How Many Topics? Medium Dataset

Overview of Research Work done in Girl Script Summer Of Code 2020

Problem

We need to split the articles to have optimal number of topics.
while splitting data, if we end up having too few topics it would mean an "overly broad" classification on the other hand choosing too many topics would mean "over-clustering".

Understanding Topic Modeling

What is Topic Modeling

Unsupervised ML concept which is used for Natural Language processing to discover underlying thematic structure in a text corpus, where the output is commonly presented as a report of the top terms appearing in each topic. The report not only provides the code for different algorithms but also takes a dive deep into why we used those methods and hence makes the project a Research Project.

Algorithms Researched

1. LDA : Latent Dirichlet Allocation
2. LSA : Latent Semantic Analysis
3. NMF : Non Negetive Matrix Factorization

How we Approached the Problem

1.Collecting Data
2. Pre processing Data
3. Cleaning Data
4. BOW
5. TFIDF Vectorization
6. Running LDA
7. Visualizing
8. Coherence Score
   - c-v measure
   - umass score
9. Hyperparameter Tuning
10. Deciding upon Final Model

LDA : Overview

Dataset

- posts tagged AI, Machine Learning, Datascience or Aritificial Intelligence on Medium,
- user information (Followers, Following)
- publication
- tags
- content of blog
- time frame September 2017 to September 2018

Link :

https://www.kaggle.com/aiswaryaramachandran/medium-articles-with-con

Pre Process and Cleaning Data

1. Used only the Columns that were important to us
   - subtitle
   - text
   - title
2. Deleted rows with languages except English

3. Reduced Dataset left with 3 Columns and ~20000 rows

Lemmitization & Tokenization

Processing

```
processed_titles[30:40]

34    [meta, model, meta, meta, model, deep, learn]
35    [meta, model, meta, meta, model, deep, learn]
36              [tip, data, scienc, team, succeed]
37              [tip, data, scienc, team, succeed]
38              [tip, data, scienc, team, succeed]
39                                  [trust, trust]
40                                  [trust, trust]
41                                  [trust, trust]
42                                  [trust, trust]
43                                  [trust, trust]
Name: title, dtype: object
```

Perform lemmatization and stem preprocessing steps on the data set

```
original document:
Machine Learning Made Easy: What it is and How it Works
['Machine', 'Learning', 'Made', 'Easy:', 'What', 'it', 'is', 'and', 'How', 'it', 'Works']


 tokenized and lemmatized document:
['machin', 'learn', 'easi', 'work']
```

BOW & TFIDF

1

Bag of words is a frequency count of the words occuring in the preprocessed_docs

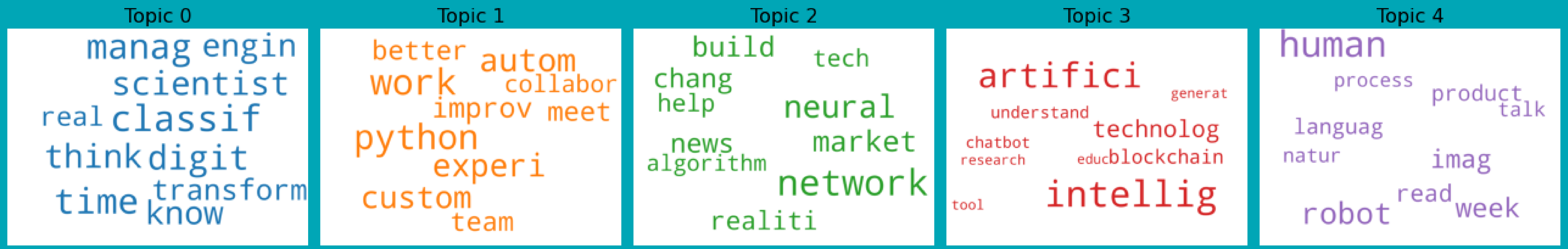
2

Set a convenient time. No need for an agenda. Keep it casual and talk about anything under the sun.

3

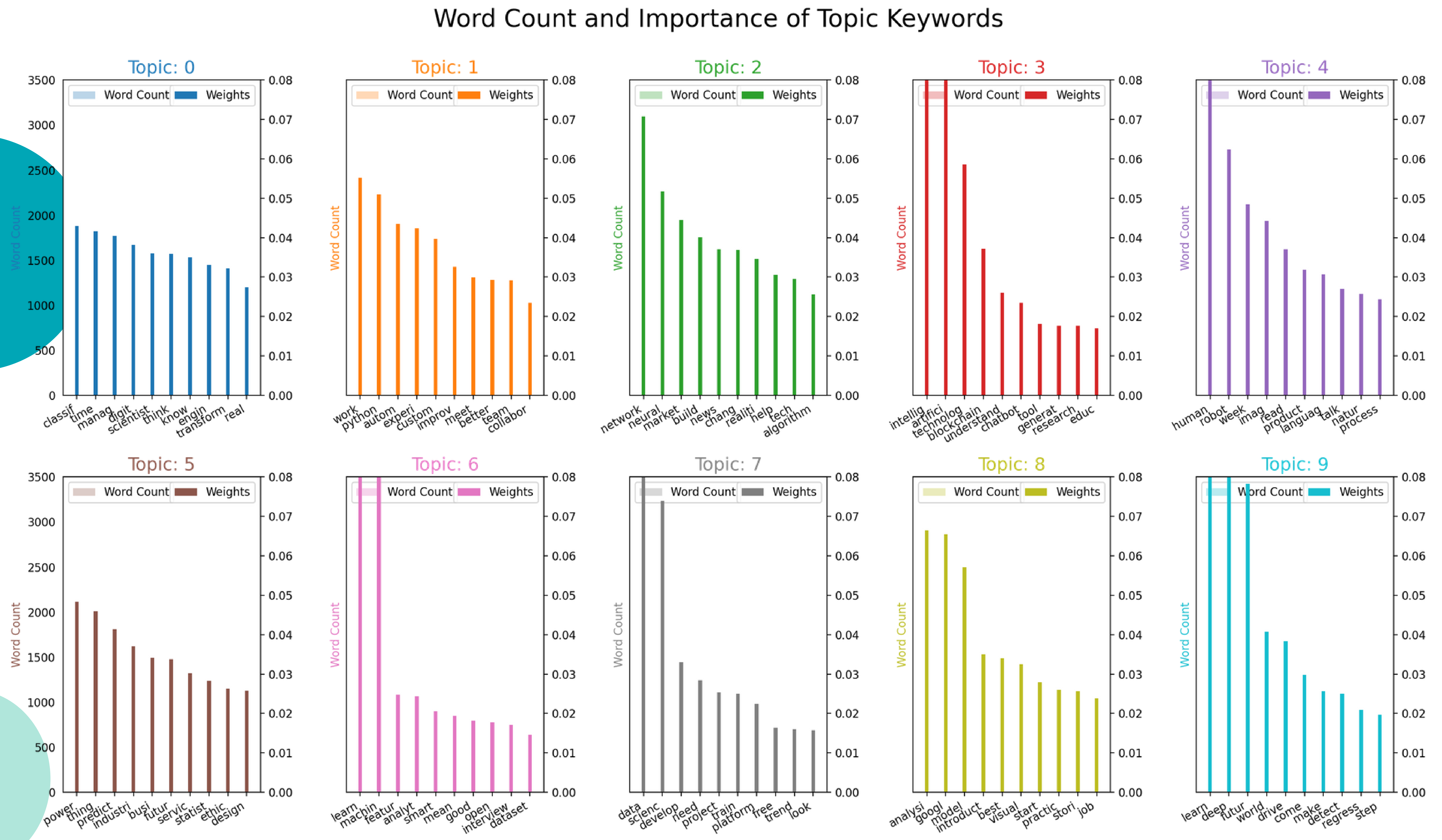
Here are some questions to get the ball rolling: “What made you smile this week?”, “What TV show are you currently watching?” or “What are you doing this weekend?”

Visualization

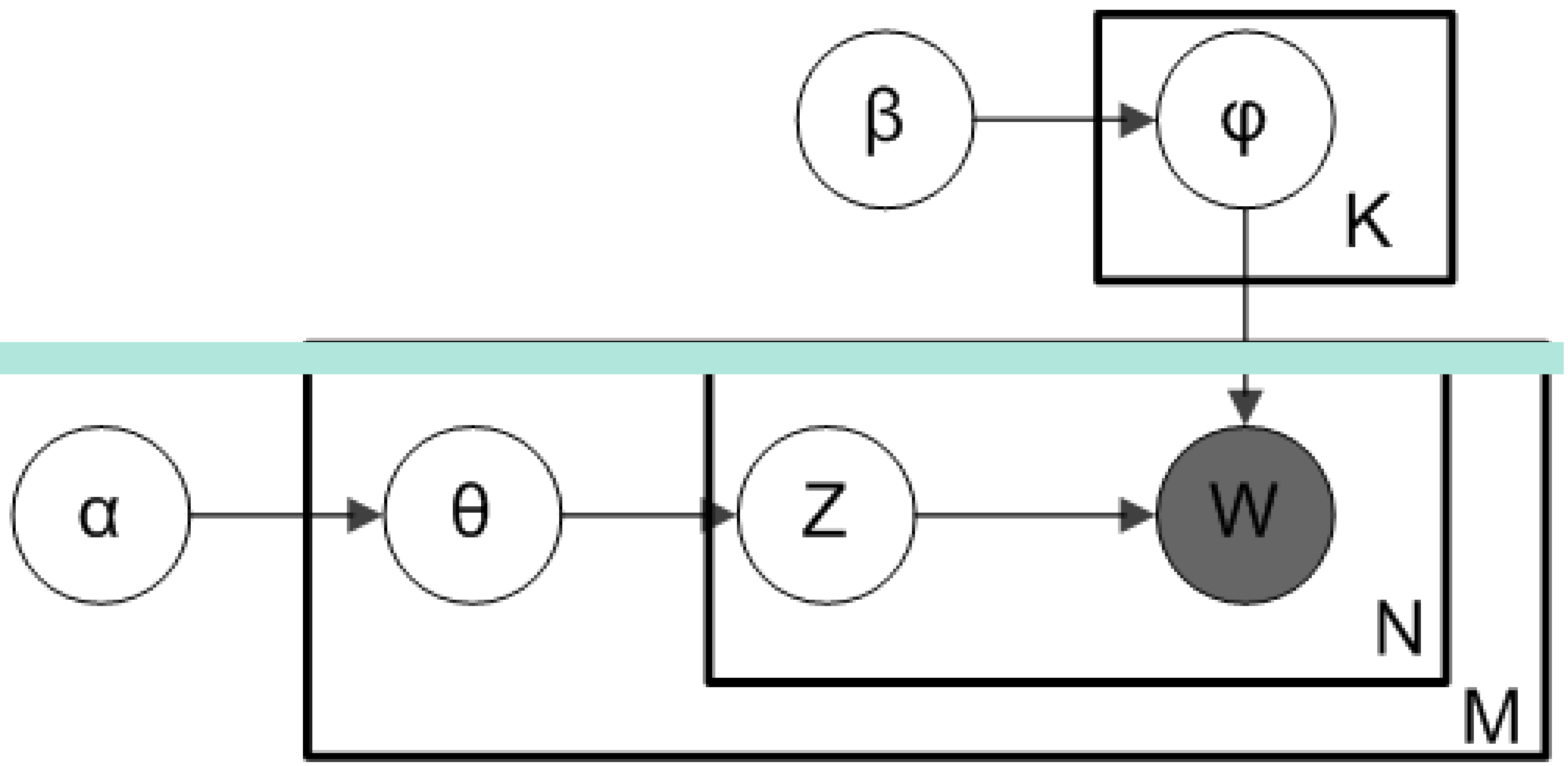


Wordcloud of Top N words in each topic

Word count vs Weights of topics



LDA Model



Above is what is known as a plate diagram of an LDA model where:

α is the per-document topic distributions,

β is the per-topic word distribution,

θ is the topic distribution for document *m*,

φ is the word distribution for topic *k*,

z is the topic for the *n*-th word in document *m*, and

w is the specific word

Topic Distribution generated by the model

```
For topic 1, the words are:  {'work': 0.05513545, 'python': 0.0
51052157, 'autom': 0.043568745, 'experi': 0.04241096, 'custom':
0.039756108, 'improv': 0.032565076, 'meet': 0.029995862, 'bette
r': 0.02930747, 'team': 0.02919189, 'collabor': 0.023429735}
```

Evaluating the Model

Eye Balling Models

- Top N words
- Topics / Documents

Intrinsic Evaluation Metrics

- Capturing model semantics
- Topics interpretability

Human Judgements

- What is a topic

Extrinsic Evaluation Metrics

Is model good at performing predefined tasks

Coherence Score

Using c_v measure

```
Coherence Score:

0.6197147350242858
```

Using UMass Measure

```
Coherence Score:

-17.40738475474645
```

Coherence Score

Using c_v measure

```
Coherence Score:

0.6197147350242858
```

Using UMass Measure

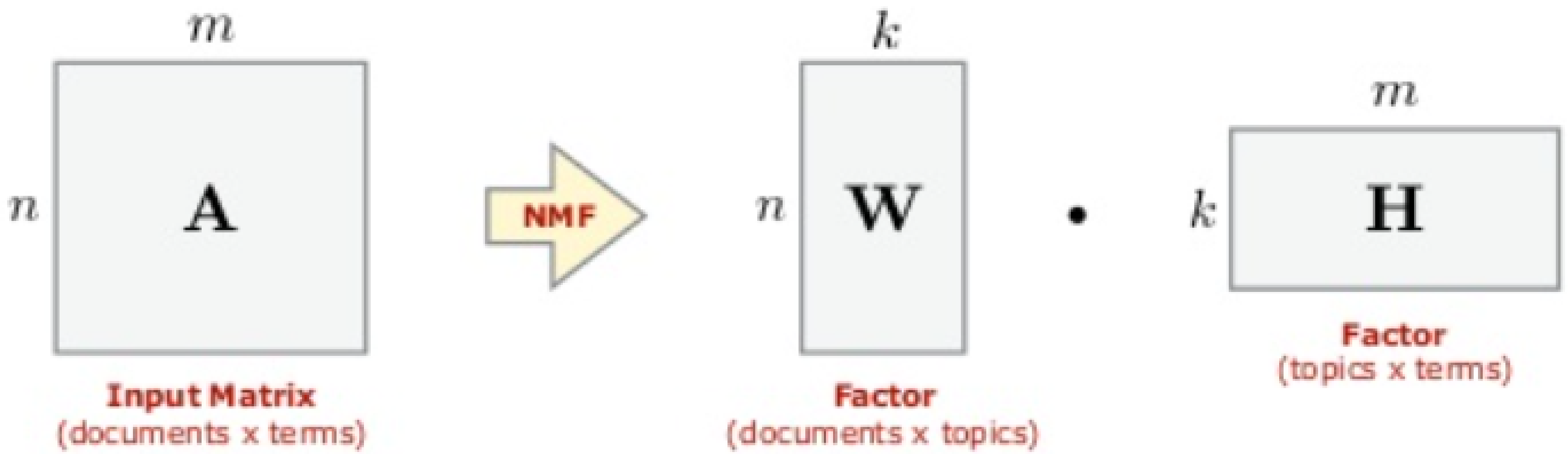
```
Coherence Score:

-17.40738475474645
```

NMF

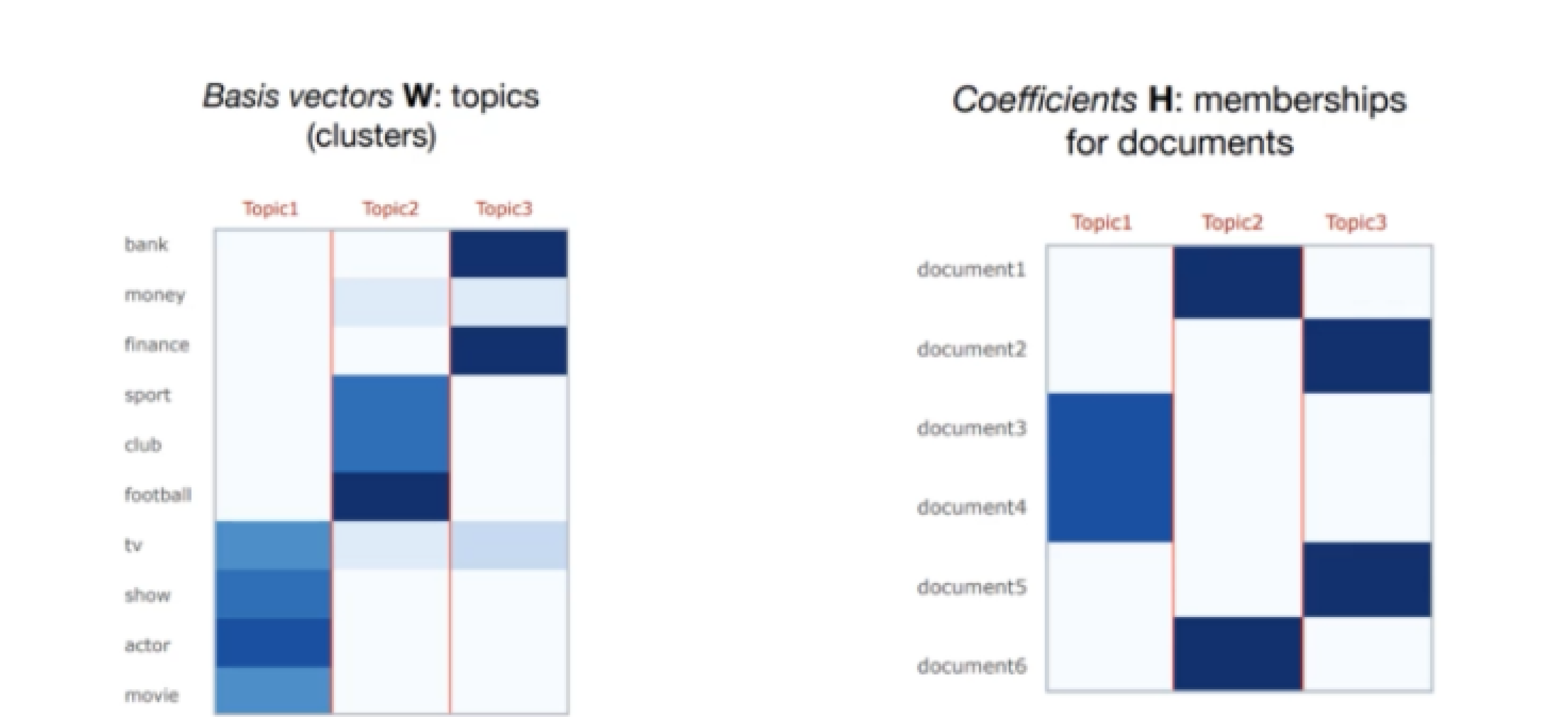
Non Negative Matrix Factorization for Topic Modelling

- Non-negative Matrix Factorization (NMF): Family of linear algebra algorithms for identifying the latent structure in data represented as a non-negative matrix (Lee & Seung, 1999).
- NMF can be applied for topic modeling, where the input is a document-term matrix, typically TF-IDF normalized.
- **Input:** Document-term matrix **A**; User-specified number of topics *k*.
- **Output:** Two *k*-dimensional factors **W** and **H** approximating **A**.



Steps

-form a term documented matrix with tfidf vectorization



Steps

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer(max_df=0.95,min_df=2,stop_words='english')
dtm=tfidf.fit_transform(npr[npr['language']=='en']['text'])
```

```
dtm
```

```
<257655x299679 sparse matrix of type '<class 'numpy.float64'>'
	with 73661900 stored elements in Compressed Sparse Row format>
```

```
from sklearn.decomposition import NMF
nmf_model=NMF(n_components=9, random_state=42)
nmf_model.fit(dtm)
```

```
for index,topic in enumerate(nmf_model.components_):
    print(f"top 15 words # {index}")
    print([tfidf.get_feature_names()[i] for i in topic.argsort()[-15:]])
    print('\n')
```

```
top 15 words # 0
['way', 'know', 'life', 'things', 'don', 'robots', 'time', 'world', 'work', 'humans', 'think', 'just', 'like', 'human
e']


top 15 words # 1
['variable', 'variables', 'value', 'test', 'models', 'set', 'features', 'function', 'linear', 'values', 'data', 'trai
taset', 'regression', 'model']
```