NLP ENGLISH TOPIC MODELING T27

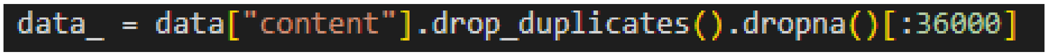
NAME	ID
نور محمد محمد عبدالعزيز الحوت	20191700702
محمد أيمن فاروق عبدالعزيز	20191700511
محمد عبدالله عبدالحكيم سالم	20191700835
علي الدين عمر علي عبدالفتاح	20191700389
سيف أيمن أحمد عبدالخالق	20201700391

DATA PREPARATION & PREPROCESSING:

The raw dataset “**articles1.csv**” contains various articles that require preprocessing to enhance the quality and extract meaningful features.

Data Preparation:

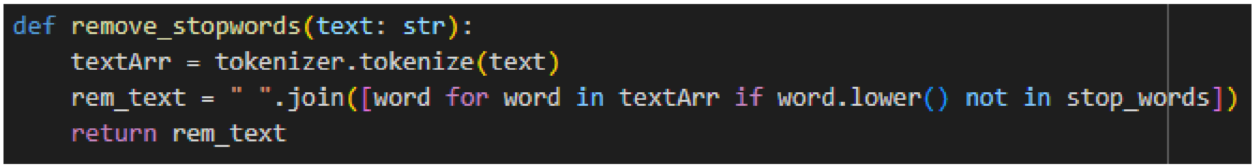
- **Dataset**: articles1.csv
- **Steps**:
  - Uploaded the dataset from the local machine.
  - Removed duplicates and null values.
  - Limited the dataset to 36,000 articles for processing efficiency.



```
data_ = data["content"].drop_duplicates().dropna()[:36000]
```

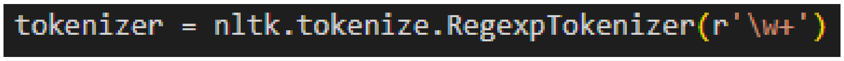
Preprocessing Steps:

- **Stopword Removal:** Using NLTK to remove common English stopwords.



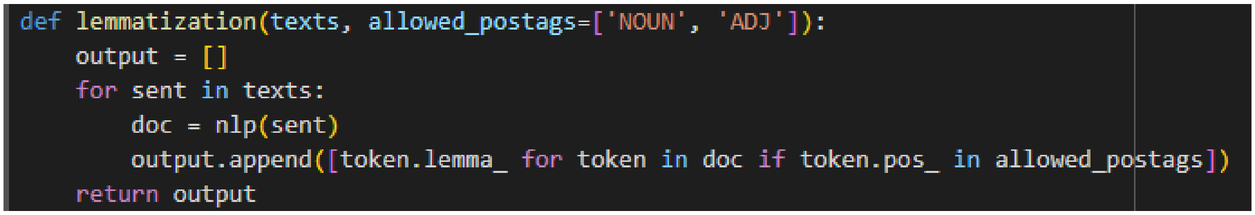
```
def remove_stopwords(text: str):
    textArr = tokenizer.tokenize(text)
    rem_text = " ".join([word for word in textArr if word.lower() not in stop_words])
    return rem_text
```

- **Tokenization:** Tokenize the text to separate words.



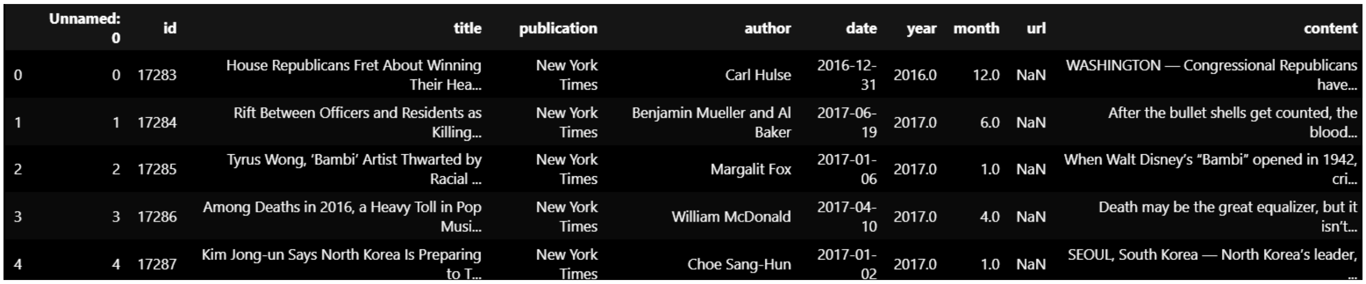
```
tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
```

- **Lemmatization:** Reduce words to their base or root form using SpaCy.



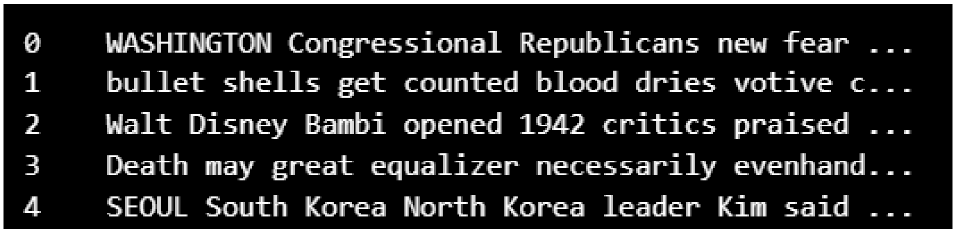
```
def lemmatization(texts, allowed_postags=['NOUN', 'ADJ']):
    output = []
    for sent in texts:
        doc = nlp(sent)
        output.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
    return output
```

Raw Data in .csv file:



	Unnamed: 0	id	title	publication	author	date	year	month	url	content
0	0	17283	House Republicans Fret About Winning Their Hea...	New York Times	Carl Hulse	2016-12-31	2016.0	12.0	NaN	WASHINGTON — Congressional Republicans have...
1	1	17284	Rift Between Officers and Residents as Killing...	New York Times	Benjamin Mueller and Al Baker	2017-06-19	2017.0	6.0	NaN	After the bullet shells get counted, the blood...
2	2	17285	Tyrus Wong, 'Bambi' Artist Thwarted by Racial ...	New York Times	Margalit Fox	2017-01-06	2017.0	1.0	NaN	When Walt Disney's "Bambi" opened in 1942, cri...
3	3	17286	Among Deaths in 2016, a Heavy Toll in Pop Musi...	New York Times	William McDonald	2017-04-10	2017.0	4.0	NaN	Death may be the great equalizer, but it isn't...
4	4	17287	Kim Jong-un Says North Korea Is Preparing to T...	New York Times	Choe Sang-Hun	2017-01-02	2017.0	1.0	NaN	SEOUL, South Korea — North Korea's leader, ...

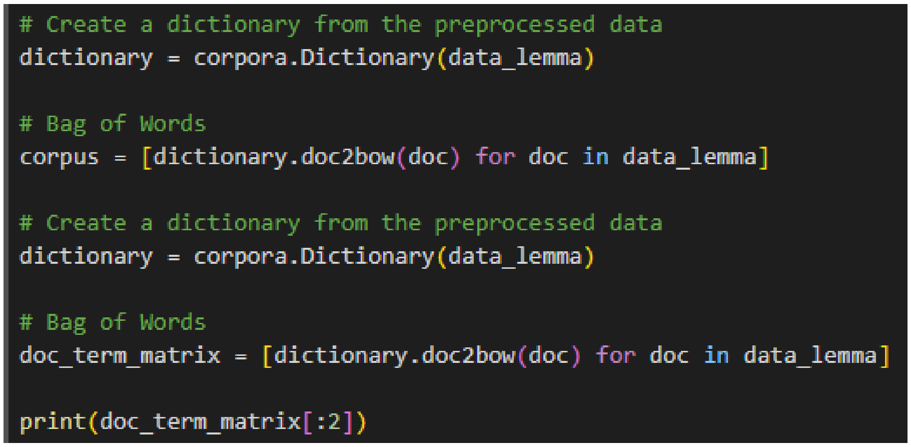
Data after removing duplicates, null values & stopwords:



```
0    WASHINGTON Congressional Republicans new fear ...
1    bullet shells get counted blood dries votive c...
2    Walt Disney Bambi opened 1942 critics praised ...
3    Death may great equalizer necessarily evenhand...
4    SEOUL South Korea North Korea leader Kim said ...
```

FEATURES EXTRACTION:

After preprocessing, tokens were converted into numerical representations using the Bag of Words to create understandable features for the model.



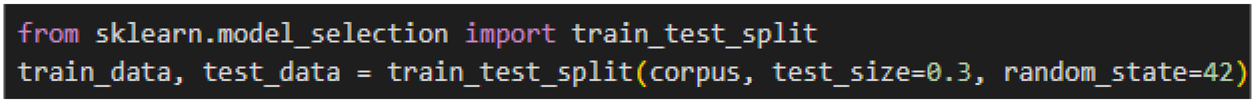
```
# Create a dictionary from the preprocessed data
dictionary = corpora.Dictionary(data_lemma)

# Bag of Words
corpus = [dictionary.doc2bow(doc) for doc in data_lemma]

# Create a dictionary from the preprocessed data
dictionary = corpora.Dictionary(data_lemma)

# Bag of Words
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data_lemma]

print(doc_term_matrix[:2])
```

MODEL TRAINING & TESTING:

The dataset is split into training and testing sets using a 70-30 ratio. The LDA model is trained on the training data and evaluated on the test data.

Train-Test Split:

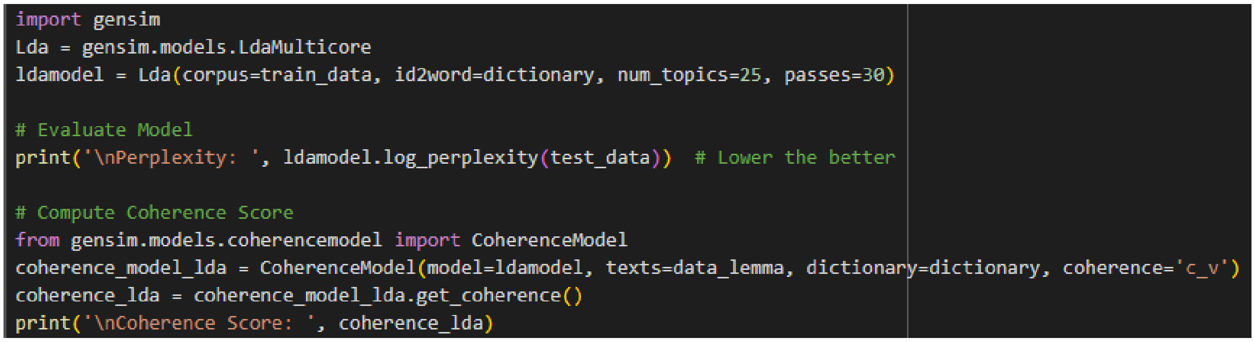
The data after conversion and encoding into the corpus is divided into training and testing sets with a 70-30 split.



```
from sklearn.model_selection import train_test_split
train_data, test_data = train_test_split(corpus, test_size=0.3, random_state=42)
```

LDA Model Training:

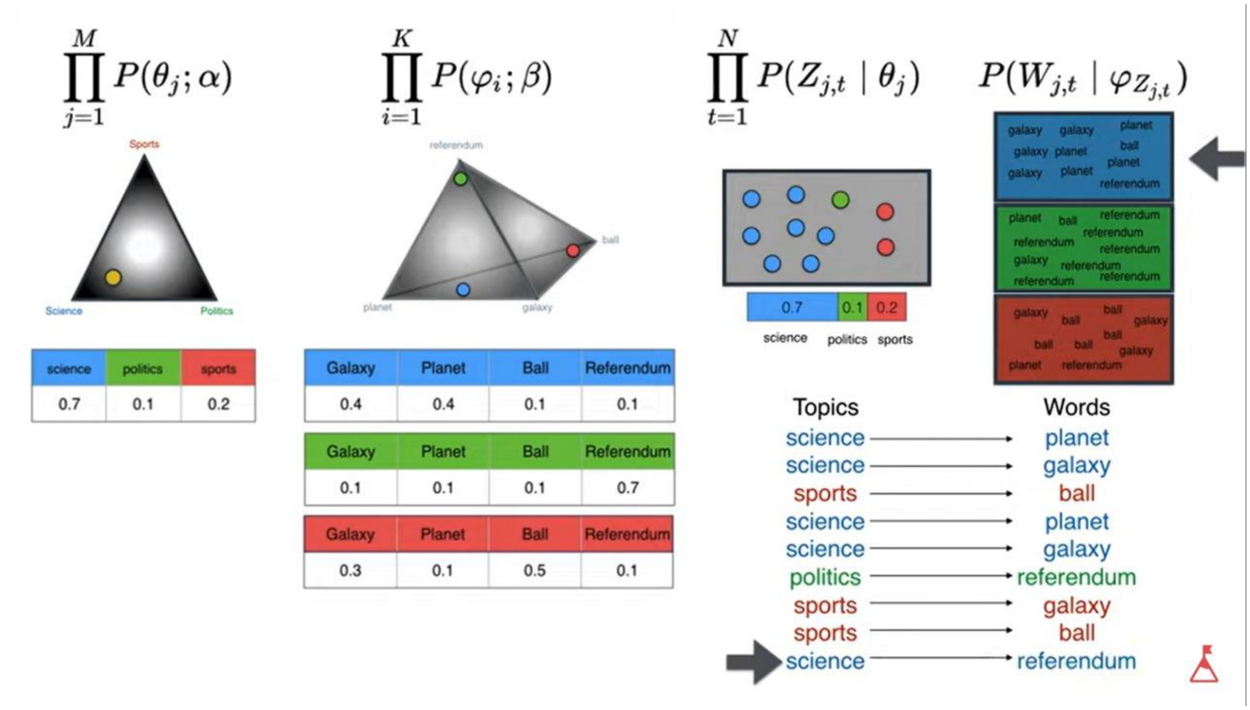
- **Algorithm**: Latent Dirichlet Allocation (LDA) using Gensim's LdaMulticore.
- **Parameters**:
  - **num_topics=25**: Number of topics to extract.
  - **passes=30**: Number of passes through the corpus during training.



```
import gensim
Lda = gensim.models.LdaMulticore
ldamodel = Lda(corpus=train_data, id2word=dictionary, num_topics=25, passes=30)

# Evaluate Model
print('\nPerplexity: ', ldamodel.log_perplexity(test_data))  # Lower the better

# Compute Coherence Score
from gensim.models.coherencemodel import CoherenceModel
coherence_model_lda = CoherenceModel(model=ldamodel, texts=data_lemma, dictionary=dictionary, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

Model Evaluation Results:

- **Perplexity**: -8.188782166071922
- **Coherence Score**: 0.5330073028833112



$\prod_{j=1}^{M} P(\theta_j;\alpha)$
$\prod_{i=1}^{K} P(\varphi_i;\beta)$
$\prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j)$
$P(W_{j,t} \mid \varphi_{Z_{j,t}})$
Sports
Science
Politics
referendum
ball
planet
galaxy
galaxy galaxy planet galaxy planet ball planet galaxy planet referendum
planet ball referendum referendum referendum referendum galaxy referendum referendum referendum
galaxy ball ball galaxy ball ball ball galaxy planet referendum
0.7
0.1
0.2
science
politics
sports

science	politics	sports
0.7	0.1	0.2

Galaxy	Planet	Ball	Referendum
0.4	0.4	0.1	0.1

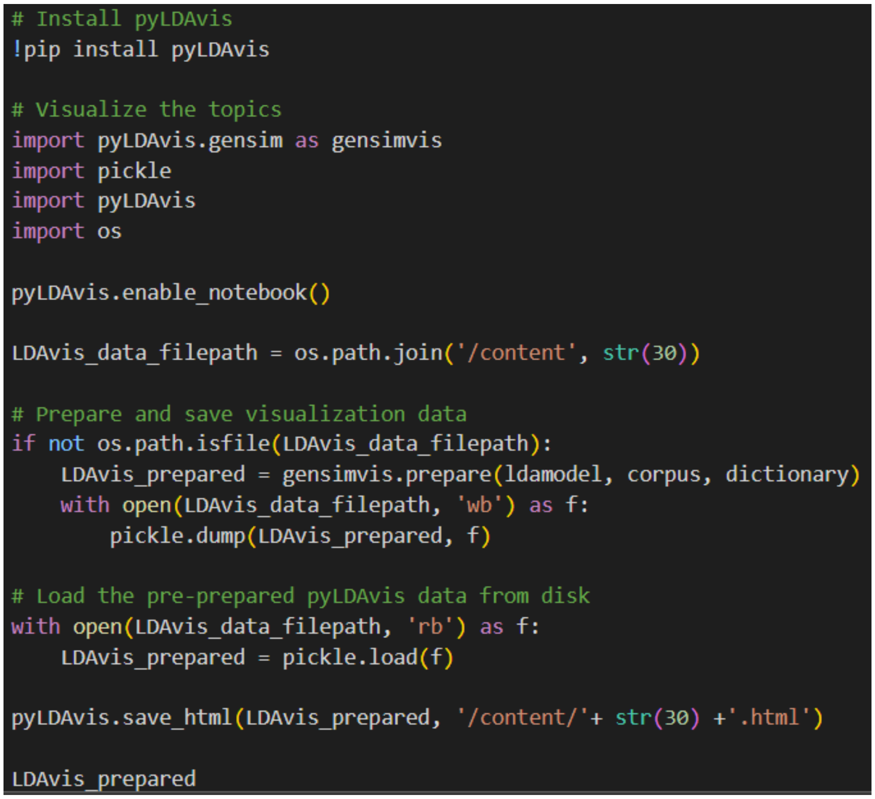
Galaxy	Planet	Ball	Referendum
0.1	0.1	0.1	0.7

Galaxy	Planet	Ball	Referendum
0.3	0.1	0.5	0.1

Topics	Words
science	planet
science	galaxy
sports	ball
science	planet
science	galaxy
politics	referendum
sports	galaxy
sports	ball
science	referendum

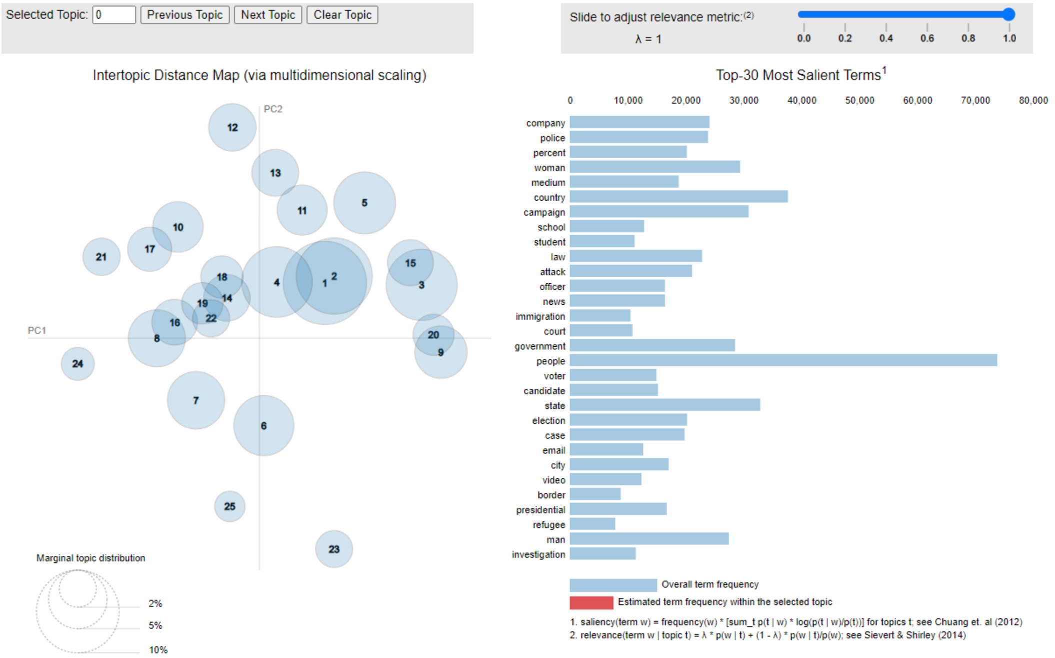
RESULTS VISUALIZATION:

The topics are visualized using PyLDAvis to provide insights into the model's findings.



```
# Install pyLDAvis
!pip install pyLDAvis

# Visualize the topics
import pyLDAvis.gensim as gensimvis
import pickle
import pyLDAvis
import os

pyLDAvis.enable_notebook()

LDAvis_data_filepath = os.path.join('/content', str(30))

# Prepare and save visualization data
if not os.path.isfile(LDAvis_data_filepath):
    LDAvis_prepared = gensimvis.prepare(ldamodel, corpus, dictionary)
    with open(LDAvis_data_filepath, 'wb') as f:
        pickle.dump(LDAvis_prepared, f)

# Load the pre-prepared pyLDAvis data from disk
with open(LDAvis_data_filepath, 'rb') as f:
    LDAvis_prepared = pickle.load(f)

pyLDAvis.save_html(LDAvis_prepared, '/content/'+ str(30) +'.html')

LDAvis_prepared
```

Visualization Interface:



Selected Topic: 0

Slide to adjust relevance metric:(2)

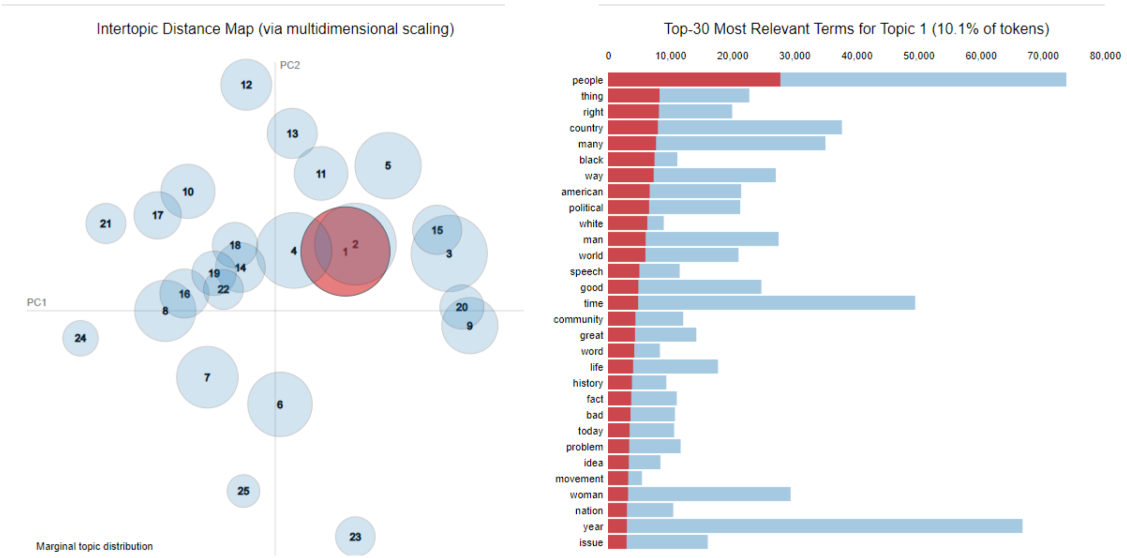
λ = 1

Intertopic Distance Map (via multidimensional scaling)

Top-30 Most Salient Terms[1]

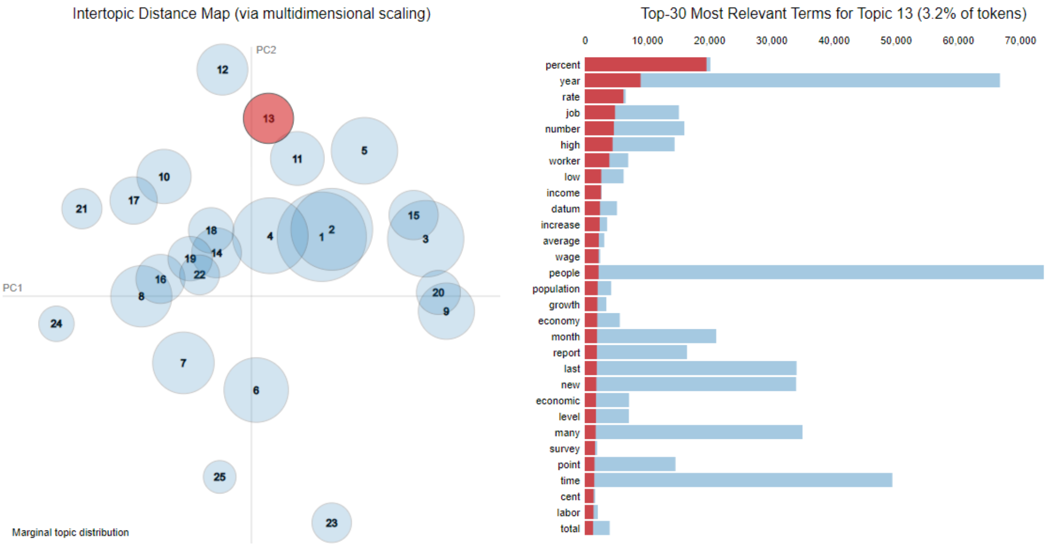
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

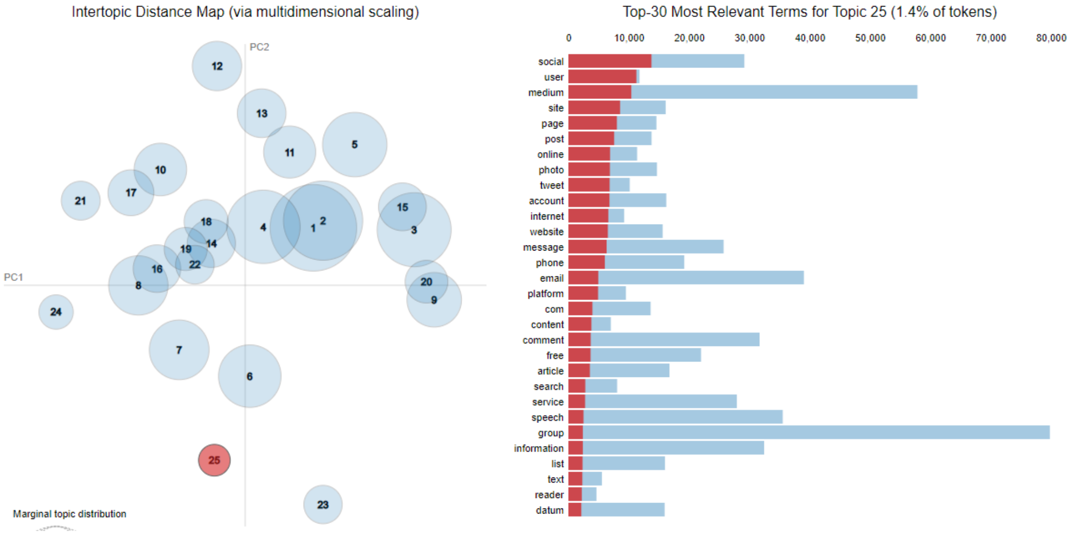
Topics Visualization:



Intertopic Distance Map (via multidimensional scaling)

Top-30 Most Relevant Terms for Topic 1 (10.1% of tokens)


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
12
13
11
5
10
17
21
18
4
1
2
15
3
19
14
22
16
8
20
9
24
7
6
25
23
Marginal topic distribution
Top-30 Most Relevant Terms for Topic 13 (3.2% of tokens)
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
percent
year
rate
job
number
high
worker
low
income
datum
increase
average
wage
people
population
growth
economy
month
report
last
new
economic
level
many
survey
point
time
cent
labor
total


Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
12
13
11
5
10
21
17
18
4
1
2
15
3
19
14
22
16
8
20
9
24
7
6
25
23
Marginal topic distribution
Top-30 Most Relevant Terms for Topic 25 (1.4% of tokens)
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
social
user
medium
site
page
post
online
photo
tweet
account
internet
website
message
phone
email
platform
com
content
comment
free
article
search
service
speech
group
information
list
text
reader
datum