposed clustering topic models, I introduce a novel approach that uses a combination of a semantic and a lexical cluster-term importance.

2.3.1. Semantic Importance

Semantic term importance is estimated similar to (cite Top2Vec), but, since we have access to a probabilistic, non-spherical model, and cluster boundaries are not hard, topic vectors are estimated from the responsibility-weighted average of document embeddings.

$$t_k = \frac{\sum_d r_{kd} \cdot x_d}{\sum_d r_{kd}}$$

where t_k is the embedding of topic k and x_d is the embedding of document d. Let the embedding of term j be w_j. The semantic importance of term j for cluster k is then:

$$s_{kj} = \cos(t_k, w_j)$$

2.3.2. Lexical Importance

Instead of relying on a tf-idf-based measure for computing the valence of a term in a corpus, an information-theoretical approach is used. Theoretically, we can estimate the lexical importance of a term for a cluster, by computing the mutual information of the term's occurrence with the cluster's occurrence. Due to its convenient interpretability properties, I opt for using normalized pointwise mutual information (NPMI), which has been historically used for phrase detection (cite) and topic-coherence evaluation (cite).

We calculate the pointwise mutual information by taking the logarithm of the fraction of conditional and marginal word probabilities:

$$\text{pmi}_{kj} = \log_2 \frac{p(v_j|z_k)}{p(v_j)}$$

where $p(v_j|z_k)$ is the conditional probability of word j given the presence of topic z, and $p(v_j)$ is the probability of word j occurring.

A naive approach might include estimating these probabilities empirically:

$$p(v_j) = \frac{n_j}{\sum_i n_i}, \text{ and } p(v_j \mid z_k) = \frac{n_{jt}}{\sum_i n_{it}}$$

where n_j is the number of times word j occurs, n_{jt} is the number of times word j occurs in cluster t.

This would, however, overestimate the importance of rare words in the clusters where they appear. We can therefore opt for a mean-a-posteriori estimate under a symmetric dirichlet prior with an α *smoothing* parameter, which is analyticaly tractable:

$$p(v_j) = \frac{n_j + \alpha}{N\alpha + \sum_i n_i}, \text{ and } p(v_j \mid z_k) = \frac{n_{jt} + \alpha}{N\alpha + \sum_i n_{it}}$$

where N is the size of the vocabulary. In further analysis, $\alpha = 2$ will be used. Since regular PMI scores have no lower bound, we normalize them to obtain NPMI:

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

2.3.3. Combined Term Importance

To balance the semantic proximity of keywords to topic embeddings and cluster-term occurrences, a I introduce a combined approach, which consists of the geometric mean of min-max normalized lexical and semantic scores:

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

3. Experimental Methods

Since one of the main strengths of clustering approaches, that they can supposedly find the number of clusters in the data, and are not given this information a-priori, a good clustering topic model

should be able to faithfully replicate a human-assigned clustering of the data, and should be able to describe these clusters in a manner that is human-interpretable. I will therefore utilize datasets with gold-standard labels. In this section I will outline the criteria and considerations taken into account when designing an evaluation procedure:

1. The number of clusters in the topic model should preferably be not too far from the number of gold categories.
2. Preferably, if two points are in the same gold category, they should also belong together in the predicted clustering, while points that do not, shouldn't.
3. For topic modelling purposes, it is often preferable that the number of clusters is not overly large. Topic models should, in theory, aid the understanding of a corpus. Using a topic model becomes impractical when the number of topics one has to interpret is over a couple hundred.
4. Topics should be distinct and easily readable.

3.1. Datasets

In order to evaluate these properties, I used a number of openly available datasets with gold-standard category metadata. This included all clustering tasks from the new version of the Massive Text Embedding Benchmark `MTEB(eng, v2)` (cite). To avoid evaluating on the same corpus twice, the P2P variants of the tasks where used. In addition an annotated Twitter topic-classification dataset, and a BBC News dataset was used.

Dataset	Document Length	Corpus Size	Clusters
	N characters	N documents	N unique gold labels
ArXivHierarchicalClusteringP2P	1008.44±438.01	2048	23
BiorxivClusteringP2P.v2	1663.97±541.93	53787	26
MedrxivClusteringP2P.v2	1981.20±922.01	37500	51
StackExchangeClusteringP2P.v2	1091.06±808.88	74914	524
TwentyNewsgroupsClustering.v2	32.04±14.60	59545	20
TweetTopicClustering	165.66±68.19	4374	6
BBCNewsClustering	1000.46±638.41	2224	5

Table 1: Descriptive statistics of the datasets used for evaluation
Document length is reported as mean±standard deviation

3.2. Models

To compare Topeax with existing approaches, it was run on all corpora alongside BERTopic and Top2Vec. Implementations were sourced from the Turftopic (cite) Python package. For the main analysis, default hyperparameters were used from the original BERTopic and Top2Vec packages respectively, as these give different clusterings, despite having the same pipeline. All models were run with both the `all-MiniLM-L6-v2`, and the slightly larger and higher performing `all-mpnet-base-v2` sentence encoders (cite sbert) to to control for embedding size and quality. The models were fitted without filtering for stop words and uncommon terms, since state-of-the art topic models are able to handle such information without issues (cite S3).

3.3. Metrics

For evaluating model performance, both clustering quality and topic quality was evaluated. I evaluated the faithfulness of the predicted clustering to the gold labels using the Fowlkes-Mallows index (cite). The FMI, is very similar to the F1 score for classification, in that it also intends to balance precision and recall. Unlike F1, however, FMI uses the geometric mean of these quantities:

$$\text{FMI} = \frac{N_{\text{TP}}}{\sqrt{(N_{\text{TP}} + N_{\text{FP}}) \cdot (N_{\text{TP}} + N_{\text{FN}})}}$$

where N_{TP} is the number of pairs of points that get clustered together in both clusterings (true positives), N_{FP} is the number of pairs that get clustered together in the predicted clustering but not in the gold labels (false positives) and N_{FN} is the number of pairs that do not get clustered together in the predicted clustering, despite them belonging together in the gold labels (false negatives).

For topic quality, I adopt the methodology of (cite S3), with minor differences. I use GloVe embeddings (cite GloVe) for evaluating internal word embedding coherence instead of Skip-gram. As such, topic quality was evaluated on topic diversity d, external word embedding coherence C_{ex} using the `word2vec-google-news-300` word embedding model, as well as internal word embedding coherence C_{in}' with a GloVe model trained on each corpus. Ideally a model should both have high intrinsic and extrinsic coherence, and thus an aggregate measure of coherence can give a better estimate of topic quality: $\bar{C} = \sqrt{C_{\text{in}} \cdot C_{\text{ex}}}$. In addition an aggregate metric of topic quality can be calculated by taking the geometric mean of coherence and diversity $I = \sqrt{\bar{C} \cdot d}$.

3.4. Robustness checks
1. Hyperparameters (perplexity)
2. Corpus Subsampling

4. Results

4.1. Cluster Recovery