Vector Space Models

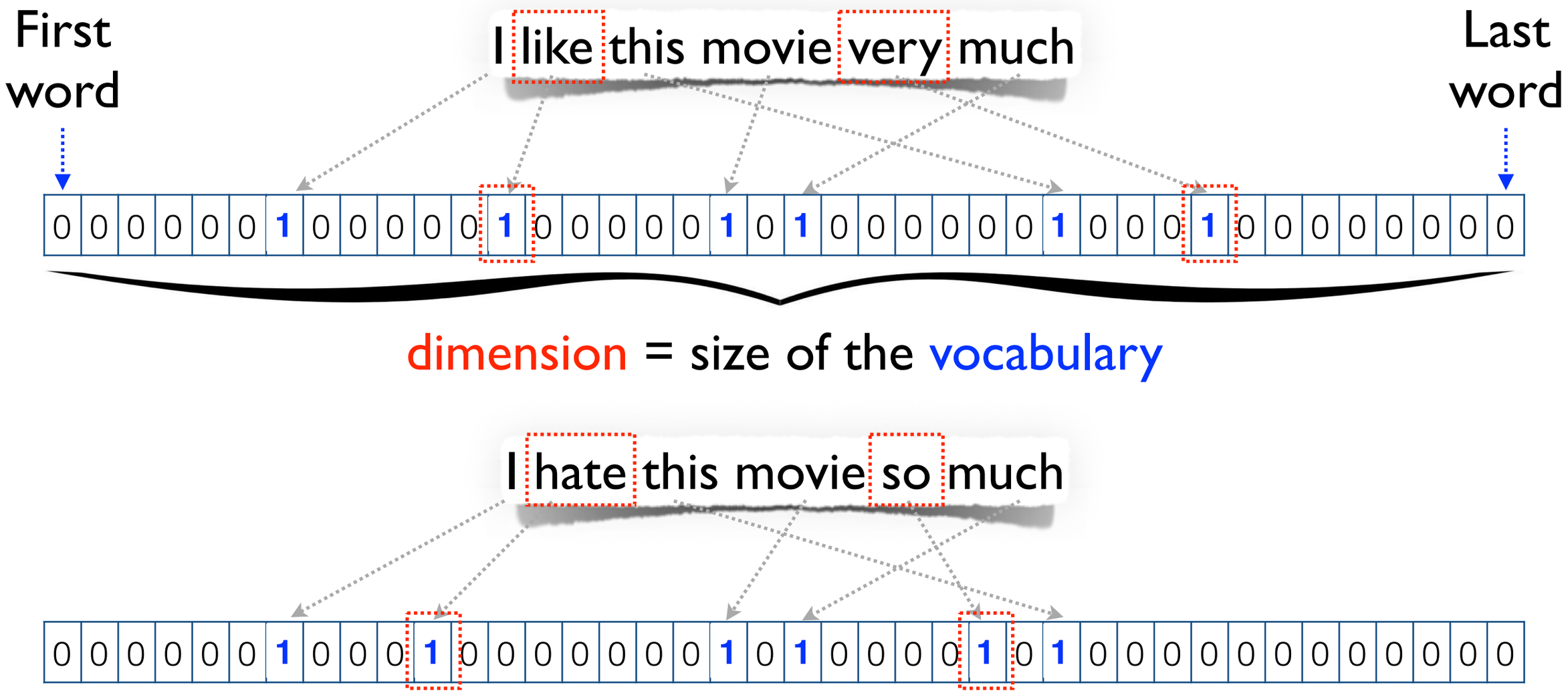
Practical Approaches to Data Science with Text

Emory University

Jinho D. Choi

Document Representation

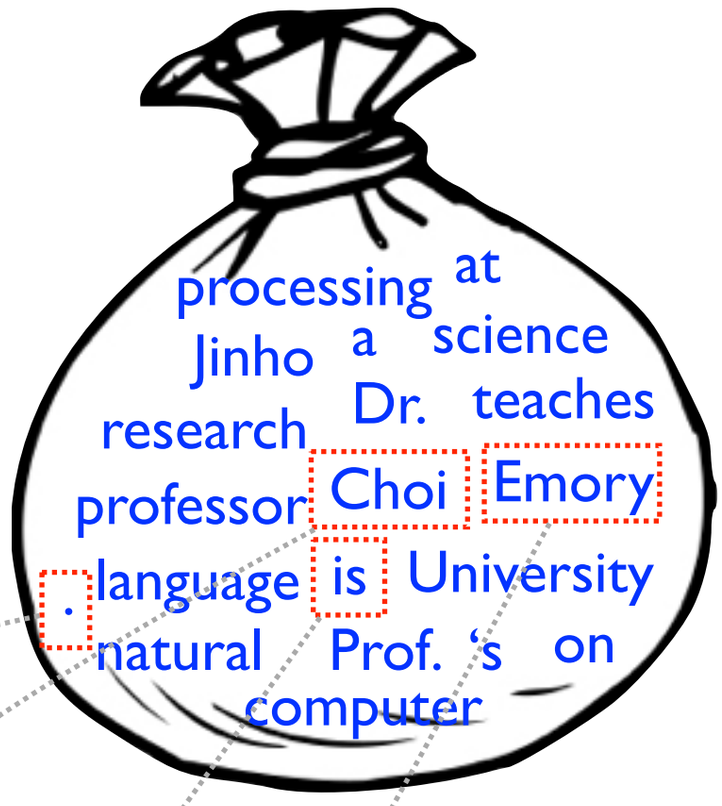
How to represent a document in a vector space?



Bag-of-Words

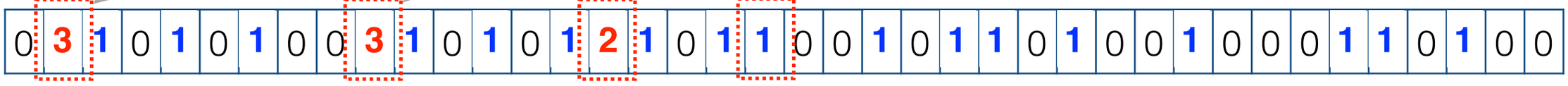
Bag-of-Words

Jinho Choi is a professor at Emory University .
Prof. Choi teaches computer science .
Dr. Choi 's research is on natural language processing .



"." as important as "Choi"?

"is" more important than "Emory"?



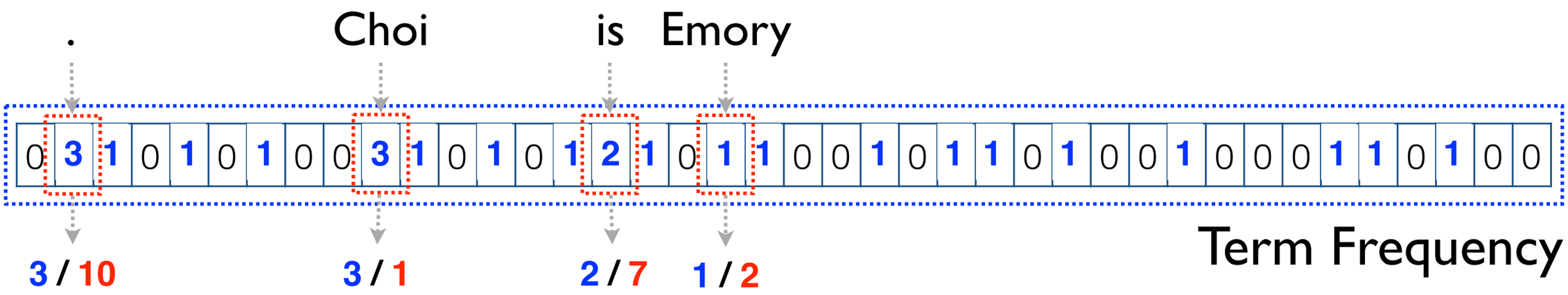
Are all words equally important?

Bag-of-Words

Jinho Choi is a professor at Emory University .

Prof. Choi teaches computer science .

Dr. Choi 's research is on natural language processing .



Out of 10 documents:

"."	appears in	10 documents
"Choi"	appears in	1 document
"is"	appears in	7 documents
"Emory"	appears in	2 documents

Document Frequency

TF-IDF

Given a set of documents *D*:

Term Frequency of *w* in *d* ∈ *D* = # of times that *w* appears in *d*

Document Frequency of *w* ∈ *D* = # of documents that *w* appears

$$\mathrm{tf} \cdot \mathrm{idf}_{w,d} = \mathrm{tf}_{w,d} \cdot \log \frac{|D|}{\mathrm{df}_w}$$

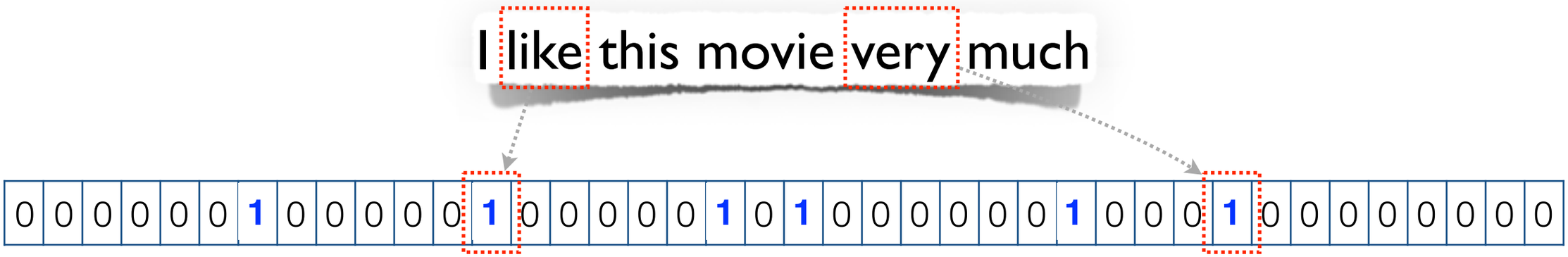
Inverse Document Frequency

sublinear $$\mathrm{wf}_{w,d} = \begin{cases} 1 + \log \mathrm{tf}_{w,d} & \text{if } \mathrm{tf}_{w,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

normalized $$\mathrm{ntf}_{w,d} = \alpha + (1 - \alpha) \frac{\mathrm{tf}_{w,d}}{\mathrm{tf}_{\max}(d)}$$

Document Similarity

Compare two documents in a vector space?

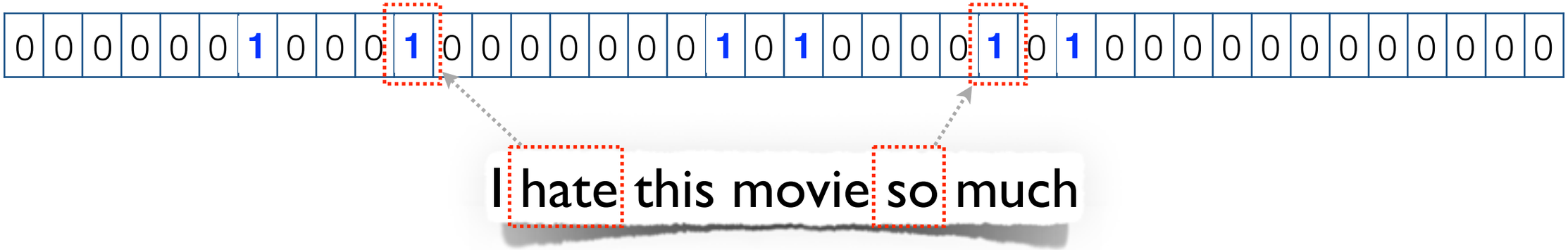


Euclidean distance

$$\|p - q\| = \sqrt{2}$$

Cosine similarity

$$\frac{p \cdot q}{\|p\|\|q\|} = \frac{4}{\sqrt{6} \cdot \sqrt{6}} = \frac{2}{3}$$



Document Similarity

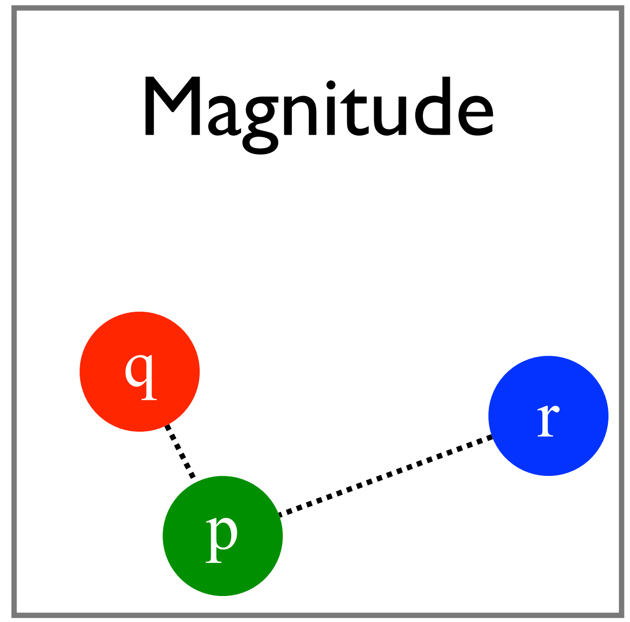
I like this movie very much

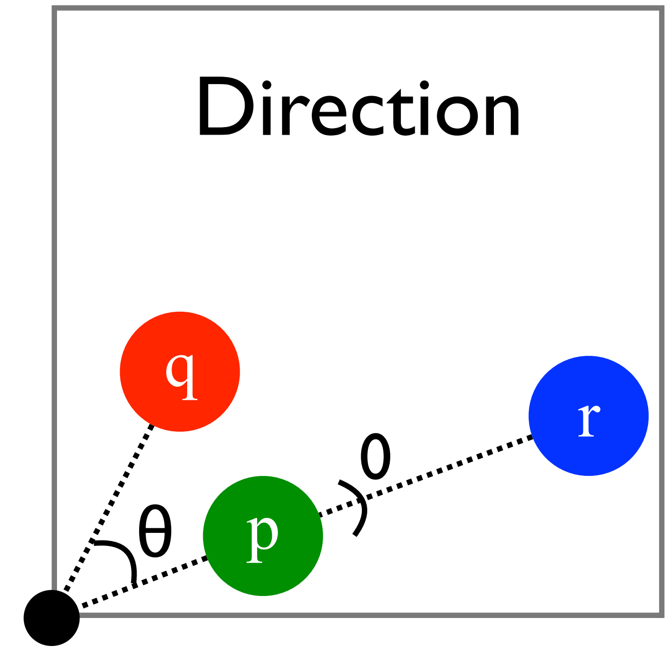
I hate this movie so much

I like this movie very much
I like this movie very much
I like this movie very much

euc(p, q) = 2
euc(p, r) = 4.90

cos(p, q) = 0.67
cos(p, r) = 1





0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0

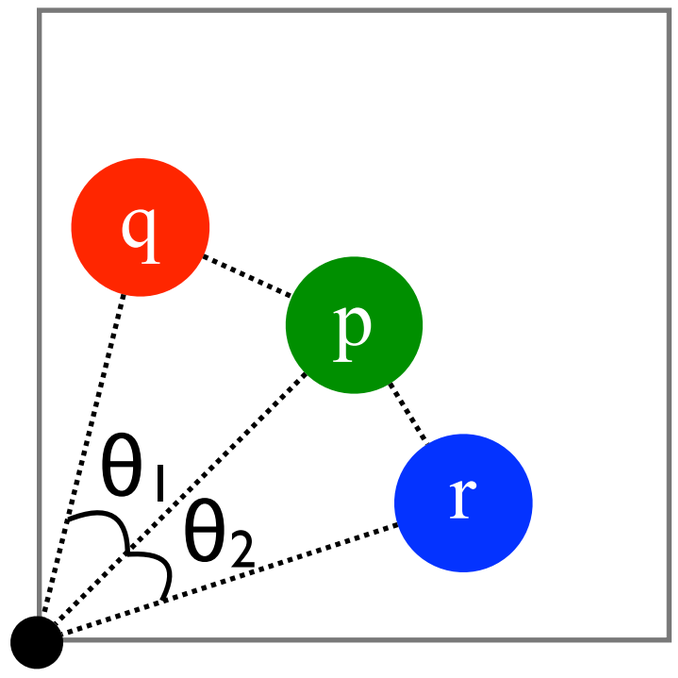
0	0	0	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	3	0	3	0	0	0	0	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0

Document Similarity

I like this movie very much

I hate this movie so much vs. I love this movie so much

$$\text{euc}(p, q) \approx \text{euc}(p, r)$$



$$\cos(p, q) \approx \cos(p, r)$$

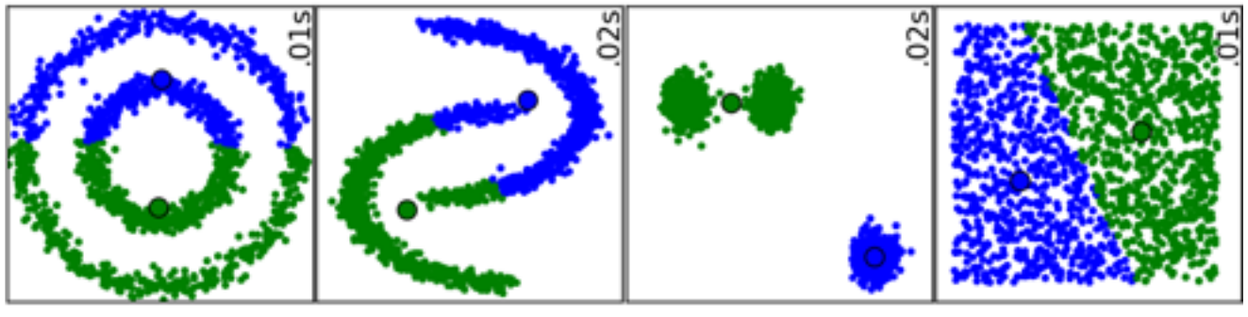
Shouldn't this be more similar?

Represent documents using word embeddings!

Clustering

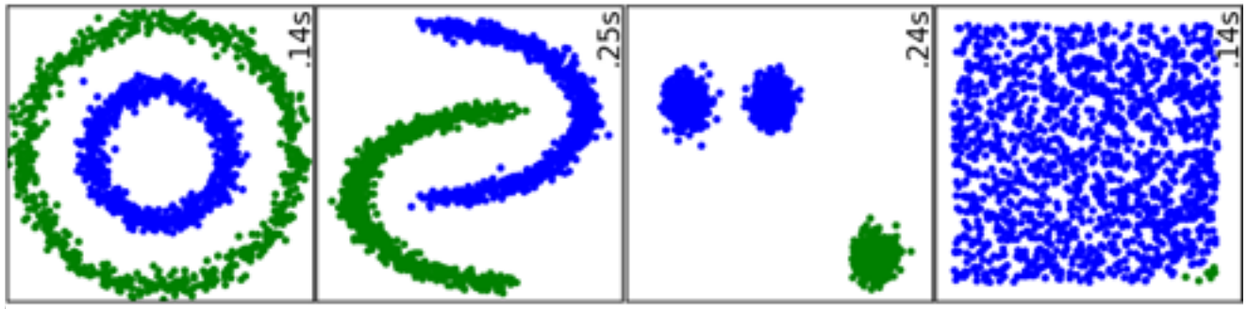
Group vectors together by their similarities.

Partition-based



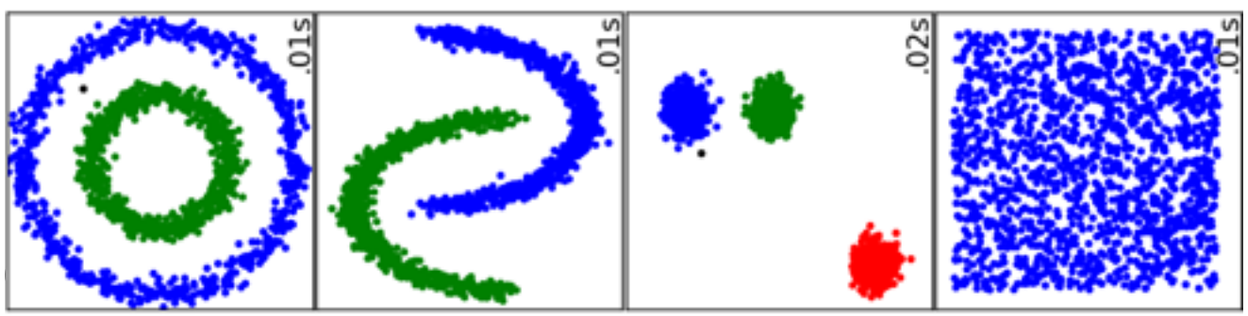
K-means

Hierarchical



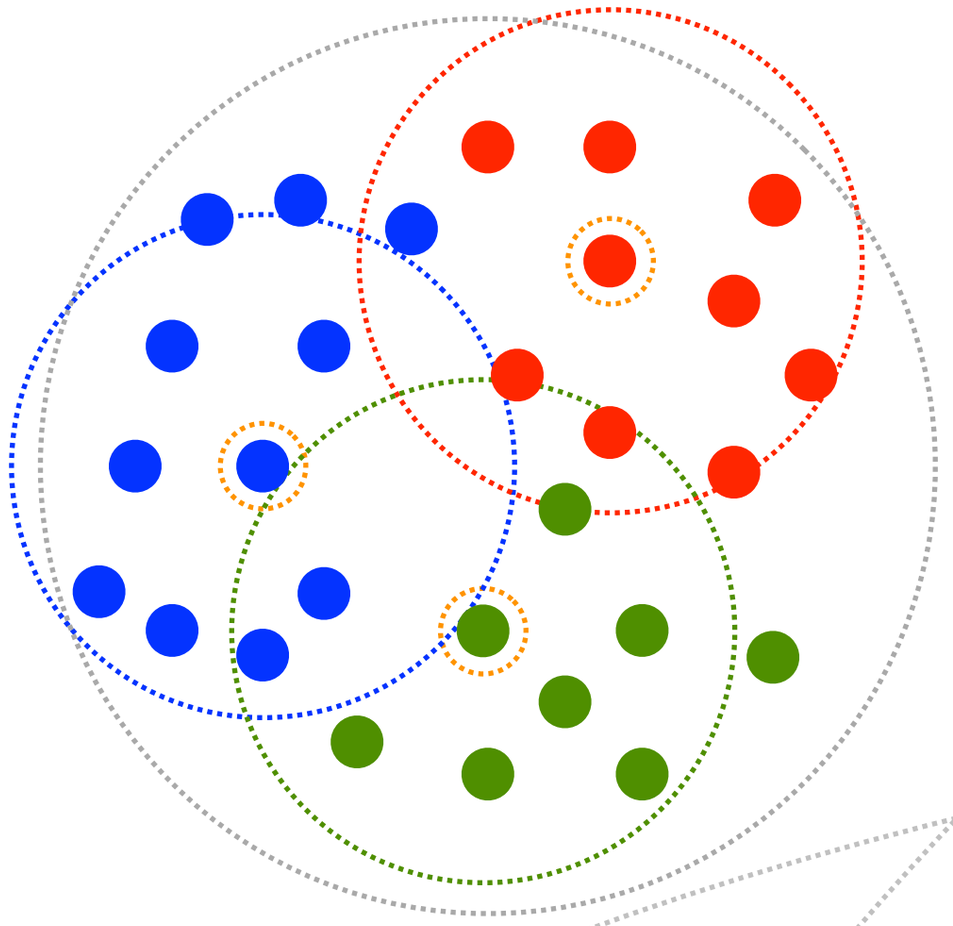
Agglomerative

Density-based



DBSCAN

K-Means Clustering



best we can do?

Pick random *k* vectors.

These represent centroids.

For each vector, group it with its nearest centroid.

Find centroids.

Centroids don't have to be the actual vectors.

Repeat until converges.

Expectation Maximization

EM algorithm!

K-Means++ Clustering

Introduce a random vector c as the first centroid.

Measure the distance between every vertex x to its nearest centroid c.

$$D(x) = \min_{\forall c} dist(x, c)$$

Measure the distance probability for each vertex x.

$$P(x) = \frac{D(x)^2}{\sum_{\forall x} D(x)^2}$$

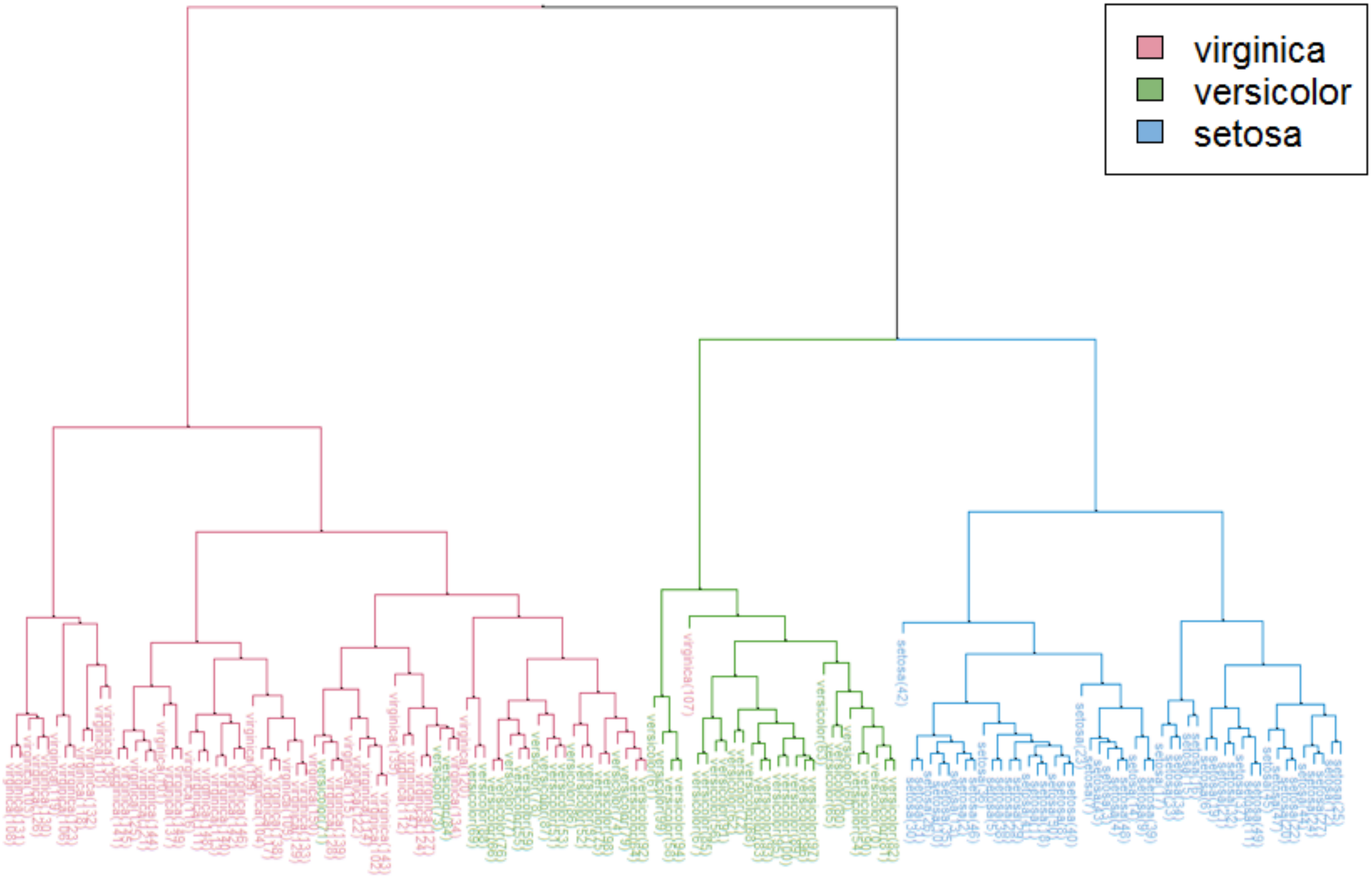
Measure the cumulative probability for each vertex x.

$$C(x_0) = 0 \quad C(x_i) = \sum_{j=1}^{i} P(x_j)$$

Pick a random number r in (0, 1].

Choose x_i as the next centroid s.t. $C(x_{i-1}) < r \leq C(x_i)$

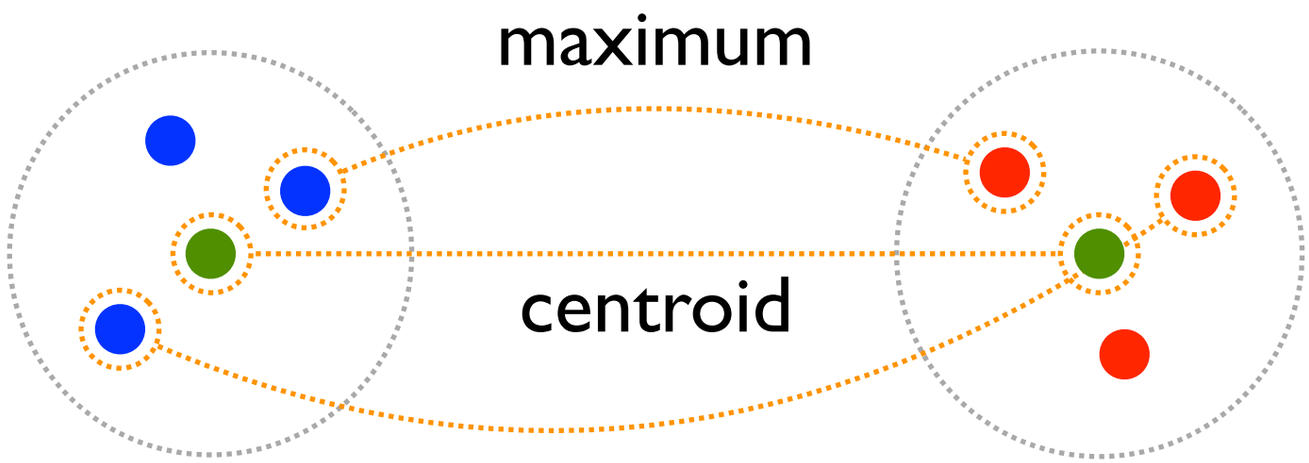
Hierarchical Agglomerative Clustering



Hierarchical Agglomerative Clustering

Initially, each vector becomes a cluster.

Measure the similarity between every pair of clusters.

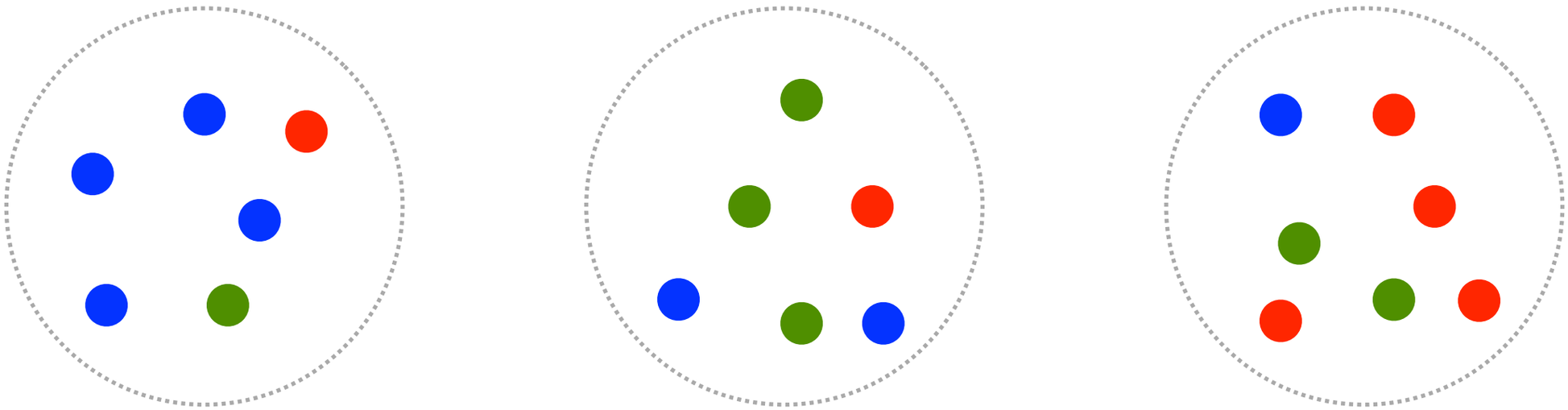


Create a new cluster by merging two clusters that are most similar.

Measure the similarity between the new cluster and every other cluster.

Purity Score

How to evaluate clusters?



4 3 4

Count of the genre with the maximum documents

Purity = (4 + 3 + 4) / 19