Understanding ISE2

Process

1. **Preprocessed data** → Converted all your cleaned documents into token lists.
2. **Created dictionary & corpus** → Converted words into numeric IDs and counts.
3. **Trained LSA model** → Found latent *topics* across all documents using **Singular Value Decomposition (SVD)**.
4. **Generated embeddings:**
    - `document_embeddings.npy` → Each document becomes a numeric vector of length `N_TOPICS` (e.g. 130).

      Example:

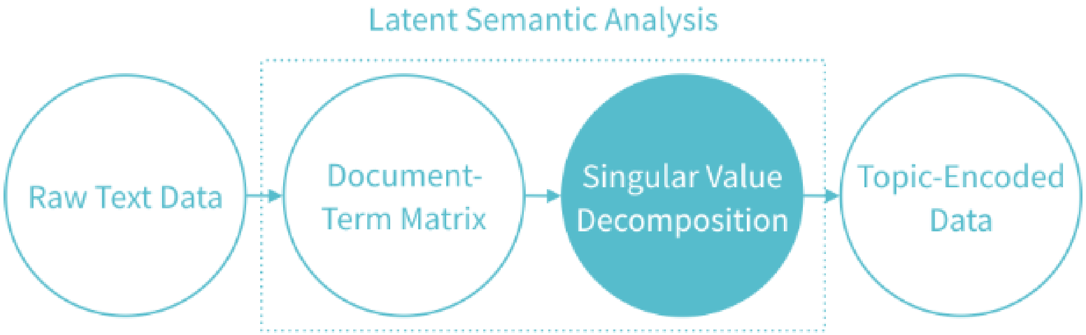
      ```
      Document 0 → [0.21, -0.03, 0.17, ..., 0.09]
      Document 1 → [0.18, -0.02, 0.22, ..., 0.11]
      ```

    - `term_embeddings.npy` → Each word becomes a numeric vector representing its relation to topics.

So after that step, you had **dense vector representations** (embeddings) for every document.

Latent Semantic Analysis

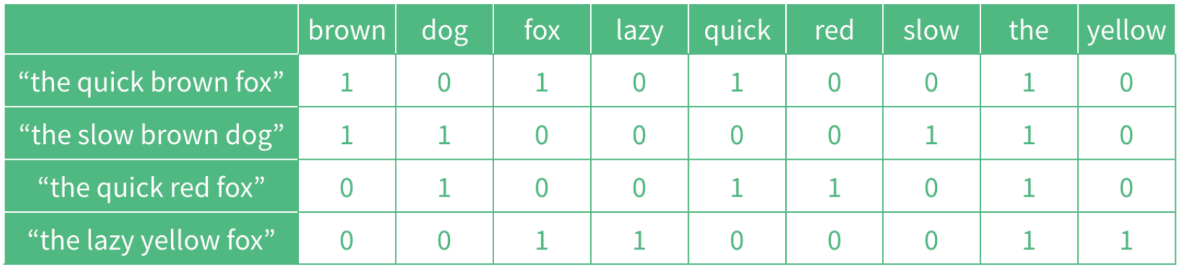
- LSA uses bag of word(BoW) model, which results in a term-document matrix



1. Document Term Matrix

   If term is present or in the document

   A basic idea of a Document-Term Matrix is that documents can be represented as points in Euclidean space aka **vectors**.



	brown	dog	fox	lazy	quick	red	slow	the	yellow
“the quick brown fox”	1	0	1	0	1	0	0	1	0
“the slow brown dog”	1	1	0	0	0	0	1	1	0
“the quick red fox”	0	1	0	0	1	1	0	1	0
“the lazy yellow fox”	0	0	1	1	0	0	0	1	1

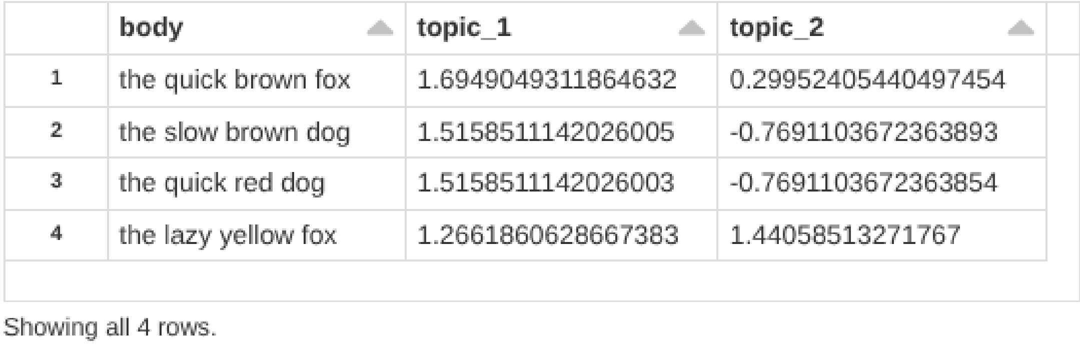
2. Singular Value Decomposition

   Use -

   a. reduces the dimension of the original data

   b. transforms the data to be encoded using latent, or hidden, variables

   c. for LSA, these latent variables represent topics



	body	topic_1	topic_2
1	the quick brown fox	1.6949049311864632	0.29952405440497454
2	the slow brown dog	1.5158511142026005	-0.7691103672363893
3	the quick red dog	1.5158511142026003	-0.7691103672363854
4	the lazy yellow fox	1.2661860628667383	1.44058513271767

Showing all 4 rows.

But… how are the topics found??

Topic Modeling

Topic modeling is a text mining technique which provides methods for identifying co-occurring keywords to summarize large collections of textual information. It helps in discovering hidden topics in the document, annotate the documents with these topics, and organize a large amount of unstructured data.

so basically what we want to do,

! can add some visualisation to check at the end to check the res with comparision to the data that we have (excel sir gave for A4)

- unsupervised text analytics algorithm
- There is a possibility that, a single document can associate with multiple themes.

Text Classification vs Topic Modeling

Text Classification

- supervised
- text document or article classified into a pre-defined set of classes

Topic Modeling

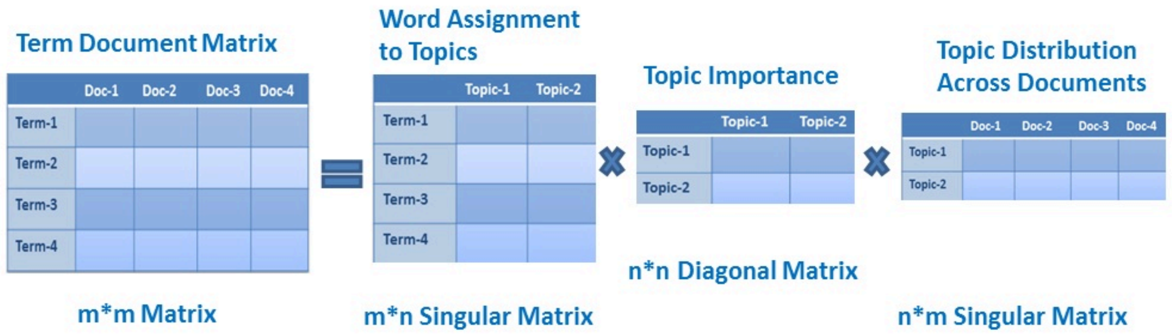
- unsupervised
- process of discovering groups of co-occurring words in text documents

- These group co-occurring related words makes "topics”
- set of possible topics are unknown

How they work together? →

Topic modeling can be used to solve the text classification problem. Topic modeling will identify the topics presents in a document" while text classification classifies the text into a single class.

SVD is a matrix factorization method →



$$M=U\Sigma V^*$$

- M is an m×m matrix
- U is a m×n left singular matrix
- Σ is a n×n diagonal matrix with non-negative real numbers.
- V is a m×n right singular matrix
- V* is n×m matrix, which is the transpose of the V.

Optimum number of topics -

How to find?

1. One way to determine the optimum number of topics is to consider each topic as a cluster and find out the effectiveness of a cluster using the Silhouette coefficient.
2. **Topic Coherence -** realistic way
   uses - latent variable models
   Each generated topic has a list of words. find average/median of pairwise word similarity scores of the words in a topic. The high value of topic coherence score model will be considered as a good topic model.
   what is considered high value?

By products of LSA

The LSA generates a few byproducts that are useful for analysis:

- the **dictionary** or the set of all words that appear at least once in the **body**
- the **encoding matrix** used to encode the documents into topics. The encoding matrix can be analyzed to identify the hidden topics underlying the dataset.the encoding matrix used to encode the documents into topics. The encoding matrix can be analyzed to identify the hidden topics underlying the dataset.
  we can do this part also - check out **Plot Topic Encoded Data** from link
  and the last part of that article too

O/P of LSA from the code

1. doc_embeddings → how much a doc is related to a topic
   Used for - Find similar documents, cluster topics, visualize groups
2. term_embeddings → how much a term is related to a topic
   Used for - Find similar terms, build concept networks, synonym analysis

KD Tree

Main idea - to see how similar documents are by comparing the embeddings from LSA. so mainly, here we are using - Cosine DIstance for the same

Cosine Distance

This is like to concept of distance we use to see corelation like mahalanubis or eucledian

But simply using *distance* (like Euclidean distance) isn't ideal, because:

- Long documents naturally have bigger numbers.
- Shorter documents have smaller numbers — even if they talk about the same thing.

So instead of comparing *how far apart* two vectors are,

we compare **how much they point in the same direction —** that's what **cosine similarity** does.

Cosine Similarity, Mathematical

Why use it?

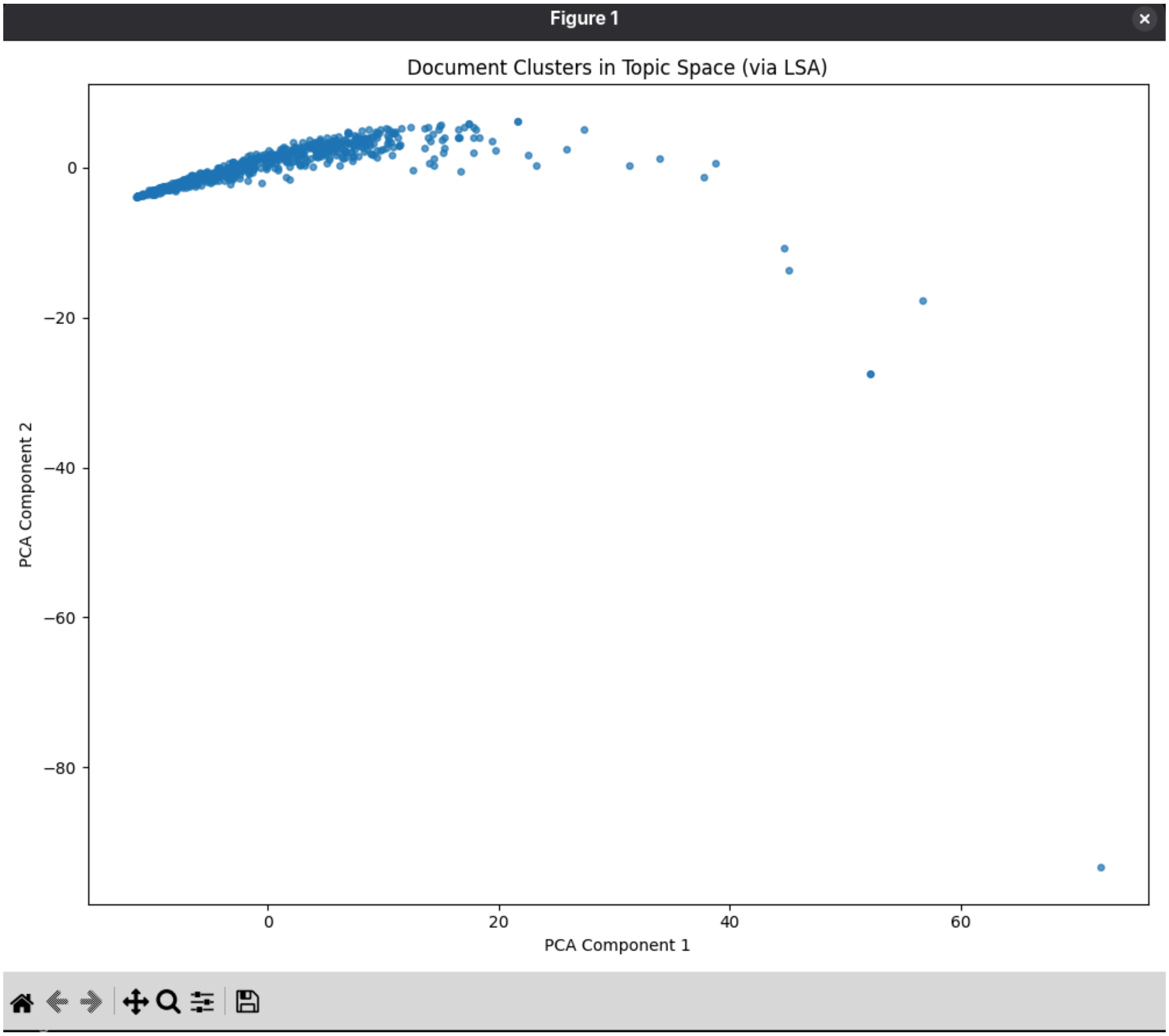
- Each **document embedding** and **term embedding** is a vector in high-dimensional topic space.
- Two vectors that **point in the same direction** → talk about similar topics or concepts.
- So, cosine similarity tells you **how semantically close** two documents or words are —

  not how long or big they are.

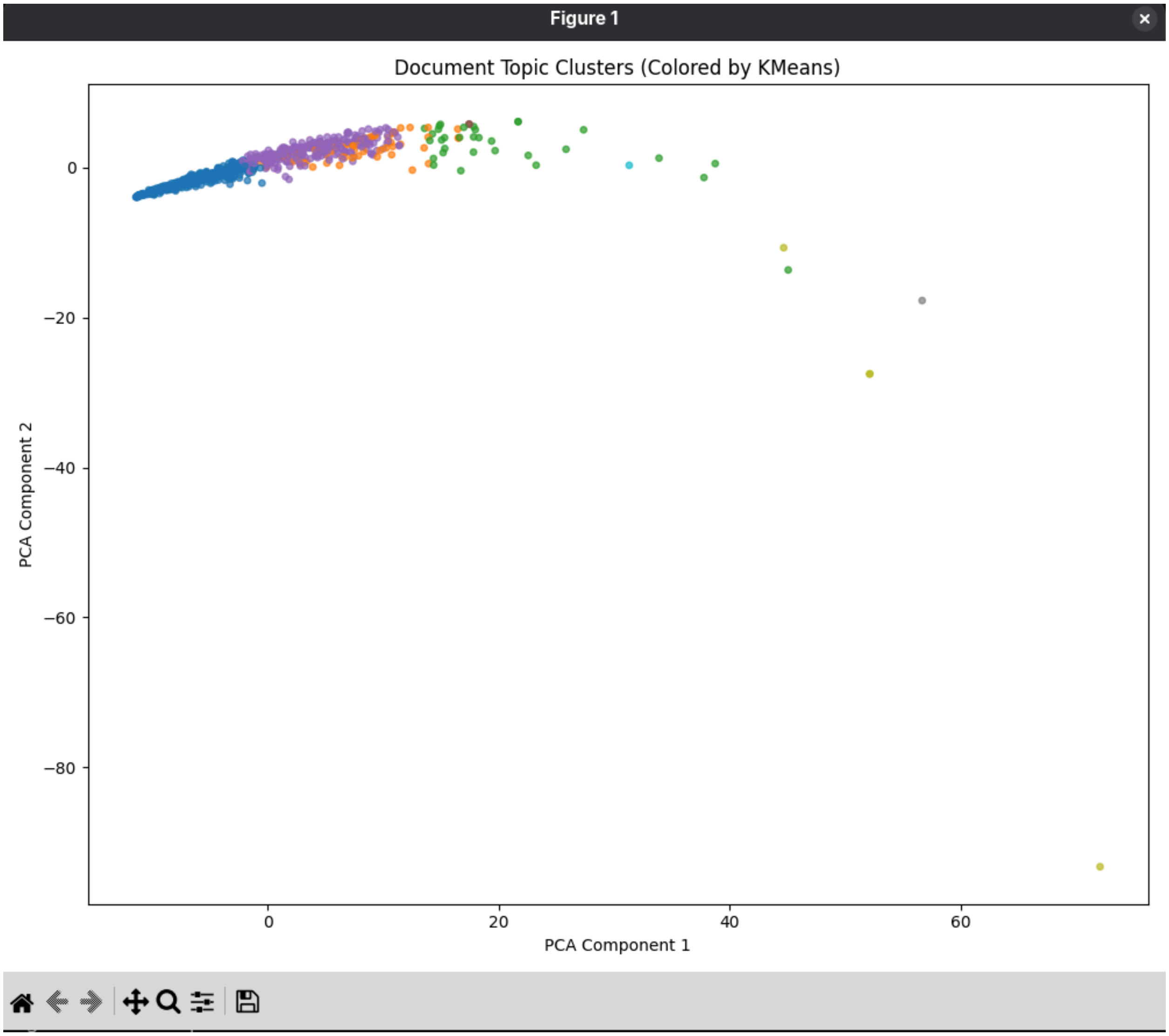
SO end product

dataset of **document-document associations** discovered automatically from your corpus.

Visualisation

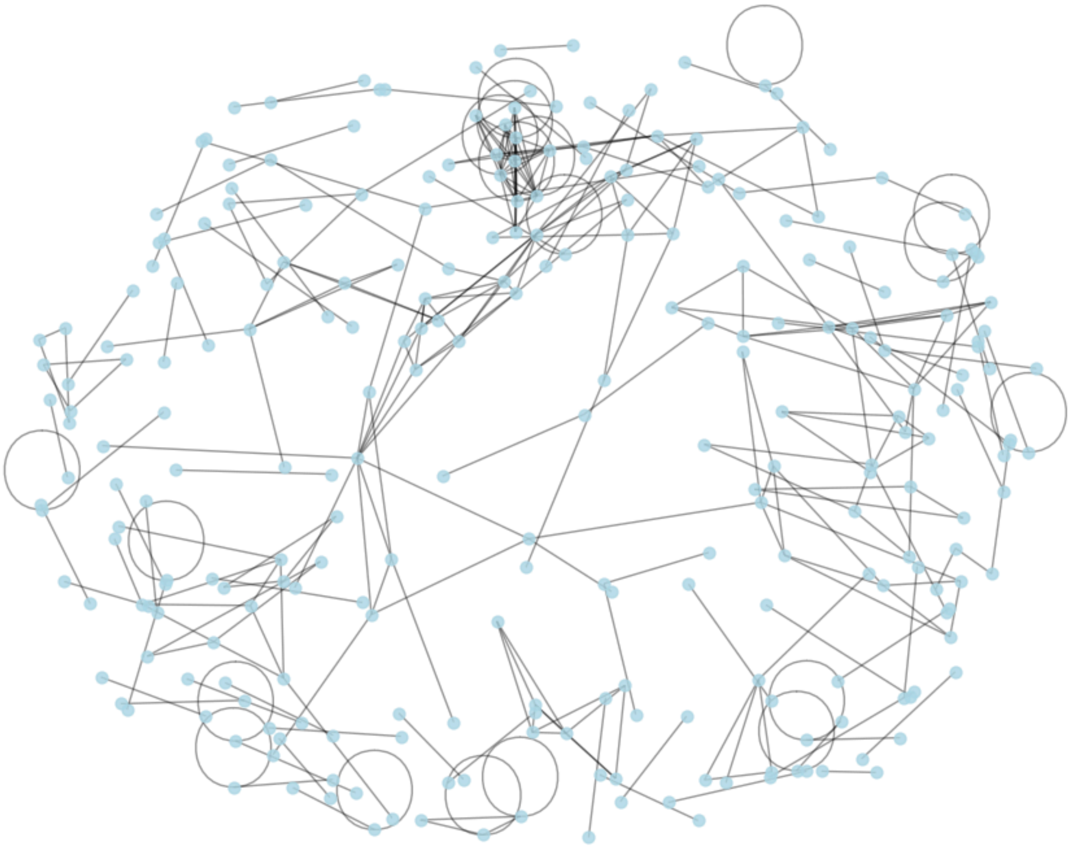


- PCA (Principal Component Analysis) takes your 130-topic vector and **squashes it down to 2D** — so you can plot it on a normal chart.
  Component 1 -
  Component 2 -
- Each dot = one document.
- Dots that are **close together** have **similar topic mixtures** (they talk about the same kinds of things).
- Dots that are **far apart** are about totally different topics.

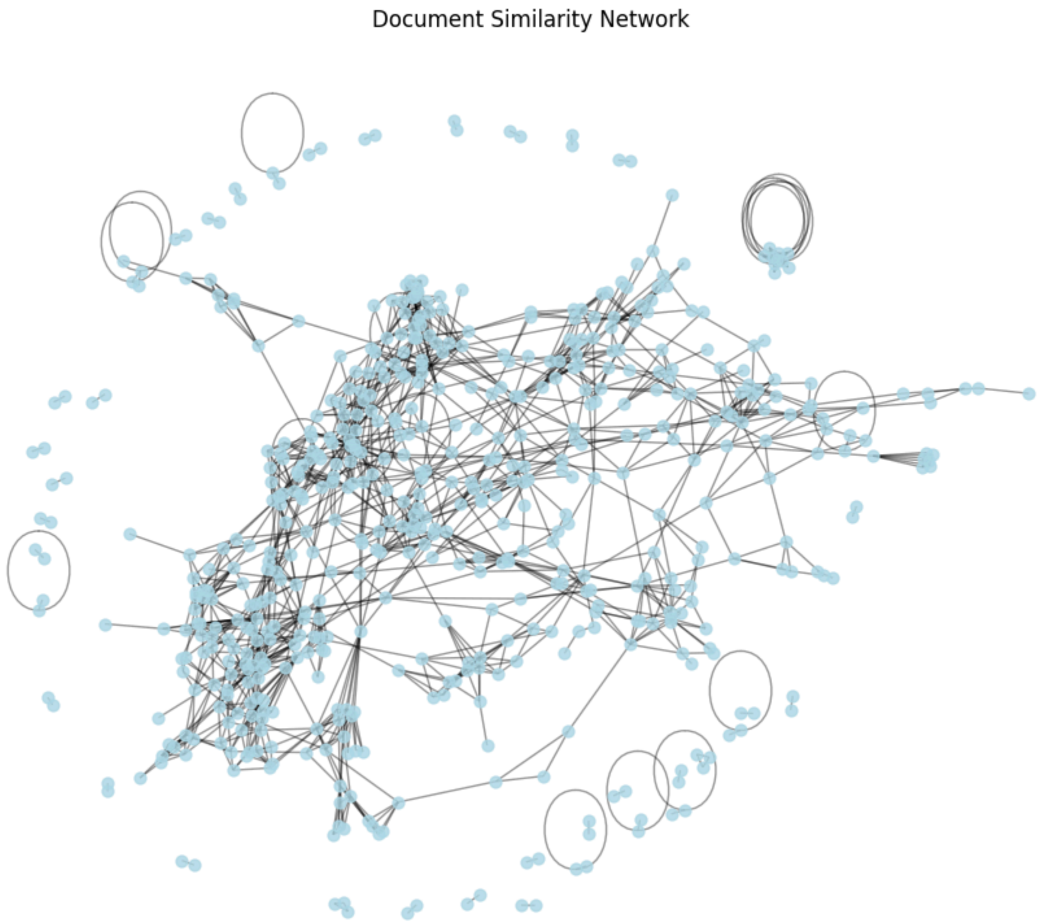


- KMeans tries to find "centers" in your topic space — basically, automatic groups of similar documents.
- Each color = one cluster (a group of similar docs).

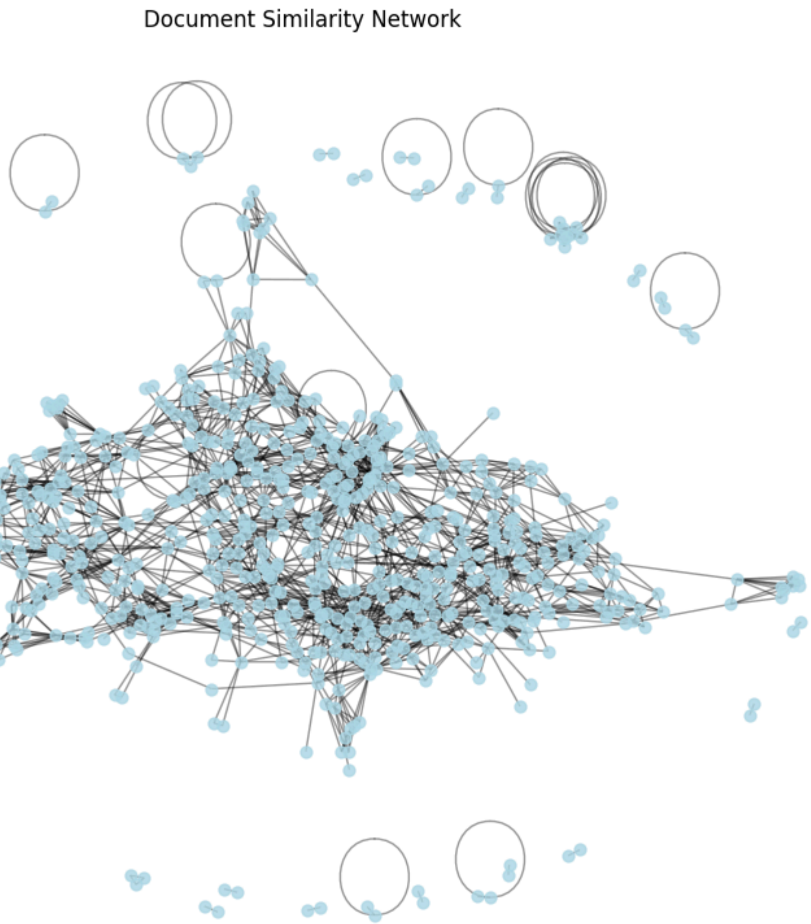
Document Similarity Network



- Here, we take only **strong relationships** (similarity > 0.9).
- Each document becomes a **node**, and if two documents are very similar, we draw a **line (edge)** between them.



when deg of similarity is dec to 0.8



dec more to 0.7

PPT Division

Page 1 -
- Title Slide

Page 2 -
- Anshuman’s Part

Page 3 & 4 - LSA
- Related Theory

- Models + their significance

Page 5 & 6 -

- Explain procedure - Huma
- compare the 2 - Aditi

Page 7 -

- Analysis and Conclusions