JM2050 – Natural Language Processing

Topic Modeling

October 10th, 2024

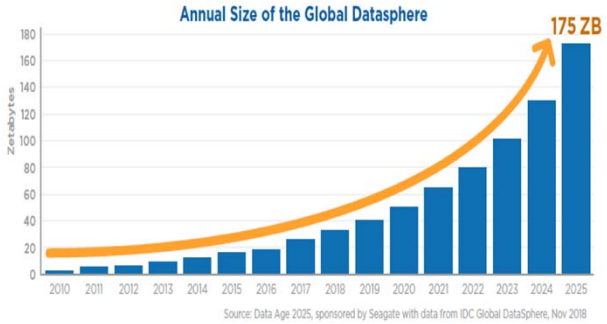
Prof.dr.ir. U. Kaymak

Recap

- NLP Tasks
- Part-of-speech (POS) tagging
- Text classification
- Named entity recognition
- Dialogue systems
- Linguistic summaries
- Healthcare applications

[2]

By 2025, the world is forecasted to produce 175 ZB of data



90% Unstructured

https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/

https://www.forbes.com/sites/bernardmarr/2019/10/16/what-is-unstructured-data-and-why-is-it-so-important-to-businesses-an-easy-explanation-for-anyone/

1 zettabyte is 1,000,000,000,000,000,000,000 gigabytes (10^{21})

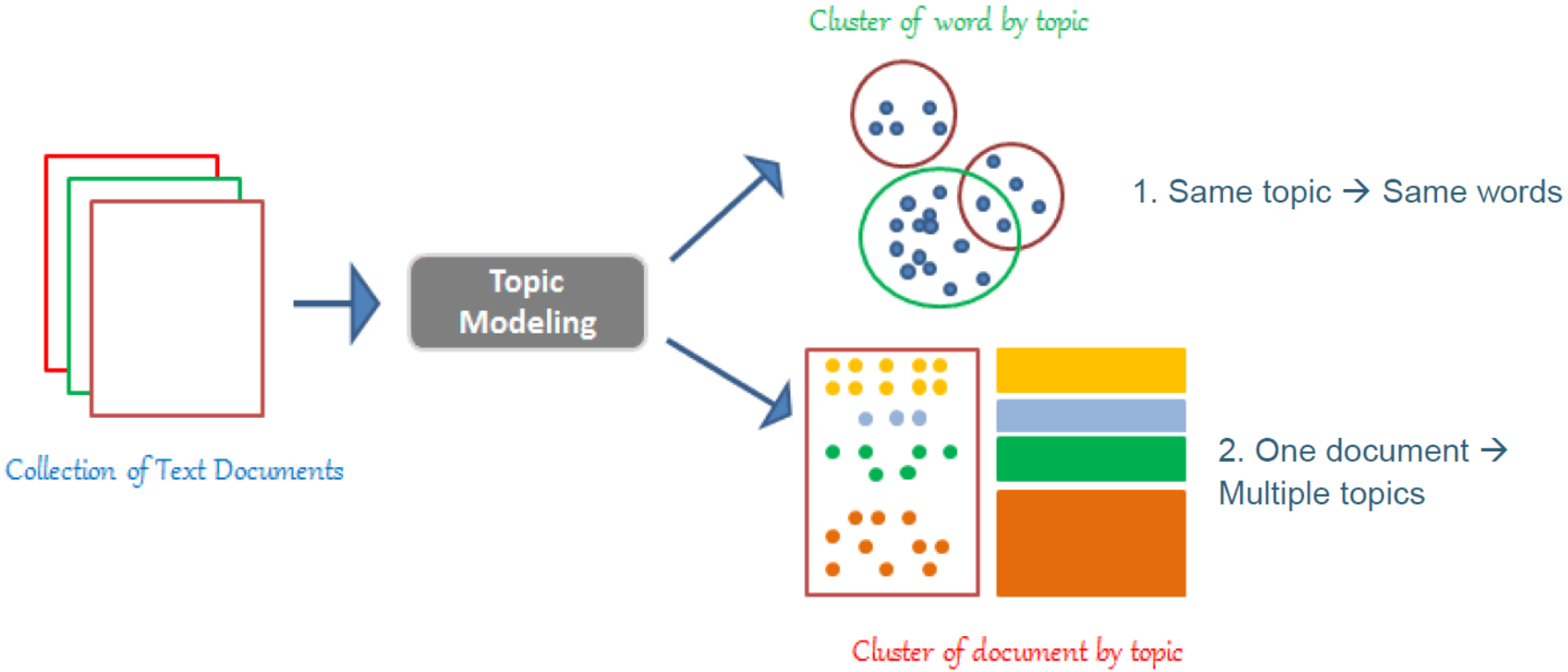
JADS Jheronimus Academy of Data Science

What is a topic model

"A **statistical** model to discover **latent** topics from a **corpus**"

- Statistical model → probabilities
- Corpus → collection of documents
- Latent topics → clusters of associated words

What is a topic model?



5

What are topic models

Topic model = an (unsupervised) statistical model to discover latent topics from a corpus.

Input:
corpus + number of topics

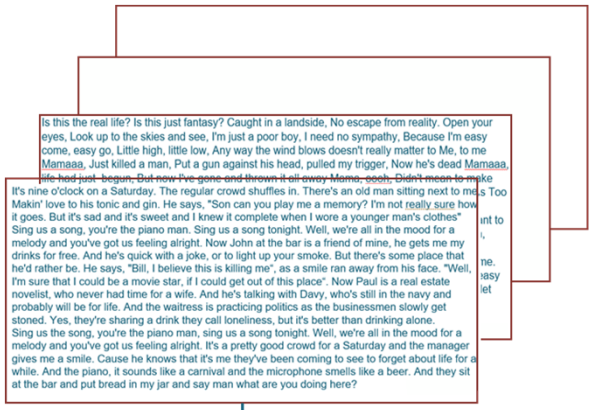
Output:
Two matrices:
1. **P(word|topic)** (*words x topics*)
2. **P(topic|document)** (*topics x documents*)

Traditional topic modeling algorithms return two matrices. This need not be the case for all models.

Per topic, the top *n* words are typically used to represent a topic

6

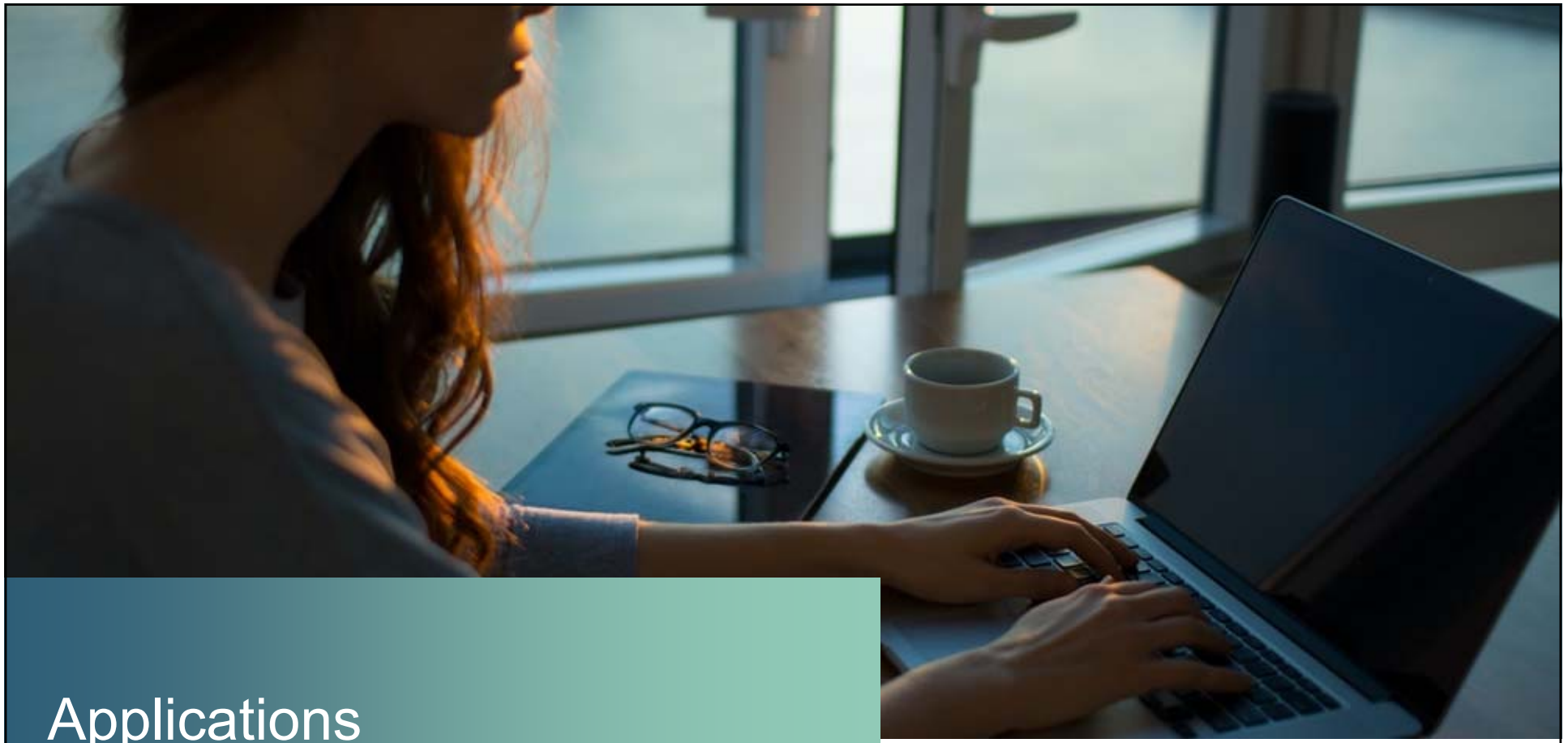
Fictional example of output with 3 topics and 10 words




```
[(0,
  '0.0219*"real" + 0.0143*"life" +
0.0126*"fantasy" + 0.0071*"reality" +
0.0067*"galileo" + 0.0055*"poor" +
0.0051*"boy" + 0.005*"mama" + 0.005*"mia"
+ 0.005*"matters"'),

 (1,
  '0.0167*"submarine" + 0.011*"yellow" +
0.0065*"sailed" + 0.0064*"sun" +
0.0062*"sea" + 0.0055*"waves" +
0.0047*"friends" + 0.0046*"sky" +
0.0046*"live" + 0.0045*"sailed" '),

 (2,
  '0.0064*"piano" + 0.0045*"man" +
0.0035*"crowd" + 0.0035*"saturday" +
0.0031*"tonic" + 0.0031*"gin" +
0.0023*"bar"  + 0.0018*"drinking" +
0.0017*"beer" + 0.0016*"melody"+
0.0015*"sitting" ')]
```



Applications

Applications - Classification

Classification of academic journals' archives

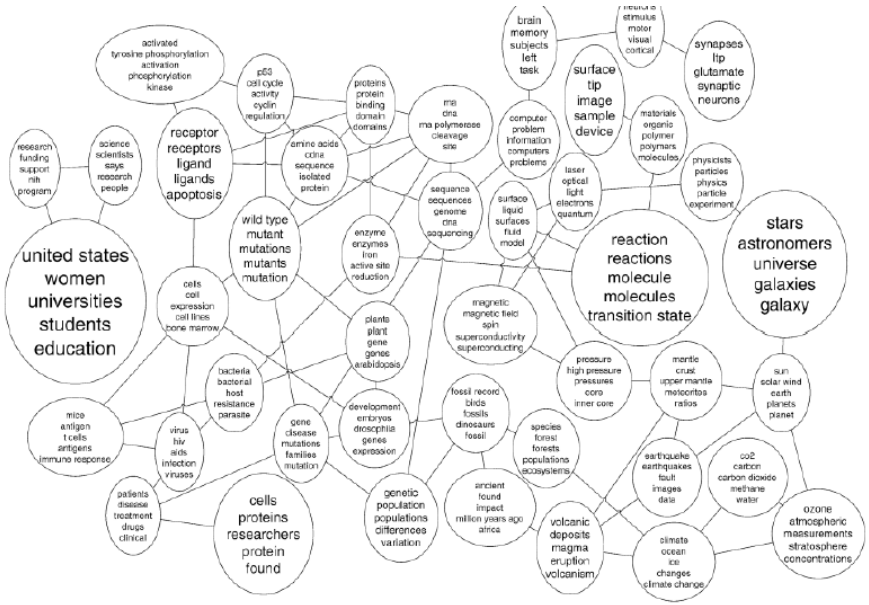


FIG. 2. *A portion of the topic graph learned from* 16,351 *OCR articles from Science* (1990–1999). *Each topic node is labeled with its five most probable phrases and has font proportional to its popularity in the corpus.* (*Phrases are found by permutation test.*) *The full model can be found in* http://www.cs.cmu.edu/~lemur/science/ *and on STATLIB.*

Applications – Content Recommendation

E.g. newspapers

How?

Approach:

1. Identify the topics in a reader's articles
2. Recommend other similar articles

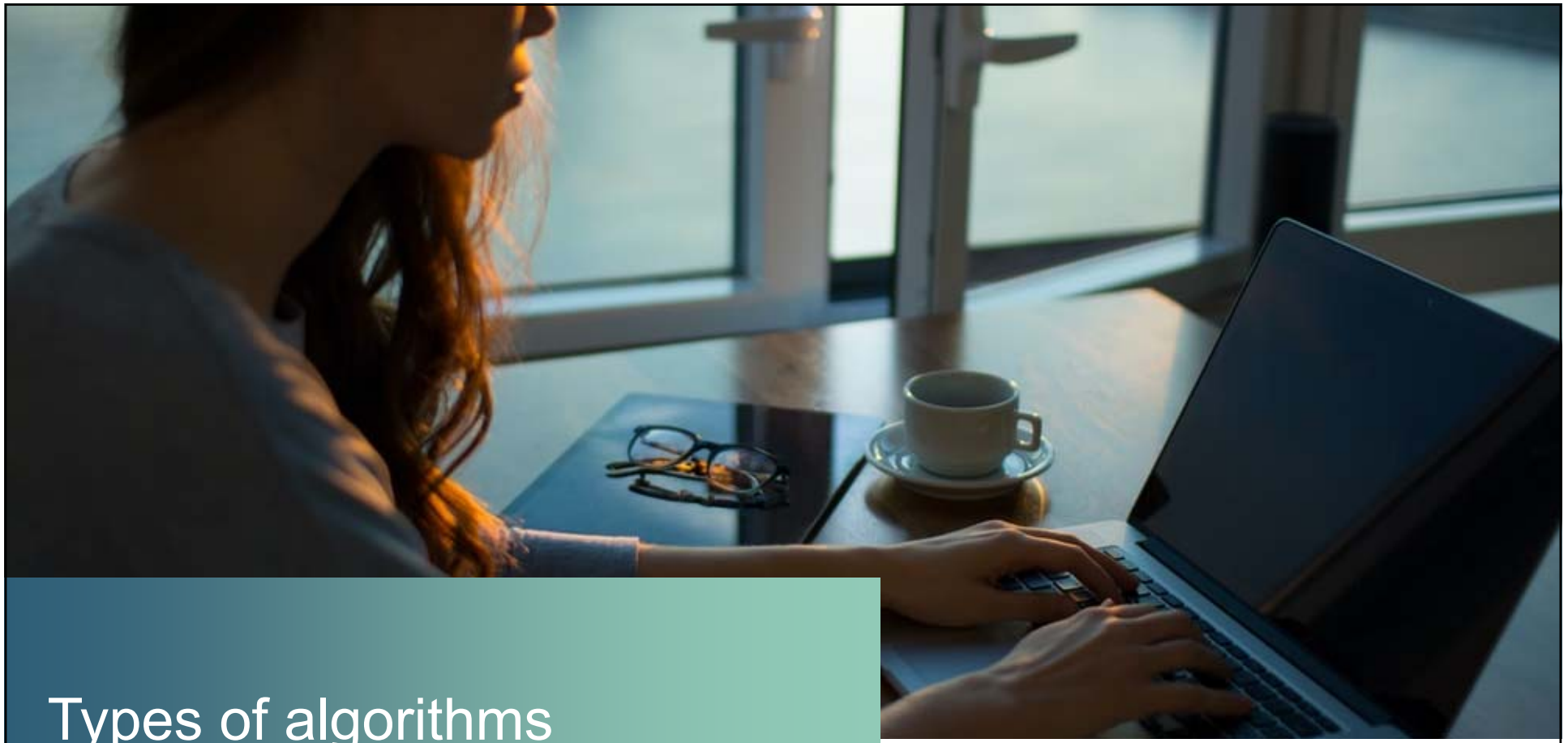
Applications – Understanding Themes in Large Datasets

E.g. Patient feedback

How?

Approach:

1. Cluster feedback in different groups (e.g. positive and negative).
2. For each group, identify common topics in feedback.



Types of algorithms

Types of topic models

Probabilistic models: assume that each document is generated by a probabilistic process, where each word in the document is generated by a topic (e.g. LDA[1]).

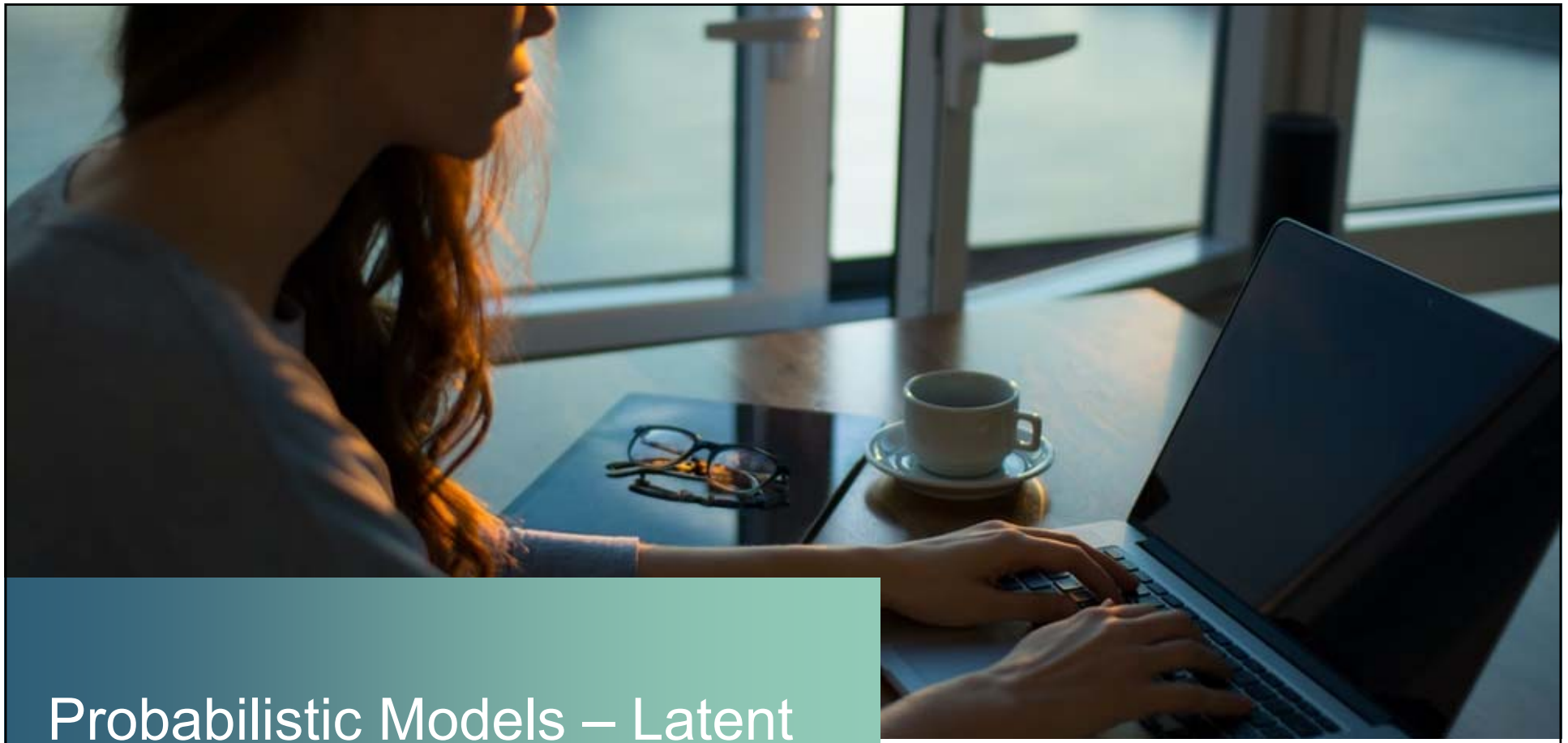
Factorization models: These models factorize the document-term matrix into two matrices, which can be interpreted as a document-topic matrix and a topic-term matrix (e.g. LSA[3]).

Deep learning-based models: These models use neural networks to learn the underlying structure of the documents and extract topics (e.g. BERTopic).

1) Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

2) Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794.

3) Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284.

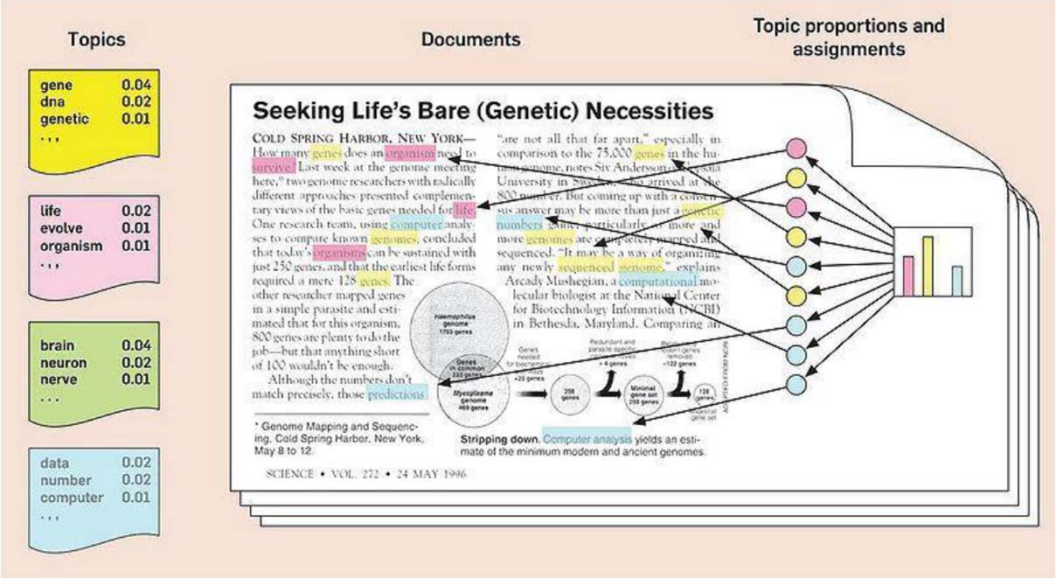


Probabilistic Models – Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA)

LDA assumes documents can be created by sampling from:

- a distribution of topics over documents.
- a distribution of words over topics.



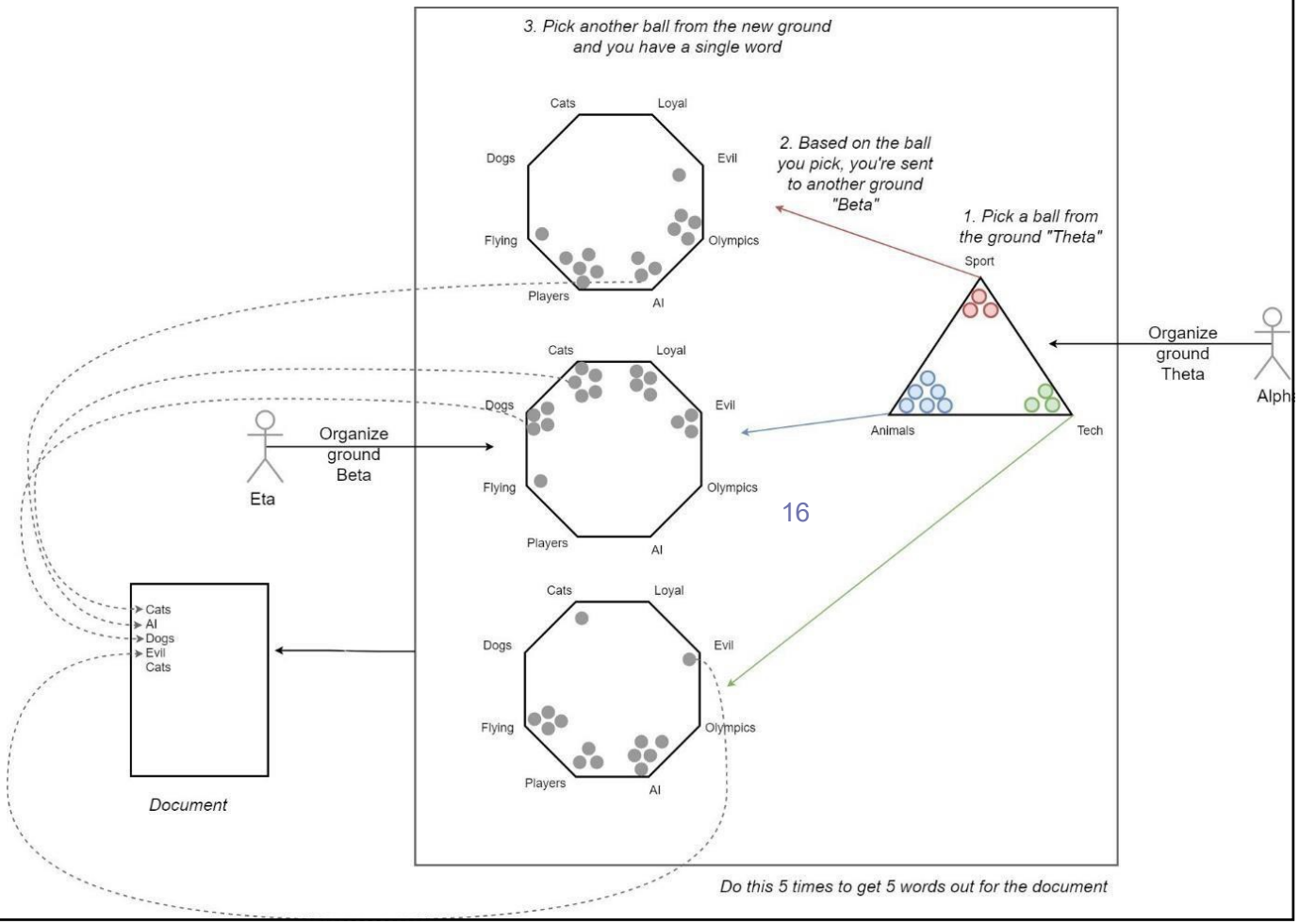
LDA

How a document is generated:

- α organizes the ground θ (=topic distribution) and then you go and pick a ball from θ.
- Based on what you pick, you're sent to ground β (=topic-word distribution). β is organized by η.
- Now you pick a word from β and put it into the document.

Iterate this process N times to get N words out.

(Ganegedara, 2018)



LDA

Plate notation of the generative probabilistic process

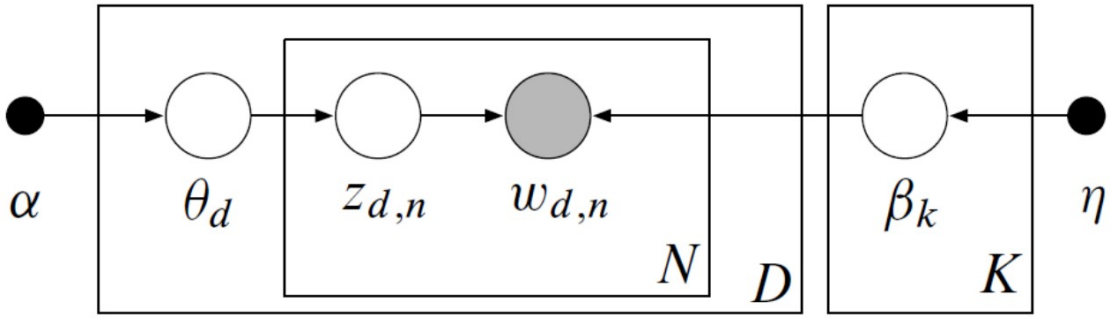


Plate: repetition of variables

Node:
- Grey: observed
- White: hidden
- Black: parameter

$w_{d,n}$ nth word in document d
β_K distribution over the vocabulary for topic K
θ_d topic distribution for document d
$z_{d,n}$ topic assignment for the nth word in document d
α parameter of prior distribution over θ
η parameter of prior distribution over β

17

We don’t know these distributions

Observed: documents in corpus

Hidden: topics; document topic distributions ; topic word distributions; word topic

LDA solves the computational problem of inferring the hidden topic structure from the observed documents

LDA Algorithmic Steps

1. **Initialize the model parameters**: randomly assign each word in the vocabulary to a topic. The document-topic distribution is then initialized by assigning each document a uniform distribution over the topics.
2. **Iterate through the documents**: loop over each document in the corpus. For each document, the topic distribution is sampled first. Then, for each word in the document, the topic for the word is sampled. Finally, the topic-word distribution for the topic is updated.
3. **Repeating step 2 until the model converges**: continue to iterate through the documents until the topic distributions for the documents do not change significantly from one iteration to the next. This is typically done by setting a maximum number of iterations or by monitoring the change in the topic distributions.
4. **Calculating the topics**: look at the topic-word distributions. The words that are most likely to be assigned to a topic are the words that define that topic. This can be done by looking at the top words in the topic-word distribution for each topic.

19

LDA – How to solve this?

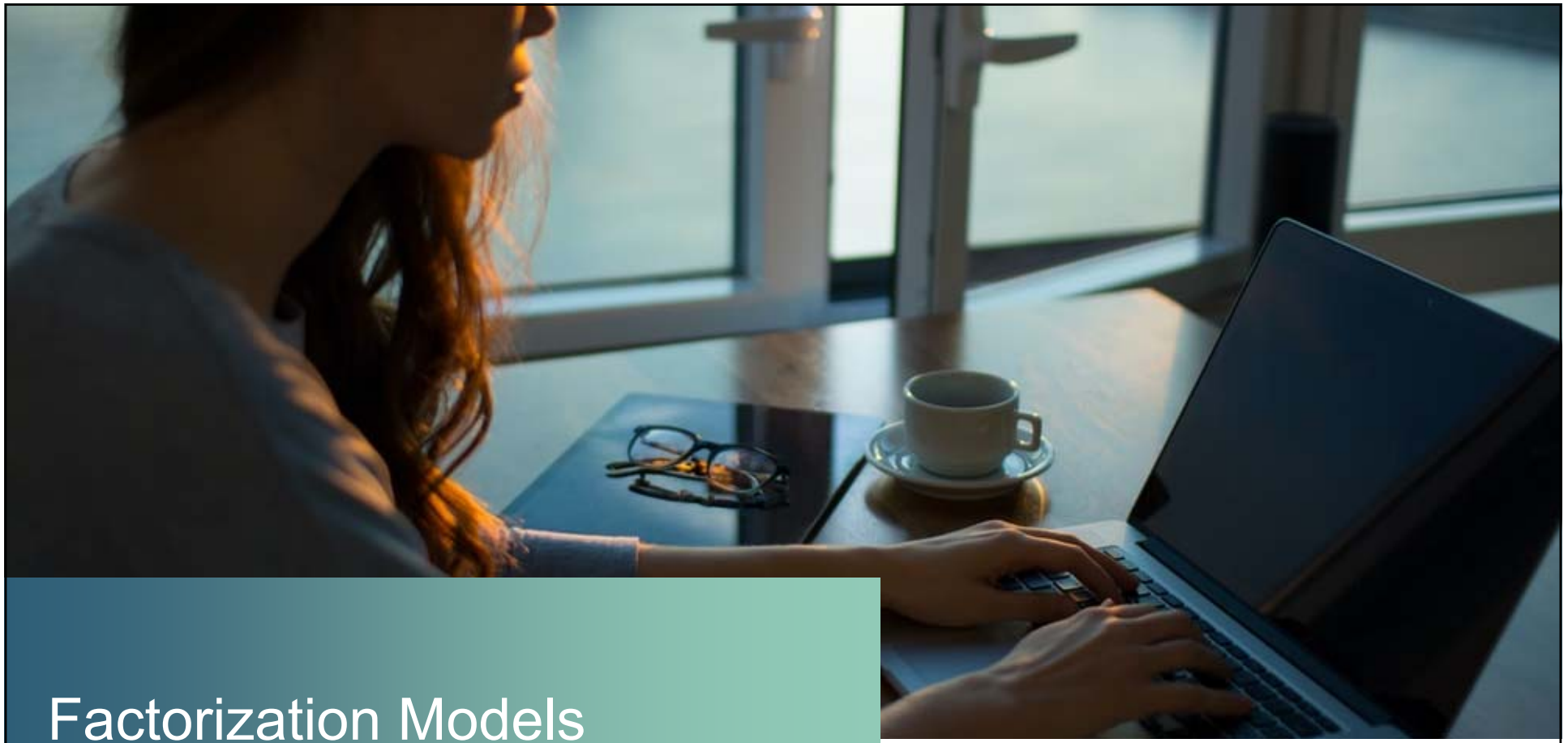
Compute posterior distribution

- Conditional probability of the hidden variables, given the observed variables
- Same as: posterior = conditional probability of topic structure given observed documents

$$p(z,\theta,\beta|w)=\frac{p(z,\theta,\beta,w)}{p(w)}$$

Intractable to compute for all topic structures

Instead, approximate the posterior distribution with a variational inference, expectation maximization or sampling method



Factorization Models

Another approach for finding topics is based on a projection of the data into a lower-dimensional space

Before discussing this branch of topic models, firstly we will discuss two common dimensionality reduction techniques.

An important concept is *factorization* and is easiest explained with numbers.

Factorizing numbers: take a number and represent it as its factors (e.g. $8 = 4 \times 2$)

Matrices can also be factorized into constituent matrices.

Singular Value Decomposition

The first method *Singular Value Decomposition*, decomposes a matrix into its constituent parts.

The singular value decomposition of a matrix $\boldsymbol{A}$ is the factorization of $\boldsymbol{A}$ into the product of three matrices $\boldsymbol{A} = \boldsymbol{UDV^T}$.

Where:

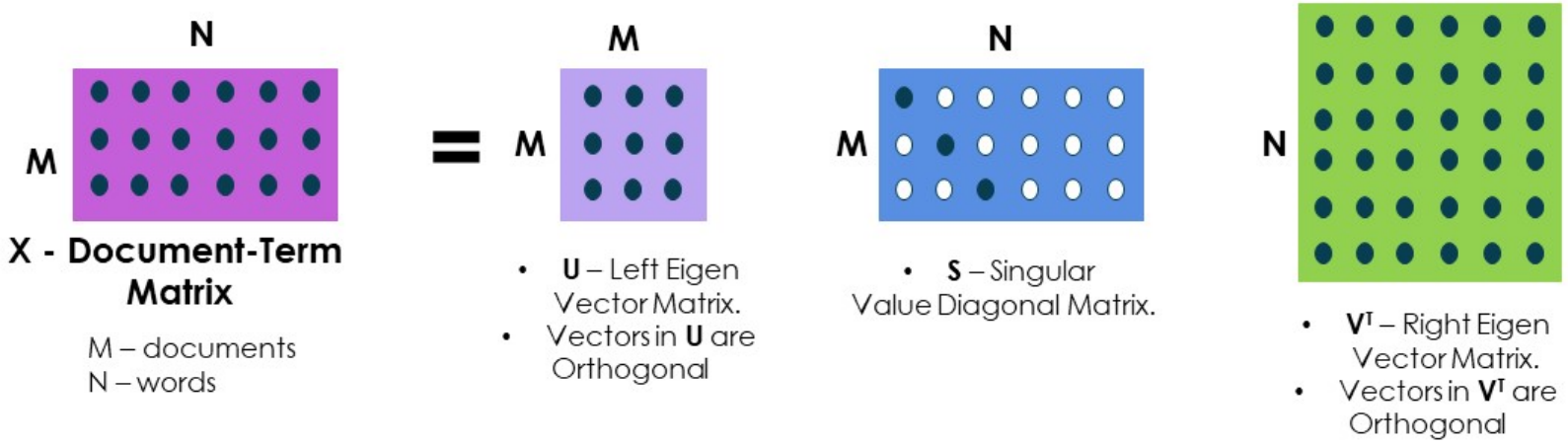
- the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal,
- the matrix $\boldsymbol{D}$ is diagonal with positive real entries.

The weights in the diagonal are called the *singular values*, which are the absolute values of the matrix' eigen values.

Informally: the singular values indicate how important each dimension is.

23

Singular Value Decomposition



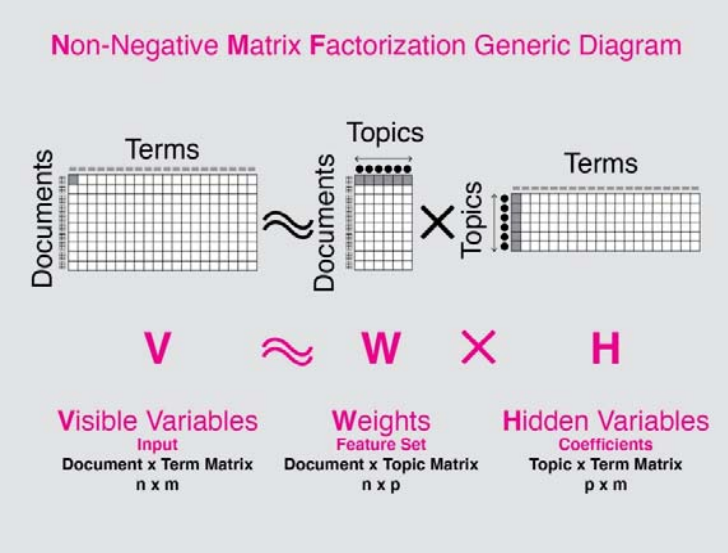
Source: https://www.datajango.com/topic-modeling-and-singular-value-decomposition-svd/

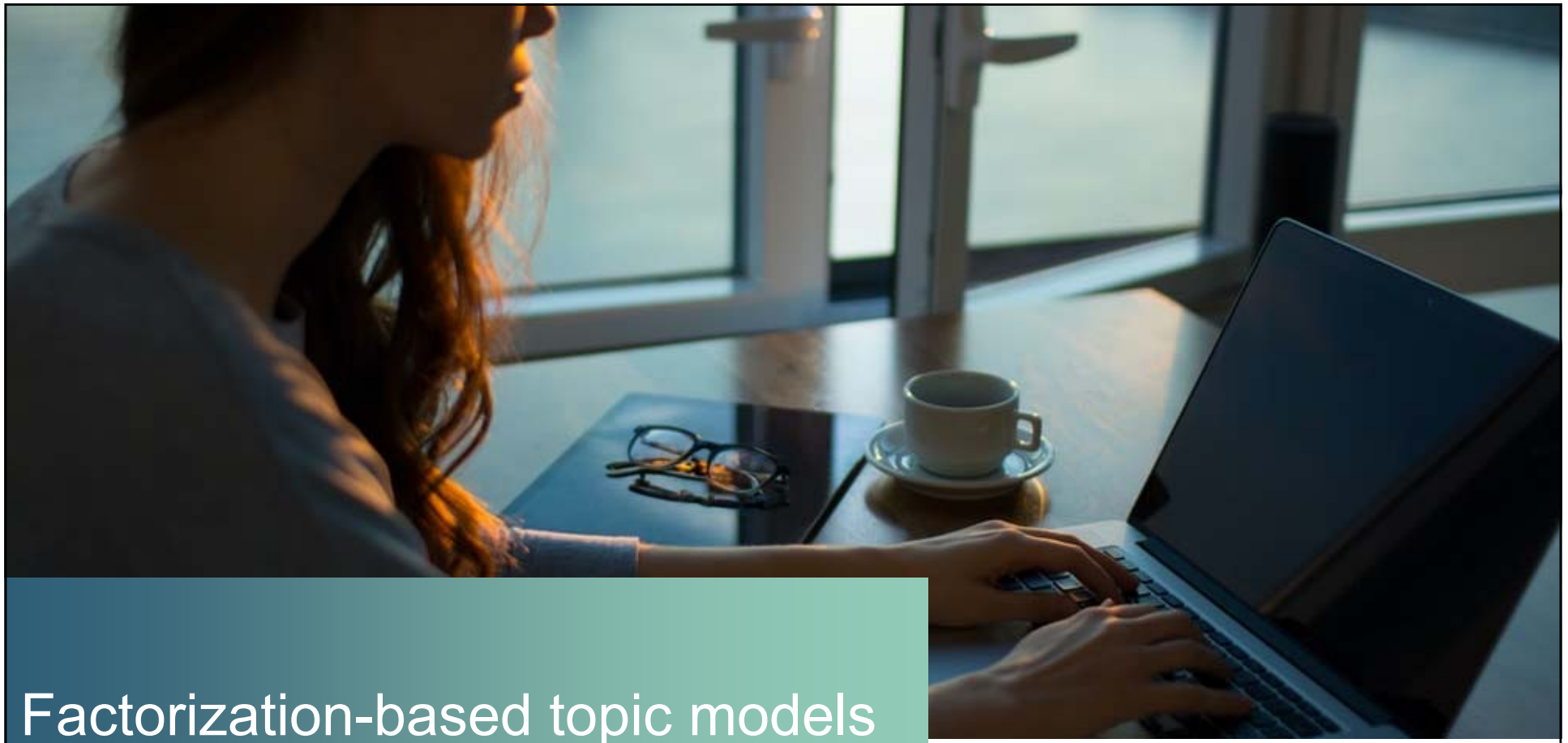
Non-Negative Matrix Factorization (NMF)

NMF also decomposes a matrix into its constituent parts.

NMF approximates a matrix $\boldsymbol{X}$ with a low-rank matrix approximation such that $\boldsymbol{X} \approx \boldsymbol{WH}$.

NMF requires its factors to be *non-negative*, instead of requiring *orthogonal* factors.





Factorization-based topic models

Latent Semantic Analysis (LSA)

Also called Latent Semantic Indexing (LSI), this is one of the earliest topic modeling algorithms based on matrix factorization.

1. Create a document-term matrix.
2. Apply global weighting.
3. Perform singular value decomposition and return the $\boldsymbol{U}$ $(documents \times number\ of\ topics)$ and $\boldsymbol{V}$ $(documents \times number\ of\ topics)$ matrices.

Note: the number of topics are selected after training.

Example of making a document-term matrix

Given the following documents:

1. *'Time flies like an arrow'*
2. *'Fruit flies like banana'*
3. *'Rose rose to put rose roes on her rows of roses'*

After stop word removal and lowercasing:

1. *'time flies like arrow'*
2. *'fruit flies like banana'*
3. *'rose rose put rose roes rows roses'*

	arrow	banana	flies	fruit	like	put	roes	rose	roses	rows	time
Doc 1	1	0	1	0	1	0	0	0	0	0	1
Doc 2	0	1	1	1	1	0	0	0	0	0	0
Doc 3	0	0	0	0	0	1	1	3	1	1	0

28

Global-term weights

- Common global-term wights are used to weight words according to their presence in other corpus documents.

- Let us define:
- LTW_{ij}: the number of times word *i* occurs in document *j*.
- m: the number of words
- n : the number of documents
- $b(LTW_{ij})$: $\begin{cases} 1, LTW_{ij} > 0 \\ 0, LTW_{ij} = 0 \end{cases}$
- p_{ij}: $\frac{tf_{ij}}{\sum_j tf_{ij}}$

Common global-term-weighting mechanisms

Name	Formula
Entropy	$1 + \frac{\sum_j p_{ij} \log_2(LTW_{ij})}{\log_2 n}$
IDF	$\log_2 \frac{n}{\sum_j b(LTW_{ij})}$
Normal	$\frac{1}{\sqrt{\sum_j LTW_{ij}^2}}$
ProbIDF	$\log_2 \frac{n - \sum_j b(LTW_{ij})}{\sum_j b(LTW_{ij})}$

29

Example of calculating IDF

IDF $\log_2 \frac{n}{\sum_j b(LTW_{ij})}$

	arrow	banana	flies	fruit	like	put	roes	rose	roses	rows	time
Doc 1	1	0	1	0	1	0	0	0	0	0	1
Doc 2	0	1	1	1	1	0	0	0	0	0	0
Doc 3	0	0	0	0	0	1	1	1	1	1	0

	arrow	banana	flies	fruit	like	put	roes	rose	roses	rows	time
Doc 1	$log2\frac{3}{1}$	0	$log2\frac{3}{2}$	0	$log2\frac{3}{2}$	0	0	0	0	0	$log2\frac{3}{1}$
Doc 2	0	$log2\frac{3}{1}$	$log2\frac{3}{2}$	$log2\frac{3}{1}$	$log2\frac{3}{2}$	0	0	0	0	0	0
Doc 3	0	0	0	0	0	$log2\frac{3}{1}$	$log2\frac{3}{1}$	$log2\frac{3}{1}$	$log2\frac{3}{1}$	$log2\frac{3}{1}$	0

	arrow	banana	flies	fruit	like	put	roes	rose	roses	rows	time
Doc 1	1.43	0	0.72	0	0.72	0	0	0	0	0	1.43
Doc 2	0	1.43	0.72	1.43	0.72	0	0	0	0	0	0
Doc 3	0	0	0	0	0	1.43	1.43	1.43	1.43	1.43	0

30

LSA pipeline

Start with a corpus

Create a document-term matrix

	d_1	d_2	...	d_N
w_1	0	0		2
w_1	1	0		1
...				
w_M	0	1		0

Calculate global term weights

	d_1	d_2	...	d_N
w_1	0	0		1.61
w_1	0.32	0		0.95
...				
w_M	0	0.12		0

Use SVD to obtain the *U* and *V* matrices

	s_1	...	s_K
d_1	0,14		0,87
d_1	0,36		0,48
...			
d_M	0,10		0,88

	w_1	...	w_S
s_1	0,02		0,01
	0,01		0,11
...			
	0,10		0,01

3
1

Fuzzy C-Means Clustering

Fuzzy C-Means (FCM) clustering is an extension of the traditional C-Means (or K-Means) clustering algorithm.

While K-Means assigns each data point to one and only one cluster, Fuzzy C-Means allows data points to belong to multiple clusters with varying degrees of membership.

For topic modeling this seems to be intuitive.

Fuzzy Latent Semantic Analysis (FLSA)

1. Create a document-term matrix.
2. Apply global weighting.
3. Perform singular value decomposition and return the ***U*** $(documents \times number\ of\ topics)$ matrix.
4. Perform fuzzy clustering on ***U*** to obtain $P(T|D)$.
5. Use matrix multiplications to find the output matrices.

Note: the created topics do depend on the number of topics.

Karami, A., Gangopadhyay, A., Zhou, B., & Kharrazi, H. (2018). Fuzzy approach topic discovery in health and medical corpora. *International Journal of Fuzzy Systems*, *20*, 1334-1345.

33

FLSA matrix multiplications

Bayes' theorem: $P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$

We are interested in finding: $P(W|T)$ *and* $P(T|D)$

We know: $P(T|D)$

Then:

1 Calculate the probability of document j

$$P(D_j) = \frac{\sum_{i=1}^{M} P(W_i, D_j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} P(W_i, D_j)}.$$

2 Calculate the probability of document j, given topic

$$\begin{aligned} P(D_j, T_k) &= (P(T_k|D_j) \otimes P(D_j))^T, \\ P(D_j|T_k) &= \frac{P(D_j, T_k)}{\sum_{j=1}^{N} P(D_j, T_k)}, \end{aligned}$$

$\otimes$ represents element-wise multiplication.

3 Calculate the probability of word i, given document

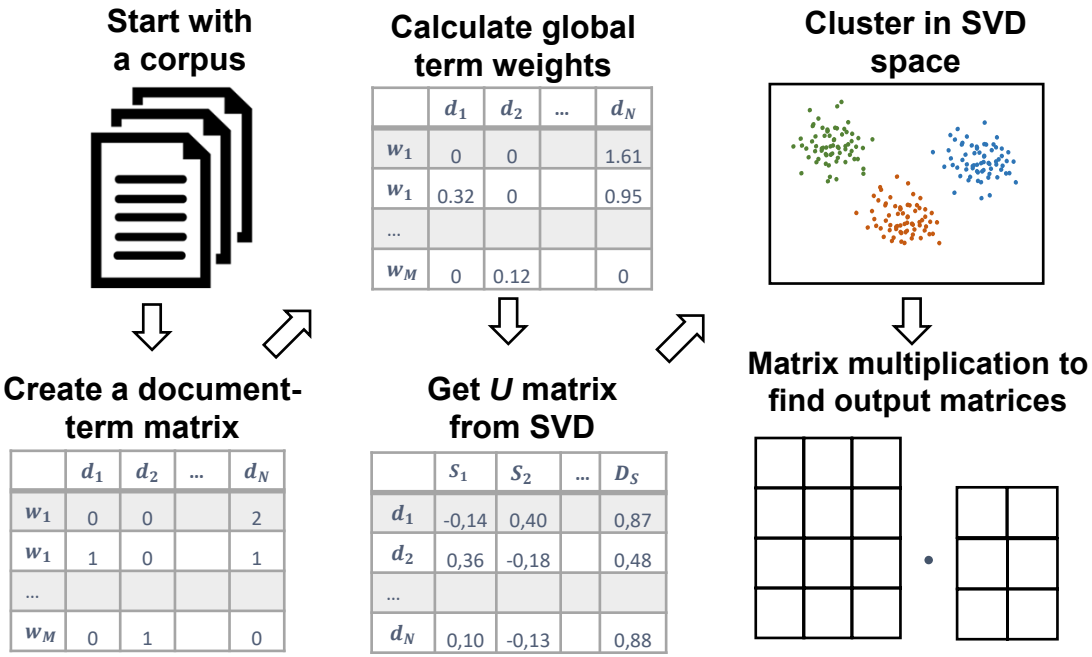
$$P(W_i|D_j) = \frac{P(W_i, D_j)}{\sum_{i=1}^{M} P(W_i, D_j)}.$$

4 Calculate the probability of word i, given topic k

$$P(W_i|T_k) = \sum_{j=1}^{N} P(W_i|D_j)^T P(D_j|T_k).$$

JADS Jheronimus Academy of Data Science

FLSA pipeline



	d_1	d_2	...	d_N
w_1	0	0		1.61
w_1	0.32	0		0.95
...				
w_M	0	0.12		0

	d_1	d_2	...	d_N
w_1	0	0		2
w_1	1	0		1
...				
w_M	0	1		0

	S_1	S_2	...	D_S
d_1	-0,14	0,40		0,87
d_2	0,36	-0,18		0,48
...				
d_N	0,10	-0,13		0,88

FLSA-W(ords)

With FLSA, the ***U*** matrix from SVD is fed to the clustering algorithm to obtain $P(T|D)$. Thus, documents are being clustered with FLSA.

Note: we are now clustering over documents, containing various topics, each containing various words. This might make clustering harder.

Solution: With FLSA-W, we cluster within the ***V*** matrix. Hence, we cluster words instead of documents.

FLSA-W

1. Create a document-term matrix.
2. Apply global weighting.
3. Perform singular value decomposition and return the $\boldsymbol{V}$ $(number\ of\ topics \times vocabulary)$ matrix.
4. Perform fuzzy clustering on $\boldsymbol{V}$ to obtain $P(T|W)$.
5. Use matrix multiplications to find the output matrices.

FLSA-W matrix multiplications

Bayes' theorem: $$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

We are interested in finding: $P(W|T)$ *and* $P(T|D)$

We know: $P(T|W)$

38

Then:

1 Calculate probability vectors:

$$P(D_j) = \frac{\sum_{i=1}^{M} P(W_i, D_j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} P(W_i, D_j)}, \quad (6)$$

$$P(W_i) = \frac{\sum_{j=1}^{N} P(W_i, D_j)}{\sum_{i=1}^{m}\sum_{j=1}^{N} P(W_i, D_j)}, \quad (7)$$

$$P(T_k) = P(T_k|W_i)P(W_i). \quad (8)$$

2 Calculate the probability of word i, given topic k

$$P(W_i, T_k) = (P(T_k|W_i) \otimes P(W_i))^T, \quad (9)$$

$$P(W_i|T_k) = \frac{P(W_i, T_k)}{\sum_{i=1}^{M} P(W_i, T_k)}. \quad (10)$$

3 Calculate the probability of topic k, given document j

$$P(W_i|D_j) = \frac{P(W_i, D_j)}{\sum_{i=1}^{M} P(W_i, D_j)}, \quad (11)$$

$$P(D_j|W_i) = \frac{(P(W_i|D_j) \otimes P(D_j))^T}{P(W_i)}, \quad (12)$$

$$P(D_j|T_k) = \sum_{i=1}^{M} P(D_j|W_i)P(W_i|T_k), \quad (13)$$

$$P(T_k|D_j) = \frac{(P(D_j|T_k) \otimes P(T_k))^T}{P(D_j)}. \quad (14)$$

FLSA-W pipeline

Start with a corpus

Create a document-term matrix

	d_1	d_2	...	d_N
w_1	0	0		2
w_1	1	0		1
...				
w_M	0	1		0

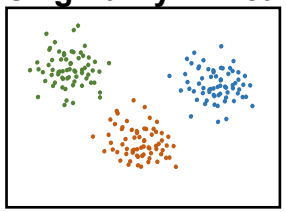
Calculate global term weights

	d_1	d_2	...	d_N
w_1	0	0		1.61
w_1	0.32	0		0.95
...				
w_M	0	0.12		0

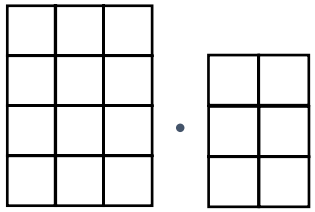
Get *V* matrix from SVD

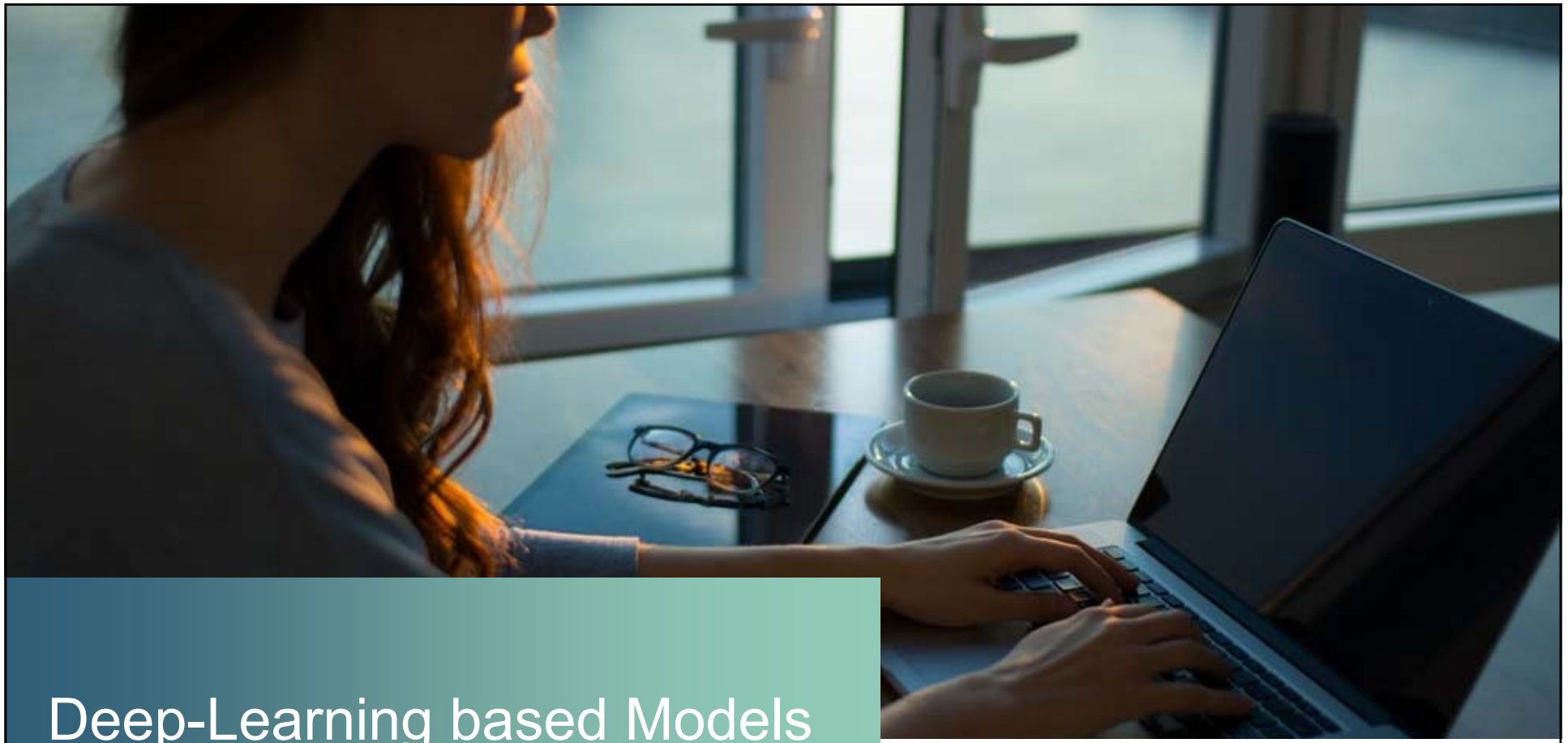
	S_1	S_2	...	D_S
w_1	-0,14	0,40		0,87
w_1	0,36	-0,18		0,48
...				
w_M	0,10	-0,13		0,88

Cluster in SVD space using Fuzzy C-means



Matrix multiplication to find output matrices

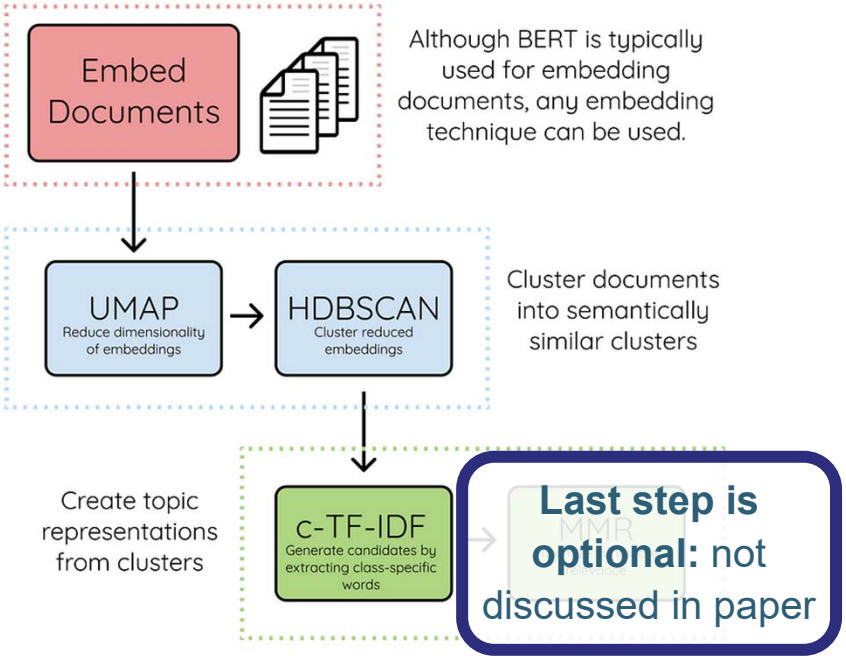




Deep-Learning based Models

BERTopic

Proposed by Maarten Grootendorst (a former JADS student)



41

BERTopic

Embedding Generation: Documents are first transformed into embeddings using BERT or another specified transformer model.

Dimensionality Reduction: Since embeddings from models like BERT can be high-dimensional, BERTopic applies dimensionality reduction, often using UMAP (Uniform Manifold Approximation and Projection).

Clustering: The reduced-dimensional embeddings are then clustered, typically using the HDBSCAN algorithm, to identify topics. Each cluster of document embeddings represents a topic.

Topic Representation: For each identified topic, BERTopic determines the most representative words or terms for that topic, often based on the c-TF-IDF (Class-based Term Frequency-Inverse Document Frequency) measure.

Maximize Marginal Relevance: *The idea behind MMR is to balance the trade-off between relevance (how closely a topic represents a document or set of documents) and diversity (how different the selected topics are from each other).*

42

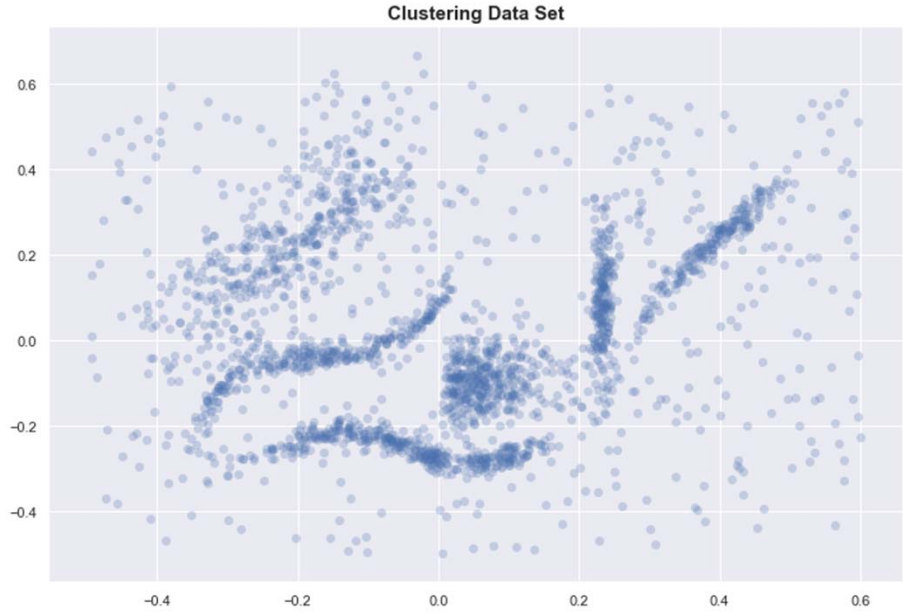
JADS Jheronimus Academy of Data Science

By default, the number of topics need not be specified to BERTopic

Hierarchical Density-Based Spatial Clustering of Applications with Noise

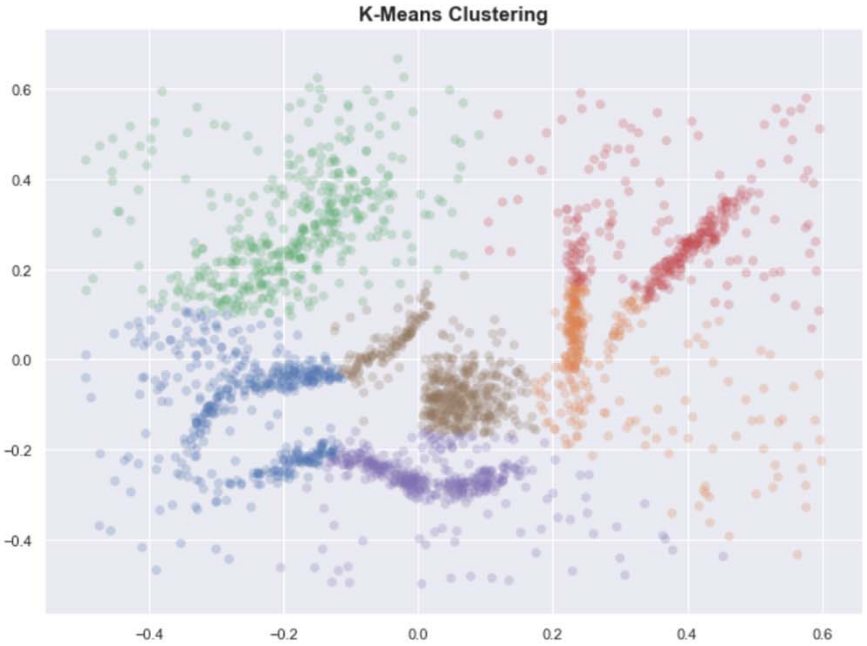
Density based clustering

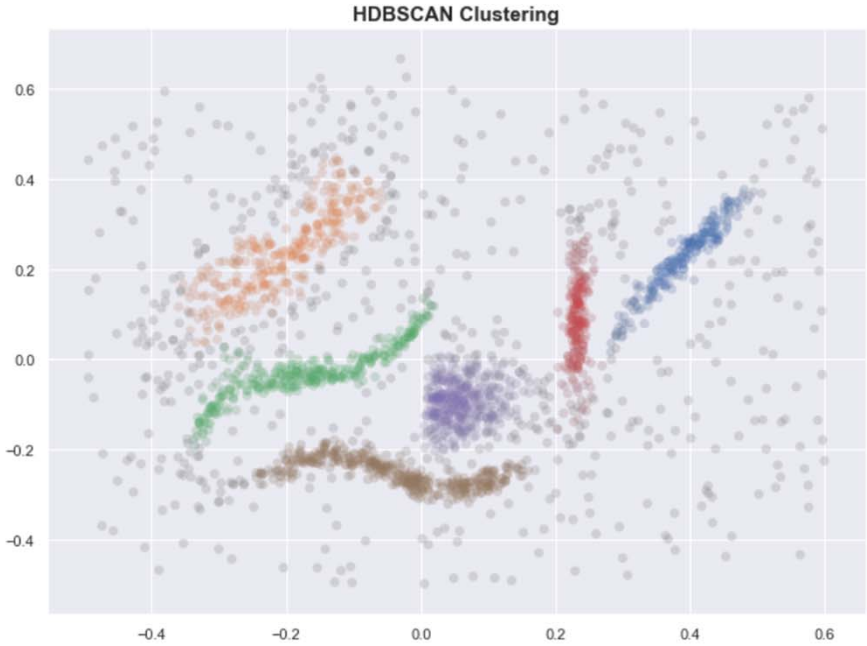
Source: https://pberba.github.io/stats/2020/01/17/hdbscan/



44

HDensityBasedSCAN



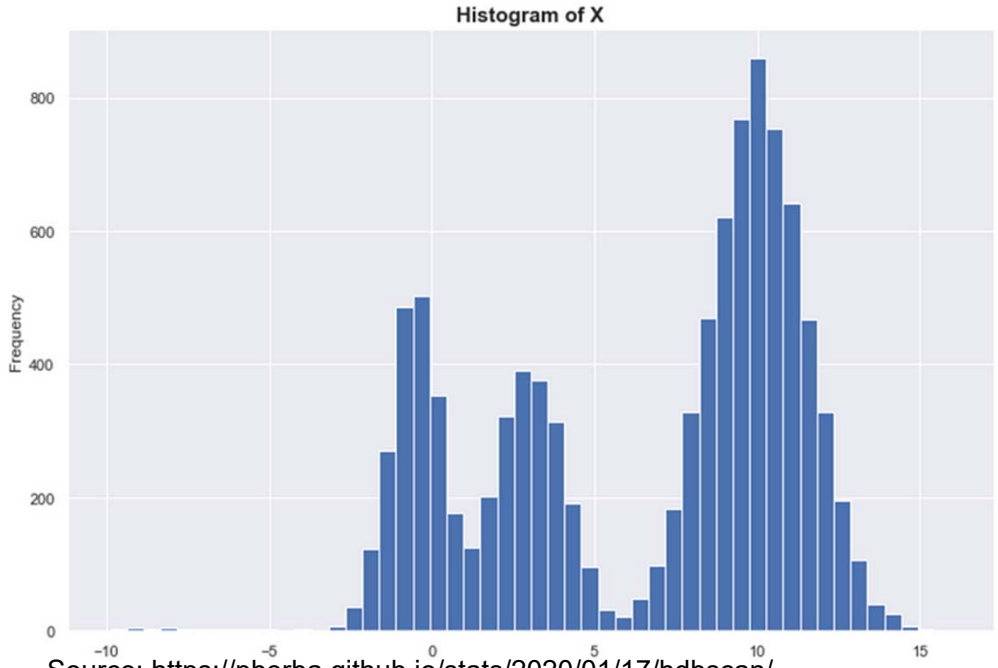


Source: https://pberba.github.io/stats/2020/01/17/hdbscan/

45

HierarchicalDBSCAN

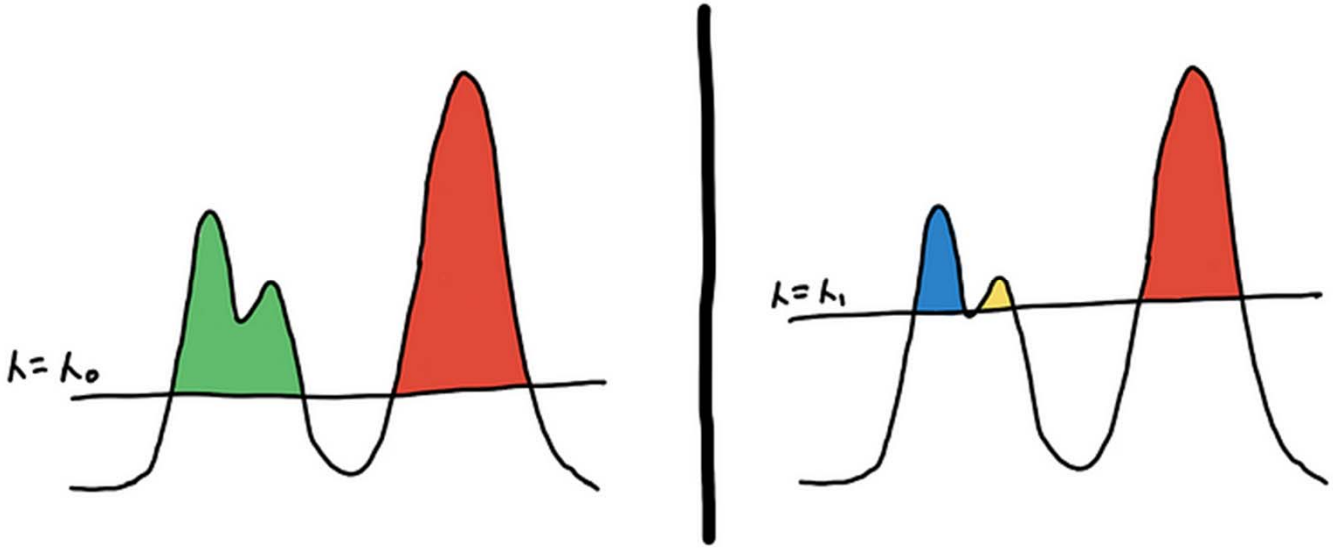
How many clusters are in the histogram below?



Source: https://pberba.github.io/stats/2020/01/17/hdbscan/

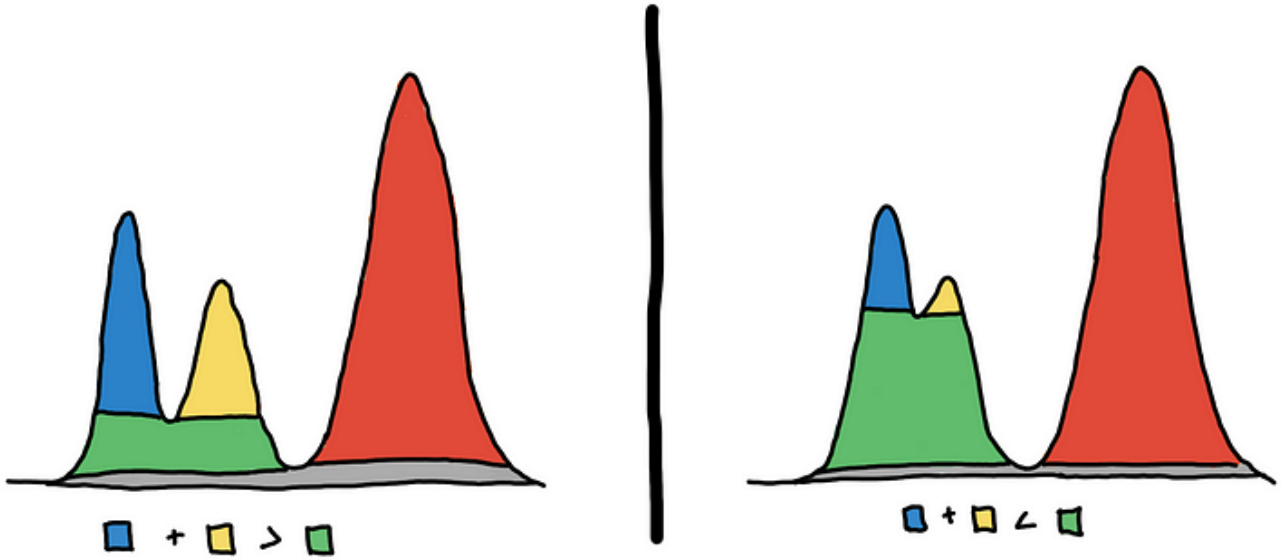
46

Based on the threshold, there are 2 or 3 clusters



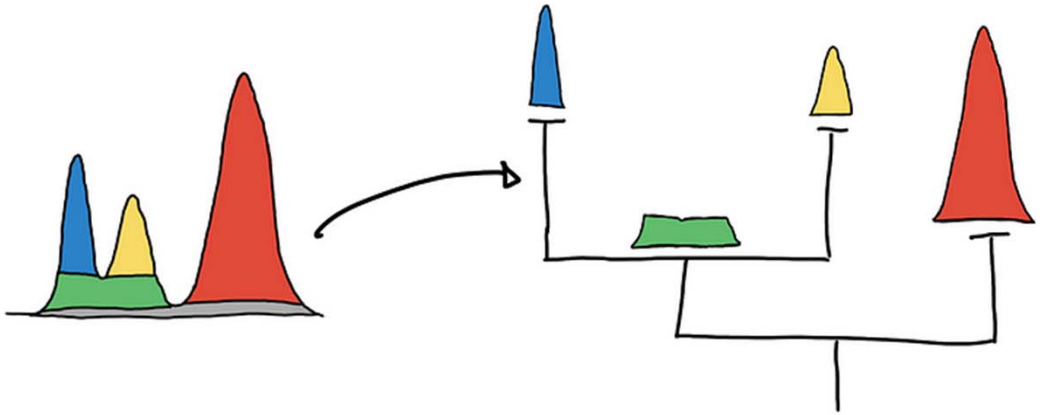
Source: https://pberba.github.io/stats/2020/01/17/hdbscan/

HDBSCAN decides the number of clusters by assessing which parts ‘persists’ more



Source: https://pberba.github.io/stats/2020/01/17/hdbscan/

By creating hierarchical trees, the model should be able to detect clusters of varying densities



Source: https://pberba.github.io/stats/2020/01/17/hdbscan/

Is it better not having to specify the number of topics?

[50]

In the end, a density threshold still needs to be defined..

And selecting the right number of topics is not always straightforward.

51

Philosophical question: What is a topic?

[52]

www.jads.nl

JADS Jheronimus Academy of Data Science

A probability distribution over all vocabulary words

A limited number of words to describing a set of documents

JADS Jheronimus Academy of Data Science


AND THE
WINNER IS...

It depends…

Probability distribution

- Not really intuitive to think of a topic as a probability distribution over all vocabulary words.

+ Probability distributions hold useful information for some tasks.

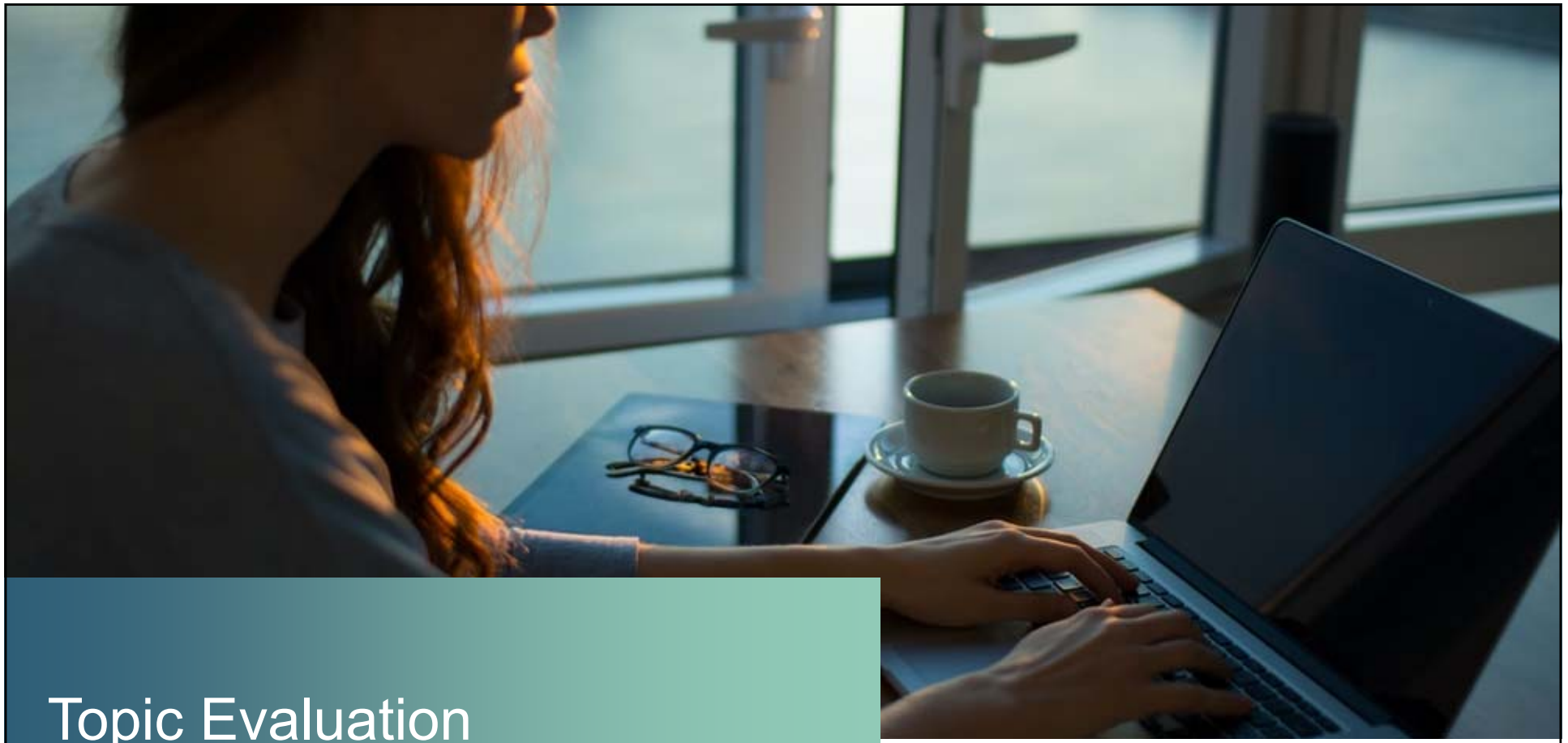
Tasks: document classification, content recommendation

A limited number of words

+ It is more intuitive to see topics as a group of words

- Without having a word-topic distribution, not all tasks can be conducted.

Tasks: document classification, content recommendation



Topic Evaluation

Topic evaluation: trade-off between human and automated evaluation

Human evaluation:

+ Direct indicator for topic quality
- Time-intensive
- Subjective

Automated evaluation

- Proxy for topic quality
+ Fast
+ Objective

Topic evaluation – what are topics?

From the word-topic matrix, for each topic the top-*n* words with the highest probability are used to represent that topic.

The produced topics are a collection (topics) of collections (words).

```
[(0,
  '0.0219*"real" + 0.0143*"life" + 0.0126*"fantasy" + 0.0071*"reality" + 0.0067*"galileo" + 0.0055*"poor" +
0.0051*"boy" + 0.005*"mama" + 0.005*"mia" + 0.005*"matters"'),

 (1,
  '0.0167*"submarine" + 0.011*"yellow" + 0.0065*"sailed" + 0.0064*"sun" + 0.0062*"sea" + 0.0055*"waves" +
0.0047*"friends" + 0.0046*"sky" + 0.0046*"live" + 0.0045*"sailed" '),

 (2,
  '0.0064*"piano" + 0.0045*"man" + 0.0035*"crowd" + 0.0035*"saturday" + 0.0031*"tonic" + 0.0031*"gin" +
0.0023*"bar"  + 0.0018*"drinking" + 0.0017*"beer" + 0.0016*"melody"+ 0.0015*"sitting" ')]
```

Would this be good topics?

```
[(0,
  '0.0219*"water" + 0.0143*"package" + 0.0126*"paracetamol" + 0.0071*"vienna" +
0.0067*"biden" + 0.0055*"ball" + 0.0051*"walk" + 0.005*"car" + 0.005*"coat" +
0.005*"spider"')]
```

No, words are unrelated

```
[(1,
  '0.0219*"senate" + 0.0143*"president" + 0.0126*"election" + 0.0071*"government"
+ 0.0067*"legislation" + 0.0055*"majority" + 0.0051*"trump" + 0.005*"bill" +
0.005*"rules" + 0.005*"state"')]
```

Yes; U.S. politics

```
[(2,
  '0.0219*"world" + 0.0143*"final" + 0.0126*"record" + 0.0071*"win" +
0.0067*"ball" + 0.0055*"referee" + 0.0051*"league" + 0.005*"team" + 0.005*"lose"
+ 0.005*"coach"')]
```

Yes; sports

Would this be a good topic?

```
[(0,
  '0.0219*"world" + 0.0143*"final" + 0.0126*"record" + 0.0071*"win" +
0.0067*"ball" + 0.0055*"referee" + 0.0051*"league" + 0.005*"team" +
0.005*"lose" + 0.005*"coach"')]

(1,
  '0.0219*"world" + 0.0143*"final" + 0.0126*"record" + 0.0071*"win" +
0.0067*"ball" + 0.0055*"referee" + 0.0051*"league" + 0.005*"team" +
0.005*"lose" + 0.005*"coach"'),

(2,
  '0.0219*"world" + 0.0143*"final" + 0.0126*"record" + 0.0071*"win" +
0.0067*"ball" + 0.0055*"referee" + 0.0051*"league" + 0.005*"team" +
0.005*"lose" + 0.005*"coach"'),

(3,
  '0.0219*"world" + 0.0143*"final" + 0.0126*"record" + 0.0071*"win" +
0.0067*"ball" + 0.0055*"referee" + 0.0051*"league" + 0.005*"team" +
0.005*"lose" + 0.005*"coach"'),

(4,
  '0.0219*"world" + 0.0143*"final" + 0.0126*"record" + 0.0071*"win" +
0.0067*"ball" + 0.0055*"referee" + 0.0051*"league" + 0.005*"team" +
0.005*"lose" + 0.005*"coach"')]
```

No, all the topics are the same

60

Inter- vs. intra-topic quality

Inter-topic -quality

```
[(0,
  '0.0219*"real" + 0.0143*"life" + 0.0126*"fantasy" + 0.0071*"reality" + 0.0067*"galileo" +
0.0055*"poor" + 0.0051*"boy" + 0.005*"mama" + 0.005*"mia" + 0.005*"matters"'),

 (1,
  '0.0167*"submarine" + 0.011*"yellow" + 0.0065*"sailed" + 0.0064*"sun" + 0.0062*"sea" +
0.0055*"waves" + 0.0047*"friends" + 0.0046*"sky" + 0.0046*"live" + 0.0045*"sailed" '),

 (2,
  '0.0064*"piano" + 0.0045*"man" + 0.0035*"crowd" + 0.0035*"saturday" + 0.0031*"tonic" +
0.0031*"gin" + 0.0023*"bar"  + 0.0018*"drinking" + 0.0017*"beer" + 0.0016*"melody"+
0.0015*"sitting" ')]
```

Intra-topic-quality

Intra-topic-quality

Intra-topic-quality

Topic evaluation: inter-topic-quality

A simple and intuitive measure for inter-topic-quality is the diversity score.

$$diversity\ score = \frac{the\ number\ of\ unique\ words}{the\ total\ number\ of\ words},$$

$$0 < diversity\ score\ \leq 1$$

An example of the diversity score

Given the topics:

```
[(0,'0.0219*"real" + 0.0143*"live " + 0.0126*"fantasy" + 0.0071*"reality" + 0.0067*"sailing" + 0.0055*"poor" +
0.0051*"friends " + 0.005*"mama" + 0.005*"sky " + 0.005*"sun"'),

 (1,'0.0167*"submarine" + 0.011*"yellow" + 0.0065*"sailed" + 0.0064*"sun" + 0.0062*"sea" + 0.0055*"waves" +
0.0047*"friends" + 0.0046*"sky" + 0.0046*"live" + 0.0045*"sailing" ')]
```

Unique words: 15 (real, live, fantasy, reality, sailing, poor, friends, mama, sky, sun, submarine, yellow, sailed, sea, waves,)

Total words: 20 (real, life, fantasy, reality, galileo, poor, boy, mama, mia, matters, submarine, yellow, sailed, sun, sea, waves, friends, sky, live, sailing)

Diversity score = 0.75 = 15/20

63

How to measure the semantic relatedness of words within a topic?

Hint: use the corpus as a benchmark

64

Intra-topic quality

Initial (pre word embeddings) approach: make manual calculations

Considerations:
- How to segment words?
- Which confirmation measure to use?
- How to compute probabilities?
- How to aggregate all values?

Röder (2015):
- Defines a four-dimensional configuration space (one dimension per consideration).
- Uses a dataset where each topic has a human interpretation score.
- Calculates scores for each combination in his configuration space.
- Calculates correlations between coherence scores and human scores.

65

Intra-topic quality

- Röder finds a new setting that correlates the highest with human interpretation.
- He calls this setting C_v.
- C_v is based on Normalized Pointwise Mutual Information (NPMI).

The NPMI is calculated by finding the log-probabilities of words *p* and *q* cooccurring in the corpus within a given sliding window (number of words around a target word to consider) for all words *p and q,* $(p \neq q)$.

NPMI:

$$\mathrm{NPMI(w)} = \frac{1}{n(n-1)} \sum_{q=2}^{n} \sum_{p=1}^{q-1} \frac{\log \frac{P(w_p, w_q)}{P(w_p)P(w_q)}}{-\log P(w_p, w_q)}$$

Where: *n* = the *n* most likely words in a topic

66

Intra-topic quality

C_v coherence:

- uses a sliding window of size 110,
- to calculate the probabilities of each word combination,
- based on the NPMI,
- and reports the arithmetic mean for all topics.

(How) Can we use word embeddings to calculate topic coherence?

(Ding et. al., 2018)

Pair-wise word embedding topic coherence (WEC_{pw})

We define the following properties:

D the dimensionality of the embedding space,
E the row-normalized word embedding matrix for a list of N words,
N the number of (most likely) words per topic.

Then, $E \in \mathbb{R}^{N \times D}$ and $\|E_i, ;\| = 1$.

Let ⟨., .⟩ denote the inner product. Then, WEC_{pw} is defined as follows:

$$\begin{aligned} \mathrm{WEC}_{\mathrm{pw}}(\mathrm{E}) &= \frac{1}{N(N-1)} \sum_{j=2}^{N} \sum_{i=1}^{j-1} \langle E_{i,:}, E_{j,:} \rangle \\ &= \frac{\sum\{E^T E\} - N}{2N(N-1)} \end{aligned}$$

In words:

Report the average score of the topic words' word vectors' inner-products, for each combination of topic words.

69

Centroid-based word embedding topic coherence (WEC_c)

We define the following properties:

D	the dimensionality of the embedding space,
E	the row-normalized word embedding matrix for a list of N words,
N	the number of (most likely) words per topic.
t	centroid of *E*, normalized to have $\|\|t\|\| = 1$. Where $t \in \mathbb{R}^{1\times D}$

Then, WEC_C is defined as follows:

$$\mathrm{WEC_c(E)} = \frac{1}{N}\sum\{Et^T\}$$

In words:

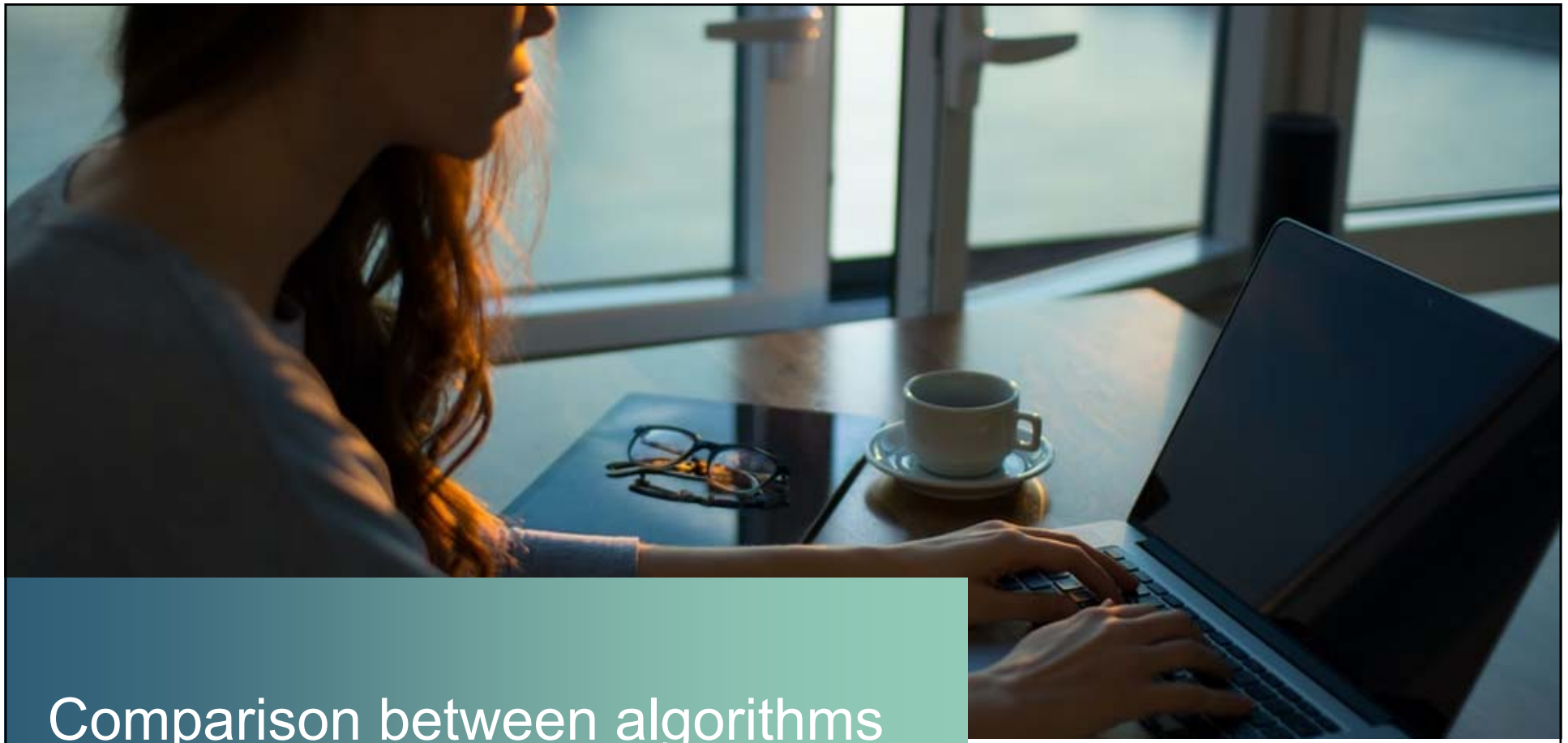
Report the average similarity with the centroid, based on each embedding's dimension {1,2,…D}.

Topic evaluation: interpretability score

The interpretability score (Dieng et. al., 2020) combines the coherence- and diversity score in one score.

$$Interpretability\ score = Coherence\ score\ \times Diversity\ score,$$

$$0 < Interpretability\ score\ \leq 1$$



Comparison between algorithms

Interpretability vs. predictive value

Context: we want to do text classification of clinical nurse notes. To create a numerical representation (that we feed to the classifier) for each document we use *topic embeddings*, indicating the extent to which a topic represented in a text.

Goal: compare the predictive value of an embedding with the topic interpretability score on a task in the hospital (hence, private dataset).

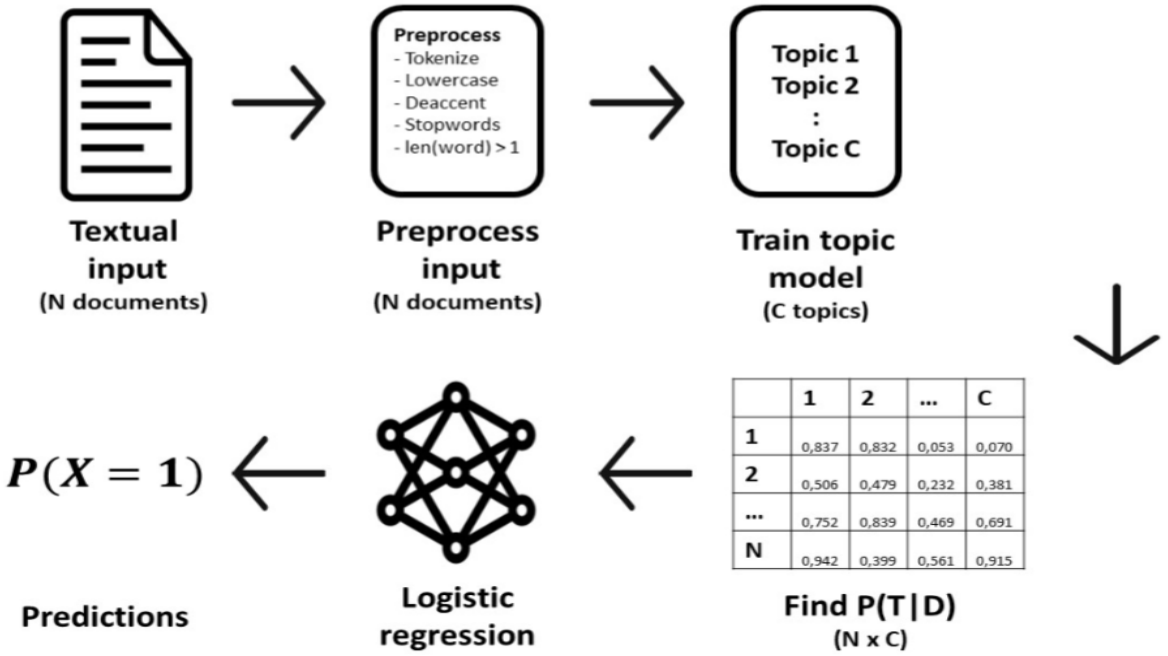
Approach: Train each algorithm with

A number of topics 5, 10, … , 100

Rijcken, E., Kaymak, U., Scheepers, F., Mosteiro, P., Zervanou, K., & Spruit, M. (2022). Topic Modeling for Interpretable Text Classification From EHRs. Frontiers in Big Data, 5.

JADS Jheronimus Academy of Data Science

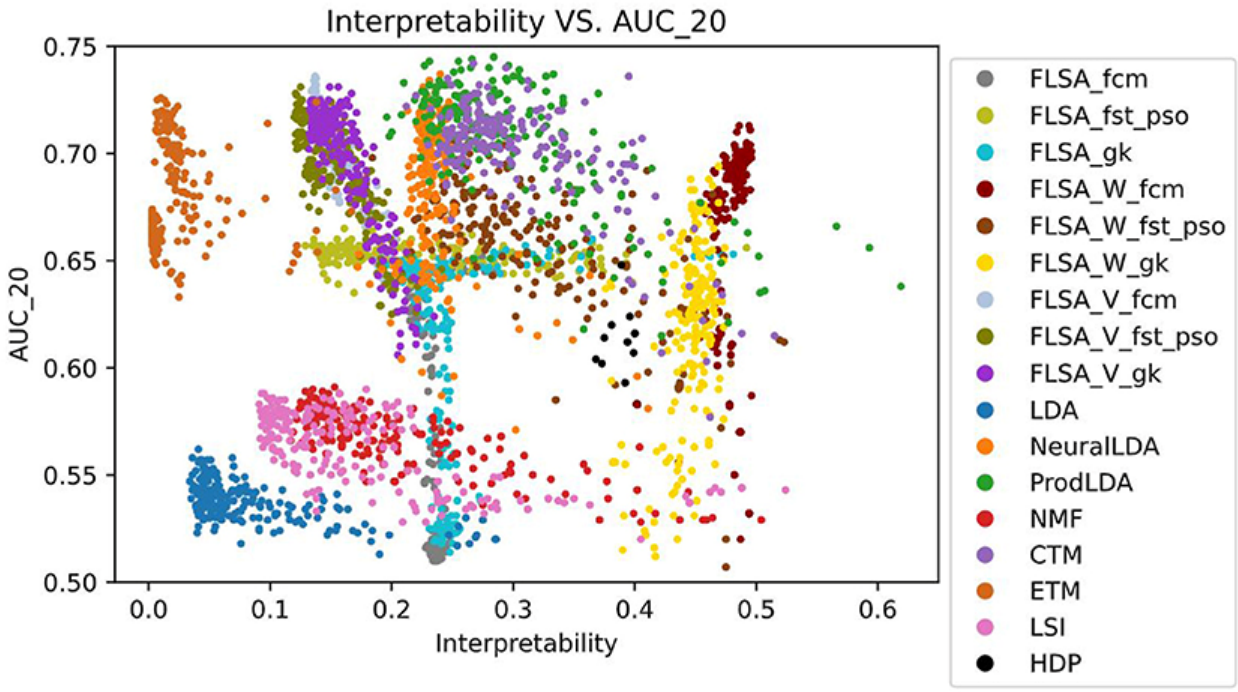
Pipeline



	1	2	...	C
1	0,837	0,832	0,053	0,070
2	0,506	0,479	0,232	0,381
...	0,752	0,839	0,469	0,691
N	0,942	0,399	0,561	0,915

74

Rijcken, E., Kaymak, U., Scheepers, F., Mosteiro, P., Zervanou, K., & Spruit, M. (2022). Topic Modeling for Interpretable Text Classification From EHRs. Frontiers in Big Data, 5.


Interpretability VS. AUC_20
AUC_20
Interpretability
0.75
0.70
0.65
0.60
0.55
0.50
0.0
0.1
0.2
0.3
0.4
0.5
0.6
FLSA_fcm
FLSA_fst_pso
FLSA_gk
FLSA_W_fcm
FLSA_W_fst_pso
FLSA_W_gk
FLSA_V_fcm
FLSA_V_fst_pso
FLSA_V_gk
LDA
NeuralLDA
ProdLDA
NMF
CTM
ETM
LSI
HDP

75

Experimental results on four open datasets

1. For four open datasets.

Dataset	Number of Texts	Unique Words	Words per Text
BBC-News	2225	2949	120.1
DBLP	54595	1513	5.4
M10	8355	1696	5.9
20News	16309	1612	48.1

2. Train topic models with the number of topics 10, 20, … , 100.
3. For various common and state-of-the-art algorithms.
4. Calculate the coherence- (C_v), diversity- and interpretability score.
5. Report the average score.

C_v

BBC-news										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	0.388	0.406	0.411	0.422	0.422	0.424	0.425	0.428	0.431	0.428
FLSA-V	-	-	-	-	-	-	-	-	-	-
FLSA	0.498	0.472	0.445	0.428	0.404	0.394	0.394	0.397	0.398	0.402
LDA	0.341	0.36	0.363	0.366	0.365	0.369	0.365	0.372	0.368	0.369
NeuralLDA	0.517	0.461	0.453	0.456	0.461	0.464	**0.471**	**0.46**	**0.466**	**0.461**
ProdLDA	**0.652**	**0.647**	**0.645**	**0.636**	**0.575**	**0.504**	0.402	0.422	0.407	0.382
LSI	0.45	0.429	0.393	0.367	0.356	0.342	0.334	0.323	0.321	0.314
ETM	0.457	0.379	0.424	0.479	0.481	0.45	0.434	0.371	0.353	0.422
NMF	0.415	0.42	0.423	0.419	0.413	0.414	0.425	0.419	0.417	0.416
DBLP										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	0.462	0.462	**0.475**	**0.476**	**0.489**	**0.497**	**0.507**	**0.51**	**0.515**	**0.515**
FLSA-V	0.303	0.393	0.437	0.459	0.468	0.469	0.473	0.47	0.46	0.449
FLSA	0.24	0.259	0.267	0.272	0.273	0.277	0.277	0.281	0.281	0.28
LDA	0.317	0.318	0.305	0.304	0.311	0.316	0.32	0.322	0.319	0.314
NeuralLDA	0.282	0.284	0.308	0.308	0.314	0.324	0.326	0.32	0.326	0.325
ProdLDA	**0.467**	**0.475**	0.457	0.428	0.422	0.417	0.415	0.408	0.401	0.396
LSI	0.269	0.258	0.266	0.265	0.258	0.253	0.247	0.245	0.245	0.244
ETM	0.169	0.136	0.108	0.108	0.116	0.126	0.13	0.136	0.138	0.142
NMF	0.332	0.357	0.361	0.357	0.35	0.347	0.342	0.337	0.331	0.325
M10										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	**0.585**	0.589	**0.607**	**0.612**	**0.612**	**0.609**	**0.61**	**0.616**	**0.619**	**0.622**
FLSA-V	0.572	**0.601**	0.591	0.59	0.58	0.577	0.58	0.573	0.574	0.575
FLSA	0.351	0.357	0.415	0.425	0.451	0.461	0.467	0.469	0.467	0.476
LDA	0.272	0.317	0.331	0.346	0.359	0.364	0.373	0.372	0.367	0.358
NeuralLDA	0.419	0.412	0.425	0.435	0.441	0.45	0.456	0.455	0.46	0.465
ProdLDA	0.423	0.393	0.406	0.412	0.414	0.434	0.44	0.445	0.452	0.46
LSI	0.331	0.332	0.31	0.309	0.304	0.312	0.319	0.33	0.334	0.34
ETM	0.24	0.162	0.144	0.138	0.134	0.139	0.138	0.136	0.134	0.146
NMF	0.328	0.34	0.336	0.342	0.341	0.342	0.345	0.346	0.348	0.35
20NewsGroup										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	0.391	0.381	0.372	0.37	0.357	0.35	0.348	0.34	0.332	0.327
FLSA-V	0.213	0.202	0.193	0.201	0.202	0.201	0.201	0.201	0.203	0.203
FLSA	0.537	0.534	0.505	0.481	0.467	0.454	0.442	0.435	0.429	0.422
LDA	0.512	0.536	0.542	0.518	0.513	0.499	0.492	0.481	0.473	0.473
NeuralLDA	0.434	0.46	0.479	0.477	0.491	0.475	0.493	0.462	0.466	0.478
ProdLDA	0.563	**0.619**	**0.603**	**0.598**	**0.572**	**0.574**	**0.562**	0.536	0.545	0.545
LSI	0.522	0.465	0.428	0.408	0.386	0.367	0.362	0.357	0.348	0.347
ETM	0.508	0.498	0.509	0.498	0.466	0.461	0.462	0.432	0.442	0.404
NMF	**0.575**	0.565	0.588	0.578	**0.572**	0.569	0.561	**0.555**	**0.551**	**0.549**

Diversity

BBC-news										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	**1.0**	**0.975**	**0.981**	**0.98**	**0.981**	**0.972**	**0.966**	**0.949**	**0.937**	**0.914**
FLSA-V	-	-	-	-	-	-	-	-	-	-
FLSA	0.703	0.602	0.622	0.659	0.68	0.695	0.711	0.713	0.706	0.702
LDA	0.414	0.324	0.301	0.277	0.247	0.245	0.237	0.234	0.226	0.225
NeuralLDA	0.898	0.897	0.877	0.822	0.793	0.759	0.734	0.696	0.679	0.654
ProdLDA	0.91	0.778	0.711	0.657	0.606	0.626	0.638	0.601	0.607	0.588
LSI	0.592	0.426	0.354	0.324	0.294	0.267	0.251	0.232	0.222	0.216
ETM	0.529	0.244	0.262	0.299	0.254	0.193	0.173	0.092	0.072	0.114
NMF	0.673	0.57	0.498	0.456	0.44	0.408	0.389	0.376	0.361	0.347
DBLP										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	**1.0**	**0.974**	**0.941**	**0.901**	**0.877**	0.837	0.797	0.753	0.703	0.656
FLSA-V	0.756	0.665	0.59	0.534	0.483	0.438	0.399	0.36	0.323	0.294
FLSA	0.44	0.307	0.237	0.195	0.176	0.161	0.15	0.14	0.134	0.13
LDA	0.787	0.863	0.882	0.883	0.875	**0.867**	**0.859**	**0.854**	**0.861**	**0.857**
NeuralLDA	0.994	0.932	0.837	0.751	0.68	0.631	0.587	0.542	0.511	0.49
ProdLDA	0.954	0.848	0.629	0.491	0.418	0.369	0.341	0.308	0.292	0.287
LSI	0.473	0.296	0.248	0.222	0.202	0.18	0.165	0.157	0.153	0.148
ETM	0.113	0.077	0.08	0.085	0.092	0.096	0.094	0.091	0.088	0.086
NMF	0.713	0.645	0.593	0.567	0.541	0.523	0.5	0.488	0.473	0.458
M10										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	**0.96**	**0.915**	**0.868**	**0.834**	**0.791**	**0.739**	0.705	0.675	0.644	0.611
FLSA-V	0.782	0.61	0.448	0.34	0.274	0.229	0.197	0.172	0.153	0.138
FLSA	0.558	0.51	0.533	0.528	0.544	0.546	0.543	0.531	0.515	0.505
LDA	0.625	0.633	0.668	0.684	0.694	0.7	**0.707**	**0.699**	**0.7**	**0.686**
NeuralLDA	0.901	0.749	0.665	0.576	0.532	0.472	0.446	0.417	0.392	0.371
ProdLDA	0.959	0.702	0.645	0.6	0.568	0.568	0.556	0.549	0.534	0.53
LSI	0.56	0.383	0.306	0.262	0.248	0.228	0.209	0.196	0.193	0.186
ETM	0.393	0.098	0.062	0.048	0.039	0.035	0.035	0.032	0.027	0.029
NMF	0.72	0.622	0.565	0.532	0.509	0.492	0.481	0.47	0.453	0.451
20NewsGroup										
	10	20	30	40	50	60	70	80	90	100
FLSA-W	**1.0**	**0.997**	**0.992**	**0.972**	**0.949**	**0.921**	**0.879**	**0.827**	**0.771**	0.723
FLSA-V	0.898	0.832	0.768	0.707	0.624	0.551	0.488	0.43	0.385	0.348
FLSA	0.694	0.641	0.567	0.534	0.506	0.475	0.455	0.445	0.426	0.414
LDA	0.689	0.706	0.721	0.718	0.722	0.732	0.735	0.735	0.743	**0.735**
NeuralLDA	0.919	0.87	0.81	0.756	0.714	0.65	0.649	0.583	0.561	0.558
ProdLDA	0.905	0.893	0.854	0.802	0.752	0.739	0.678	0.634	0.606	0.589
LSI	0.575	0.492	0.439	0.368	0.337	0.302	0.283	0.26	0.244	0.227
ETM	0.552	0.384	0.32	0.236	0.127	0.116	0.107	0.087	0.09	0.061
NMF	0.817	0.674	0.607	0.556	0.534	0.495	0.479	0.458	0.432	0.42

77

There is a plethora of topic modeling algorithms

- LDA
- ProdLDA
- NeuralLDA
- FLSA
- FLSA-W
- LSI
- NMF
- ETM
- …
- …

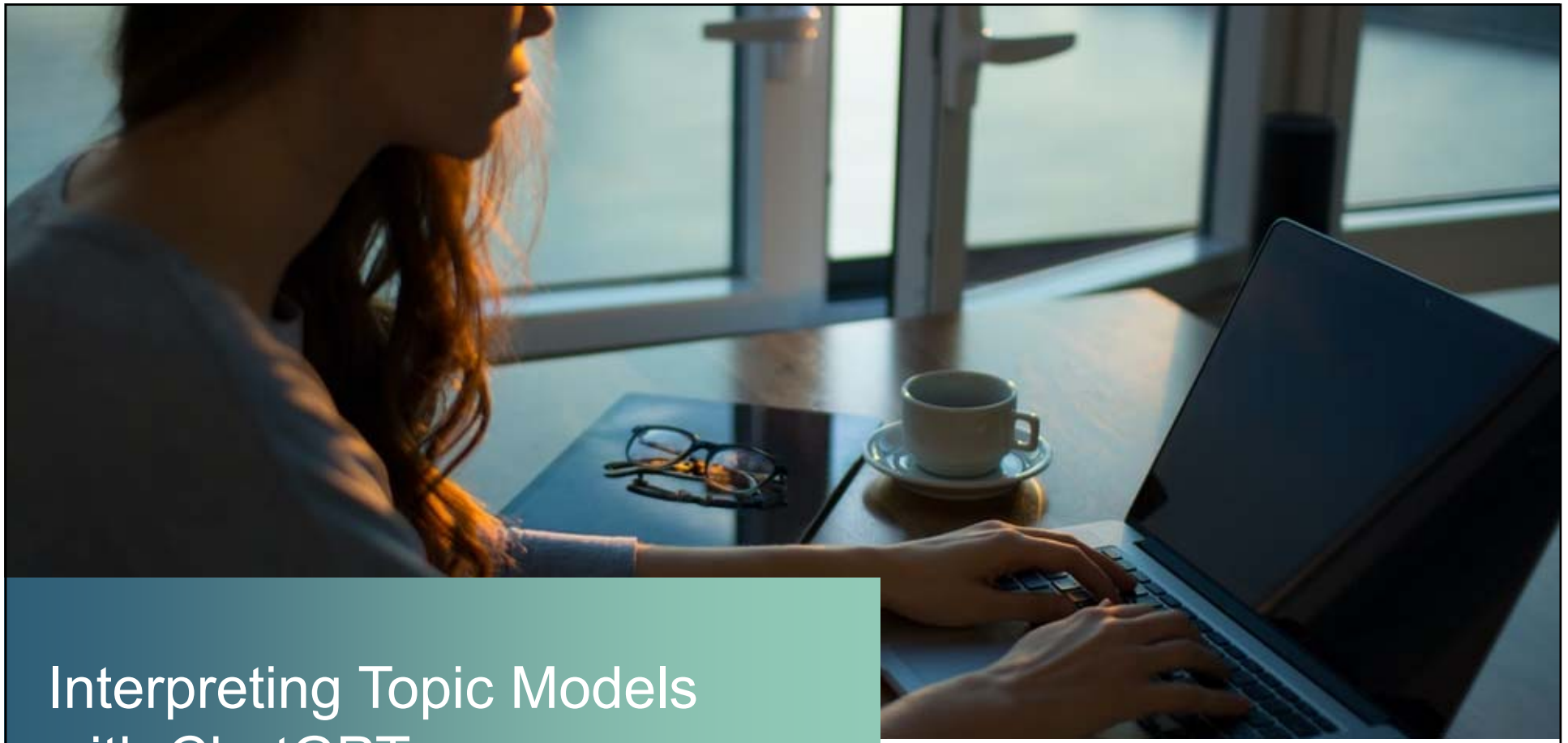
Coherence	Diversity
0.305	0.882
0.457	0.629
0.308	0.837
0.267	0.237
0.475	0.941
0.266	0.248
0.361	0.593
0.108	0.008

Is this enough information to select an algorithm for a task?

Knowing the specificness of topics, might help in algorithm selection

Topic Rank!

78



Interpreting Topic Models with ChatGPT

79

Rijcken, E., Scheepers, F., Zervanou, K., Spruit, M., Mosteiro, P., & Kaymak, U. (2023). Towards Interpreting Topic Models with ChatGPT. In *The 20th World Congress of the International Fuzzy Systems Association*.

Interpreting the topics can be difficult

'intact', 'symmetrical', 'undisturbed', 'normal', 'examination', 'normal', 'sensibility', 'person', 'facial', 'peristalsis', 'motor activity' ', 'neurological', belly', 'noises', 'extremities', 'reflexes', 'arms'

'patient', 'admission', 'suicidality', 'thoughts', 'suicidal', 'partner', 'year', 'good', 'gives', 'woman', 'admitted', 'said ', 'plans', 'depressed', 'contact', 'treatment', 'finished', 'gloomy', 'mg'

'person', 'mother', 'parents', 'patient', 'good', 'father', 'year', 'admission', 'speech', 'voices', 'complaints', 'sometimes', 'hallucinations' ', 'psychosis', 'diagnostics', 'examination', 'anamnesis', 'thoughts', 'tells', 'whereby'

We explore the potential of using ChatGPT to interpret the output generated by topic models

We follow a three-step approach

1. Create topics (LDA & FLSA-W)*
2. Interpreting Topics with a Domain Expert and ChatGPT
3. Compare the generated summaries with those produced by a domain expert

- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.
- Rijcken, E., Scheepers, F., Mosteiro, P., Zervanou, K., Spruit, M., & Kaymak, U. (2021). A comparative study of fuzzy topic models and LDA in terms of interpretability. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-8). IEEE.

JADS Jheronimus Academy of Data Science

Step 2: Prompts needed to be specified

Initial prompt:

"Which common denominator does the following set of words have <TOPICS>?"

Step 2: Prompts needed to be specified

Initial prompt:

"Which common denominator does the following set of words have <TOPICS>?"

→ **Meaningless answer:** *'the words come from a medical setting'*

Step 2: Prompts needed to be specified

Initial prompt:
"Which common denominator does the following set of words have <TOPICS>?"

→ **Meaningless answer:** *'the words come from a medical setting'*

Specified prompt:
Given the fact that the following words all originate from the electronic patient record of a psychiatric department, what common denominator do the following words have? Please be as specific as possible. <TOPIC>"

Step 2 (part 2):
Ask the domain expert to assign labels to each topic

Step 3 - Comparing Summaries Generated by ChatGPT and the Domain Expert

- We show both answers to the domain expert (hers and ChatGPT's).
- Then, we ask three questions for each topic/label:

1. **Do you agree with ChatGPT's output?** Answers: Yes, No

2. **Which description is better?** Answers: domain expert, ChatGPT, approximately equally good, NA

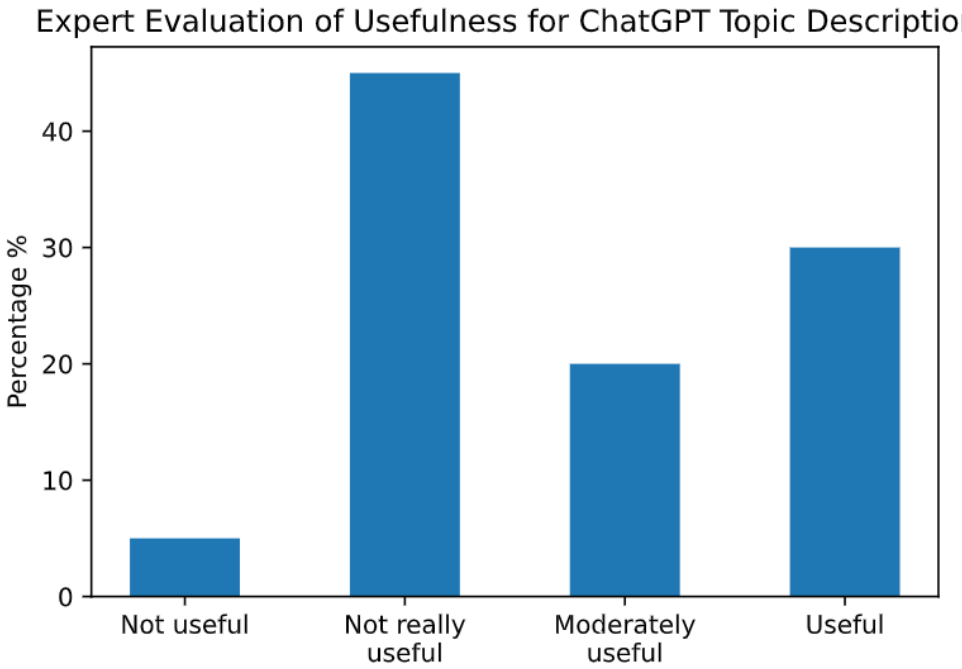
3. **How useful is ChatGPT's answer?** Answers: not useful, not really useful, moderately useful, useful

Data

Clinical notes from the Psychiatry department of the University Medical Center Utrecht.

- 4280 notes,
- 1481 words per document on average.

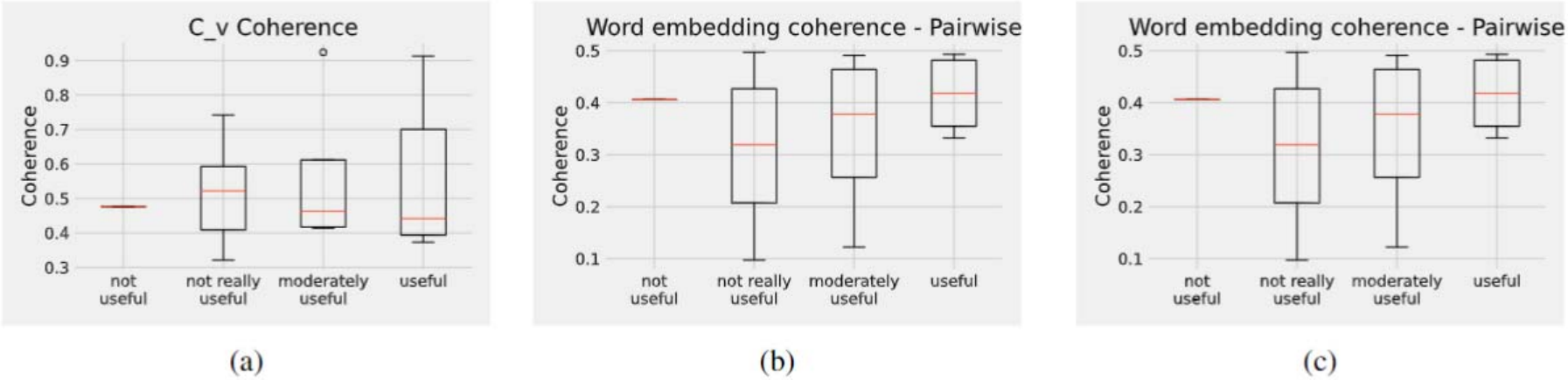
Results



Findings

1) The domain expert assigned a label to 75% of all topics.
2) In 20% of the unassigned topics, the domain expert found ChatGPT's description to be useful.
3) In 55% of all the topics, the domain expert did not agree with ChatGPT's topic description.
4) For 35%, of the topics, the domain expert considered her own description to be better than ChatGPT's description, while she found 30% of the topics to be equally good, and in 5% of the cases, ChatGPT's description was considered better.

Coherence score vs. usefulness



(a) (b) (c)

Insights per rating

Rating	Domain Expert	ChatGPT	Topic
Useful	Psychoses	Treatment of psychosis in young people, including the role of parents and family in the history and diagnosis.	*'person', 'mother', 'parents', 'patient', 'good', 'father', 'year', 'admission', 'issue', 'voices', 'complaints', 'sometimes', 'hallucinations', 'psychosis', 'diagnostics', 'examination', 'anamnesis', 'thoughts', 'tells', 'whereby'*
Moderately useful	Anamnesis	Examination and admission of a patient, taking into account anamnesis, referrer, additional somatic and psychological factors, and legal orientation.	*'examination', 'recording', 'anamnesis', 'referrer', 'additional', 'somatic', 'known', 'affect', 'year-old', 'preserved', 'means', 'recorded', 'intact ', 'attract', 'intoxications', 'consciousness', 'psychomotor', 'intelligence', 'legal', 'orientation'*
Not really useful	Social skills	Treatment of psychological problems and aggression in a person and their functioning in daily activities.	*'person', 'good', 'problems', 'risk', 'contact', 'general activities of daily living', 'functioning', 'goes', 'aggression', 'psychic', 'functions', 'social', 'skills ', 'occupational', 'conversation', 'gives', 'nurse', 'department', 'going', 'present'*
Not useful	Serious mental illnesses	Psychopharmacology	*'mg', 'admission', 'dp', 'organization', 'medication', 'person', 'treatment', 'complaints', 'ect', 'patient', 'day', 'clozapine', 'effect', 'lithium', 'good', 'disorder', 'psychotic', 'depression', 'start', 'burden'*


Jheronimus
Academy
of Data Science