Identifying topics in unlabeled documents: a methodological guide

Łukasz Nawaro, Kristóf Gyódi, Michał Paliński, Katarzyna Śledziewska, Maciej Wilamowski

Executive summary

This guide is a methodological companion of the report “Towards a Human-Centric Internet: Challenges and Solutions”. The main goal of this study is to develop a visualization tool enabling the exploration of key technology challenges and related policy issues. Based on a text-mining methodology, we have examined and identified the specific topics discussed in a wide range of written media shared on social media platforms.

In this methodological guide, we describe various methods that could be used to automatically generate topics, optionally augmented with expert analysis. Later, we present how these methods can be benchmarked to find the one most suitable for our NGI dataset and its umbrella topics. The benchmarking method is based on a labeled news dataset: Reuters-21578. We examine how various unsupervised topic detection methods (Latent Dirichlet Allocation, Pachinko Allocation, t-SNE, doc2vec, SVD and bag-of-words, combined with suitable clustering algorithms such as k-means, Gaussian mixtures, and HDBSCAN) perform on this dataset.

We show the results and justify the choice of the model: t-SNE embeddings clustered with Gaussian mixtures. We also demonstrate that HDBSCAN clustering is a robust alternative to expert analysis, although with some demonstrable disadvantages. The main report presents a description of all narrow topics identified, as well as a deep dive into one umbrella topic. In this report, the descriptions of umbrella topics focus on the similarities and differences between the main and the alternative methods of assigning topics. The interactive results presenting both methods are available online: https://ngitopics.delabapps.eu.

Acknowledgement: This Report is part of a project that has received funding from the **European Union's Horizon 2020 research and innovation programme under grant agreement N°825652**

Introduction to topic modelling

Our main goal is to identify narrow topics discussed online within each of the six wide umbrella topics[1]. These specific topics are unknown, as our datasets contain thousands of documents without labels. One way to solve this problem is by reading these articles and assigning the topics manually. However, especially in the case of a large number of articles a human cannot perform this task well: not only is reading speed limited to a few hundred words per minute, but also our memory is imperfect. Topic modelling methods, however, can produce good-quality topics within a few seconds or minutes – depending on computing power, dataset size, and the chosen method. Topic modelling assigns a topic or multiple topics with given probabilities to a document. Therefore, such models are apt to find general themes in large collections of articles.

The grouping or clustering of documents can be performed in various ways. Giving such a task to various humans would yield us multiple different groups of clusters. Similarly, various algorithms have different ideas regarding what constitutes an optimal model. They may disagree on fundamentals: for example, should the clusters be of a similar size? The same algorithm with other settings can return vastly different clusters. The single best algorithm performing well regardless of dataset and metric may not exist. Still, we can test multiple algorithms with various settings on a labeled dataset which we know is similar to the dataset with unknown topics we attempt to cluster. It is likely that the superior method on the labeled dataset will generalize well to the other datasets.

Generative statistical models

One of the most widely used topic modelling methods is Latent Dirichlet Allocation (LDA). It is based on the assumption that topics can be represented by distribution over words, and documents – over topics (Blei et al., 2003). The generative process begins with drawing a Dirichlet distribution over topics, later it draws a topic index for each word, and finally draws the word from the topic (Hoffman et al., 2010). The iterative process assigns words to different topics until it finds a steady state. Finally, LDA provides us with shares of topics for each document. The corpus-level parameters in the *tomotopy* Python package we used are alpha and eta: the former determines the document-topic Dirichlet distribution and the latter concerns topic-word distribution.

Li and McCallum (2006) introduced Pachinko Allocation (PA), with a purported advantage over LDA in ”discover[ing] a large number of fine-grained, tightly-coherent topics”. It is a generalized form of LDA which can find smaller topics within larger topics, comparable to hierarchical LDA or Correlated Topic Model (CTM), but with some advantages over them. CTM’s flaws include quadratic complexity with regards to the covariance matrix parameter

[1] The umbrella topics are Environment, Sustainability & Resilience, Decentralising Power & Building Alternatives, Public Space & Sociality, Privacy, Identity & Data Governance, Trustworthy Information Flows, Cybersecurity & Democracy, Access, Inclusion & Justice. You can read more about our take on them in https://ngitopics.delabapps.eu/report.pdf.

estimations, and hierarchical LDA is less flexible in topic path sampling (Li and McCallum, 2006). The *tomotopy* parameters for PA are corresponding to LDA, with an additional parameter of subalpha: alpha for sub-topics.

A naïve way of assigning a single topic to a document is assigning the topic with the highest probability, as used in multiple research papers in recent years (Curiskis et al. (2020), Hagen (2018), Asghari et al. (2018), Jędrzejowicz and Zakrzewska (2017)). Another way is using a clustering technique (Bui et al., 2018), as the document is assigned to several topics with given probabilities and clustering joins together documents assigned to these topics in a similar way. We will explore clustering algorithms in the next section.

Clustering algorithms

Clustering algorithms can be broadly split into distance-based (such as K-means and CLARA/CLARANS) and density-based (DBSCAN/HDBSCAN) (Güngör and Özmen, 2017). Density-based clustering searches for areas with a high density of observations, which are separated by low-density areas (Kriegel et al., 2011). Simpler methods take into account only distance between observations (Patra et al., 2011) – observations which are close to each other are in the same cluster regardless of whether they are separated by an empty space or by a region dense with observations.

K-means finds K clusters, minimizing the squared distances between points and centroids of the cluster (Hackeling, 2017). First, each data point is assigned to the nearest of the K means – which are initially random. Second, the means are updated to be equal to the mean of the points in their cluster. The two steps are repeated until the steady state is reached (MacKay, 2003). Gaussian mixtures are a generalized case of k-means (Lücke and Forster, 2019), which use an expectation-maximization algorithm and can find overlapping clusters. Components are modeled using a multivariate normal distribution, initialized with k-means (Ouyang et al., 2004). A similarity between k-means and Gaussian mixtures is the importance of initial optimization settings (Su and Dy, 2007), but initialization algorithms effective in real-life applications – like k-means++ – were proposed (Arthur and Vassilvitskii, 2007) and are commonly used.

DBSCAN, introduced by Ester et al. (1996), is the oldest density-based clustering algorithm (Khan et al., 2014). It considers observations which are within a given distance denoted by epsilon to be connected. If the number of such observations is below the set threshold, the observation is assumed to be noise (Schubert et al., 2017). Noise is the part of the input to the algorithm which is meaningless and does not generate insights. The algorithm promises to be applicable in high-dimensional settings with noise (Khan et al., 2014). Early improvements of the algorithm focused on computational speed (El-Sonbaty et al., 2004), but also robustness to hyper-parameters (Yu et al., 2005). McInnes et al. (2017) published HDBSCAN library, which apart from low sensitivity to hyper-parameters has the advantage of allowing for clusters of different densities by “perform[ing] DBSCAN over varying epsilon values”.

Dimensionality reduction

Recent developments in natural language processing suggest that methods based on word embeddings or dimensionality reduction can result in more accurate classification of documents than generative statistical models. However, such methods do not produce lists of topics and topic words, but more abstract vector representations of documents. Therefore, in order to form the groups of documents, such techniques must always be used with a clustering algorithm for this purpose.

Bag-of-words matrices inform how often a particular word appears in a document. Their drawbacks include disregarding grammatical structure (Verberne et al., 2010), but it does not preclude them from being used in a topic modelling context (Blei et al., 2003). Instead of pure counts of words in documents, usually tf-idf transformation – a product of the number of occurrences of a term in the document and logarithm of the number of all documents divided by the number of documents in which the term occurs. In the framework of information theory, it means ”the amount of information of a term weighted by its occurrence probability” (Aizawa, 2003). Bag-of-words matrices are sparse, containing zeros in a vast majority of the rows. Computational constraints and characteristics of various models call for effective transformation of this high-dimensional matrix to a lower-dimensional space, while maintaining as much information about the observations as possible. Clustering algorithms are not as effective on a bag-of-words matrix as on a smaller matrix with a reduced number of dimensions.

Singular value decomposition, based on the work of Beltrami and Jordan among others (Stewart, 1993), decomposes a matrix A into a product of three matrices U, Σ, and V transposed. U and V are unitary matrices, which means that the product of the matrix and its transpose is an identity matrix, and Σ is a diagonal matrix (Golub and Reinsch, 1971). After discarding all but k largest singular values from the Σ matrix, we get the best least-squares k-dimensional approximation of the original matrix. This is the base for Latent Semantic Analysis (Dumais, 2004), which aims to find associations between documents (Hofmann, 2013). LSA performs poorly on a very low and very high number of dimensions – its peak performance is achieved near 300 dimensions (Landauer et al., 1998).

Van der Maaten and Hinton (2008) presented t-distributed Stochastic Neighbor Embedding (t-SNE), which has grown to be one of the most often used unsupervised learning techniques in multiple fields, such as data about a single cell (Zhou and Jin, 2020). t-SNE does not suffer from the ”crowding problem” of its predecessor (SNE). The problem causes points close to each other in the high-dimensional space to be put too far from each other in the low-dimensional representation. A low-dimensional distribution with heavier tails, in t-SNE’s case t-distribution, is a computationally effective solution to the problem. t-SNE applied on text data may deliver well-separated clusters like in Sikorskiy et al. (2018), although clusters can be misleadingly found by t-SNE also in random data (Wattenberg et al., 2016). Typically, t-SNE is used to reduce the number of dimensions to 2 or 3 with non-interpretable axes.

Perplexity is the crucial hyperparameter in t-SNE. Van der Maaten and Hinton (2008) claim that t-SNE is "fairly robust to changes in perplexity", although others argue that manual or automatic work in selecting the optimal perplexity is usually required (Cao and Wang, 2017). Interpretation of perplexity is a measure of "the effective number or neighbors" (Van der Maaten and Hinton, 2008). The larger the perplexity, the more neighboring observations are taken into account, so the global structure becomes more important (Wattenberg et al., 2016) at the cost of local structure.

Kobak and Berens (2018) suggested to first run t-SNE on high perplexity settings to find global structure and use the output as initialization for low perplexity settings – the process they call *perplexity annealing*, which achieves slightly better results than initialization with PCA. Their later work (Kobak and Berens, 2019) uses *perplexity averaging* based on Lee et al. (2015) and available in t-SNE implementations. Perplexity averaging uses a multi-scale kernel (Koban and Berens, 2019):

$$\frac{1}{\sigma_i} exp(-\frac{d^2}{2\sigma_i^2}) + \frac{1}{\tau_i} exp(-\frac{d^2}{2\tau_i^2})$$

Finding the right way to gain insights from t-SNE is more art than science. Some issues are clusters which are not clearly separable (especially with low perplexity) and meaningless distances between clusters despite clear connection in data (Wattenberg et al., 2016). In order to put observations into separate clusters, using a clustering algorithm is required. Despite the fact that in t-SNE distances between points are ignored in favor of joint probabilities both in high-dimensional and low-dimensional spaces (Van der Maaten and Hinton, 2008), which may deteriorate results of distance-based clustering, methods such as k-means are used in various applications in conjunction with low-dimensional embeddings. These applications include tumor prognosis (Abdelmoula et al., 2016), safety decision support systems (Dhalmahapatra et al., 2019), partial discharge faults (Kumar et al., 2020), and SARS-CoV-2 mutation datasets (Hozumi et al., 2021).

Document vectors

Word embeddings assign a multi-dimensional vector to each word. Arithmetic operations on vectors can be performed: a well-known example is *king + woman - man = queen*. The similarity of vectors can be calculated using the cosine function, enabling the analysis of related terms. Mikolov, Chen, Corrado and Dean (2013) introduced word2vec, which "preserve[s] the linear regularities among words". There are two architectures proposed: continuous bag-of-words, which predicts the word based on its nearest surroundings; and more commonly used continuous skip-gram, which predicts the word's nearest surroundings based on the word. The model is based on Feedforward Neural Net Language Model (NNLM), but with the hidden layer removed and the "projection layer shared for all words" – only the input and output layers are left unchanged. The removal of the hidden layer significantly reduces the computational cost.

Documents consist of words, but how to get document meaning from (or using) word vectors is not straightforward. Early approaches like averaging words or matrix-vector

operations to combine vectors with parsing were superseded by doc2vec, introduced by Le and Mikolov (2014). The output of doc2vec consists not only of word vectors but also of vectors that represent a paragraph or a document. The method does not have limitations regarding text length or the necessity to use tuning or parse trees, and combines classifying a random sample of words based on paragraph token (distributed bag-of-words) with a model similar to continuous bag-of-words, but with another token representing the paragraph. The authors find that doc2vec is a state-of-the-art solution with only 3.82% error rate compared to 8.10% for a bag-of-words model on an information retrieval task. Performance on sentiment analysis on reviews from Rotten Tomatoes and IMDB is also superior to traditional approaches like Support Vector Machines and other types of neural networks. Other authors also find that doc2vec is a robust solution for document clustering: in patent clustering, Kim et al. (2020) use k-means on Doc2vec achieving an accuracy of 0.3548, close to a proposed non-standard technique of deep embedded clustering (0.3714). Zhang et al. (2018) use the output of doc2vec to cluster descriptions of Web APIs with k-means, achieving superior results to i.a. the combination of LDA with k-means.

Reuters analysis

In order to ascertain what the optimal method is, we need ground truth: the desired classification outcome. We have no valuable ground truth regarding topics in our datasets, as classification schemes vary between websites, if a website uses categories at all.

There are multiple labelled datasets used in the literature. Examples of the most common ones are *20newsgroups* and *Reuters*, another dataset notable for including news articles is *BBC News* (Greene and Cunningham, 2006). Their goal is to provide comparable environments for testing machine learning methods, both supervised and unsupervised. Values of a chosen metric are then compared to ascertain which method prevails. Results are reproducible and characteristics of a dataset are known, allowing other researchers to build on methodology.

BBC News consists of 2225 news articles: the five categories included are business, entertainment, sport, politics, and tech. They are fairly balanced and easily separable, so we rejected it for being too simple: there is not enough challenge to distinguish between well-performing and poor-performing algorithms.

The 20newsgroups dataset contains 18846 posts from 20 Usenet groups. Cleaning of the dataset can be done in multiple ways. Removing headers is obvious and easy to perform precisely, but removing footers and quotes is imprecise and results in many empty documents. There are no standardized rules of writing in Usenet groups. Some terms are very specific to a particular target group and are rarely used otherwise. Moreover, while the target groups are balanced, some of them are close in theme to each other, while others are highly dissimilar: there are five computing groups, including two very similar groups (comp.os.ms-windows.misc and comp.windows.x), two groups on religion (alt.atheism and soc.religion.christian), and only one group on medical science – although there are four scientific groups in total.

Our chosen approach is to test the clustering methods on the Reuters dataset[2]. It has been used in machine learning research since the 90s. Applications of the Reuters dataset range from classical papers on traditional ML algorithms like Support Vector Machines (Dumais, 1998) or Bayesian probabilistic generative models (McCallum & Nigam, 1998) to gradient boosting (Zdrojewska et al., 2018) and novel term weighting strategies (Dogan and Uysal, 2020). Its properties resemble our datasets: it has multiple topics of unequal size, contains technical terms, and it is not too sensitive to cleaning methods or prone to volatile results.

If at least 80% of articles in a category belonged to another category, these categories were merged. After this procedure, articles which belong to more than 1 category or to a category with less than 10 articles were discarded. In the final Reuters dataset, 9096 articles are split into 44 categories.

[2] https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

In our experiment, we can compare the 44 groups created by the tested methodologies to the 44 ground truth groups. We treated the testing as an unsupervised problem: the labels were hidden from the algorithm. The only information the algorithm was given was the number of desired clusters, equal to the number of ground truth categories.

There are multiple metrics which measure the quality of the assignment. The assignment is a multiclass problem (there are more than 2 classes), but not a multi-label problem – there is only one target class per document. Consequently, precision (ratio of true positives to the sum of true positives and false positives: a positive in a ground truth topic is a document located in a cluster which is assigned to the topic) can be defined in three basic ways:

- micro-average: precision is calculated "globally". All true positives are added together and divided by the sum of all true positives and all false positives (i.e. the number of all articles). This method checks for the largest number of articles correctly assigned, but the importance of small groups is low.
- macro-average: precision is calculated for each class separately. The final macro-averaged precision is the arithmetic mean of all precision values in classes. All classes are equally important.
- weighted: precision is also calculated for each class separately. The final weighted precision is the average of precision values weighted by the number of articles in each class: the number of true positives (articles assigned to a correct cluster) and false negatives (articles assigned to an incorrect cluster). This is a compromise approach, which assigns more importance to larger classes like micro-average, but does not treat large and small classes equally like macro-average.

$$ARI = \frac{\binom{n}{2}(TP+TN) - ((TP+FP)(TP+FN)+(FN+TN)(TP+TN))}{\binom{n}{2}^2 - ((TP+FP)(TP+FN)+(FN+TN)(FP+TN))}$$

Two other methods commonly used in the literature on multiclass classification are adjusted Rand index (ARI) and Normalized Mutual Information (NMI). ARI is defined (Santos & Embrechts, 2009) as:where *TP* is the number of true positives, *FP* – false positives, *FN* – false negatives, and *TN* – true negatives. ARI's goal is to compute similarity between two partitions of a dataset. If one of the partitions is the ground truth, it becomes an appropriate metric. Adjustment for chance takes place so that independent random labelling results in a value close to 0 and partitioning identical to the ground truth returns 1.

NMI is an information-theoretic measure based on Shannon entropy. Entropy is based on probabilities that randomly chosen nodes are assigned to particular classes. The formula for mutual information is (Emmons et al. 2009):

$$I(X,Y) = \sum_x \sum_y P(x,y) log(\frac{P(x,y)}{P(x)P(y)})$$

It is further normalized to the range between 0 and 1 to be comparable with other metrics. All metrics are computed using the *scikit-learn* Python library (Pedregosa et al. 2011).

A further consideration is how to match the ground truth groups to the new clusters. Naturally, the 44 clusters produced by the algorithm contain combinations of articles that are in different ground truth groups. Assigning clusters to ground truth topics can be done in two ways. The simpler one is assigning them to the best group overall -- if in a cluster there are 300 documents in group A, 200 documents in group B and 150 documents in group C, it is always assigned as representing group A. However, one could argue that if there are *n* clusters and *n* categories, the function assigning groups should be bijective, i.e. if there was already a larger cluster of documents which all belong to group A, the formerly described cluster should be assigned to B. Such assignment is done using Kuhn-Munkres (KM) algorithm. Naturally, algorithms achieve higher micro-averaged precision scores with the first method, although not necessarily with the macro-averaged and weighted precision: results will be worsened by zero precision for some ground truth groups. As described in the next sections, the two methods of assignment will differently affect the various clustering methods. HDBSCAN works better with KM, as it produces unevenly-sized clusters. On the other hand, distance-based k-means and Gaussian mixtures can assign articles from a large topic to a few different clusters. However, with Kuhn-Munkres assignment, a cluster can be assigned to a topic only once, which means that even when the clusters are perfectly consistent and contain one topic only, only the largest cluster will match the ground truth, while the smaller clusters will be wrongly assigned to different ground truth topics. As the differences in cluster sizes are small in distance-based methods, they will be at a disadvantage.

There is randomness inherent in the methods. Initialisation influences distance-based clustering and generative statistical models. We compute 15 iterations for seeds ranging from 0 to 14 (inclusive) and take the mean score of a particular metric across iterations as the final result. It reduces randomness and allows for reproducibility.

The methods we tested on the dataset are the following:

- **Bag-of-words (word count / tf-idf) + k-means; SVD normalized / not normalized + k-means / Gaussian mixtures**. The high-dimensional matrices are not applicable for other clustering algorithms than k-means due to computational complexity. These matrices lose as little information as possible: only the word order is ignored. The bag-of-words matrices are tested in two ways: untransformed and transformed by tf-idf. The matrix from SVD is used in two versions: not normalized and normalized to unit norm. Both versions of the SVD matrix are clustered with two distance-based clustering methods.
- **t-SNE (single perplexity / perplexity annealing / perplexity averaging) + k-means / Gaussian mixtures / HDBSCAN**. Single perplexities range from 5 to 150, covering a larger range of perplexities than suggested by van der Maaten and Hinton (2008). Kobak and Berens' perplexity annealing and averaging was also used. Perplexity annealing ranged from 20-5 to 300-150; equivalent values were used for averaging. Additionally, averaging with one or two yet smaller perplexity values was tested, to bring the average down and more in line with single perplexity values. For example, for the 20-5 case perplexity 2 was added, while for 300-150 – 10 and 2. (See

Appendix A1 for a full list of perplexity settings.) HDBSCAN does not allow for setting the number of clusters explicitly, but has a few parameters related to it, such as minimum cluster size (mcs). The optimal value of this particular parameter is proportional to the size of the dataset. To get clusters, we either use the default method (*labels_* of the HDBSCAN object) or *get_clusters* method of the single linkage tree if there are more clusters than ground truth topics. The optimal epsilon value is chosen to bring the number of clusters as close to 44 as possible. Values of minimum_cluster_size and epsilon parameters (varying across iterations) are in Appendix A2.

- **Doc2vec + k-means**. Vector sizes we tested are 10, 15, 25, 50, 75 and 100. The higher the number of epochs, the better the results, but the improvement is lower and lower with each additional epoch. The number of epochs ranges from 300 to 500 in increments of 25. Document vectors of particular size are then clustered with k-means for similar reasons as SVD and bag-of-words matrices.
- **LDA / PA + maximum assignment (naïve) / k-means**. Both LDA and PA models were used with default settings from *tomotopy* Python package. The numbers of topics we tested are 44 (equal to the number of ground truth topics), 50, 60, 80 and 100. The lowest number of topics was used for the naive assignment method; all resulting probabilities matrices, also for 44 topics, had k-means applied to them.

Table 1 presents the five methodologies with the highest precision, calculated for each combination of ground truth assignment (maximum or KM) and the five (the remaining ones are presented in Appendix A3) measures for precision:

Table 1. Best mean results of particular methods on a given dataset

	ranking 1	**ranking 2**	**ranking 3**	**ranking 4**	**ranking 5**
Kuhn-Munkres micro-averaged precision	***t-SNE HDBSCAN (0.5623)***	PA naive (0.3146)	SVD (0.3133)	LDA naive (0.3054)	**t-SNE Gaussian (0.2666)**
maximum micro-averaged precision	**t-SNE Gaussian (0.8045)**	SVD (0.8043)	t-SNE k-means (0.8029)	*t-SNE HDBSCAN (0.795)*	PA naive (0.7638)
Kuhn-Munkres macro-averaged precision	***t-SNE HDBSCAN (0.3536)***	SVD (0.2725)	t-SNE k-means (0.2406)	**t-SNE Gaussian (0.24)**	PA k-means (0.2377)

maximum macro-averaged precision	***t-SNE HDBSCAN (0.3302)***	SVD (0.2474)	**t-SNE Gaussian (0.2102)**	t-SNE k-means (0.2035)	PA k-means (0.1988)
Kuhn-Munkres weighted precision	SVD (0.828)	***t-SNE HDBSCAN (0.8279)***	t-SNE k-means (0.8184)	**t-SNE Gaussian (0.8169)**	LDA k-means (0.8095)
maximum weighted precision	SVD (0.7752)	**t-SNE Gaussian (0.7658)**	t-SNE k-means (0.7647)	t-SNE HDBSCAN (0.7582)	PA naive (0.7189)
Kuhn-Munkres NMI	***t-SNE HDBSCAN (0.5644)***	**t-SNE Gaussian (0.5236)**	t-SNE k-means (0.5233)	SVD (0.5226)	PA naive (0.4986)
maximum NMI	t-SNE k-means (0.695)	**t-SNE Gaussian (0.6929)**	SVD (0.6769)	t-SNE HDBSCAN (0.6556)	PA naive (0.6027)
Kuhn-Munkres ARI	***t-SNE HDBSCAN (0.4188)***	PA naive (0.2193)	LDA naive (0.2007)	SVD (0.1526)	PA k-means (0.1308)
maximum ARI	**t-SNE Gaussian (0.8269)**	t-SNE k-means (0.8244)	SVD (0.798)	PA naive (0.7738)	LDA naive (0.7734)

We use the following process, taking into account results in Table 1:

- The first method that is clearly rejected is **doc2vec**. In no metric does it enter the top 5 of best algorithms. Its relatively best results are achieved in Kuhn-Munkres weighted precision and maximum NMI, but they are still inferior.
- Second, we reject generative statistical models (**LDA** and **PA**). Their results are particularly good with Kuhn-Munkres assignment in micro-averaged precision and NMI, but for all metrics there exists a preferable topic modelling method. Creating topics with k-means rarely results in an improvement over the naïve method, and whenever generative statistical models achieve a good score, it is with the naïve method.
- SVD delivers a robust performance, but apart from weighted precision (regardless of assignment method) t-SNE with some clustering algorithm achieves a better result. Moreover, the difference in Kuhn-Munkres weighted precision between SVD and t-SNE with HDBSCAN is negligible. Consequently, we should reject **SVD**.
- We are left with three clustering algorithms using t-SNE embeddings. As expected, with unevenly-sized clusters, HDBSCAN prevails over distance-based methods with

Kuhn-Munkres assignment. The difference between distance-based clustering algorithms, k-means and Gaussian mixtures, is minor. In this work, we want to check issues arising in the NGI datasets. They are well-structured and concern carefully selected topics. Had HDBSCAN been a clearly superior method, we would have chosen it. But as results are comparable[3] and we know that there is little actual noise in the NGI datasets thanks to the careful choice of terms in umbrella topics, **distance-based methods are preferred**. HDBSCAN's advantage is identifying the number of clusters automatically, so we include it **as a robustness check** for possible applicability in similar tasks.

It is now necessary to choose optimal settings for the distance-based method. In both NMI (Table 2) and ARI (Table 3) with maximum assignment method, it is the single-perplexity t-SNE clustering method which clearly prevails. The optimal perplexity is in the range 50-75. The two distance-based clustering methods, k-means and Gaussian mixtures, achieve comparable scores: in ARI, Gaussian mixtures are slightly better, in NMI, the order is reversed.

Table 2. Maximum NMI best results

Perplexity	Perplexity type	cluster	mean	std
50	**single**	k-means	0.695013	0.007431
60	**single**	**Gaussian mixtures**	0.692948	0.011142
75	**single**	k-means	0.690748	0.005370
60	**single**	k-means	0.690212	0.005230
50	**single**	**Gaussian mixtures**	0.689442	0.005638

Table 3. Maximum ARI best results

Perplexity	Perplexity type	cluster	mean	std
60	**single**	**Gaussian mixtures**	0.826931	0.015148
90	**single**	k-means	0.824371	0.010343
75	**single**	k-means	0.823584	0.007872

[3] HDBSCAN is worse on maximum assignment method, better on Kuhn-Munkres assignment method, and while HDBSCAN's top results are good, there is a large number of settings with which it performs poorly. Compare Appendix B4 and B5 with C1 and C2 to see higher stability of distance-based clustering methods.

50	**single**	k-means	0.822905	0.013291
150-30	annealing	k-means	0.822256	0.008979

25 best results (by mean result on all iterations) for each metric are available in Appendices B1-B10.

The main method will be **t-SNE, single perplexity 50, Gaussian mixtures**. The difference between k-means and Gaussian mixtures is usually small, Gaussian mixtures achieve higher scores in 7 out of 10 available metrics (see Table 1 and Appendix A3), and Gaussian mixtures may be preferable in some settings due to more information possible to infer, especially with overlapping clusters. In our work, it makes no difference, but future research may find this property useful. As we want to distinguish general topics and leave specifics to expert analysis, we choose **15** as the number of components (clusters) in Gaussian mixtures. Perplexity is within the optimal range for the Reuters dataset and the clearly superior result in NMI.

HDBSCAN with a similar embedding and clustering pipeline serves a robustness check in this deliverable. HDBSCAN's results on the Reuters dataset are close to distance-based clustering even with maximum assignment method in weighted and micro-averaged precision metrics, which shows that it may be strong in identifying small and coherent clusters. Minimum number of points in a cluster was chosen to be the number of articles in the dataset divided by 250: in the Reuters dataset, the typical values of minimum cluster size for best results ranged from 25 to 50. The t-SNE settings from the main pipeline with single perplexity 50 are suitable for HDBSCAN as well, achieving top or second best results and being robust to parameter change (see Appendix C1 for NMI and C2 for ARI)

There is no guarantee that the optimal solution for one dataset will be the optimal solution for another dataset; we can only say that this is our best guess. More importantly though, we reject solutions which do not work on the Reuters dataset: both those which are reasonable, but do not work for the particular dataset, and those which simply are not adequate for text mining purposes in the problem of clustering articles. The latter rejections are crucial.

Clustering results

A detailed analysis with the main method is presented in the main deliverable, available at: https://ngitopics.delabapps.eu/report.pdf.

In this chapter, three major insights will be demonstrated:

- HDBSCAN finds the majority of valuable clusters identified by Gaussian mixtures with expert analysis
- Expert analysis is often superior to automated analysis, as HDBSCAN clusters are sometimes meaningful
- HDBSCAN noise removal is too aggressive

For each umbrella topic, three groups of clusters will be shown to demonstrate the respective insight:

- Clusters which are found both by Gaussian mixtures with expert analysis and HDBSCAN
- Clusters which are found only by one of these methods
- Clusters which are found by Gaussian mixtures with expert analysis, but considered to be noise by HDBSCAN

Consistency between Gaussian mixtures with expert analysis and HDBSCAN is rather common. To keep descriptions short, only a few examples will be shown.

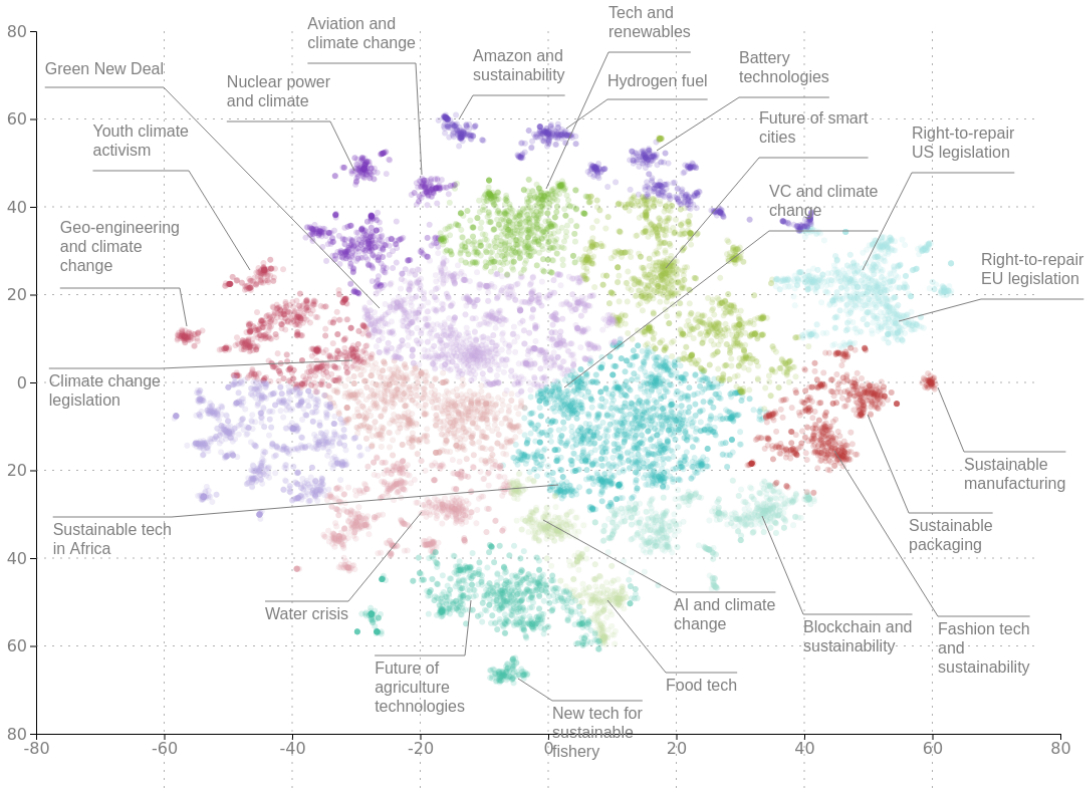
In the following sections, the upper chart contains Gaussian mixtures + expert analysis; while the lower chart is HDBSCAN. Click them to see their interactive version.

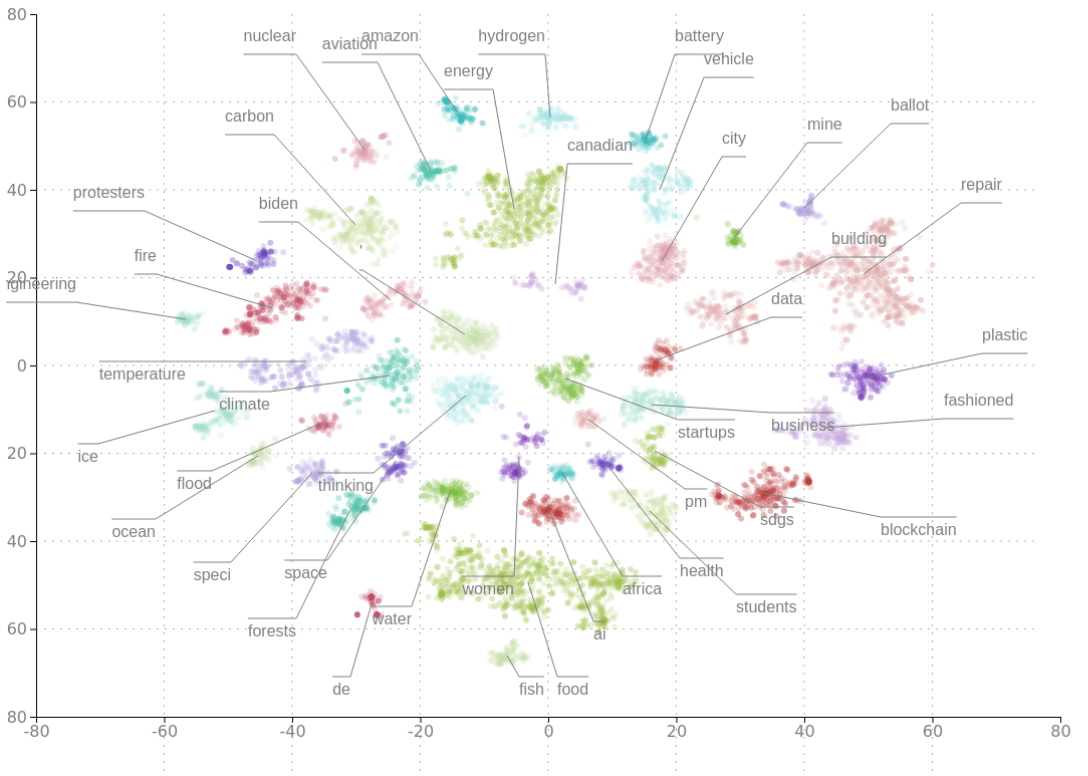
Points are positioned on the map according to the t-SNE embedding. Their color represents the cluster they are assigned to. The color palette is random and the similarity of colors should not be considered meaningful (e.g. clusters with different shades of green are not related to each other). Note that articles in the “noise” (meaningless input) cluster in HDBSCAN were removed from the HDBSCAN chart.

In the case of Gaussian mixtures, the maps present topic tags prepared with expert analysis. The online versions of maps also include topic keywords that were chosen from the list of 30 top words (ranked by: the number of occurrences in the cluster divided by square root of the number of occurrences in the whole umbrella topic). The occurrences were counted for the stemmed (root) form. For clarity, the keywords are words from the dataset which return a given root form when stemmed. Cluster names were also assigned by expert analysis.

For HDBSCAN, both cluster names and keywords (top 5 words) were chosen automatically. These automatic topic tags are presented on the maps.

Environment, Sustainability & Resilience





Both methods

One classical element: “water” (or “water crisis” in expert analysis) is on the bottom of the chart: in HDBSCAN just over a large cluster “food”.

Blockchain is a cluster which was found by expert analysis and HDBSCAN.

The top contains a cluster related to (renewable) energy surrounded by e.g. aviation, hydrogen, and battery technologies. Gaussian mixture puts it into one cluster, split into smaller topicsby expert analysis, HDBSCAN considers them separate clusters.

Only one method

In HDBSCAN, “fire” and “ice” are not only opposite classical elements, but also two topics placed close to each other on the left. Neither was deemed to be a crucial social topic based on expert analysis (top), especially as the “fire” topic concerns Australia more generally.

The large HDBSCAN cluster on the bottom about food belongs to two different topics in Gaussian mixtures. The left part of it concerns agriculture, and the right side is strictly about food.

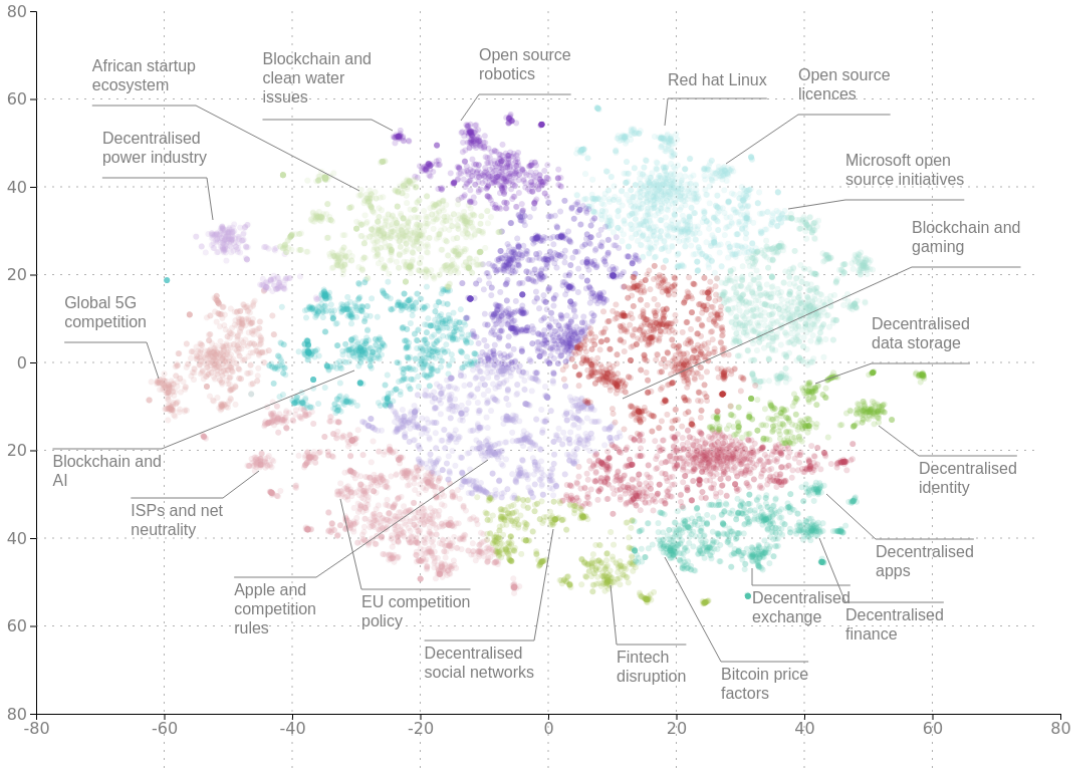
A large topic on the right is “right to repair”, common to HDBSCAN and Gaussian mixtures, split by expert analysis to EU and US parts. Geographical considerations were lost by the automatic clustering methods. This could be expected: in density-based clustering with a sufficiently large epsilon, a path between points representing articles can be found. A smaller epsilon would not be able to distinguish any valuable clusters. In distance-based clustering, the automatic clusters alone are simply too large.

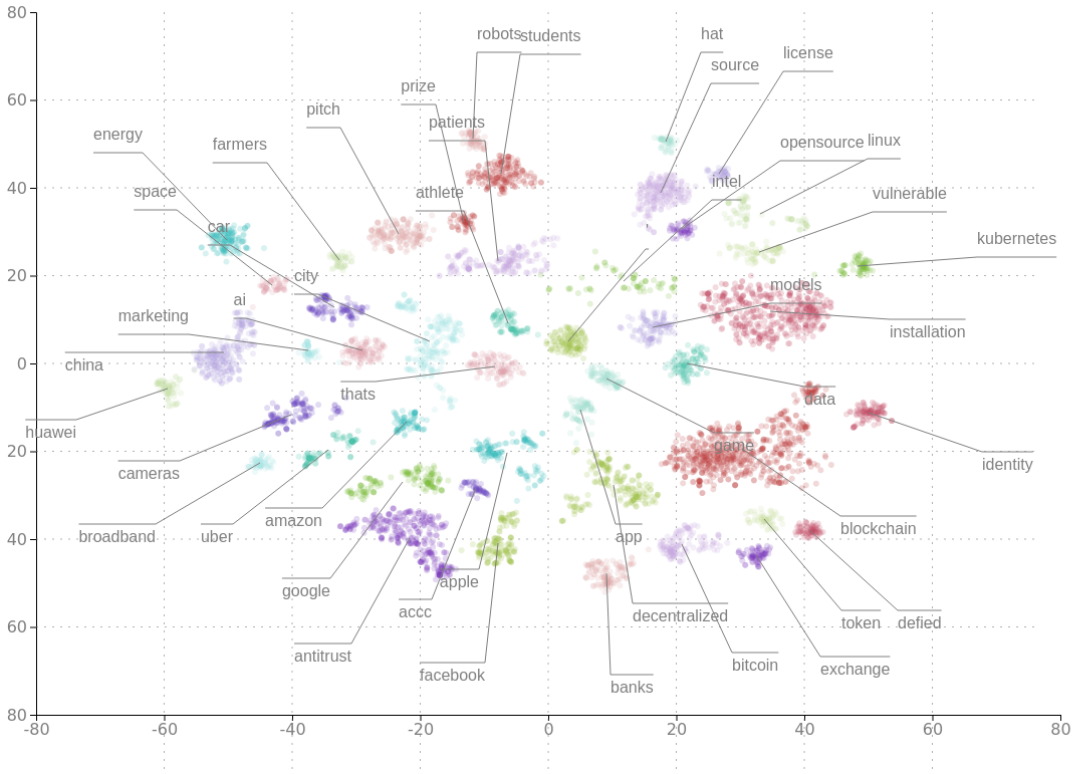
HDBSCAN topics about “thinking” or the too general “climate” cluster would be better classified as noise.

Noise

The small manufacturing cluster on the right, found by expert analysis, was deemed to be noise by HDBSCAN.

Decentralising Power & Building Alternatives





Both methods

Decentralised identity, open source robotics and decentralised power industry are examples of clusters found both by HDBSCAN and expert analysis. Fintech disruption was also found by HDBSCAN. Although its name is “banks”, keywords make it clear that this is a general fintech cluster.

Left side of the chart has a cluster about China, in HDBSCAN additionally split into a generic “China” topic and a smaller cluster on Huawei and 5G, which was also found by expert analysis.

Only one method

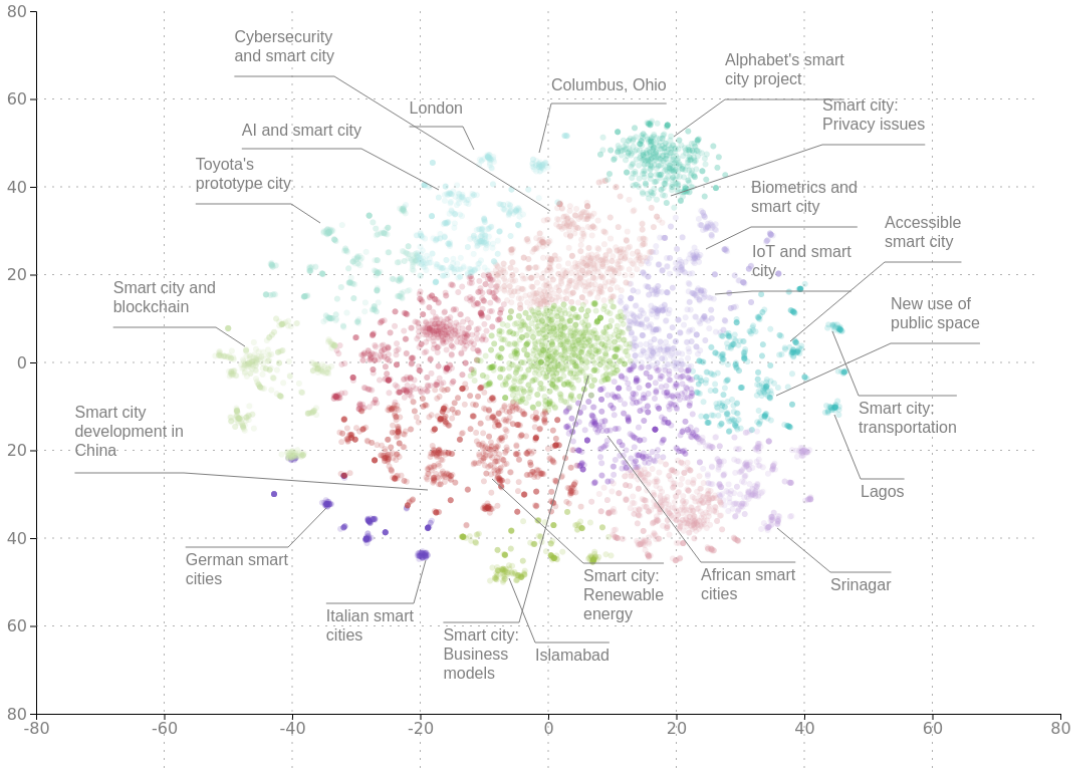
The HDBSCAN cluster about “antitrust” at the bottom is surrounded by documents on Google and Facebook. Expert analysis found that the “antitrust” cluster has a part about EU policies.

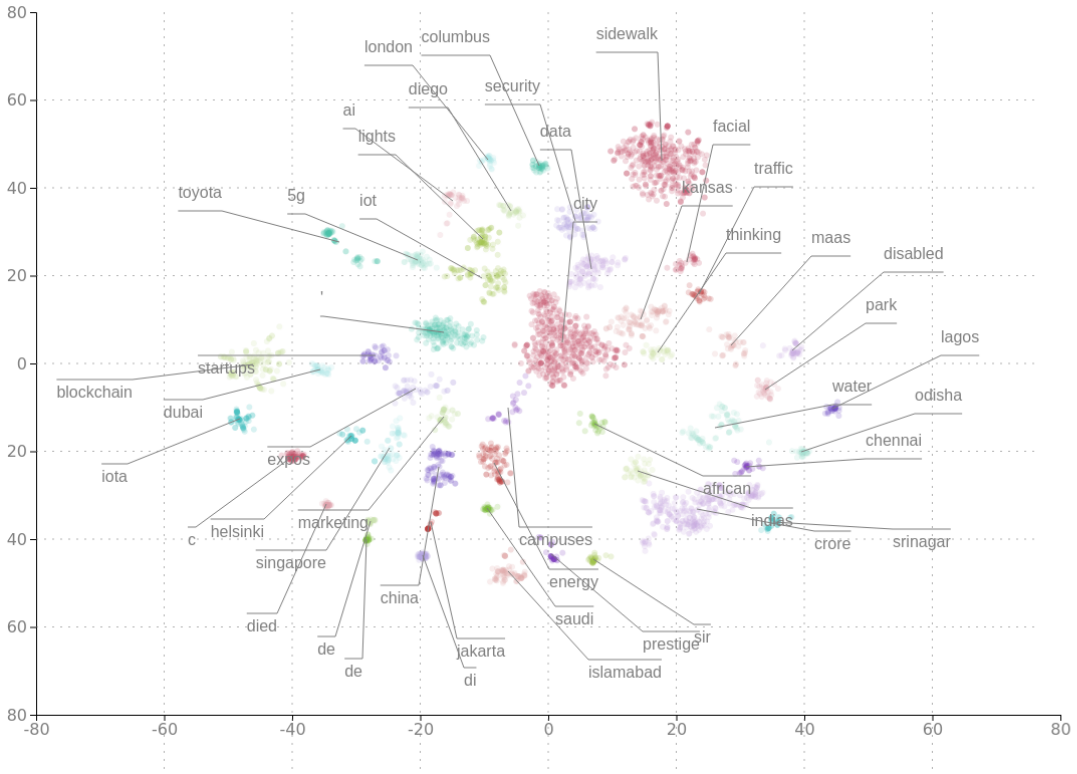
HDBSCAN’s clusters “pitch” and “installation” are rather general and not insightful.

Noise

HDBSCAN wrongly assigned articles near (-25, 40) to noise, despite the fact that they constitute a clear and meaningful cluster on startups in Africa.

Public Space & Sociality





Both methods

Blockchain and related cryptocurrency IOTA are on the left side of the chart, next to Dubai, which invests heavily in new technologies, including blockchain. They were grouped into one cluster by Gaussian mixtures, but assigned keywords show also these smaller subtopics.

African smart cities were found by HDBSCAN and expert analysis, even though they are in a noisy environment. Another example of a notably similar cluster is “Islamabad” in the bottom of the charts. Expert analysis and HDBSCAN found valuable clusters on Toyota’s prototype city (top left) and cybersecurity in smart cities (5, 35).

The cluster on Alphabet’s smart city project (Sidewalk Labs) is identical in Gaussian mixtures, expert analysis, and HDBSCAN.

Only one method

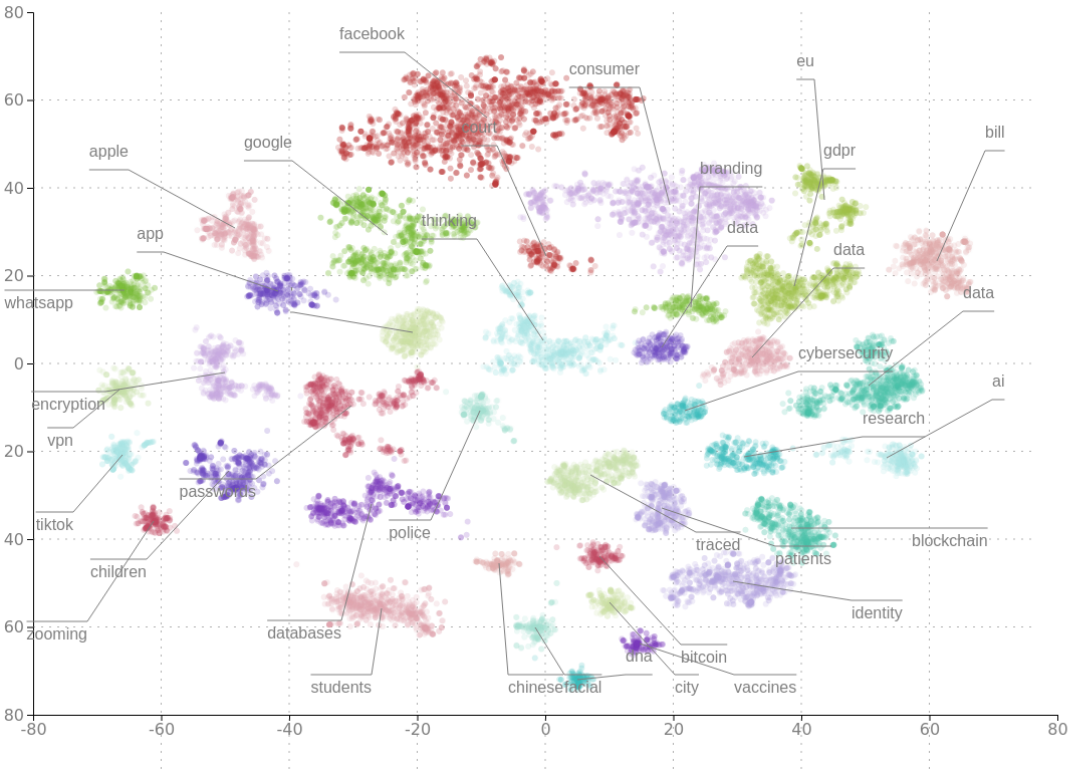
The general topic in the middle of the chart has no discernable general topic in HDBSCAN, although a part of it concerns business models according to expert analysis and infrastructure in general by Gaussian mixtures.

The HDBSCAN clusters in the middle (“kansas”, “city”, “thinking”) add little value.

Noise

Transportation in smart cities found by expert analysis in the right of the chart belongs to noise according to HDBSCAN.

Privacy, Identity & Data Governance



Both methods

In the top of the charts, the cluster on Facebook is notably similar between Gaussian mixtures and HDBSCAN.

WhatsApp, TikTok and Zoom occupy the left side of the charts. They are grouped by HDBSCAN and expert analysis in a similar manner.

The topic “Student data privacy” is well visible in both expert analysis and HDBSCAN.

Only one method

Although the similarity between the cluster on Facebook between two automatic methods is clear, expert analysis in this area is beneficial: the topic about political campaigns, while strongly connected to Facebook, deserves to be noted.

Gaussian mixtures split the articles on the left side of the charts into two clusters: cyan – more technical, about encryption; and purple – focused on issues faced by end users in their daily life: childrearing and work.

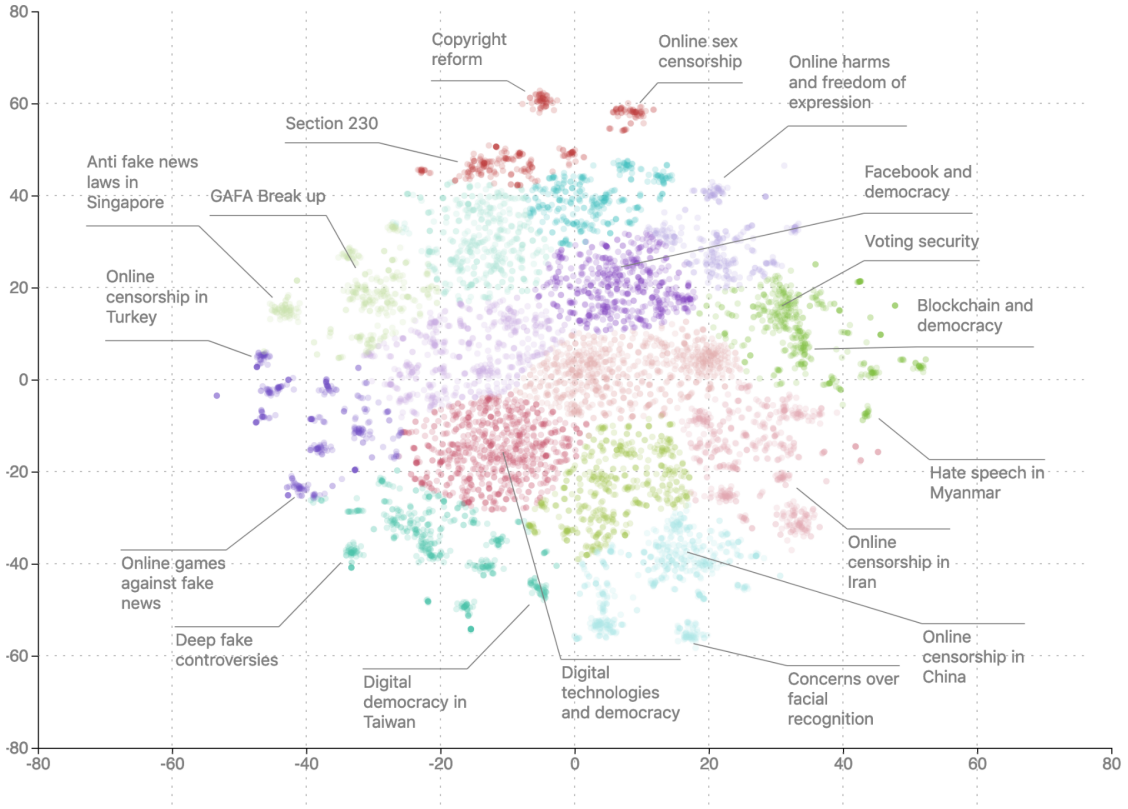
Bottom right (closer to center) contains clusters on medical issues: HDBSCAN found contact tracing and patients or healthcare in general. In expert analysis, the overarching theme of pandemic was found.

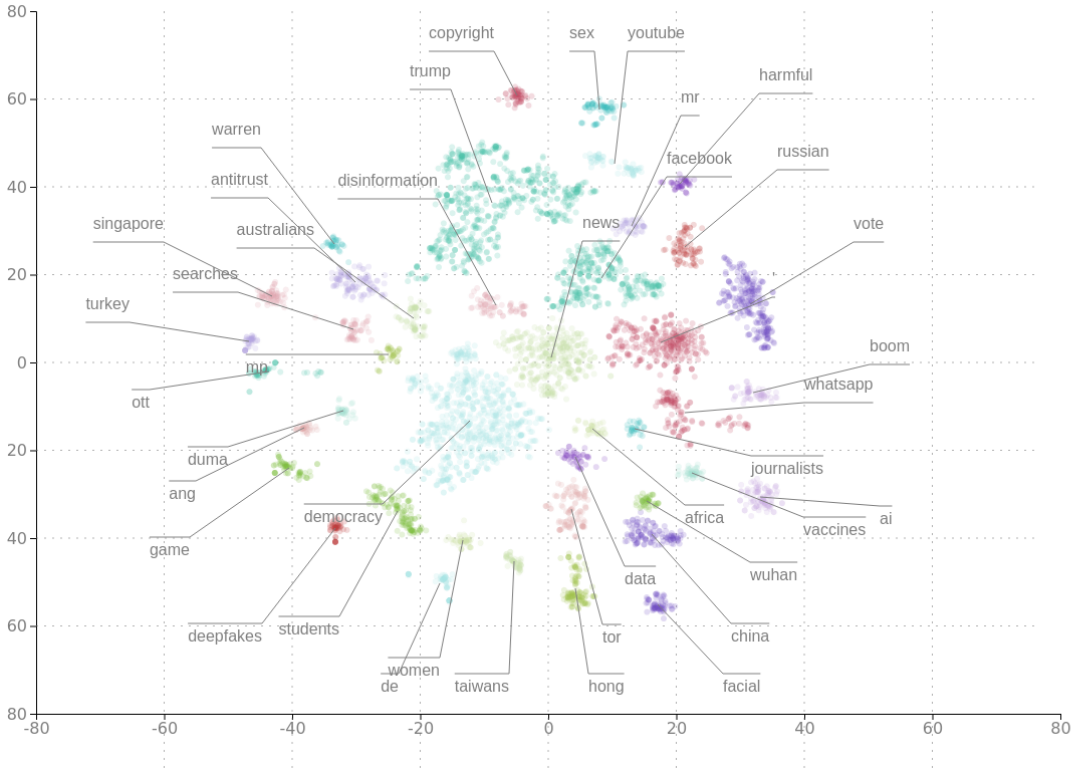
In the top right, the broad “eu” HDBSCAN cluster contains articles about both EU-US Privacy Shield and post-Brexit GDPR, different topics which deserve their own clusters.

Noise

Articles about Aadhaar, the digital ID assigned to Indian citizens, disappear from the bottom right of the HDBSCAN chart.

Trustworthy Information Flows, Cybersecurity & Democracy





Both methods

HDBSCAN and expert clusters on Asian countries (Singapore and Turkey), and online media in general, are on the left side of the chart.

Bottom right of the HDBSCAN chart contains topics related to China: Taiwan, Hong Kong, Wuhan (especially censoring information about the coronavirus), facial recognition and China itself, usually about censorship on social media. They are treated as a coherent group by Gaussian mixtures. Facial recognition is not an exclusively Chinese issue, keywords for the HDBSCAN cluster include Clearview, a controversial American facial recognition company. The overarching theme is correctly found by Gaussian mixtures, while expert analysis and HDBSCAN provide detailed subtopics.

The “democracy” topic is similar in HDBSCAN and Gaussian mixtures, although Gaussian mixtures define it a little bit broader, while HDBSCAN considers articles on the outside of the Gaussian cluster as noise.

Only one method

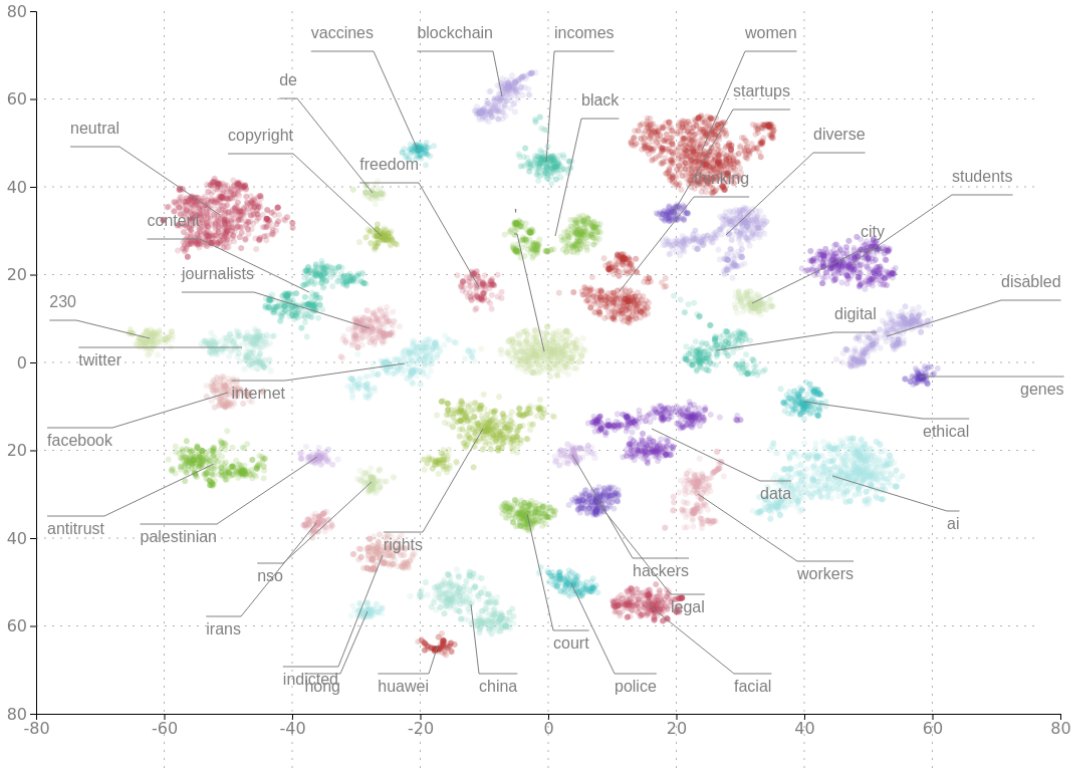
The expert topic about “section 230” is treated by HDBSCAN as a part of a broader “Trump” cluster. This cluster is very general, containing articles assigned by Gaussian mixtures to three separate clusters (USA, copyright issues, hate speech).

Noise

Hate speech in Myanmar, an expert topic on the right of the chart, was classified as noise by HDBSCAN due to being small.

Some parts of the Gaussian copyrights cluster at the top were deemed to be noise by HDBSCAN, including proposed Polish legislation on limiting tech companies from banning content on their platforms and articles about the Parler social network.

Access, Inclusion & Justice



Both methods

China is a topic of interest in this group of articles as well, with a general China cluster at the bottom. Similarly to the “Trustworthy Information Flows, Cybersecurity & Democracy” umbrella topic, facial recognition is placed close to China, but Gaussian mixtures treat it as a separate cluster. Large topics found by HDBSCAN can be located also in expert analysis.

A broad cluster on the judiciary system is above the “China” topic. HDBSCAN considers a part of it as noise, but finds two coherent clusters tagged “court” and “legal” as well. The first cluster was alsofound by expert analysis.

In the bottom right, there are similar clusters about AI found by Gaussian mixtures and HDBSCAN, with no further split required by expert analysis. “Ethical coding” is also common to HDBSCAN and expert analysis, in Gaussian mixtures it is a part of a cluster about inclusivity, just like “blockchain” forms a part of Gaussian “income inequality” cluster.

Another interesting group of HDBSCAN clusters is located in the top right: clusters on diversity and women surrounded the cluster on startups. The gender inequality cluster is very clear in all methods.

Only one method

HDBSCAN clusters in this umbrella topic tend to be small and consistent. One exception is “black”, the right part of which is a “racial inequality” cluster according to expert analysis. The left part of it is different: it focuses on art, often (but not exclusively) created by artists from the Black community, but not particularly concerning social ideas.

Noise

The Palantir controversy regarding American ICE in the bottom right is not visible in the HDBSCAN chart, due to being a part of the noise cluster.

Conclusion

Our results present two robust methods of finding clusters of articles. Automatic clustering with Gaussian mixtures proved to deliver a satisfactory general overview of umbrella topics, while expert analysis augmented the results with valuable insights, some of which could not be found by either distance- or density-based clustering.

HDBSCAN's clusters are of reasonable quality for an automated method: any expert topics were also found by HDBSCAN. An advantage of HDBSCAN is its elasticity in producing differently-sized clusters. Consequently, for exploratory research without expert knowledge, HDBSCAN is the method to choose. The methodology is therefore applicable in an even wider range of problems.

However, we showed that expert analysis is a valuable extension of automated clustering. T-SNE finds intricate structure in the documents, and assigning documents will never be perfect if the algorithm does not know whether to search for articles concerning the same geographical area, the same company, the same technology, or the same social issue. Social and EU-related topics which can be identified with expert analysis were joined together by HDBSCAN and Gaussian mixtures. The prime example is the right-to-repair topic in the "Environment, Sustainability & Resilience" umbrella topic, which is focused on EU and US legislation in different places. Both automatic clustering methods see a fairly coherent cluster on right-to-repair.

It is necessary to discuss limitations of applicability of the main method. First, the methodology has been shown to work only on news articles. Social media posts or comments may have different characteristics and it does not have to be the case that t-SNE with distance-based clustering delivers the best results. Second, if one wants to find clusters as coherent as possible, is not worried about losing some articles, or expects the dataset to be noisy, HDBSCAN's results on Kuhn-Munkres assignment show that it may be the optimal algorithm. Third, Gaussian mixtures are not superior in finding small topics: increasing the number of clusters seems to make clusters in the middle of the chart rather random. This problem is common with distance-based methods. We mitigate this problem by expert analysis, while the use of HDBSCAN is another possible solution.

For a deeper understanding of the topics, please read the main deliverable: https://ngitopics.delabapps.eu/report.pdf.

References

Abdelmoula, W. M., Balluff, B., Englert, S., Dijkstra, J., Reinders, M. J., Walch, A., ... & Lelieveldt, B. P. (2016). Data-driven identification of prognostic tumor subpopulations using spatially mapped t-SNE of mass spectrometry imaging data. *Proceedings of the National Academy of Sciences*, *113*(43), 12244-12249.

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, *39*(1), 45-65.

Asghari, M., Sierra-Sosa, D., & Elmaghraby, A. (2018). Trends on health in social media: Analysis using Twitter topic modeling. In *2018 IEEE international symposium on signal processing and information technology (ISSPIT)* (pp. 558-563). IEEE.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.

Bui, Q. V., Sayadi, K., Amor, S. B., & Bui, M. (2017). Combining latent dirichlet allocation and k-means for documents clustering: Effect of probabilistic based distance measures. In *Asian Conference on Intelligent Information and Database Systems* (pp. 248-257). Springer, Cham.

Cao, Y., & Wang, L. (2017). Automatic selection of t-SNE Perplexity. *arXiv preprint arXiv:1708.03229*.

Curiskis, S. A., Drake, B., Osborn, T. R., & Kennedy, P. J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing & Management*, *57*(2), 102034.

Dhalmahapatra, K., Shingade, R., Mahajan, H., Verma, A., & Maiti, J. (2019). Decision support system for safety improvement: An approach using multiple correspondence analysis, t-SNE algorithm and K-means clustering. *Computers & Industrial Engineering*, *128*, 277-289.

Dogan, T., & Uysal, A. K. (2020). A novel term weighting scheme for text classification: TF-MONO. *Journal of Informetrics*, *14*(4), 101076.

Dumais, S. (1998). Using SVMs for text categorization. *IEEE Intelligent Systems*, *13*(4), 21-23.

Dumais, S. (2004). Latent semantic analysis. *Annual review of information science and technology*, *38*(1), 188-230.

El-Sonbaty, Y., Ismail, M. A., & Farouk, M. (2004). An efficient density based clustering algorithm for large databases. In *16th IEEE international conference on tools with artificial intelligence* (pp. 673-677). IEEE.

Emmons, S., Kobourov, S., Gallant, M., & Börner, K. (2016). Analysis of network clustering algorithms and cluster quality metrics at scale. *PloS one*, *11*(7), e0159161.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD* (Vol. 96, No. 34, pp. 226-231).

Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. In *Linear algebra* (pp. 134-151). Springer, Berlin, Heidelberg.

Greene, D., & Cunningham, P. (2006). Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning* (pp. 377-384).

Güngör, E., & Özmen, A. (2017). Distance and density based clustering algorithm using Gaussian kernel. *Expert Systems with Applications*, *69*, 10-20.

Hackeling, G. (2017). *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd.

Hagen, L. (2018). Content analysis of e-petitions with topic modeling: How to train and evaluate LDA models?. *Information Processing & Management*, *54*(6), 1292-1307.
Hoffman, M., Bach, F., & Blei, D. (2010). Online learning for latent Dirichlet allocation. *Advances in neural information processing systems*, *23*, 856-864.
Hofmann, T. (2013). Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*.
Jędrzejowicz, J., & Zakrzewska, M. (2017). Word embeddings versus LDA for topic assignment in documents. In *International Conference on Computational Collective Intelligence* (pp. 357-366). Springer, Cham.
Hozumi, Y., Wang, R., Yin, C., & Wei, G. W. (2021). UMAP-assisted K-means clustering of large-scale SARS-CoV-2 mutation datasets. *Computers in biology and medicine*, *131*, 104264.
Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014). DBSCAN: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)* (pp. 232-238). IEEE.
Kim, J., Yoon, J., Park, E., & Choi, S. (2020). Patent document clustering with deep embeddings. *Scientometrics*, 1-15.
Kobak, D., & Berens, P. (2018). The art of using t-SNE for single-cell transcriptomics. bioRxiv.
Kobak, D., & Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, *10*(1), 1-14.
Kriegel, H. P., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(3), 231-240.
Kumar, H., Shafiq, M., Hussain, G. A., Kumpulainen, L., & Kauhaniemi, K. (2020, October). Classification of PD Faults Using Features Extraction and K-Means Clustering Techniques. In *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)* (pp. 919-923). IEEE.
Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.
Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196). PMLR.
Lee, J. A., Peluffo-Ordóñez, D. H., & Verleysen, M. (2015). Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, *169*, 246-261.
Li, W., & McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning* (pp. 577-584).
Lücke, J., & Forster, D. (2019). k-means as a variational EM approximation of Gaussian mixture models. *Pattern Recognition Letters*, *125*, 349-356.
MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.
McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).
McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Ouyang, M., Welsh, W. J., & Georgopoulos, P. (2004). Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, *20*(6), 917-923.
Patra, B. K., Nandi, S., & Viswanath, P. (2011). A distance based clustering method for arbitrary shaped clusters in large datasets. *Pattern Recognition*, *44*(12), 2862-2870.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825-2830.
Santos, J. M., & Embrechts, M. (2009). On the use of the adjusted Rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks* (pp. 175-184). Springer, Berlin, Heidelberg.
Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, *42*(3), 1-21.
Sikorskiy, S., Metsker, O., Yakovlev, A., & Kovalchuk, S. (2018). Machine learning based text mining in electronic health records: cardiovascular patient cases. In *International Conference on Computational Science* (pp. 818-824). Springer, Cham.
Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM review*, *35*(4), 551-566.
Su, T., & Dy, J. G. (2007). In search of deterministic methods for initializing K-means and Gaussian mixture clustering. *Intelligent Data Analysis*, *11*(4), 319-338.
Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).
Vassilvitskii, S., & Arthur, D. (2006). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035).
Verberne, S., Boves, L., Oostdijk, N., & Coppen, P. A. (2010). What is not in the Bag of Words for Why-QA?. *Computational Linguistics*, *36*(2), 229-245.
Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to use t-SNE effectively. *Distill*, *1*(10), e2.
Yu, X., Zhou, D., & Zhou, Y. (2005). A new clustering algorithm based on distance and density. In *Proceedings of ICSSSM'05. 2005 International Conference on Services Systems and Services Management, 2005.* (Vol. 2, pp. 1016-1021). IEEE.
Zdrojewska, A., Dutkiewicz, J., Jędrzejek, C., & Olejnik, M. (2018). Comparison of the novel classification methods on the reuters-21578 corpus. In *International Conference on Multimedia and Network Information System* (pp. 290-299). Springer, Cham.
Zhang, X., Liu, J., Cao, B., Xiao, Q., & Wen, Y. (2018). Web service recommendation via combining Doc2Vec-based functionality clustering and DeepFM-based score prediction. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)* (pp. 509-516). IEEE.
Zhou, B., & Jin, W. (2020). Visualization of single cell RNA-Seq data using t-SNE in R. In *Stem Cell Transcriptional Networks* (pp. 159-167). Humana, New York, NY.

Appendix

A1) List of t-SNE settings used

Perplexity averaging:
20, 5, 2
30, 7, 2
35, 8, 3
50, 12, 2
50, 5, 3
70, 10, 4
80, 12, 2
100, 15, 3
100, 20, 2
100, 20, 8, 2
100, 4, 2
150, 30, 6, 2
150, 50, 4
150, 7, 2
200, 40, 3
200, 60, 5
225, 75, 10, 3
250, 100, 5
250, 125, 6
250, 90, 6
300, 150, 10, 2

Perplexity averaging and perplexity annealing:
20, 5
30, 7
35, 8
50, 12
50, 5
70, 10
80, 12
100, 15
100, 20
100, 20
100, 4
150, 30
150, 50
150, 7
200, 40
200, 60

225, 75
250, 100
250, 125
250, 90
300, 150

Single perplexity:
4
5
7
8
10
12
15
20
30
40
50
60
75
90
100
125
150

A2) List of HDBSCAN settings used

Minimum cluster sizes: 25, 35, 50, 75, 100, 150, 200, 300

Base for epsilon: 0.50, 0.52, 0.54, 0.56, 0.58, 0.60, 0.62, 0.64, 0.66, 0.68, 0.70, 0.72, 0.74, 0.76, 0.78, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, 0.94, 0.96, 0.98, 1.00, 1.02, 1.04, 1.06, 1.08, 1.10, 1.12, 1.14, 1.16, 1.18, 1.20, 1.22, 1.24, 1.26, 1.28, 1.40, 1.80, 2.20, 2.60, 3.00, 3.40, 3.80, 4.20, 4.60, 5.00, 5.40, 5.80, 6.20, 6.60, 7.00, 7.40, 7.80

Epsilons for iterations:
0 - 0.4533, 0.4715, 0.4896, 0.5077, 0.5259, 0.5440, 0.5621, 0.5803, 0.5984, 0.6165, 0.6347, 0.6528, 0.6709, 0.6891, 0.7072, 0.7253, 0.7435, 0.7616, 0.7797, 0.7979, 0.8160, 0.8341, 0.8523, 0.8704, 0.8885, 0.9067, 0.9248, 0.9429, 0.9611, 0.9792, 0.9973, 1.0155, 1.0336, 1.0517, 1.0699, 1.0880, 1.1061, 1.1243, 1.1424, 1.1605, 1.2693, 1.6320, 1.9947, 2.3573, 2.7200, 3.0827, 3.4453, 3.8080, 4.1707, 4.5333, 4.8960, 5.2587, 5.6213, 5.9840, 6.3467, 6.7093, 7.0720
1 - 0.4600, 0.4784, 0.4968, 0.5152, 0.5336, 0.5520, 0.5704, 0.5888, 0.6072, 0.6256, 0.6440, 0.6624, 0.6808, 0.6992, 0.7176, 0.7360, 0.7544, 0.7728, 0.7912, 0.8096, 0.8280, 0.8464, 0.8648, 0.8832, 0.9016, 0.9200, 0.9384, 0.9568, 0.9752, 0.9936, 1.0120, 1.0304, 1.0488, 1.0672, 1.0856, 1.1040, 1.1224, 1.1408, 1.1592, 1.1776, 1.2880, 1.6560, 2.0240, 2.3920,

2.7600, 3.1280, 3.4960, 3.8640, 4.2320, 4.6000, 4.9680, 5.3360, 5.7040, 6.0720, 6.4400, 6.8080, 7.1760
2 - 0.4667, 0.4853, 0.5040, 0.5227, 0.5413, 0.5600, 0.5787, 0.5973, 0.6160, 0.6347, 0.6533, 0.6720, 0.6907, 0.7093, 0.7280, 0.7467, 0.7653, 0.7840, 0.8027, 0.8213, 0.8400, 0.8587, 0.8773, 0.8960, 0.9147, 0.9333, 0.9520, 0.9707, 0.9893, 1.0080, 1.0267, 1.0453, 1.0640, 1.0827, 1.1013, 1.1200, 1.1387, 1.1573, 1.1760, 1.1947, 1.3067, 1.6800, 2.0533, 2.4267, 2.8000, 3.1733, 3.5467, 3.9200, 4.2933, 4.6667, 5.0400, 5.4133, 5.7867, 6.1600, 6.5333, 6.9067, 7.2800
3 - 0.4733, 0.4923, 0.5112, 0.5301, 0.5491, 0.5680, 0.5869, 0.6059, 0.6248, 0.6437, 0.6627, 0.6816, 0.7005, 0.7195, 0.7384, 0.7573, 0.7763, 0.7952, 0.8141, 0.8331, 0.8520, 0.8709, 0.8899, 0.9088, 0.9277, 0.9467, 0.9656, 0.9845, 1.0035, 1.0224, 1.0413, 1.0603, 1.0792, 1.0981, 1.1171, 1.1360, 1.1549, 1.1739, 1.1928, 1.2117, 1.3253, 1.7040, 2.0827, 2.4613, 2.8400, 3.2187, 3.5973, 3.9760, 4.3547, 4.7333, 5.1120, 5.4907, 5.8693, 6.2480, 6.6267, 7.0053, 7.3840
4 - 0.4800, 0.4992, 0.5184, 0.5376, 0.5568, 0.5760, 0.5952, 0.6144, 0.6336, 0.6528, 0.6720, 0.6912, 0.7104, 0.7296, 0.7488, 0.7680, 0.7872, 0.8064, 0.8256, 0.8448, 0.8640, 0.8832, 0.9024, 0.9216, 0.9408, 0.9600, 0.9792, 0.9984, 1.0176, 1.0368, 1.0560, 1.0752, 1.0944, 1.1136, 1.1328, 1.1520, 1.1712, 1.1904, 1.2096, 1.2288, 1.3440, 1.7280, 2.1120, 2.4960, 2.8800, 3.2640, 3.6480, 4.0320, 4.4160, 4.8000, 5.1840, 5.5680, 5.9520, 6.3360, 6.7200, 7.1040, 7.4880
5 - 0.4867, 0.5061, 0.5256, 0.5451, 0.5645, 0.5840, 0.6035, 0.6229, 0.6424, 0.6619, 0.6813, 0.7008, 0.7203, 0.7397, 0.7592, 0.7787, 0.7981, 0.8176, 0.8371, 0.8565, 0.8760, 0.8955, 0.9149, 0.9344, 0.9539, 0.9733, 0.9928, 1.0123, 1.0317, 1.0512, 1.0707, 1.0901, 1.1096, 1.1291, 1.1485, 1.1680, 1.1875, 1.2069, 1.2264, 1.2459, 1.3627, 1.7520, 2.1413, 2.5307, 2.9200, 3.3093, 3.6987, 4.0880, 4.4773, 4.8667, 5.2560, 5.6453, 6.0347, 6.4240, 6.8133, 7.2027, 7.5920
6 - 0.4933, 0.5131, 0.5328, 0.5525, 0.5723, 0.5920, 0.6117, 0.6315, 0.6512, 0.6709, 0.6907, 0.7104, 0.7301, 0.7499, 0.7696, 0.7893, 0.8091, 0.8288, 0.8485, 0.8683, 0.8880, 0.9077, 0.9275, 0.9472, 0.9669, 0.9867, 1.0064, 1.0261, 1.0459, 1.0656, 1.0853, 1.1051, 1.1248, 1.1445, 1.1643, 1.1840, 1.2037, 1.2235, 1.2432, 1.2629, 1.3813, 1.7760, 2.1707, 2.5653, 2.9600, 3.3547, 3.7493, 4.1440, 4.5387, 4.9333, 5.3280, 5.7227, 6.1173, 6.5120, 6.9067, 7.3013, 7.6960
7 - 0.5000, 0.5200, 0.5400, 0.5600, 0.5800, 0.6000, 0.6200, 0.6400, 0.6600, 0.6800, 0.7000, 0.7200, 0.7400, 0.7600, 0.7800, 0.8000, 0.8200, 0.8400, 0.8600, 0.8800, 0.9000, 0.9200, 0.9400, 0.9600, 0.9800, 1.0000, 1.0200, 1.0400, 1.0600, 1.0800, 1.1000, 1.1200, 1.1400, 1.1600, 1.1800, 1.2000, 1.2200, 1.2400, 1.2600, 1.2800, 1.4000, 1.8000, 2.2000, 2.6000, 3.0000, 3.4000, 3.8000, 4.2000, 4.6000, 5.0000, 5.4000, 5.8000, 6.2000, 6.6000, 7.0000, 7.4000, 7.8000
8 - 0.5067, 0.5269, 0.5472, 0.5675, 0.5877, 0.6080, 0.6283, 0.6485, 0.6688, 0.6891, 0.7093, 0.7296, 0.7499, 0.7701, 0.7904, 0.8107, 0.8309, 0.8512, 0.8715, 0.8917, 0.9120, 0.9323, 0.9525, 0.9728, 0.9931, 1.0133, 1.0336, 1.0539, 1.0741, 1.0944, 1.1147, 1.1349, 1.1552, 1.1755, 1.1957, 1.2160, 1.2363, 1.2565, 1.2768, 1.2971, 1.4187, 1.8240, 2.2293, 2.6347, 3.0400, 3.4453, 3.8507, 4.2560, 4.6613, 5.0667, 5.4720, 5.8773, 6.2827, 6.6880, 7.0933, 7.4987, 7.9040
9 - 0.5133, 0.5339, 0.5544, 0.5749, 0.5955, 0.6160, 0.6365, 0.6571, 0.6776, 0.6981, 0.7187, 0.7392, 0.7597, 0.7803, 0.8008, 0.8213, 0.8419, 0.8624, 0.8829, 0.9035, 0.9240, 0.9445,

0.9651, 0.9856, 1.0061, 1.0267, 1.0472, 1.0677, 1.0883, 1.1088, 1.1293, 1.1499, 1.1704, 1.1909, 1.2115, 1.2320, 1.2525, 1.2731, 1.2936, 1.3141, 1.4373, 1.8480, 2.2587, 2.6693, 3.0800, 3.4907, 3.9013, 4.3120, 4.7227, 5.1333, 5.5440, 5.9547, 6.3653, 6.7760, 7.1867, 7.5973, 8.0080

10 - 0.5200, 0.5408, 0.5616, 0.5824, 0.6032, 0.6240, 0.6448, 0.6656, 0.6864, 0.7072, 0.7280, 0.7488, 0.7696, 0.7904, 0.8112, 0.8320, 0.8528, 0.8736, 0.8944, 0.9152, 0.9360, 0.9568, 0.9776, 0.9984, 1.0192, 1.0400, 1.0608, 1.0816, 1.1024, 1.1232, 1.1440, 1.1648, 1.1856, 1.2064, 1.2272, 1.2480, 1.2688, 1.2896, 1.3104, 1.3312, 1.4560, 1.8720, 2.2880, 2.7040, 3.1200, 3.5360, 3.9520, 4.3680, 4.7840, 5.2000, 5.6160, 6.0320, 6.4480, 6.8640, 7.2800, 7.6960, 8.1120

11 - 0.5267, 0.5477, 0.5688, 0.5899, 0.6109, 0.6320, 0.6531, 0.6741, 0.6952, 0.7163, 0.7373, 0.7584, 0.7795, 0.8005, 0.8216, 0.8427, 0.8637, 0.8848, 0.9059, 0.9269, 0.9480, 0.9691, 0.9901, 1.0112, 1.0323, 1.0533, 1.0744, 1.0955, 1.1165, 1.1376, 1.1587, 1.1797, 1.2008, 1.2219, 1.2429, 1.2640, 1.2851, 1.3061, 1.3272, 1.3483, 1.4747, 1.8960, 2.3173, 2.7387, 3.1600, 3.5813, 4.0027, 4.4240, 4.8453, 5.2667, 5.6880, 6.1093, 6.5307, 6.9520, 7.3733, 7.7947, 8.2160

12 - 0.5333, 0.5547, 0.5760, 0.5973, 0.6187, 0.6400, 0.6613, 0.6827, 0.7040, 0.7253, 0.7467, 0.7680, 0.7893, 0.8107, 0.8320, 0.8533, 0.8747, 0.8960, 0.9173, 0.9387, 0.9600, 0.9813, 1.0027, 1.0240, 1.0453, 1.0667, 1.0880, 1.1093, 1.1307, 1.1520, 1.1733, 1.1947, 1.2160, 1.2373, 1.2587, 1.2800, 1.3013, 1.3227, 1.3440, 1.3653, 1.4933, 1.9200, 2.3467, 2.7733, 3.2000, 3.6267, 4.0533, 4.4800, 4.9067, 5.3333, 5.7600, 6.1867, 6.6133, 7.0400, 7.4667, 7.8933, 8.3200

13 - 0.5400, 0.5616, 0.5832, 0.6048, 0.6264, 0.6480, 0.6696, 0.6912, 0.7128, 0.7344, 0.7560, 0.7776, 0.7992, 0.8208, 0.8424, 0.8640, 0.8856, 0.9072, 0.9288, 0.9504, 0.9720, 0.9936, 1.0152, 1.0368, 1.0584, 1.0800, 1.1016, 1.1232, 1.1448, 1.1664, 1.1880, 1.2096, 1.2312, 1.2528, 1.2744, 1.2960, 1.3176, 1.3392, 1.3608, 1.3824, 1.5120, 1.9440, 2.3760, 2.8080, 3.2400, 3.6720, 4.1040, 4.5360, 4.9680, 5.4000, 5.8320, 6.2640, 6.6960, 7.1280, 7.5600, 7.9920, 8.4240

14 - 0.5467, 0.5685, 0.5904, 0.6123, 0.6341, 0.6560, 0.6779, 0.6997, 0.7216, 0.7435, 0.7653, 0.7872, 0.8091, 0.8309, 0.8528, 0.8747, 0.8965, 0.9184, 0.9403, 0.9621, 0.9840, 1.0059, 1.0277, 1.0496, 1.0715, 1.0933, 1.1152, 1.1371, 1.1589, 1.1808, 1.2027, 1.2245, 1.2464, 1.2683, 1.2901, 1.3120, 1.3339, 1.3557, 1.3776, 1.3995, 1.5307, 1.9680, 2.4053, 2.8427, 3.2800, 3.7173, 4.1547, 4.5920, 5.0293, 5.4667, 5.9040, 6.3413, 6.7787, 7.2160, 7.6533, 8.0907, 8.5280

A3) Best mean results of particular methods on a given dataset, ranks 6-9

	ranking 6	ranking 7	ranking 8	ranking 9

Kuhn-Munkres micro-averaged precision	t-SNE k-means (0.2583)	PA k-means (0.2372)	LDA k-means (0.2336)	doc2vec (0.2244)
maximum micro-averaged precision	LDA naive (0.7622)	LDA k-means (0.7613)	PA k-means (0.7595)	doc2vec (0.7456)
Kuhn-Munkres macro-averaged precision	LDA k-means (0.2358)	LDA naive (0.2299)	PA naive (0.2288)	doc2vec (0.21)
maximum macro-averaged precision	LDA k-means (0.1955)	PA naive (0.1776)	LDA naive (0.173)	doc2vec (0.1666)
Kuhn-Munkres weighted precision	PA k-means (0.8083)	doc2vec (0.799)	LDA naive (0.7987)	PA naive (0.7968)
maximum weighted precision	LDA naive (0.7145)	PA k-means (0.7134)	LDA k-means (0.7111)	doc2vec (0.7026)
Kuhn-Munkres NMI	LDA naive (0.4939)	PA k-means (0.4752)	LDA k-means (0.4731)	doc2vec (0.4561)
maximum NMI	LDA naive (0.597)	doc2vec (0.5969)	LDA k-means (0.5941)	PA k-means (0.5926)
Kuhn-Munkres ARI	LDA k-means (0.1197)	t-SNE Gaussian (0.1165)	t-SNE k-means (0.1077)	doc2vec (0.0977)
maximum ARI	LDA k-means (0.757)	PA k-means (0.7481)	t-SNE HDBSCAN (0.7338)	doc2vec (0.729)

B1) Maximum micro-averaged precision best results

	Perplexity	**Perplexity type**	**cluster**	**mean**	**std**

Model					
t-SNE	60	single	Gaussian mixtures	0.804507	0.005970
SVD normalized	NA	NA	Gaussian mixtures	0.804251	0.008359
SVD normalized	NA	NA	k-means	0.803489	0.008900
t-SNE	75	single	k-means	0.802866	0.003410
t-SNE	60	single	k-means	0.802272	0.003371
t-SNE	50	single	k-means	0.801693	0.003341
t-SNE	300-150	annealing	Gaussian mixtures	0.801327	0.004091
t-SNE	40	single	Gaussian mixtures	0.801246	0.006061
t-SNE	90	single	k-means	0.801019	0.003934
t-SNE	200-60	annealing	Gaussian mixtures	0.800799	0.005368
t-SNE	200-60	annealing	k-means	0.800762	0.004690
t-SNE	40	single	k-means	0.800704	0.006289
t-SNE	300-150	annealing	k-means	0.800037	0.005258
t-SNE	250-90	annealing	Gaussian mixtures	0.799927	0.006190
t-SNE	90	single	Gaussian mixtures	0.799751	0.004371
t-SNE	150-30	averaging	Gaussian mixtures	0.799230	0.005435
t-SNE	250-125	annealing	k-means	0.798945	0.003732
t-SNE	100-20	averaging	k-means	0.798373	0.006318

t-SNE	100-20	averaging	k-means	0.798373	0.006318
t-SNE	200-40	annealing	Gaussian mixtures	0.798094	0.005853
t-SNE	250-100	annealing	Gaussian mixtures	0.797911	0.006077
t-SNE	50	single	Gaussian mixtures	0.797897	0.003322
t-SNE	75	single	Gaussian mixtures	0.797596	0.003624
t-SNE	150-30	averaging	k-means	0.797596	0.004420
t-SNE	250-100	annealing	k-means	0.797332	0.005361

B2) Maximum macro-averaged precision best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
t-SNE	150-7	averaging	HDBSCAN	25.0	0.330174	0.029620
t-SNE	100-15	averaging	HDBSCAN	25.0	0.329363	0.022969
t-SNE	50-12	annealing	HDBSCAN	25.0	0.327625	0.012410
t-SNE	100	single	HDBSCAN	25.0	0.326334	0.000000
t-SNE	20	single	HDBSCAN	25.0	0.320762	0.023015
t-SNE	20	single	HDBSCAN	25.0	0.320762	0.023015
t-SNE	30	single	HDBSCAN	25.0	0.317041	0.036898
t-SNE	35-8	averaging	HDBSCAN	25.0	0.315521	0.031352
t-SNE	250-90-6	averaging	HDBSCAN	25.0	0.314879	0.044507
t-SNE	15	single	HDBSCAN	25.0	0.313909	0.017966
t-SNE	200-40	averaging	HDBSCAN	25.0	0.313537	0.024341

t-SNE	150-30	averaging	HDBSCAN	25.0	0.311942	0.032132
t-SNE	70-10	annealing	HDBSCAN	25.0	0.311173	0.030490
t-SNE	40	single	HDBSCAN	25.0	0.309488	0.000000
t-SNE	100-20	annealing	HDBSCAN	25.0	0.306497	0.030375
t-SNE	100-20	annealing	HDBSCAN	25.0	0.306497	0.030375
t-SNE	70-10	averaging	HDBSCAN	25.0	0.303497	0.000000
t-SNE	50	single	HDBSCAN	25.0	0.302998	0.000000
t-SNE	250-100	annealing	HDBSCAN	25.0	0.302227	0.001223
t-SNE	80-12	averaging	HDBSCAN	25.0	0.301348	0.013480
t-SNE	150-50	averaging	HDBSCAN	25.0	0.300281	0.021161
t-SNE	30-7	averaging	HDBSCAN	25.0	0.299624	0.016194
t-SNE	150-50-4	averaging	HDBSCAN	25.0	0.299623	0.015519
t-SNE	225-75	annealing	HDBSCAN	25.0	0.298815	0.015207
t-SNE	225-75-10-3	averaging	HDBSCAN	25.0	0.298184	0.000000

B3) Maximum weighted precision best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
SVD normalized	NA	NA	k-means	NA	0.775173	0.017024
SVD normalized	NA	NA	Gaussian mixtures	NA	0.768901	0.019268
t-SNE	60	single	Gaussian mixtures	NA	0.765803	0.013276

t-SNE	50	single	Gaussian mixtures	NA	0.764776	0.013304
t-SNE	50	single	k-means	NA	0.764653	0.008167
t-SNE	75	single	Gaussian mixtures	NA	0.762829	0.012875
t-SNE	150-50	averaging	Gaussian mixtures	NA	0.761485	0.013218
tf-idf matrix	NA	NA	k-means	NA	0.761381	0.019948
t-SNE	60	single	k-means	NA	0.760412	0.009433
t-SNE	150-30	annealing	k-means	NA	0.759555	0.007060
t-SNE	90	single	Gaussian mixtures	NA	0.759366	0.011288
t-SNE	100	single	Gaussian mixtures	NA	0.759190	0.009210
t-SNE	90	single	k-means	NA	0.758761	0.006589
t-SNE	100	single	HDBSCAN	25.0	0.758202	0.000000
t-SNE	50	single	HDBSCAN	25.0	0.758092	0.000000
t-SNE	75	single	k-means	NA	0.757993	0.008723
t-SNE	60	single	HDBSCAN	25.0	0.757683	0.000000
t-SNE	75	single	HDBSCAN	25.0	0.756831	0.000000
t-SNE	100	single	k-means	NA	0.755973	0.009590
t-SNE	20	single	Gaussian mixtures	NA	0.755224	0.010378
t-SNE	20	single	Gaussian mixtures	NA	0.755224	0.010378
t-SNE	35-8	averaging	Gaussian mixtures	NA	0.754605	0.013486
t-SNE	150-50	annealing	k-means	NA	0.754390	0.008242

SVD not normalized	NA	NA	Gaussian mixtures	NA	0.754281	0.026813
t-SNE	200-60	annealing	k-means	NA	0.752781	0.012513

B4) Maximum NMI best results

	Perplexity	**Perplexity type**	**cluster**	**mean**	**std**
Model					
t-SNE	50	single	k-means	0.695013	0.007431
t-SNE	60	single	Gaussian mixtures	0.692948	0.011142
t-SNE	75	single	k-means	0.690748	0.005370
t-SNE	60	single	k-means	0.690212	0.005230
t-SNE	50	single	Gaussian mixtures	0.689442	0.005638
t-SNE	90	single	Gaussian mixtures	0.688823	0.011250
t-SNE	90	single	k-means	0.687933	0.005673
t-SNE	75	single	Gaussian mixtures	0.687590	0.006557
t-SNE	200-60	annealing	Gaussian mixtures	0.684648	0.008243
t-SNE	200-60	annealing	k-means	0.684288	0.009258
t-SNE	200-40	annealing	Gaussian mixtures	0.682986	0.008830
t-SNE	150-50	annealing	Gaussian mixtures	0.681838	0.007070
t-SNE	40	single	Gaussian mixtures	0.681113	0.007689

t-SNE	100	single	k-means	0.680928	0.006984
t-SNE	250-125	annealing	Gaussian mixtures	0.680762	0.011050
t-SNE	150-50	annealing	k-means	0.680555	0.007252
t-SNE	150-30	annealing	Gaussian mixtures	0.680543	0.006278
t-SNE	100	single	Gaussian mixtures	0.680523	0.007599
t-SNE	40	single	k-means	0.680289	0.009119
t-SNE	150-30	averaging	Gaussian mixtures	0.680173	0.010475
t-SNE	150-30	annealing	k-means	0.680111	0.006175
t-SNE	250-90	annealing	Gaussian mixtures	0.679263	0.010219
t-SNE	125	single	k-means	0.679153	0.008537
t-SNE	250-100	annealing	Gaussian mixtures	0.678453	0.007122
t-SNE	30	single	Gaussian mixtures	0.677841	0.009716

B5) Maximum ARI best results

	Perplexity	**Perplexity type**	**cluster**	**mean**	**std**
Model					
t-SNE	60	single	Gaussian mixtures	0.826931	0.015148
t-SNE	90	single	k-means	0.824371	0.010343
t-SNE	75	single	k-means	0.823584	0.007872
t-SNE	50	single	k-means	0.822905	0.013291

t-SNE	150-30	annealing	k-means	0.822256	0.008979
t-SNE	60	single	k-means	0.821138	0.013017
t-SNE	75	single	Gaussian mixtures	0.820402	0.011922
t-SNE	150-50	annealing	Gaussian mixtures	0.819669	0.008746
t-SNE	90	single	Gaussian mixtures	0.819243	0.018460
t-SNE	50	single	Gaussian mixtures	0.818436	0.008908
t-SNE	150-50	annealing	k-means	0.818307	0.006914
t-SNE	100	single	k-means	0.817261	0.014939
t-SNE	250-125	annealing	Gaussian mixtures	0.816370	0.020426
t-SNE	150-30	annealing	Gaussian mixtures	0.815591	0.012543
t-SNE	200-60	annealing	k-means	0.815262	0.017707
t-SNE	125	single	k-means	0.813680	0.017669
t-SNE	150-50	averaging	Gaussian mixtures	0.813631	0.018191
t-SNE	250-125	annealing	k-means	0.813623	0.016584
t-SNE	250-100	annealing	Gaussian mixtures	0.813507	0.014151
t-SNE	200-60	annealing	Gaussian mixtures	0.811896	0.013389
t-SNE	250-90	annealing	Gaussian mixtures	0.811007	0.019160
t-SNE	200-40	annealing	Gaussian mixtures	0.810395	0.018925

t-SNE	100-4	averaging	Gaussian mixtures	0.809963	0.012210
t-SNE	250-90	annealing	k-means	0.809094	0.015409
t-SNE	80-12	averaging	Gaussian mixtures	0.809012	0.016266

B6) Kuhn-Munkres micro-averaged precision best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
t-SNE	75	single	HDBSCAN	35.0	0.562335	0.0
t-SNE	250-100	averaging	HDBSCAN	50.0	0.549582	0.0
t-SNE	50	single	HDBSCAN	25.0	0.540897	0.0
t-SNE	50	single	HDBSCAN	35.0	0.540787	0.0
t-SNE	75	single	HDBSCAN	25.0	0.534850	0.0
t-SNE	250-100	averaging	HDBSCAN	25.0	0.534081	0.0
t-SNE	250-125	averaging	HDBSCAN	25.0	0.533531	0.0
t-SNE	30	single	HDBSCAN	35.0	0.524736	0.0
t-SNE	60	single	HDBSCAN	25.0	0.523087	0.0
t-SNE	225-75	averaging	HDBSCAN	75.0	0.522977	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.521878	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.521878	0.0
t-SNE	40	single	HDBSCAN	25.0	0.519349	0.0
t-SNE	300-150	averaging	HDBSCAN	35.0	0.515831	0.0
t-SNE	50	single	HDBSCAN	50.0	0.515391	0.0
t-SNE	100	single	HDBSCAN	25.0	0.513522	0.0

t-SNE	50-12	averaging	HDBSCAN	35.0	0.512863	0.0
t-SNE	40	single	HDBSCAN	50.0	0.512533	0.0
t-SNE	80-12	averaging	HDBSCAN	35.0	0.510884	0.0
t-SNE	300-150	averaging	HDBSCAN	25.0	0.507256	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.504947	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.504947	0.0
t-SNE	40	single	HDBSCAN	75.0	0.504837	0.0
t-SNE	40	single	HDBSCAN	35.0	0.502419	0.0
t-SNE	90	single	HDBSCAN	75.0	0.501979	0.0

B7) Kuhn-Munkres macro-averaged precision best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
t-SNE	100	single	HDBSCAN	25.0	0.353638	0.000000
t-SNE	50-12	annealing	HDBSCAN	25.0	0.346667	0.017948
t-SNE	20	single	HDBSCAN	25.0	0.345503	0.024718
t-SNE	20	single	HDBSCAN	25.0	0.345503	0.024718
t-SNE	150-7	averaging	HDBSCAN	25.0	0.343053	0.033991
t-SNE	100-15	averaging	HDBSCAN	25.0	0.342669	0.026348
t-SNE	70-10	averaging	HDBSCAN	25.0	0.335763	0.000000
t-SNE	30	single	HDBSCAN	25.0	0.334917	0.034973
t-SNE	200-40	averaging	HDBSCAN	25.0	0.333955	0.033915
t-SNE	15	single	HDBSCAN	25.0	0.333735	0.027614
t-SNE	35-8	averaging	HDBSCAN	25.0	0.331125	0.036756

t-SNE	250-90-6	averaging	HDBSCAN	25.0	0.327998	0.046383
t-SNE	100-20	annealing	HDBSCAN	25.0	0.326345	0.024363
t-SNE	100-20	annealing	HDBSCAN	25.0	0.326345	0.024363
t-SNE	150-30	averaging	HDBSCAN	25.0	0.325770	0.039422
t-SNE	80-12	averaging	HDBSCAN	25.0	0.323727	0.018454
t-SNE	50	single	HDBSCAN	25.0	0.323650	0.000000
t-SNE	70-10	annealing	HDBSCAN	25.0	0.323485	0.038721
t-SNE	30-7	averaging	HDBSCAN	25.0	0.322784	0.024317
t-SNE	40	single	HDBSCAN	25.0	0.321582	0.000000
t-SNE	75	single	HDBSCAN	25.0	0.317318	0.000000
t-SNE	225-75-10-3	averaging	HDBSCAN	25.0	0.314969	0.000000
t-SNE	150-50	averaging	HDBSCAN	25.0	0.314573	0.026473
t-SNE	150-50-4	averaging	HDBSCAN	25.0	0.313178	0.022592
t-SNE	12	single	HDBSCAN	25.0	0.313005	0.024140

B8) Kuhn-Munkres weighted precision best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
SVD normalized	NA	NA	k-means	NA	0.827975	0.021280
t-SNE	300-150	averaging	HDBSCAN	25.0	0.827948	0.000000
t-SNE	300-150	averaging	HDBSCAN	35.0	0.826081	0.000000
t-SNE	200-60	averaging	HDBSCAN	35.0	0.825571	0.000000

tf-idf matrix	NA	NA	k-means	NA	0.825125	0.015904
t-SNE	225-75	averaging	HDBSCAN	50.0	0.824518	0.000000
SVD normalized	NA	NA	Gaussian mixtures	NA	0.821422	0.019352
t-SNE	100-15	averaging	k-means	NA	0.818445	0.011364
t-SNE	100-20	averaging	Gaussian mixtures	NA	0.816885	0.009627
t-SNE	100-20	averaging	Gaussian mixtures	NA	0.816885	0.009627
t-SNE	100	single	Gaussian mixtures	NA	0.815826	0.017304
t-SNE	40	single	Gaussian mixtures	NA	0.814985	0.007961
t-SNE	300-150	averaging	HDBSCAN	50.0	0.814542	0.000000
t-SNE	60	single	Gaussian mixtures	NA	0.814384	0.011030
t-SNE	150-30	averaging	Gaussian mixtures	NA	0.813539	0.011629
t-SNE	150-30	annealing	k-means	NA	0.813290	0.016845
t-SNE	15	single	Gaussian mixtures	NA	0.812789	0.009558
t-SNE	75	single	Gaussian mixtures	NA	0.811897	0.014369
t-SNE	70-10	averaging	Gaussian mixtures	NA	0.811855	0.004084
t-SNE	30-7	averaging	k-means	NA	0.811624	0.010518
t-SNE	75	single	k-means	NA	0.811179	0.019432
t-SNE	250-100	averaging	HDBSCAN	25.0	0.811058	0.000000

t-SNE	50	single	Gaussian mixtures	NA	0.810942	0.015268
t-SNE	150-30-6-2	averaging	k-means	NA	0.810886	0.019375
t-SNE	150-30	averaging	k-means	NA	0.810499	0.017312

B9) Kuhn-Munkres NMI best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
t-SNE	60	single	HDBSCAN	25.0	0.564408	0.000000
t-SNE	50	single	HDBSCAN	25.0	0.560827	0.000000
t-SNE	250-125	averaging	HDBSCAN	25.0	0.560082	0.000000
t-SNE	75	single	HDBSCAN	35.0	0.559601	0.000000
t-SNE	40	single	HDBSCAN	25.0	0.558308	0.000000
t-SNE	50	single	HDBSCAN	35.0	0.558144	0.000000
t-SNE	250-100	averaging	HDBSCAN	25.0	0.557944	0.000000
t-SNE	75	single	HDBSCAN	25.0	0.556198	0.000000
t-SNE	30	single	HDBSCAN	35.0	0.539546	0.000000
t-SNE	100	single	HDBSCAN	25.0	0.537292	0.000000
t-SNE	100-20	averaging	HDBSCAN	25.0	0.536934	0.000000
t-SNE	100-20	averaging	HDBSCAN	25.0	0.536934	0.000000
t-SNE	300-150	averaging	HDBSCAN	25.0	0.536169	0.000000
t-SNE	300-150	averaging	HDBSCAN	35.0	0.531442	0.000000
t-SNE	50	single	HDBSCAN	50.0	0.529693	0.000000
t-SNE	40	single	HDBSCAN	35.0	0.527512	0.000000

t-SNE	50-12	averaging	HDBSCAN	35.0	0.525750	0.000000
t-SNE	60	single	Gaussian mixtures	NA	0.523640	0.005619
t-SNE	50	single	k-means	NA	0.523269	0.003319
SVD normalized	NA	NA	k-means	NA	0.522633	0.008725
SVD normalized	NA	NA	Gaussian mixtures	NA	0.522323	0.007642
t-SNE	75	single	k-means	NA	0.522228	0.004319
t-SNE	100	single	Gaussian mixtures	NA	0.522147	0.004452
t-SNE	90	single	Gaussian mixtures	NA	0.521334	0.004038
t-SNE	200-60	annealing	Gaussian mixtures	NA	0.521188	0.003967

B10) Kuhn-Munkres ARI best results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
t-SNE	250-100	averaging	HDBSCAN	25.0	0.418811	0.0
t-SNE	250-125	averaging	HDBSCAN	25.0	0.415144	0.0
t-SNE	50	single	HDBSCAN	25.0	0.412470	0.0
t-SNE	75	single	HDBSCAN	35.0	0.407422	0.0
t-SNE	60	single	HDBSCAN	25.0	0.406951	0.0
t-SNE	40	single	HDBSCAN	25.0	0.402764	0.0
t-SNE	75	single	HDBSCAN	25.0	0.402475	0.0
t-SNE	50	single	HDBSCAN	35.0	0.401511	0.0

t-SNE	30	single	HDBSCAN	35.0	0.378636	0.0
t-SNE	250-100	averaging	HDBSCAN	50.0	0.377726	0.0
t-SNE	300-150	averaging	HDBSCAN	25.0	0.377002	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.374710	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.374710	0.0
t-SNE	100	single	HDBSCAN	25.0	0.367851	0.0
t-SNE	50	single	HDBSCAN	50.0	0.359032	0.0
t-SNE	50-12	averaging	HDBSCAN	35.0	0.355219	0.0
t-SNE	300-150	averaging	HDBSCAN	35.0	0.353165	0.0
t-SNE	225-75	averaging	HDBSCAN	50.0	0.326720	0.0
t-SNE	40	single	HDBSCAN	35.0	0.325489	0.0
t-SNE	90	single	HDBSCAN	75.0	0.321256	0.0
t-SNE	80-12	averaging	HDBSCAN	100.0	0.307561	0.0
t-SNE	225-75	averaging	HDBSCAN	75.0	0.293882	0.0
t-SNE	40	single	HDBSCAN	50.0	0.292487	0.0
t-SNE	300-150	averaging	HDBSCAN	50.0	0.287054	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.285752	0.0

C1) Maximum NMI best HDBSCAM results

	Perplexity	**Perplexity type**	**cluster**	**mcs**	**mean**	**std**
Model						
t-SNE	60	single	HDBSCAN	25.0	0.655631	0.0
t-SNE	50	single	HDBSCAN	25.0	0.651769	0.0

t-SNE	75	single	HDBSCAN	25.0	0.645279	0.0
t-SNE	40	single	HDBSCAN	25.0	0.638656	0.0
t-SNE	75	single	HDBSCAN	35.0	0.631991	0.0
t-SNE	50	single	HDBSCAN	35.0	0.628128	0.0
t-SNE	100	single	HDBSCAN	25.0	0.623089	0.0
t-SNE	30	single	HDBSCAN	35.0	0.616189	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.611815	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.611815	0.0
t-SNE	250-125	averaging	HDBSCAN	25.0	0.607633	0.0
t-SNE	250-100	averaging	HDBSCAN	25.0	0.606155	0.0
t-SNE	40	single	HDBSCAN	35.0	0.595714	0.0
t-SNE	50	single	HDBSCAN	50.0	0.590357	0.0
t-SNE	300-150	averaging	HDBSCAN	25.0	0.587003	0.0
t-SNE	50-12	averaging	HDBSCAN	35.0	0.586804	0.0
t-SNE	300-150	averaging	HDBSCAN	35.0	0.576430	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.575066	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.575066	0.0
t-SNE	70-10	averaging	HDBSCAN	25.0	0.574377	0.0
t-SNE	40	single	HDBSCAN	50.0	0.566951	0.0
t-SNE	80-12	averaging	HDBSCAN	35.0	0.560136	0.0
t-SNE	100-20	annealing	HDBSCAN	35.0	0.557253	0.0
t-SNE	100-20	annealing	HDBSCAN	35.0	0.557253	0.0
t-SNE	225-75	averaging	HDBSCAN	50.0	0.556351	0.0

C2) Maximum ARI best HDBSCAN results

	Perplexity	Perplexity type	cluster	mcs	mean	std
Model						
t-SNE	50	single	HDBSCAN	25.0	0.733793	0.0
t-SNE	60	single	HDBSCAN	25.0	0.731702	0.0
t-SNE	75	single	HDBSCAN	25.0	0.725955	0.0
t-SNE	40	single	HDBSCAN	25.0	0.715232	0.0
t-SNE	75	single	HDBSCAN	35.0	0.707823	0.0
t-SNE	50	single	HDBSCAN	35.0	0.696466	0.0
t-SNE	30	single	HDBSCAN	35.0	0.689331	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.675514	0.0
t-SNE	100-20	averaging	HDBSCAN	25.0	0.675514	0.0
t-SNE	100	single	HDBSCAN	25.0	0.673468	0.0
t-SNE	250-100	averaging	HDBSCAN	25.0	0.654589	0.0
t-SNE	250-125	averaging	HDBSCAN	25.0	0.653152	0.0
t-SNE	40	single	HDBSCAN	35.0	0.627007	0.0
t-SNE	50-12	averaging	HDBSCAN	35.0	0.622718	0.0
t-SNE	50	single	HDBSCAN	50.0	0.620941	0.0
t-SNE	300-150	averaging	HDBSCAN	25.0	0.610814	0.0
t-SNE	300-150	averaging	HDBSCAN	35.0	0.583779	0.0
t-SNE	40	single	HDBSCAN	50.0	0.576030	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.561702	0.0
t-SNE	100-20	averaging	HDBSCAN	35.0	0.561702	0.0

t-SNE	100-20	annealing	HDBSCAN	35.0	0.545915	0.0
t-SNE	100-20	annealing	HDBSCAN	35.0	0.545915	0.0
t-SNE	225-75	averaging	HDBSCAN	50.0	0.543483	0.0
t-SNE	90	single	HDBSCAN	75.0	0.540687	0.0
t-SNE	80-12	averaging	HDBSCAN	35.0	0.538723	0.0