Custering

Agenda:

- Clustering
  - Applications
  - distances
  - approaches
    - hierarchical
    - density based
  - evalaution
    - internal
    - external
  - labeling clusters

`Note:` This is for part 2.

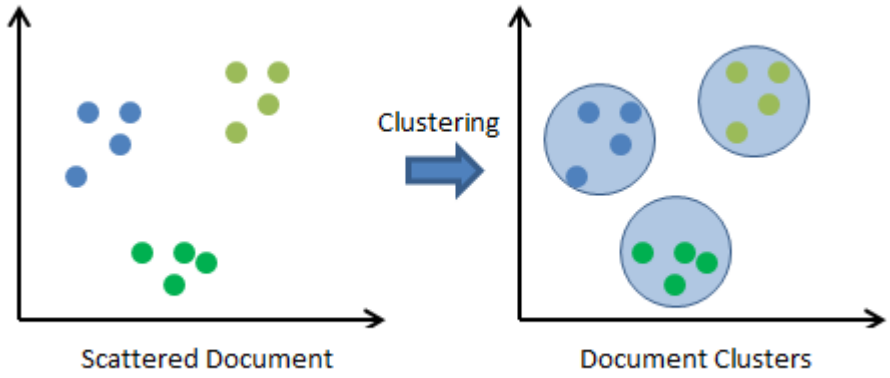
Clustering

What is clustering?

Imagine we have our documents and terms in a table

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
d1	1	0	0	1	0	0	0	0	0	1
d2	0	1	0	0	0	0	1	0	0	1
d3	0	0	1	0	0	0	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...
dn	1	0	0	1	0	0	0	0	0	1

`Clustering` is grouping documents based on their similarity.
We want `similar` documents near each other and `dissimilar` documents far from each other.



What are the `pros` and `cons`

`Pros`

- Organization
  - This is `unsupervised categorization` of documents
  - Improve recall

    - Imagine having a query `q` . If we get some documents, we can get more documents that are similar to the ones we have so the recall is increased.
  - Visualization ?
  - Efficiency
    - We need to imagine the search engine with like high level clusters with more clusters inside. The way to retrieve the best documents fast is to compare the `centroids` of the clusters with the query to get the best match.

Distances and Approaches

Distances

Imagine we have an embedding in a vector space. There are many ways to measure the distance between two points in a vector space.

- Cosine similarity
- Minkowski distance
- Hamming distance (Is used for binary vectors, it counts the number of mismatches)

Approaches

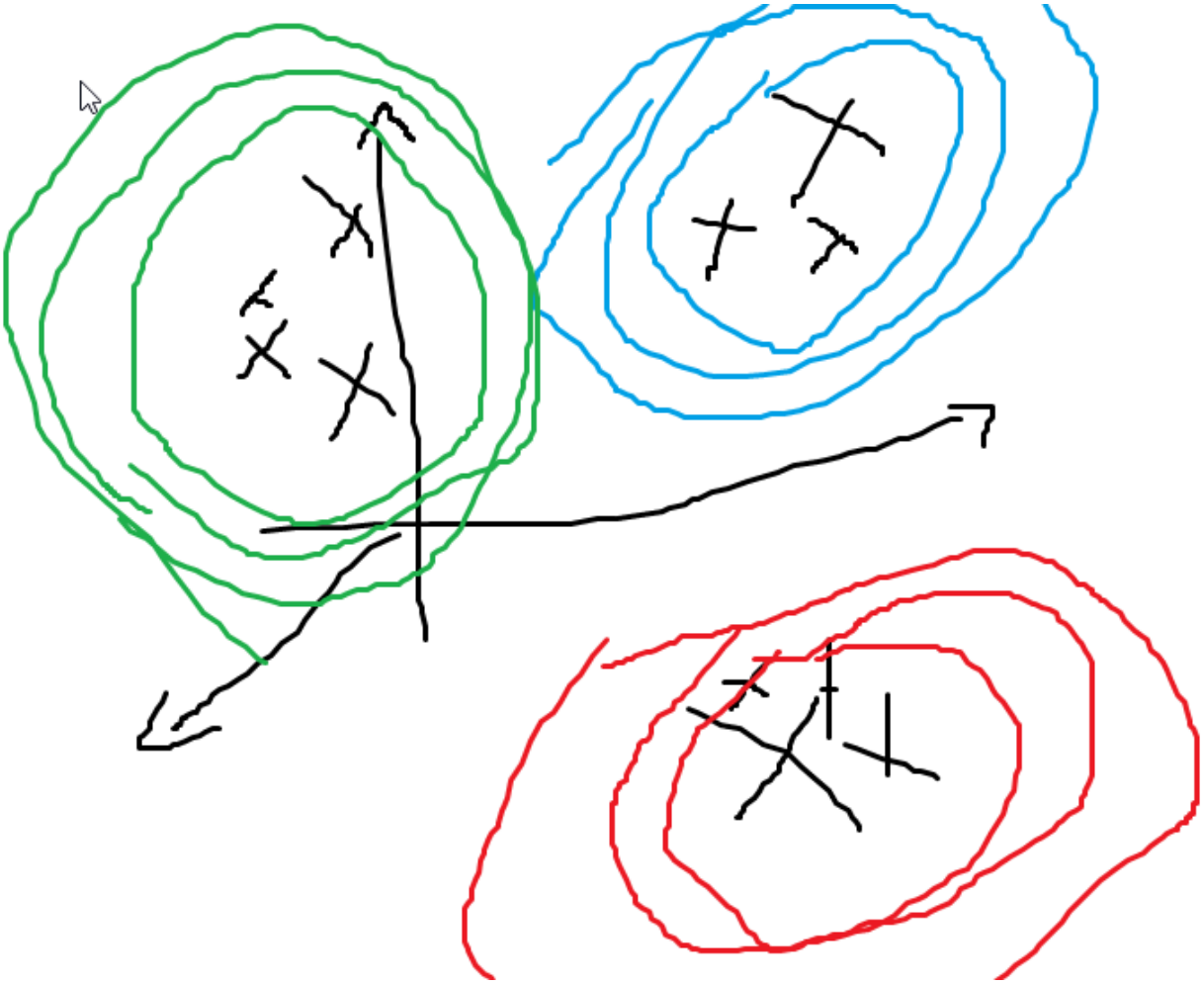
There are many approaches to clustering.

Partitioning

Is the most common approach. It is the process of partitioning the space into `k` clusters. The most common algorithm for this is `K-means` .

Statistical

Is another approach. We can think of clusters as distributions. When we check for a new point



We can see this as a table of documents and clusters

	c1	c2	c3
d1	0.1	0.2	0.7
d2	0.3	0.4	0.3
d3	0.5	0.3	0.2

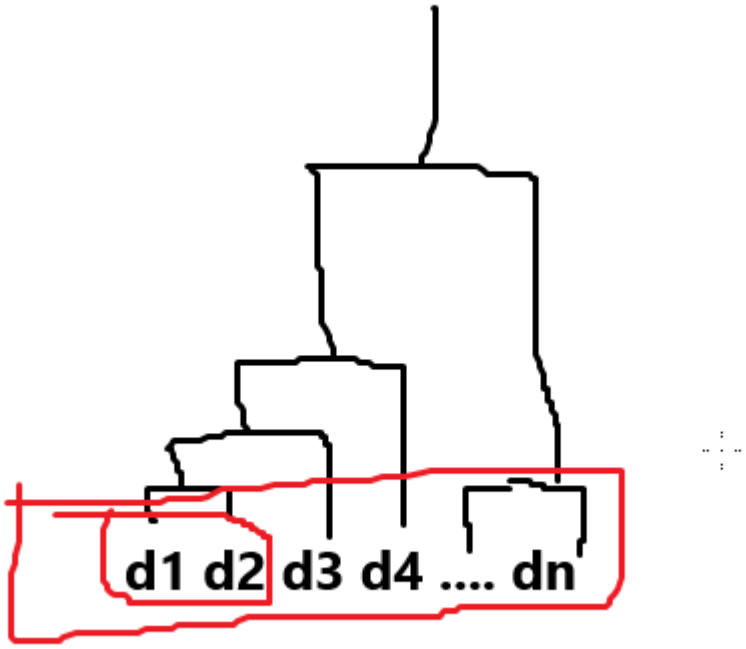
One document can be in multiple clusters. We can say that a document is in a cluster if the probability is higher than a threshold.

Hierarchical (Dendogram)

Imagine having our documents

$$d_1, d_2, d_3, d_4, d_5$$

We can use this document of documents to create a tree of clusters. This is called `dendrogram`



We have more ways to group our documents.

Let's make an example to understand it better.

Example for Hierarchical Clustering

This is a `pair-wise` distance matrix. We can use this to create a dendrogram.

We insert in each cell the `distance` of the documents.

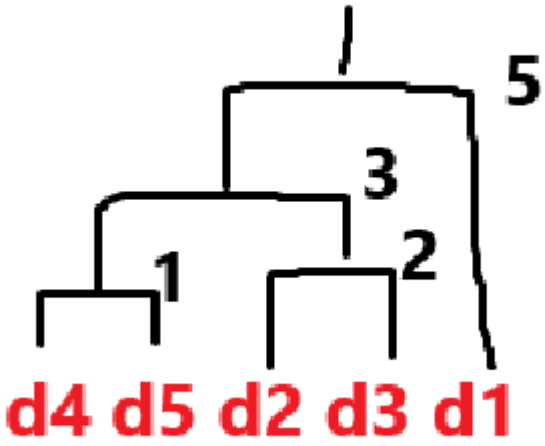
Of course, the distance of a document with itself is `0` .

	d1	d2	d3	d4	d5
d1	0	8	8	5	5
d2	1	0	2	4	3

	d1	d2	d3	d4	d5
d3	2	1	0	1	2
d4	3	2	1	0	1
d5	4	3	2	1	0

So the documents that are closest will be grouped.
Let's go in ascending

- d4 and d5 `1`
- d2 and d3 `2`
- Now we have distance 3 for d4 and d3 and d5 and d2 `3`
- now d1 and d5 `5`



Now we have what we call maximum linkage and minimum criteria.

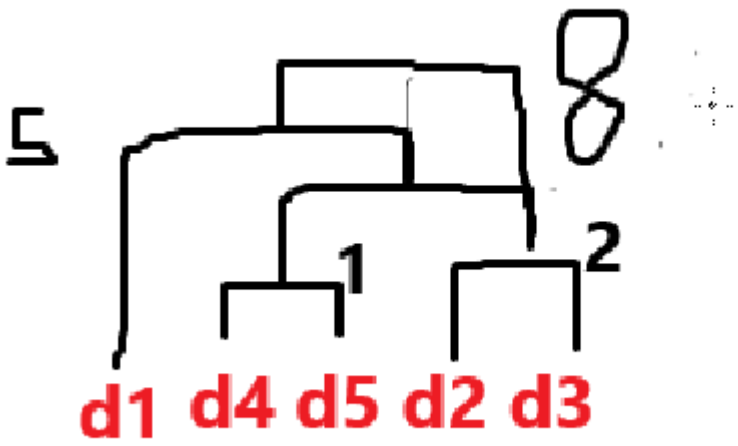
We initially have

- d4 and d5 with distance `1`
- d2 and d3 with distance `2`

We ask ourself `What is the maximum distance in this cluster?`
`Answer` : is `6` btween d3 and d5

But we see a 3 for the 2 clusters in the diagram. Heh. We have to create another diagram to handle this

Idk this is wrong



If we use `minimum link` we can find clusters with `non convex` shape. So it's strong. I would partition ad the end of the dendrogram and the find the clusters.

So, remember, `minimum link` is used when having clusters with irregular shape.

`Maximumc criteria` is not used that much because it `tends` to a `more balanced clusters` in terms of documents.

Trade-off: `Average Link` criteria. Will be explained later for the exam preparation.

Density Based

It's not very different from hierarchical with `minimum link` . It's easy to explain and `high` efficien.

We have two parameters, `k` that's the number of neighbors and `epsilon` that's the distance.

Let's make an example of `k=2` and `epsilon=3` . We are searching for 2 documents with distance at most 3.

	d1	d2	d3	d4	d5
d1	0	8	8	5	5
d2	8	0	2	4	3
d3	2	1	0	1	2
d4	3	2	1	0	1

	d1	d2	d3	d4	d5
d5	4	3	2	1	0

Let's take a documents, `d2`. `d2` has 2 neighbors with distance at most 3, `d3` and `d5`.

$$\{d2\}$$

Now go with `d3`. `d3` has 2 neighbors with distance at most 3, `d2` and `d5`.

$$\{d2, d3\}$$

this goes on until we have visit all the documents. We will end up with:

$$\{d2, d3, d5, d5\}$$

and another cluster. We can refer this as `outlier`.

$$\{d1\}$$

`Pros`:

- Efficient
- Handles outliers

`Cons`:

- `k` and `epsilon` are hard to set

Evaluation

We want to assess:

- Separation
- Cohesion

We introduce the concept of `centroid`. It's a measure that, given some documents, it's the `average` of the documents.

- d1=(1,0)
- d2=(2,0)
- d3=(4,2)

If we use the mean:

$$\bar{c1} = (\frac{7}{3}, \frac{2}{3})$$

But this is the mean. We can even use the `median`.

$$\bar{c1} = (2, 0)$$

Now, what if we want to describe the cluster.
We can take the `most central sentence` of a documents, for example. This is called `medoid`.

In our case, the `medoid` is `d2`. That's the most central document that describes the cluster.

Let's go back to evaluation.

We call `SSE` (Sum of Squared Errors) of a set of cluster `C`.

$$SSE(C) = \sum_{c_k \in C} \sum_{d \in c_k} \Delta^2(d, \bar{c_k})$$

where:

- c_k is a cluster in C
- d is a document in c_k
- Δ is the distance between d and $\bar{c_k}$
- $\bar{c_k}$ is the centroid of c_k

`Silhouette` is another measure. It can be computed at:

- document level
- cluster level
- total solution level

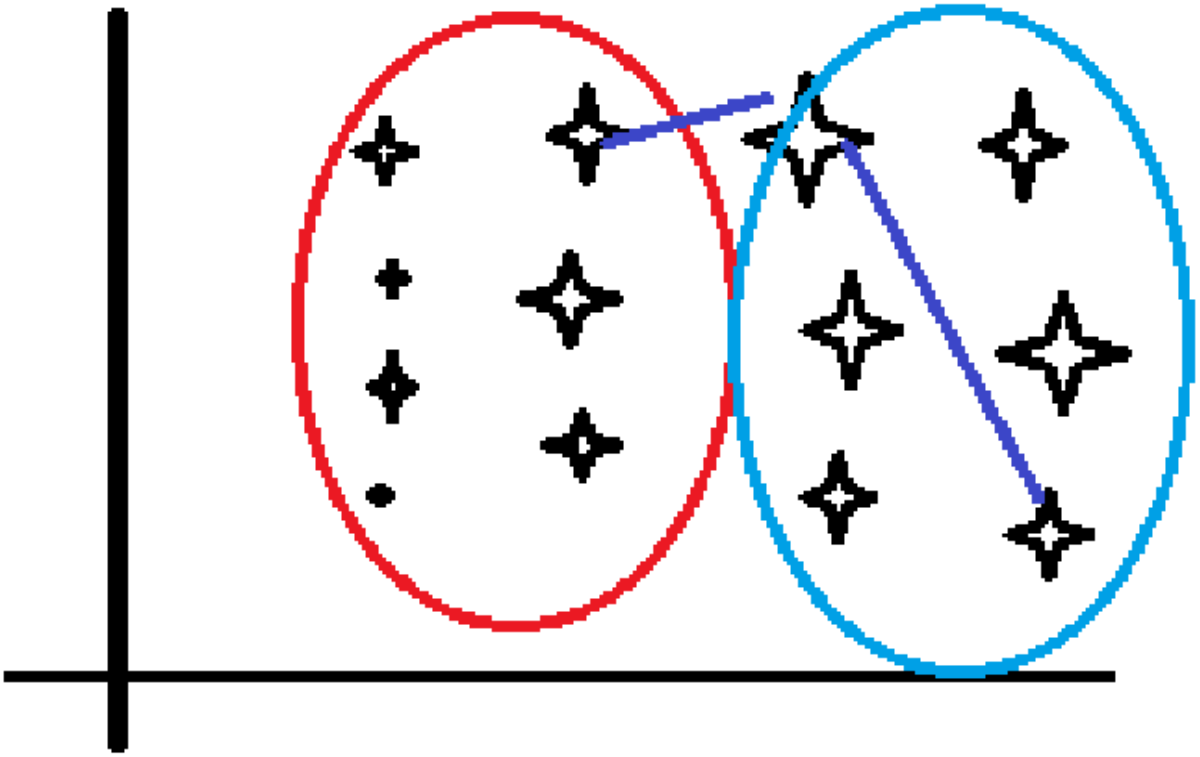
Given a document `d`, the silhouette `s` is:

$$s(d) = 1 - \frac{a(d)}{b(d)}$$

where:

- a(d) is the average distance of d to the other documents in the same cluster
- b(d) is the average distance of d to the documents in the nearest cluster
- s(d) is in [-1, 1]

Note: something like this can happen



So we can have a document in a cluster more near to another document in another cluster rather than the documents in the same cluster.

We then need to use another formula

$$s(d) = \frac{b(d)}{a(d)} - 1$$

We can have the `silhouette` at cluster level. The silhouette of a cluster is the average of the silhouette of the documents in the cluster.

And the silhouette of the total solution is the average of the silhouette of the clusters. Of course.

Exercise

	d1	d2	d3	d4	cluster
d1	0	2	1	3	c1
d2	2	0	3	2	c1
d3	1	3	0	4	c1

	d1	d2	d3	d4	cluster
d4	3	2	4	0	c2

Let's calculate silhouette at document level.

For d1:

- $$a(d1) = \frac{2+1}{2} = 1.5$$
- $$b(d1) = 3$$
- $$s(d1) = 1 - \frac{1.5}{3} = 0.5$$

For d2:

- $$a(d2) = \frac{2+3}{2} = 2.5$$
- $$b(d2) = 2$$
- $$s(d2) = \frac{2}{2.5} - 1 = -0.2$$

For d3

- $$s(d3) = 0.5$$

For cluster 1:

- $$s(c1) = \frac{0.5 + (-0.2) + 0.5}{3} = 0,2\bar{6}$$

Singleton cluster are clusters that have only one document. Usually it's 0 the silhouette.

External Evaluation

We can use these techniques when we actually have a `ground truth`.
The measures we can use are:

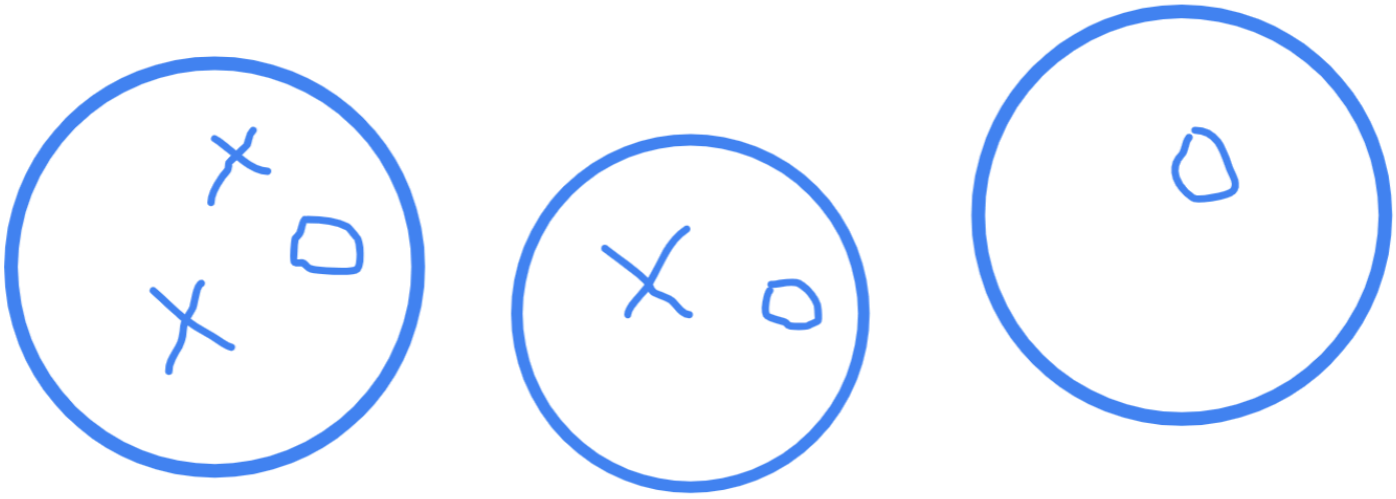
- Purity
- Rand Index

Purity

Imagine having 6 documents and clustering solution

	true	cluster
d1	g1	c1
d2	g1	c1
d3	g2	c1
d4	g2	c2
d5	g2	c3
d6	g1	c3

g is the group.



Purity is computed as:

$$purity(C, G) = \frac{1}{n} \sum_{c_k \in C} \max_j |c_k \cap g_j|$$

where:

- C is the clustering solution
- G is the ground truth
- c_k is a cluster in C

- g_j is a group in G
- n is the number of documents
- |c_k \cap g_j| is the number of documents in c_k that belong to g_j

So witht hsi example is:

$$purity = \frac{1}{6}(2 + 1 + 1) = \frac{4}{6} = \frac{2}{3}$$

If we want good alignment, we would go for `0.75` at least for purity.

Rand Index

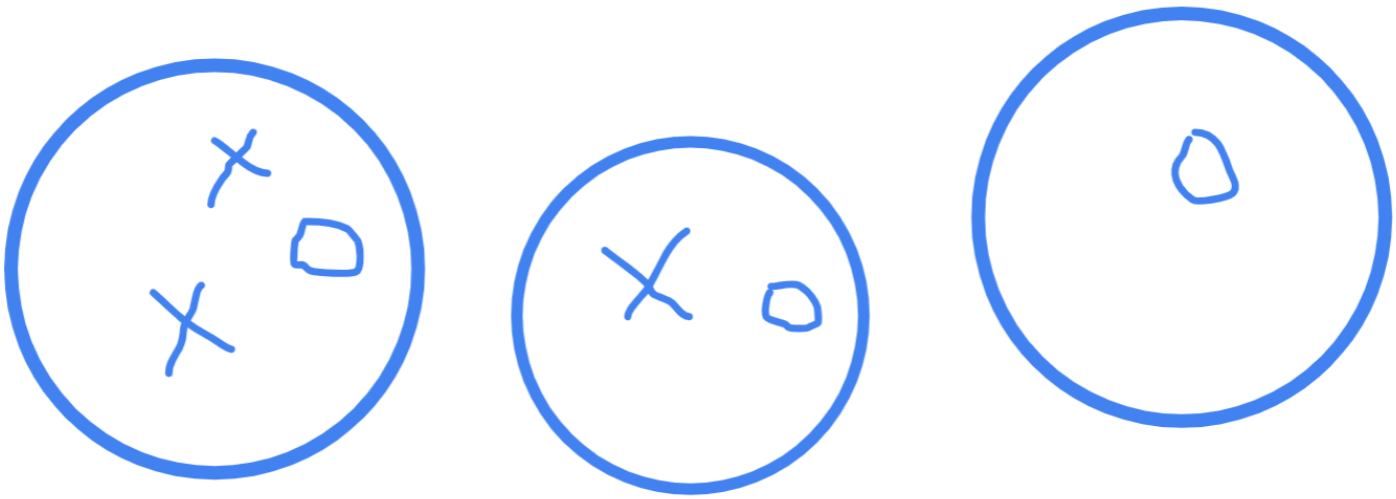
Ran Index resemble our confusion matrix. Is not like the clasic one. We don't enumerate documents but we look at pair of documents that belong to the same cluster or to different clusters, and have the same group or different group.

pairs	same cluster	different cluster
same group	`TP`	`FP`
different group	`FN`	`TN`

`Accuracy | Rand Index`：

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Let's see our previous example



pairs	same cluster	different cluster
same group	1	5
different group	3	6

To understand this, look at pairs 2 by 2, like first `x` with the other `x` and other `o`. Then we have to look at the other `x` with the other `x` and other `o`. ecc...
This will lead to the result above.

$$\frac{1+6}{1+6+5+3} = \frac{7}{15}$$

`Note:` Scikit-learn has a function to compute the `rand index` but it has a correction that tends to decrese the value

Labeling Clusters

Imagine having to have some clusters already made. Now we have to set a label for each of them.

An initial strategy could be to select the `medoid` of each cluster, or top 2 or top 3, it depends on the resources and requirements.

Then, having the `center of the cluster` (medoid).

	t1	t2	...	tn
centroid1	0.1	0.2	...	0.7
centroid2	0.3	0.4	...	0.3
...	...	...	...	...
centroidk	0.5	0.3	...	0.2

We can use `PCA` to reduce the dimensionality of the data. Yet, this can be a problem because we can lose some information and we are seeing this as an `embedding`.

Another strategy is to take the terms with the `higher score`. But, remember, this could lead to a problem. The problem is that a `term` can appear in more than one cluster and this can lead to a `confusion`. So, not only `the score` should be high, but also the score for the terms should be `unique` for each cluster.

This is a solution.

Another solution is the `feature selection` in a more supervised way. I didn't fully understand this.