Case Study- Spectral Clustering

Now Let’s apply spectral clustering to real-world examples. News websites like Google Newsgroup articles by topics. Let's see how we can construct a grouping like that.

Air strikes as Syria declares truce over

Air strikes hit rebel-held parts of the Syrian city of Aleppo, hours after the military declares the current cessation of violence is over.

9 minutes ago | Middle East

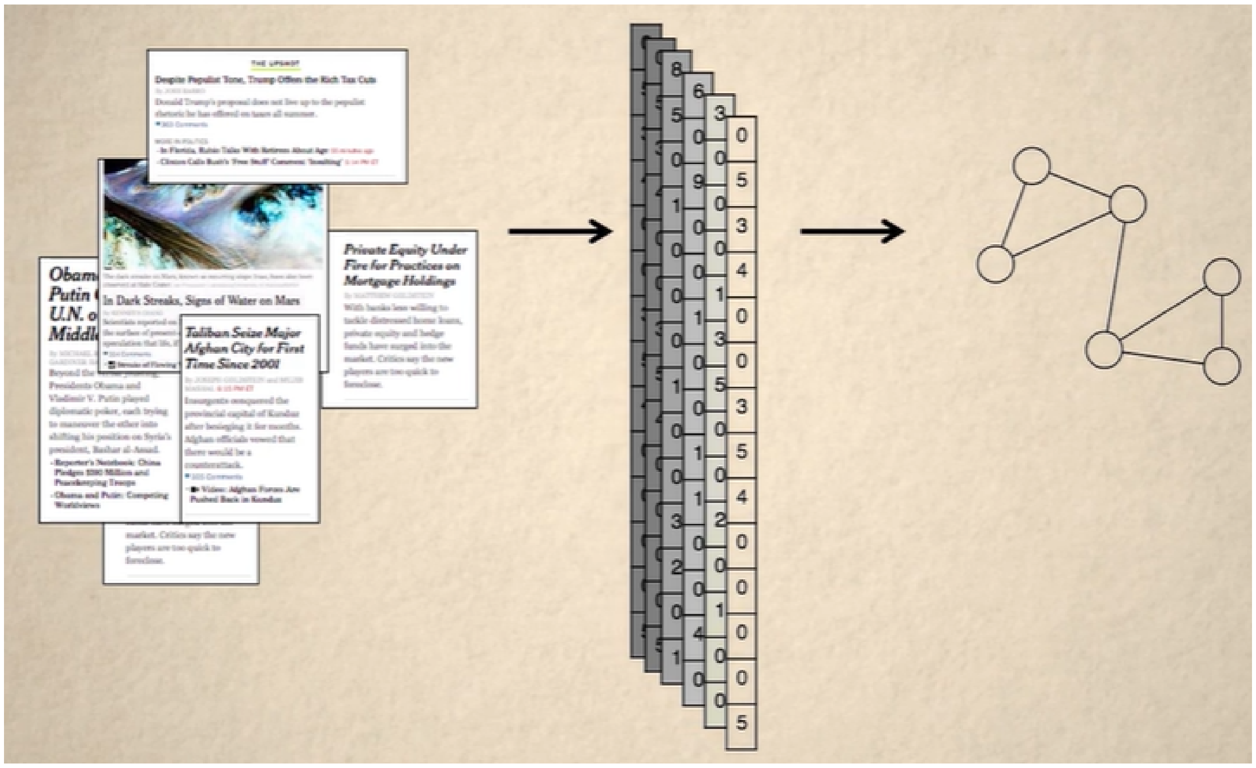
Assad 'never bombed his people'

Assad 'must be more responsible'

Ceasefire 'test for everyone'

Syria's theatre of nightmares

We collected 60 news articles from The Guardian, Independent, BBC and CNN. On those websites, the articles had the tags Brexit, USA, Middle East, Africa, Euro 2016. And for our analysis, we ignore the text. We only use the words in the articles. First, we will build a feature vector for each article. Then we compute a similarity graph for each node as an article.

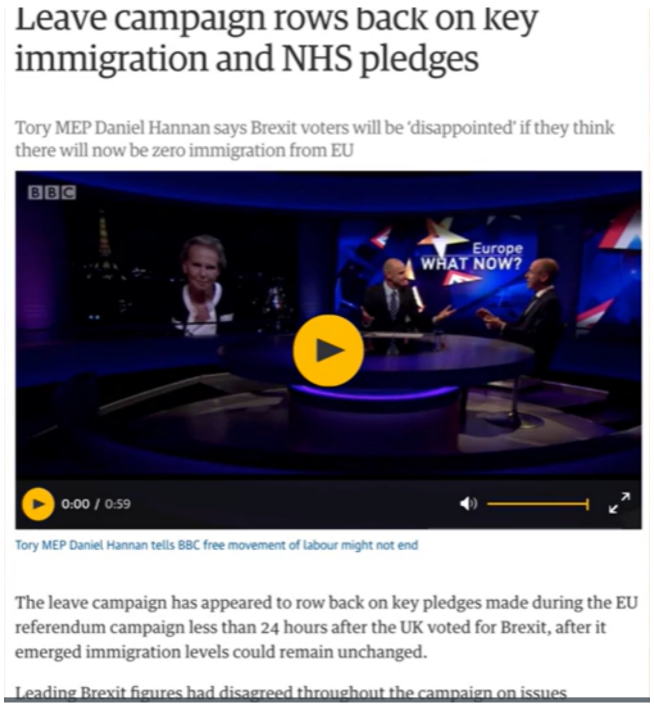


Then we compute the Laplacian eigenvectors and this gives us a visualization and a cluster.

Let’s start by creating a vector for each article that is present.

Let us take an example - this article was published in The Guardian right after the Brexit referendum in the UK on the 23rd of June, 2016.

The article has 720 words. We use a library to extract all entities mentioned in the article like names, organizations, and locations.



Leave campaign rows back on key immigration and NHS pledges

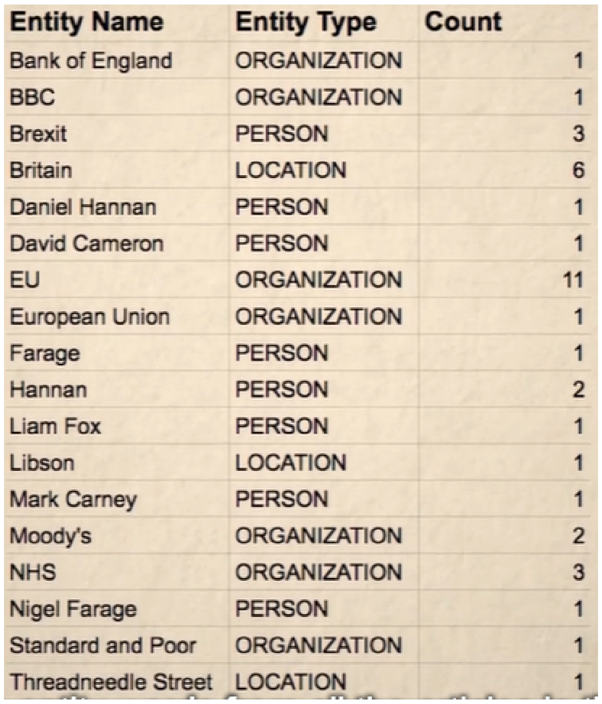
Tory MEP Daniel Hannan says Brexit voters will be ‘disappointed’ if they think there will now be zero immigration from EU

Tory MEP Daniel Hannan tells BBC free movement of labour might not end

The leave campaign has appeared to row back on key pledges made during the EU referendum campaign less than 24 hours after the UK voted for Brexit, after it emerged immigration levels could remain unchanged.

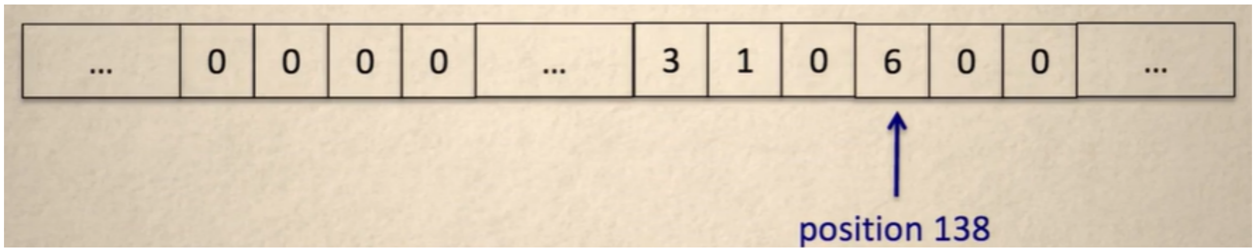
Leading Brexit figures had disagreed throughout the campaign on issues

Here are the entities detected in the article and how often they occur. Bank of England and BBC occur once, Brexit occurs three times, Britain six times, and so on.



Entity Name	Entity Type	Count
Bank of England	ORGANIZATION	1
BBC	ORGANIZATION	1
Brexit	PERSON	3
Britain	LOCATION	6
Daniel Hannan	PERSON	1
David Cameron	PERSON	1
EU	ORGANIZATION	11
European Union	ORGANIZATION	1
Farage	PERSON	1
Hannan	PERSON	2
Liam Fox	PERSON	1
Libson	LOCATION	1
Mark Carney	PERSON	1
Moody's	ORGANIZATION	2
NHS	ORGANIZATION	3
Nigel Farage	PERSON	1
Standard and Poor	ORGANIZATION	1
Threadneedle Street	LOCATION	1

So, how can we use this to compute similarities between articles?
→We collect all the entity words from all the articles in the dictionary. There are 1063 words in total. Each word is assigned an ID number between zero and 1062. Now, each article can be represented as a vector of 1062 columns, one for each entity. For example, the word Britain has ID 138 and it occurs six times in our article. So the column vector of our article has a value of 6 in position 138.



To make these counts more meaningful, we convert them into TF-IDF format.
TF-IDF stands for Term-Frequency- Inverse Document Frequency and it indicates the importance of a word.

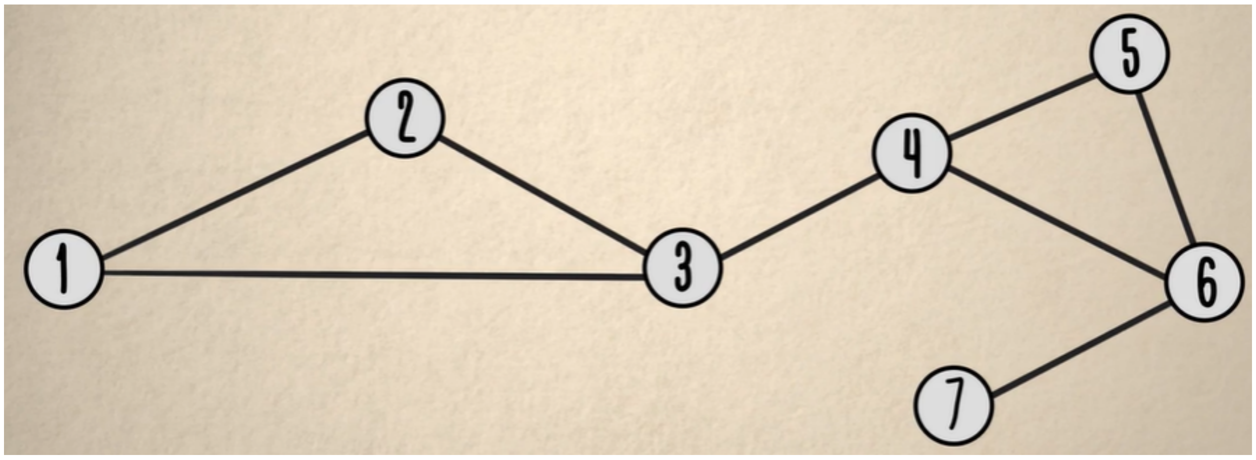
This value increases with the number of times a word occurs in the article, but decreases if the word occurs often in many articles.

The term frequency is the number of times a word occurs in the article i.e our count is divided by the number of entities in this article. So if Britain occurs 6 times in an article with 200 entities, then the term frequency would be 6 divided by 200. We divide this value by the number of articles this word occurs in, for example.35.

$$\frac{\text{term-frequency in article}}{\text{document frequency}} = \frac{6}{200} \times \frac{1}{35}$$

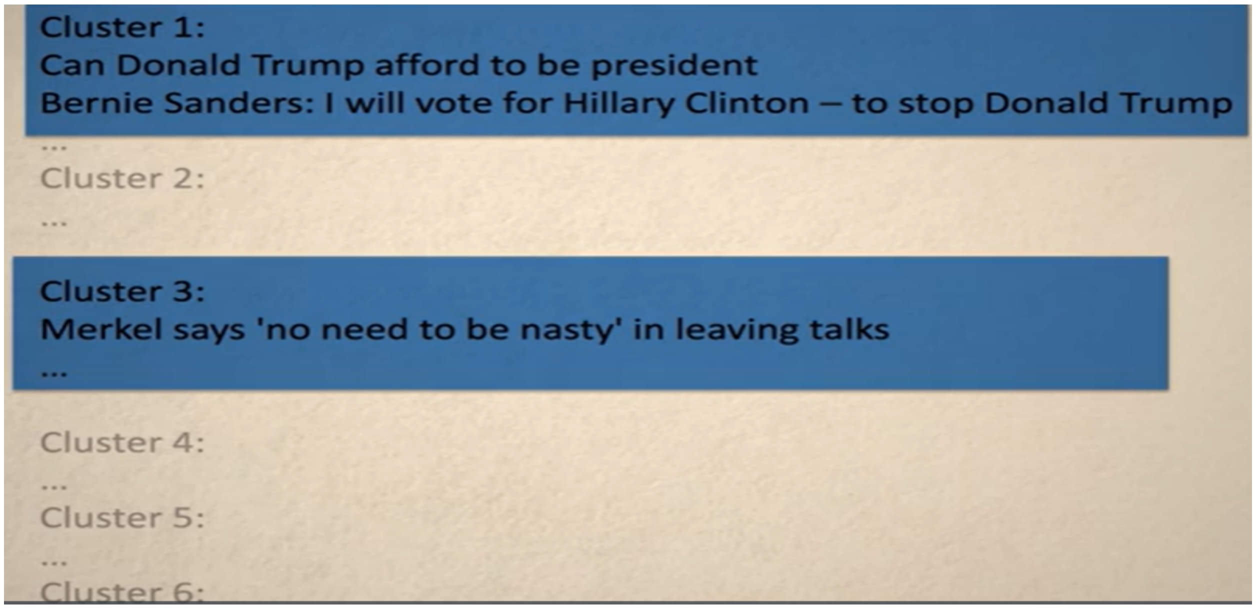
We use our TF-IDF vectors to compute similarities between all articles.
For any two articles, we sum up the square differences of their TF-IDF vectors and exponentiate that.

Now we build a graph of articles by connecting each article to its 10 closest articles. Then we apply spectral clustering and we get 7 clusters.

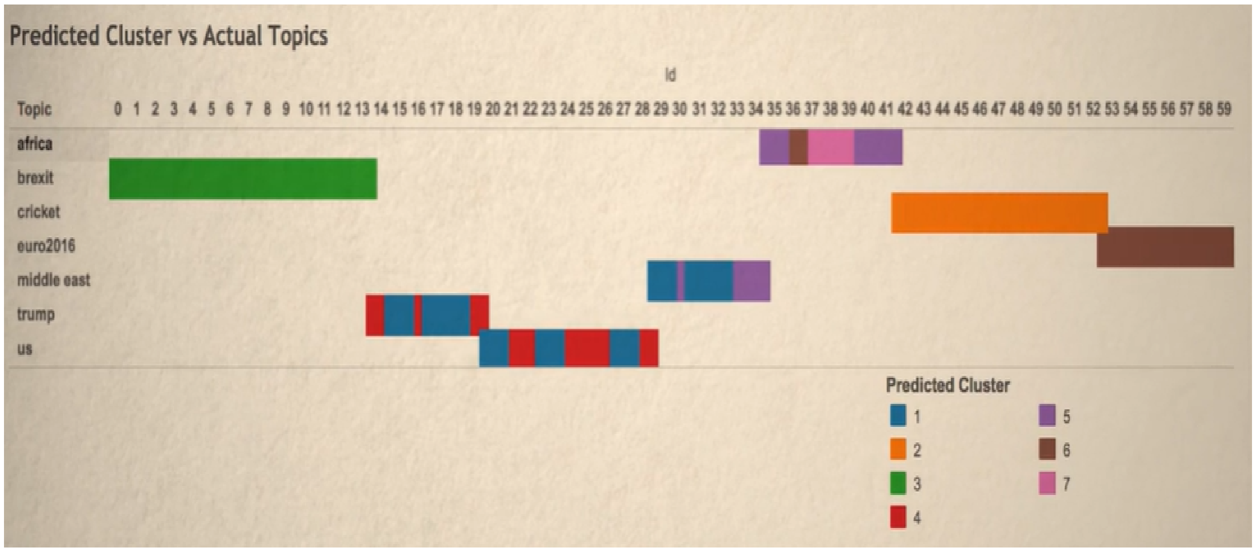


The clusters contain articles like:

$$\text{similarity}(\text{article } 1,\ \text{article } 2) = \exp\left(- \sum_{\text{entity } j=0}^{2063} (\underbrace{x_j}_{\text{tf-idf value for article 1 and entity j}} - \underbrace{y_j}_{\text{tf-idf value for article 2 and entity j}})^2 \right)$$

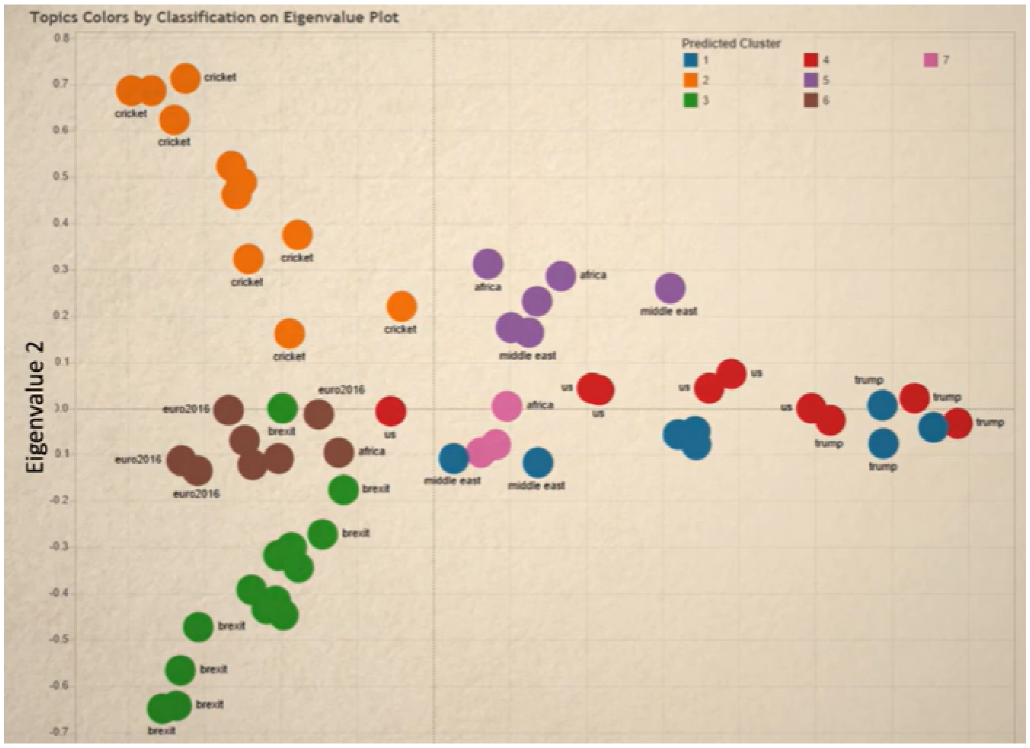

Cluster 1:
Can Donald Trump afford to be president
Bernie Sanders: I will vote for Hillary Clinton – to stop Donald Trump
...
Cluster 2:
...
Cluster 3:
Merkel says 'no need to be nasty' in leaving talks
...
Cluster 4:
...
Cluster 5:
...
Cluster 6:

Indeed the clusters correspond pretty much to the tags given by the news websites. Here we see how the articles of each tag are clustered, one color for each cluster. The tags that are put together here are USA and Trump and indeed these articles contain very similar words.



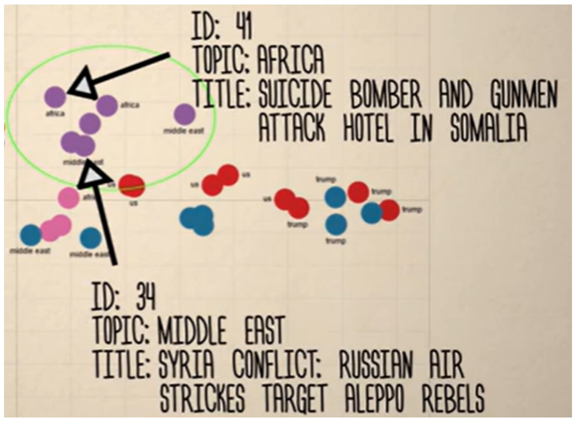
Let’s use eigenvectors of the second and third smallest eigenvalues to plot the data. Each article now has 2 coordinates, the entry in each of these eigenvectors.
Each dot is an article and the colors indicate the predicted clusters. Here we label some articles by their tags from the news websites and this plot shows the relationships between articles.

Even without cluster labeling, we see different directions emerging.



The top left is articles on cricket and the bottom left are articles on Brexit. Towards the right, we get a set of articles on the USA and Trump.

Let's have a look at the purple group on the right. It contains articles on the Middle East and Africa. For example, one talks about an attack on a hotel in Somalia and one about a Russian airstrike in Syria. These are articles about war and attacks.



One middle east article is positioned much closer to the center and the European topics. Its title is ‘The Orchestra of Syrian Musicians. When there is violence, you have to make music’. It talks about musicians from a former Syrian national orchestra who were refugees and were reunited to go on a European tour. This article also mentions how the conductor is still waiting in the U.S. It mentions the public opinion towards refugees. In short, it touches upon many of the other topics. So it's drawn to the middle

of the plot. Here we already see some clusters emerging, but some clusters are still somewhat close together. They get separated when we use other Eigenvectors.
In summary, we've seen how spectral embedding arranges news articles by content and finds topic groups. It reveals whether there is any group structure in our data and it also illustrates relationships in the data.