December 5, 2024

UC San Diego

Non-Negative Matrix Factorization (NMF) vs. BERTopic for Topic Modeling

Group 14

Sarah Borsotto, John Driscoll, Taylor Martinez, Thuy Nguyen

DSC 210 FA'24 Numerical Linear Algebra

Section 1: Introduction.

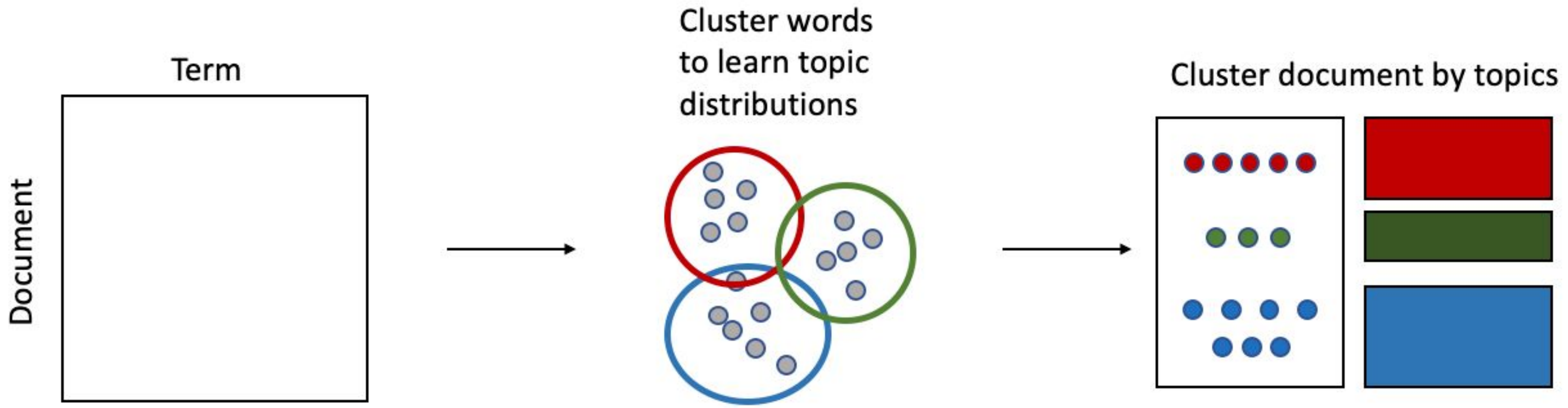
Background

- 402.89 million terabytes of data generated daily –
  80% unstructured
- Humans struggle to process; machines struggle to interpret

What is Topic Modeling?

- Unsupervised machine learning technique
- Identify patterns in text to summarize themes

How Topic Modeling Works



Applications

Topic modeling acts as the foundation for many NLP tasks

Thematic Analysis

Extracts topics from text input and derives important information from search engine query

Content Recommendations

Identify topics user is interested in and recommend relevant content

Sentiment Analysis

Understand overall sentiment of text

Trend Analysis

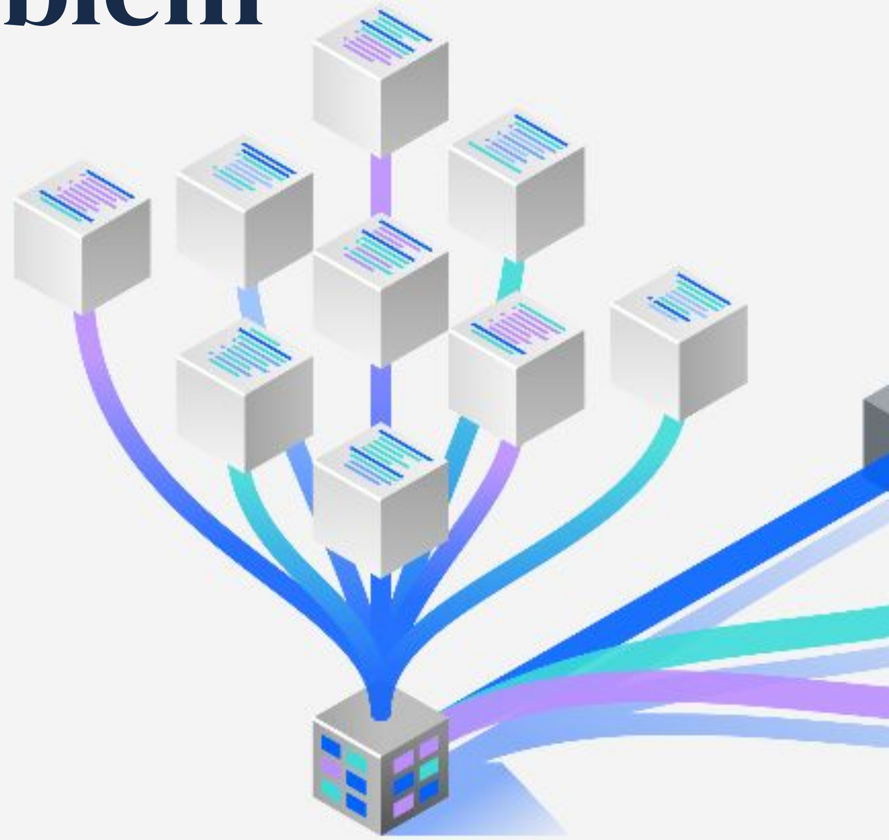
Identify which topics are trending in a specific domain

Section 2: Problem Formulation & Relation to Linear Algebra.

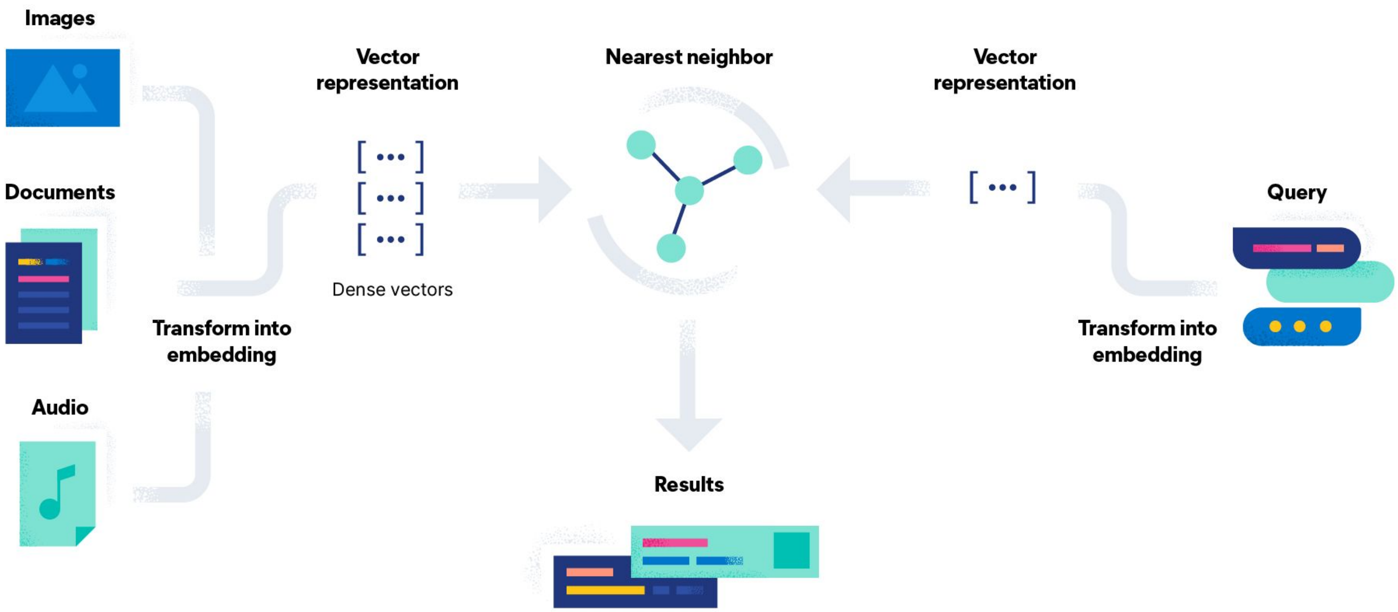
The Problem

Overview

- High Level Idea: an attempt to automate document indexing and retrieval
- Evolution of technology – social media and real-time text generation
  - Need to handle shorter, noisier text
  - Temporal dynamics



Mathematical Formulation

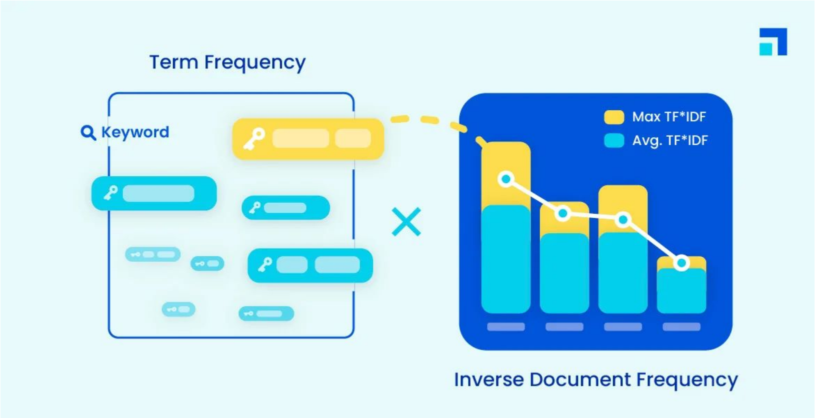


An example of information retrieval

Relation to Numerical Linear Algebra:

TF-IDF

- Measures the importance of words in a corpus
- Combines Term Frequency (TF) and Inverse Document Frequency (IDF)



Term Frequency–Inverse Document Frequency

Term Frequency

Amount of times a specific
word appears in a document

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

Where **t** is the number of times a term occurs in the document **d**

Inverse Document Frequency

How often a particular
word appears in a document

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d : d \in D \text{ and } t \in d\}|}$$

Where **N** represents total number of documents in a corpus and **{d : d ⊆ D and t ⊆ d}** is the total number of documents that contain t

TF-IDF Example

Sentence A: The car is driven on the road

Word	TF A	TF B	IDF	TF*IDF A	TF*IDF B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d : d \in D \text{ and } t \in d\}|}$$

Sentence B: The truck is driven on the highway

Word	TF A	TF B	IDF	TF*IDF A	TF*IDF B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

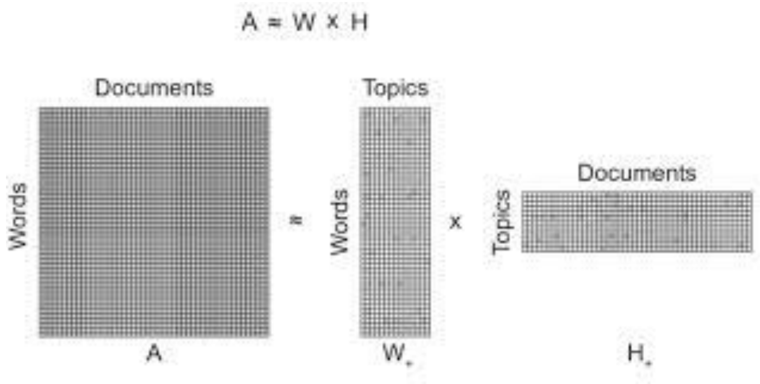
Higher TF-IDF score shows a word is highly relevant to that specific document while being less common in other documents in the corpus

Section 3:
Linear Algebra Approach.

An NLA Approach: Non-Negative Matrix Factorization (NMF)

What is NMF?

Traditional, linear-algebra-based machine learning technique that decomposes a term-document matrix into two smaller non-negative matrices representing topics and word associations



Non-Negative Matrix Factorization (NMF) Example

Using NMF to create personalized food recommendations

$$V = \begin{array}{c} \\ \\ \end{array}\begin{array}{cccccc|c} John & Alice & Mary & Greg & Peter & Jennifer & \\ \end{array}$$

$$V = \left(\begin{array}{cccccc} 0 & 1 & 0 & 1 & 2 & 2 \\ 2 & 3 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 3 & 4 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right) \begin{array}{c} Vegetables \\ Fruits \\ Sweets \\ Bread \\ Coffee \end{array}$$

The initial V matrix

Perform NMF

$$W = \begin{array}{ccc} Component1 & Component2 & Component3 \\ \end{array}$$

$$W = \left(\begin{array}{ccc} 2.19 & 0. & 0.03 \\ 1.53 & 3.13 & 0.11 \\ 0.61 & 1.58 & 0. \\ 0.01 & 0. & 1.88 \\ 0.47 & 0. & 0. \end{array}\right) \begin{array}{c} Vegetables \\ Fruits \\ Sweets \\ Bread \\ Coffee \end{array}$$

$$H = \begin{array}{cccccc} John & Alice & Mary & Greg & Peter & Jennifer \\ \end{array}$$

$$H = \left(\begin{array}{cccccc} 0. & 0.43 & 0. & 0.42 & 0.96 & 0.86 \\ 0.64 & 0.66 & 0.34 & 0. & 0.18 & 0.22 \\ 0. & 1.06 & 1.58 & 2.12 & 0.52 & 0.53 \end{array}\right) \begin{array}{c} Component1 \\ Component2 \\ Component3 \end{array}$$

The W and H matrices after NMF

Optimizing NMF:

Minimizing the Difference

Frobenius Norm

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(|a_{ij}|)^2}$$

Kullback-Liebler Divergence

$$\text{kl\_div}(x, y) = \begin{cases} x\log\left(\frac{x}{y}\right) - x + y, & \text{if } x > 0, y > 0, \\ y, & \text{if } x = 0, y \geq 0, \\ \infty, & \text{otherwise.} \end{cases}$$

Optimizing NMF:

Iterative Algorithms

Coordinate Descent Solver

$$\text{minimize } \phi(x_1, x_2, \ldots, x_p) \text{ for } x_i \in \Omega_i$$

$$x_i^{k+1} = \text{argmin}_\xi \, \phi(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, \xi, x_{i+1}^k, \ldots, x_p^k).$$

Multiplicative Update Solver

$$W^{\text{new}} = \left[W + S \odot \left(AH^T - WHH^T\right)\right]_+$$
$$H^{\text{new}} = \left[H + S' \odot \left(W^T A - W^T W H\right)\right]_+,$$

$$S = W \oslash (WHH^T), \quad S' = H \oslash (W^T W H).$$

Experiment Setup and Results

Dataset

20NewsGroup

- A collection of approximately 20,000 documents distributed amongst 20 different newsgroups
- Distribution is roughly even across topics
- Commonly used dataset for text classification

Tools

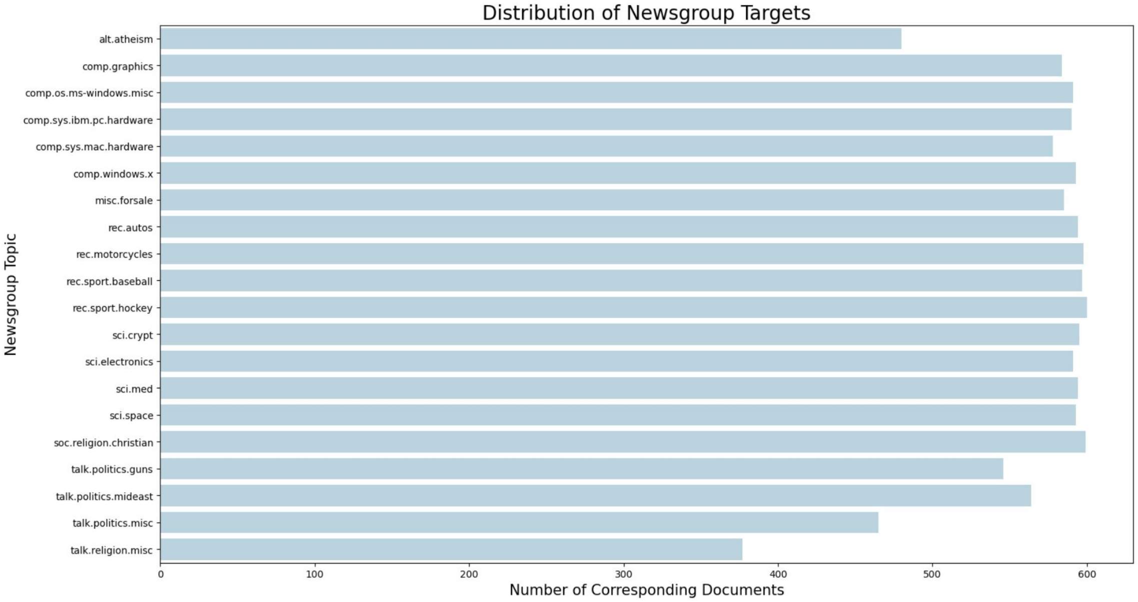
- Implemented our code in Jupyter Notebooks
- Tracked version control through Github

Libraries: sklearn, seaborn, pandas, numpy, nltk, wordcloud, bertopic, gensim

Evaluation

1. Subjective Judgement of Topic Representations
2. Visualizations of Topic Distributions
    a. Word Clouds
    b. Topic Visualizers
    c. Class Distributions
3. Coherence Score
4. Diversity Score

Exploratory Data Analysis



Themes

```
rec.sport.hockey            600
soc.religion.christian      599
rec.motorcycles             598
sci.crypt                   595
sci.med                     594
sci.space                   593
misc.forsale                585
comp.graphics               584
talk.politics.mideast       564
talk.politics.guns          546
```

Data Preprocessing

Clean Data → Lemmatize → Tokenize → Remove Stop Words

Example

```
'From: bmdelane@quads.uchicago.edu (brian manning delaney)\nSubject: Brain Tumor Treatment (thanks)\nReply-To: bmde
lane@midway.uchicago.edu\nOrganization: University of Chicago\nLines: 12\n\nThere were a few people who responded t
o my request for info on\ntreatment for astrocytomas through email, whom I couldn\'t thank\ndirectly because of mai
l-bouncing probs (Sean, Debra, and Sharon).  So\nI thought I\'d publicly thank everyone.\n\nThanks! \n\n(I\'m sure
glad I accidentally hit "rn" instead of "rm" when I was\ntrying to delete a file last September. "Hmmm... \'News?\'
What\'s\nthis?"....)\n\n-Brian\n'
```

```
'bmdelane quad uchicago brian manning delaney brain tumor treatment reply bmdelane midway uchicago organization chi
cago line people responded request info treatment astrocytomas email couldn thank directly mail bouncing probs sean
debra sharon thought publicly thank sure glad accidentally hit instead trying delete file september hmmm news bria
n'
```

NMF Implementation

```python
# Initialize TfidfVectorizer
vectorizer = TfidfVectorizer(max_df=0.95, min_df=2, max_features=1000)

# Fit and transform the processed abstracts into TF-IDF
tfidf = vectorizer.fit_transform(newsgroup_df['processed_documents'])
```

```python
def do_nmf(tfidf, n_topics):
    # Specify the number of topics
    nmf_model = NMF(n_components=n_topics)
    W = nmf_model.fit_transform(tfidf)  # Document-topic matrix   (n_samples, n_components)
    H = nmf_model.components_          # Topic-term matrix       (n_components, n_features)
    return W, H, nmf_model
```

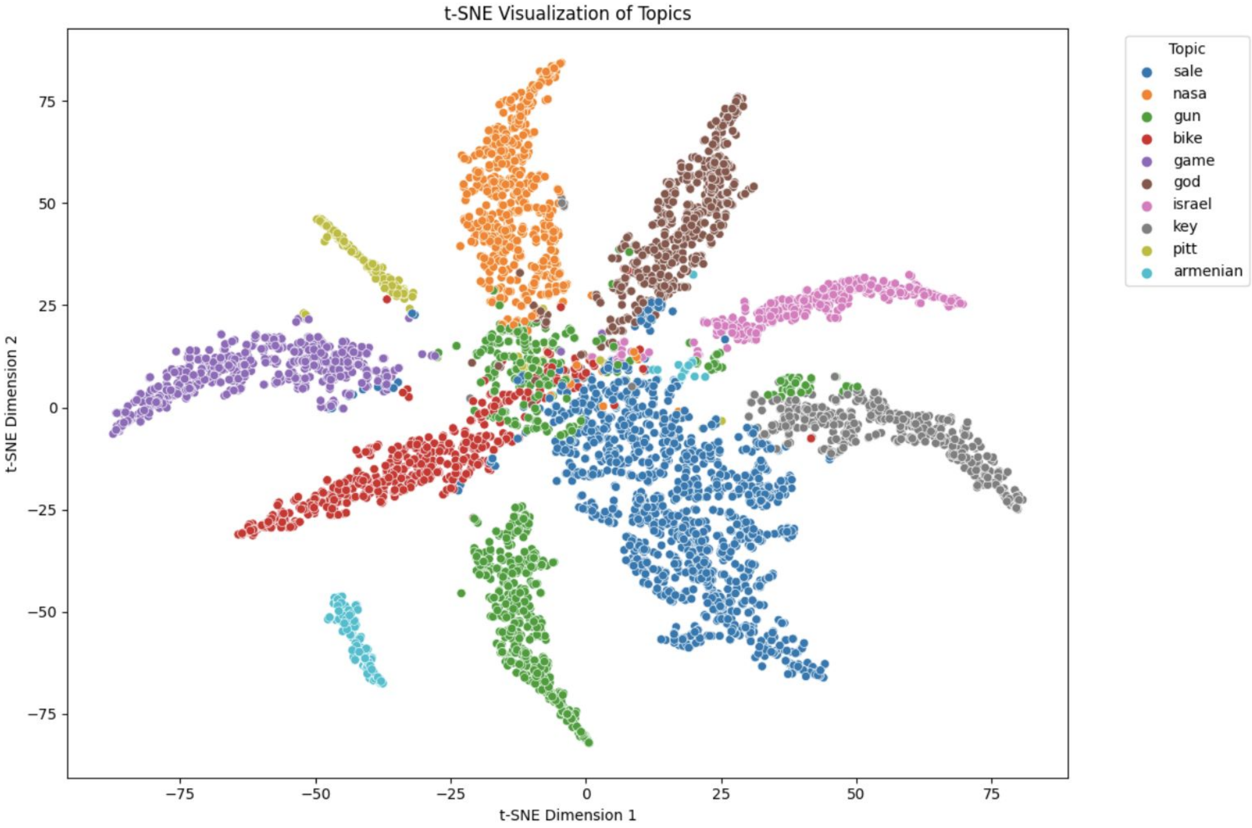
```python
# we chose a subset of 10 topics from our 20newsgroup dataset
n_topics = 10
W, H, nmf_model = do_nmf(tfidf, n_topics)
```

Our Results

```
Topic #1:
say,believe,faith,bible,christ,people,church,jesus,christian,god
Topic #2:
crypto,phone,netcom,algorithm,government,escrow,encryption,clipper,chip,key
Topic #3:
software,new,distribution,offer,mail,host,posting,file,graphic,sale
Topic #4:
reply,computer,science,soon,univ,pittsburgh,bank,gordon,geb,pitt
Topic #5:
win,year,playoff,season,play,nhl,player,hockey,team,game
Topic #6:
state,right,peace,policy,jewish,palestinian,jew,arab,israeli,israel
Topic #7:
pat,alaska,jpl,moon,orbit,digex,access,gov,space,nasa
Topic #8:
handgun,crime,criminal,control,law,right,weapon,firearm,people,gun
Topic #9:
azeri,genocide,greek,turkey,serdar,argic,turk,armenia,turkish,armenian
Topic #10:
bnr,helmet,rider,riding,writes,dog,ride,motorcycle,dod,bike
```

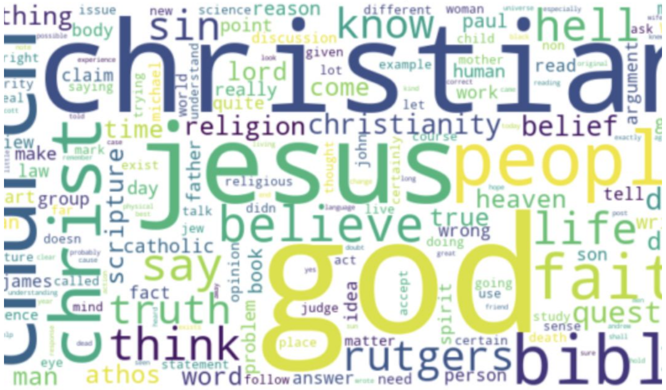
Themes

```
rec.sport.hockey
soc.religion.christian
rec.motorcycles
sci.crypt
sci.med
sci.space
misc.forsale
comp.graphics
talk.politics.mideast
talk.politics.guns
```

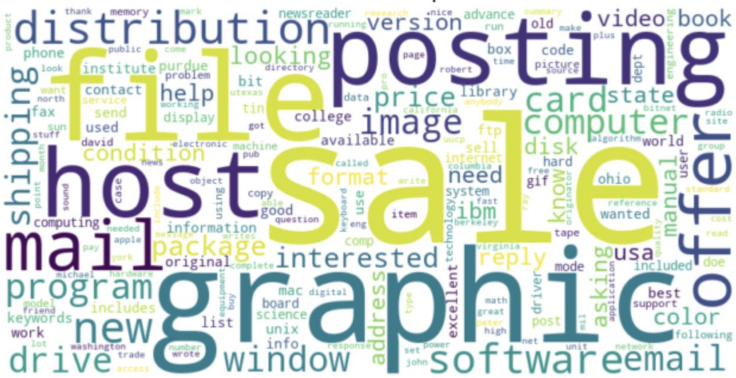

t-SNE Visualization of Topics
75
50
25
0
−25
−50
−75
t-SNE Dimension 2
−75
−50
−25
0
25
50
75
t-SNE Dimension 1
Topic
sale
nasa
gun
bike
game
god
israel
key
pitt
armenian

Clusters

Word Cloud for Topic 0

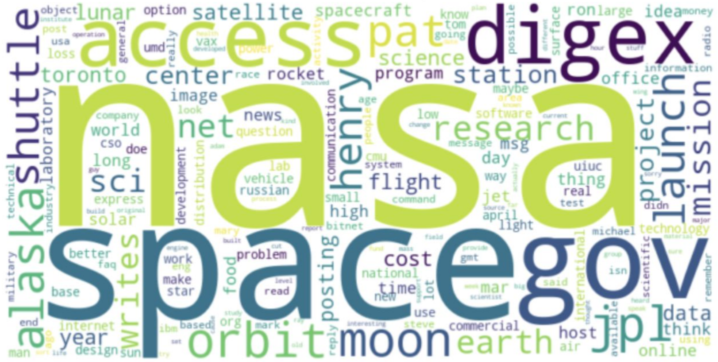


Word Cloud for Topic 3

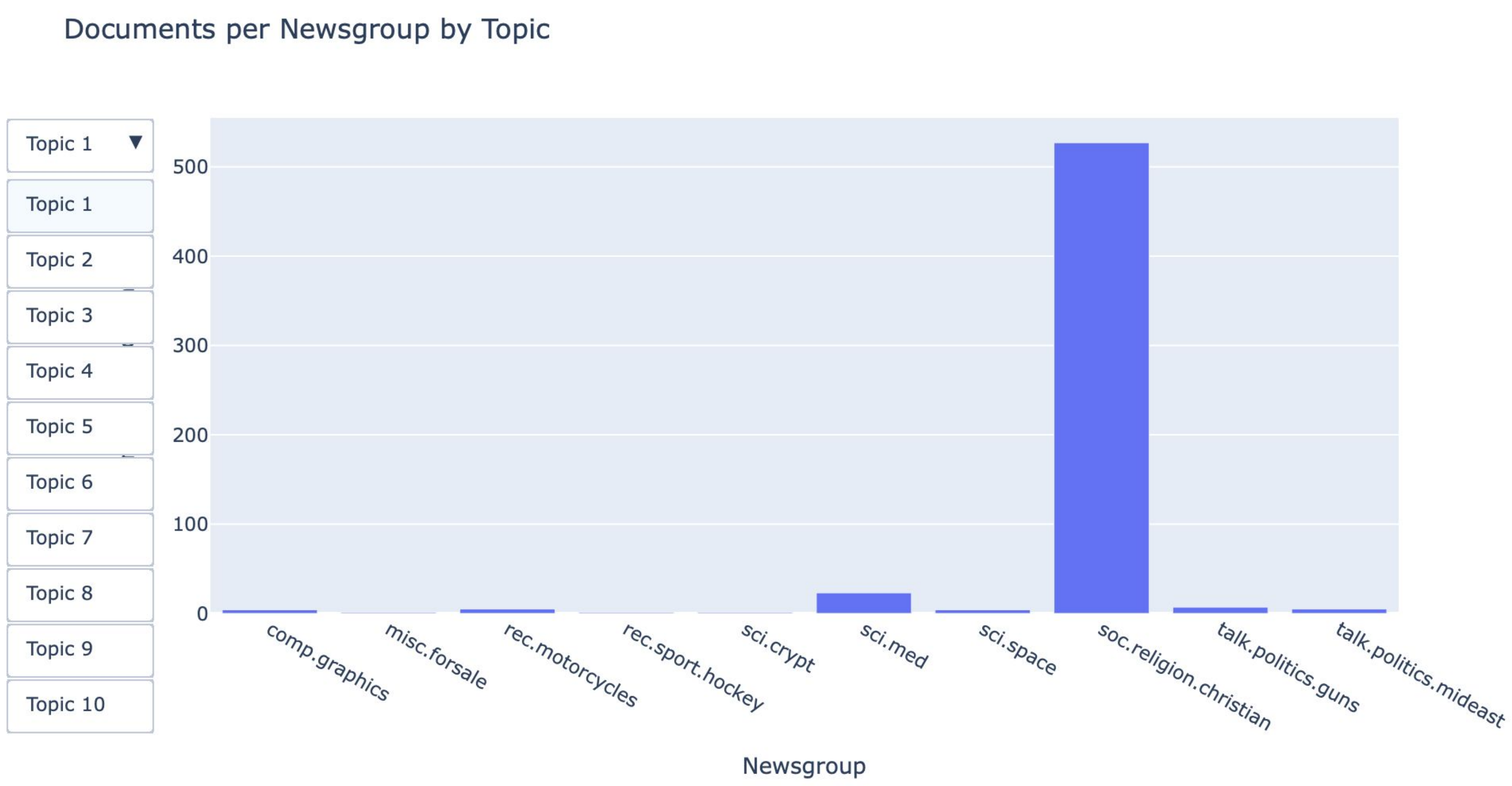


Word Cloud for Topic 4

Word Cloud for Topic 6



Word Clouds



Topic Distribution

Coherence

- Measures the degree to which the words within a topic are related to each other
- Doesn't use the model itself - based on the documents in the training set
- Four steps: segmentation, probability calculation, confirmation measure, and aggregation

0.43

Diversity

- Measures how distinct topics are from one another
- More diverse topics cover more themes
- Lots of different metrics: jaccard, embedding based, etc.
- Use the proportion of unique words for the top 10 words

0.98

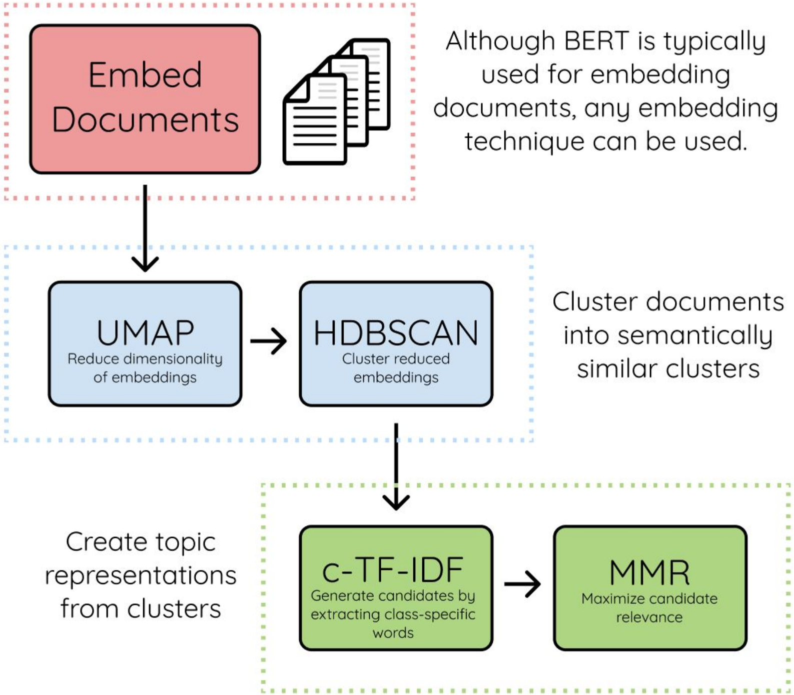
Section 4:
State of the Art (SOTA).

A Modern Approach: BERTopic

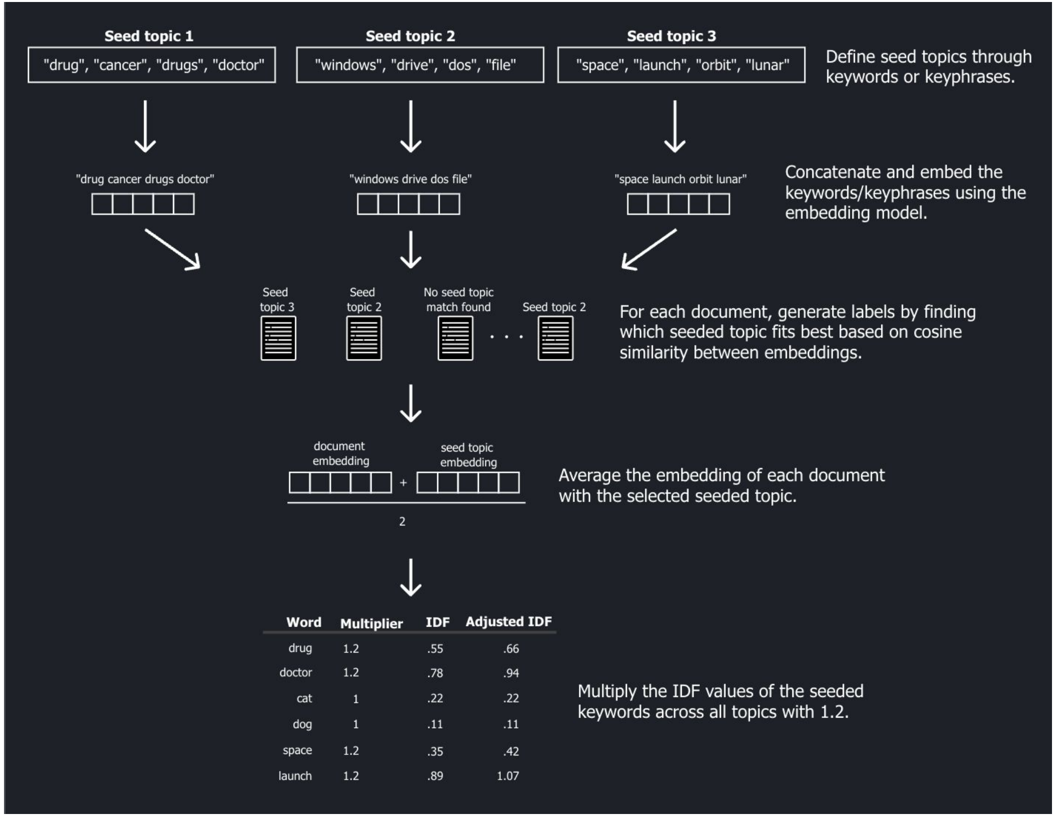
What is BERTopic?

Transformer-based machine learning model that uses Bidirectional Encoder Representations from Transformers (BERT) embeddings

How Does BERTopic Work?



How Does BERTopic Work?



NMF vs. BERTopic

NMF

- Computationally efficient and scalable
- Easy to implement
- Matrix operations - more likely to capture incoherent topics
- User must define # of topics in advance
- Each document can contain several topics

BERTopic

- Computationally intensive
- Allows for multilingual analysis
- Uses embeddings so no data preprocessing necessary
- Automatically finds # of topics
- Can prune topics but lose fidelity
- Each document assigned to 1 topic, don't receive probabilities of each

BERTopic Implementation

```python
# Pre-calculate embeddings
data = newsgroup_df["processed_documents"]

embedding_model = SentenceTransformer("all-MiniLM-L6-v2")
embeddings = embedding_model.encode(data, show_progress_bar=True)

#method for dimensionality reduction
umap_model = UMAP.UMAP(#n_neighbors=50,
                  #min_dist=0.5,
                  #metric='cosine',
                  random_state=35)

#higherarchial clustering method
hdbscan_model = HDBSCAN(min_cluster_size=50,
                        #min_samples=30
                        )

#vectorizer to create matrix from corpus
vectorizer_model = TfidfVectorizer(min_df=1, max_
```

```python
topic_model = BERTopic(

  # Pipeline models|
  embedding_model=embedding_model,
  umap_model=umap_model,
  hdbscan_model=hdbscan_model,
  vectorizer_model=vectorizer_model,

  # Hyperparameters
  top_n_words = 10,
  verbose=True
)

# Train model
topics, probs = topic_model.fit_transform(data, embeddings)
```

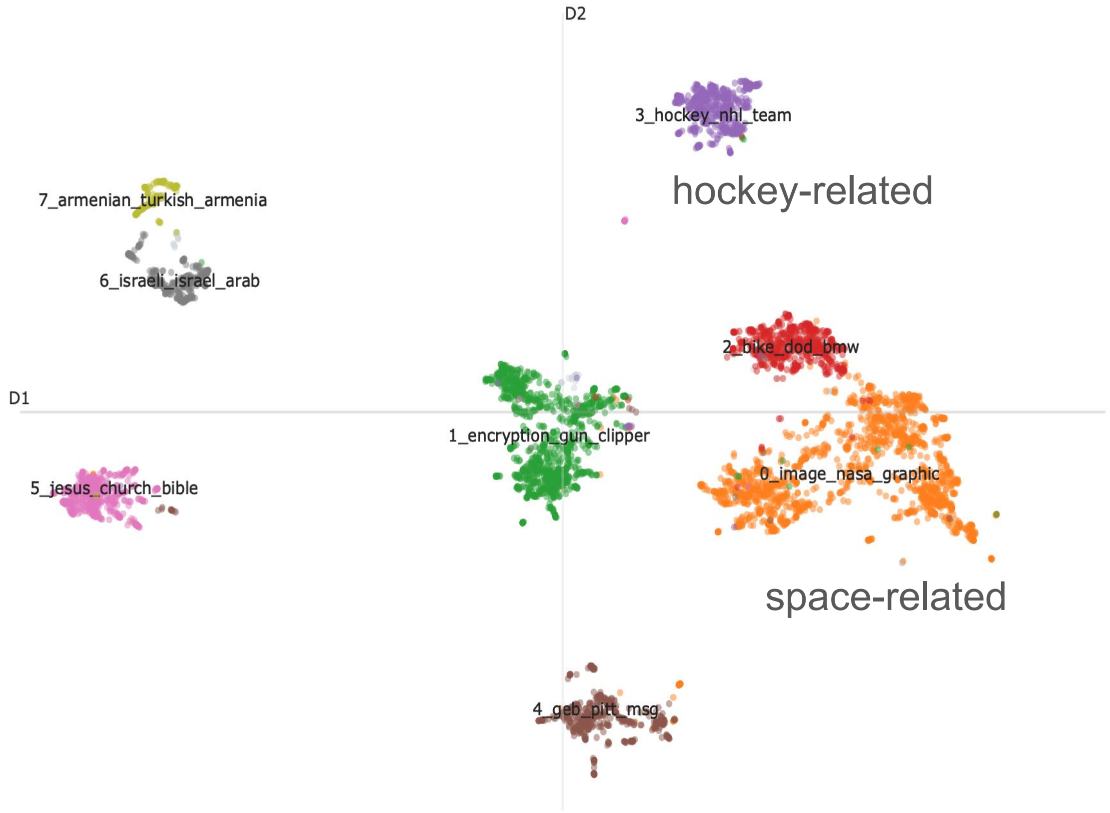
Our Results

1	[encryption, gun, clipper, firearm, escrow, chip, privacy, government, nsa, crypto]
2	[bike, dod, motorcycle, bmw, rider, ride, behanna, helmet, riding, nec]
3	[nhl, hockey, team, playoff, player, season, puck, espn, lemieux, islander]
4	[jesus, bible, church, christ, faith, sin, christianity, scripture, athos, catholic]
5	[geb, pitt, msg, dyer, patient, disease, food, candida, gordon, health]
6	[sale, shipping, printer, disk, manual, forsale, floppy, ohio, item, brand]
7	[nasa, orbit, launch, satellite, spacecraft, shuttle, moon, mission, jpl, lunar]
8	[image, jpeg, polygon, format, gif, vga, algorithm, pixel, window, mode]
9	[israeli, israel, arab, jew, lebanese, palestinian, cpr, gaza, bony, hernlem]
10	[armenian, turkish, armenia, turk, azerbaijani, azerbaijan, turkey, argic, azeri, serdar]

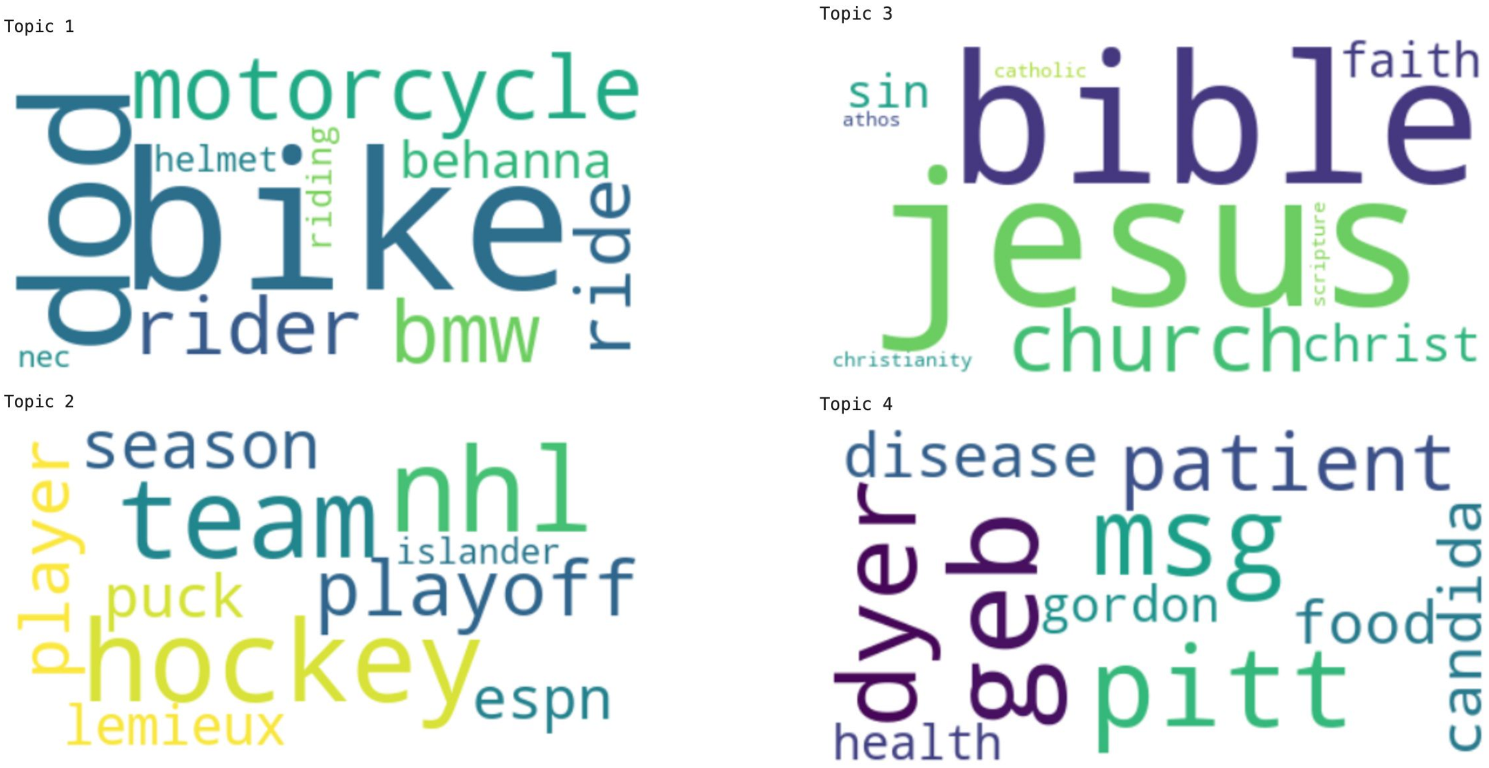
Themes

```
rec.sport.hockey
soc.religion.christian
rec.motorcycles
sci.crypt
sci.med
sci.space
misc.forsale
comp.graphics
talk.politics.mideast
talk.politics.guns
```

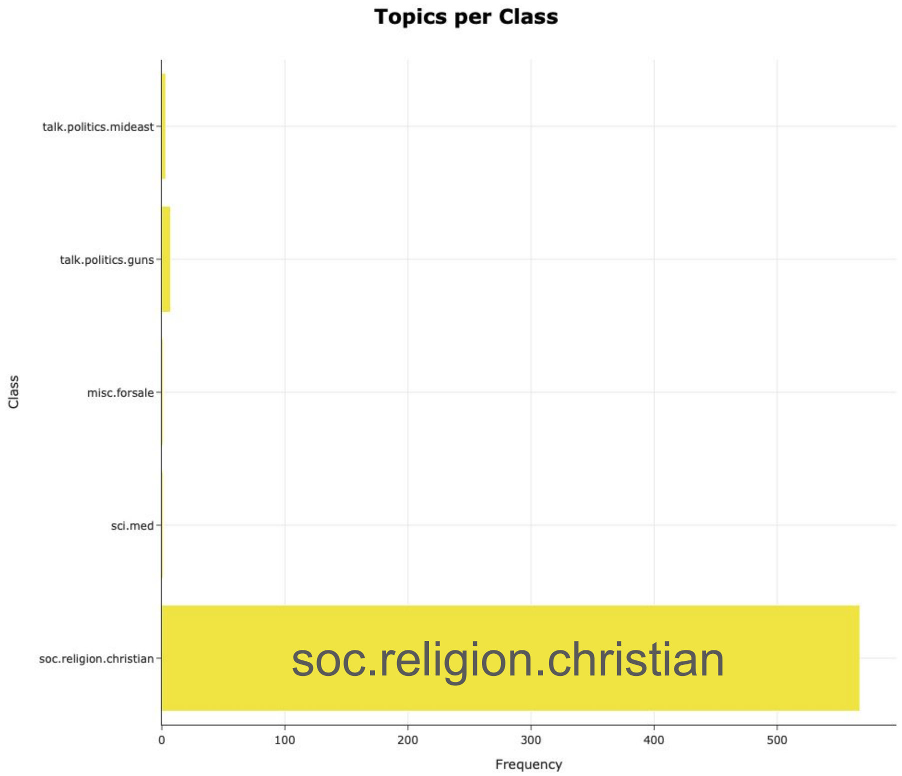
Clusters



D2
D1
3_hockey_nhl_team
hockey-related
7_armenian_turkish_armenia
6_israeli_israel_arab
2_bike_dod_bmw
1_encryption_gun_clipper
0_image_nasa_graphic
5_jesus_church_bible
space-related
4_geb_pitt_msg


Topic 1
motorcycle
dod
bike
helmet
riding
behanna
ride
rider
bmw
nec
Topic 2
season
player
team
nhl
islander
puck
playoff
hockey
espn
lemieux
Topic 3
faith
catholic
sin
athos
bible
jesus
scripture
church
christ
christianity
Topic 4
disease
patient
dyer
geb
msg
candida
gordon
food
pitt
health

Word Clouds



Theme 3

[jesus, bible, church, christ, faith, sin, christianity, scripture, athos, catholic]

Topic Distribution

Coherence

- How related are words within each topic?
- Uses documents, not model
- segmentation, probability calculation, confirmation measure, aggregation

0.51

Diversity

- How distinct are topics?
- More diverse topics cover more themes
- Proportion of unique words in top 10

0.99

Section 5: Concluding Remarks.

NMF

- NMF had more cluster overlap
- NMF showed more variability in document distribution per topic
- NMF model had a better representation of our original themes

Measure	NMF	BERTopic
Coherence	0.43	0.51
Diversity	0.98	0.99

BERTopic

- BERTopic has more defined group clusters
- BERTopic was able to map topics to their newsgroup documents better
- BERTopic does better than NMF in terms of Coherence and Diversity

Conclusion

NMF appears to better extract abstract themes, like religion and sport, whereas BERTopic appears to better extract more specific topics, like christianity and hockey.

References

[1] How hdbscan works. https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html.
[2] 20 newsgroups. http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html, 1999. Last modified: 9 September 1999.
[3] What percentage of data is unstructured? 3 must-know statistics. https://edgedelta.com/company/blog/what-percentage-of-data-is-unstructured, 2024.
[4] A. Bhangale. Introduction to topic modelling with lda, nmf, top2vec and bertopic. https://medium.com/blend360/introduction-to-topic-modelling-with-lda-nmf-top2vec-and-bertopic-ffc3624d44e4,2023.
[5] H. M. A. Subakti and N. Hariadi. The performance of bert as data representation of text clustering. J Big Data, 2022.

References

[6] Bindel. Numerics for data science, 2018-06-21. https://www.cs.cornell.edu/~bindel/class/sjtu-summer18/lec/2018-06-21.pdf, 2018.

[7] C. Goyal. Part 15: Step by step guide to master nlp – topic modelling using nmf. https://www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf/, 2024.

[8] R. Egger and J. Yu. A topic modeling comparison between lda, nmf, top2vec, and bertopic to demystify twitter posts. Frontiers in Sociology, 7, 05 2022.

[9] F. Chiusano. Two minutes nlp — learn tf-idf with easy examples. https://medium.com/nlplanet/two-minutes-nlp-learn-tf-idf-with-easy-examples-7c15957b4cb3#:~:text=Term%20Frequency%20(TF)%3A%20how,common%20words%20have%20low%20scores., 2022.

References

[10] FreeCodeCamp. How to process textual data using tf-idf in python. 2024. Accessed: 2024-12-02.
[11] Geeks for Geeks. Non-negative matrix factorization. https://www.geeksforgeeks.org/non-negative-matrix-factorization/, 2023.
[12] Geeks for Geeks. Topic modeling examples. https://www.geeksforgeeks.org/topic-modeling-examples/, 2024.
[13] M. Grootendorst. Bertopic: Visualization, 2024. Accessed: 2024-12-04.
[14] ProjectPro. A beginner's guide to topic modeling nlp. https://www.projectpro.io/article/topic-modeling-nlp/801, 2024.
[15] Qualtrics. Topic modeling: definition, benefits and use cases. https://www.qualtrics.com/experience-management/research/topic-modeling/.

References

[16] S. Kim. Let us extract some topics from text data — part iv: Bertopic. https://towardsdatascience.com/let-us-extract-some-topics-from-text-data-part-iv-bertopic-46ddf3c91622, 2022.

[17] B. Slawski. Semantic topic modeling for search queries at google. https://gofishdigital.com/blog/semantic-topic-modeling/, 2016.

[18] SOAX. How much data is generated every day? https://soax.com/research/data-generated-per-day, 2024.

[19] H. Vijay. Dimensionality reduction : Pca, tsne, umap. https://aurigait.com/blog/blog-easy-explanation-of-dimensionality-reduction-and-techniques/, May 2023.

Thank You!