In [1]:
```
#Group 8 - Wk9 - TopicModel
#Group Member: Athena Zhang, Pratheek Praveen Kumar, Weifeng Li, Wenke Yu, Ziqiao Wei
```

In [4]:
```
!pip install spacy
!pip install pyLDAvis
!pip install --upgrade gensim
```

```
Requirement already satisfied: spacy in c:\users\student\anaconda3\lib\site-packages (3.3.0)
Requirement already satisfied: jinja2 in c:\users\student\anaconda3\lib\site-packages (from spacy) (2.11.1)
Requirement already satisfied: typing-extensions<4.0.0.0,>=3.7.4; python_version < "3.8" in c:\users\student\anaconda3\lib\site-pa
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\student\anaconda3\lib\site-packages (from spacy) (3.0.6)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (4.42.1)
Requirement already satisfied: numpy>=1.15.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (1.21.6)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (2.27.1)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (1.0.2)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.9 in c:\users\student\anaconda3\lib\site-packages (from spacy) (3.0.9)
Requirement already satisfied: pathy>=0.3.5 in c:\users\student\anaconda3\lib\site-packages (from spacy) (0.6.1)
Requirement already satisfied: setuptools in c:\users\student\anaconda3\lib\site-packages (from spacy) (45.2.0.post20200210)
Requirement already satisfied: typer<0.5.0,>=0.3.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (0.4.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (3.3.0)
Requirement already satisfied: packaging>=20.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (21.3)
Requirement already satisfied: thinc<8.1.0,>=8.0.14 in c:\users\student\anaconda3\lib\site-packages (from spacy) (8.0.15)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (1.0.7)
Requirement already satisfied: wasabi<1.1.0,>=0.9.1 in c:\users\student\anaconda3\lib\site-packages (from spacy) (0.9.1)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\student\anaconda3\lib\site-packages (from spacy) (2.0.6)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\student\anaconda3\lib\site-packages (from spacy) (2.0.7)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\student\anaconda3\lib\site-packages (from spacy) (2.4.3)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.9.0,>=1.7.4 in c:\users\student\anaconda3\lib\site-packages (from spacy)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in c:\users\student\anaconda3\lib\site-packages (from spacy) (0.7.7)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\student\anaconda3\lib\site-packages (from jinja2->spacy) (1.1.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\student\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->s
Requirement already satisfied: idna<4,>=2.5; python_version >= "3" in c:\users\student\anaconda3\lib\site-packages (from requests<
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\student\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0
Requirement already satisfied: charset-normalizer~=2.0.0; python_version >= "3" in c:\users\student\anaconda3\lib\site-packages (f
2)
Requirement already satisfied: smart-open<6.0.0,>=5.0.0 in c:\users\student\anaconda3\lib\site-packages (from pathy>=0.3.5->spacy
Requirement already satisfied: click<9.0.0,>=7.1.1 in c:\users\student\anaconda3\lib\site-packages (from typer<0.5.0,>=0.3.0->spac
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\student\anaconda3\lib\site-packages (from packaging>=20.0->spa
Requirement already satisfied: zipp>=0.5; python_version < "3.8" in c:\users\student\anaconda3\lib\site-packages (from catalogue<2
Requirement already satisfied: colorama; platform_system == "Windows" in c:\users\student\anaconda3\lib\site-packages (from click<
(0.4.3)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in c:\users\student\anaconda3\lib\site-packages (from cl
cy) (3.10.1)
Requirement already satisfied: pyLDAvis in c:\users\student\anaconda3\lib\site-packages (3.3.1)
Requirement already satisfied: funcy in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (1.17)
Requirement already satisfied: numpy>=1.20.0 in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (1.21.6)
Requirement already satisfied: jinja2 in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (2.11.1)
Requirement already satisfied: future in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (0.18.2)
Requirement already satisfied: joblib in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (0.14.1)
Requirement already satisfied: sklearn in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (0.0)
Requirement already satisfied: numexpr in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (2.7.1)
Requirement already satisfied: pandas>=1.2.0 in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (1.3.5)
Requirement already satisfied: gensim in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (4.2.0)
Requirement already satisfied: scipy in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (1.4.1)
Requirement already satisfied: setuptools in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (45.2.0.post20200210)
Requirement already satisfied: scikit-learn in c:\users\student\anaconda3\lib\site-packages (from pyLDAvis) (1.0.2)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\student\anaconda3\lib\site-packages (from jinja2->pyLDAvis) (1.1.1)
Requirement already satisfied: pytz>=2017.3 in c:\users\student\anaconda3\lib\site-packages (from pandas>=1.2.0->pyLDAvis) (2019.3
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\student\anaconda3\lib\site-packages (from pandas>=1.2.0->pyLDAvi
Requirement already satisfied: smart-open>=1.8.1 in c:\users\student\anaconda3\lib\site-packages (from gensim->pyLDAvis) (5.2.1)
Requirement already satisfied: Cython==0.29.28 in c:\users\student\anaconda3\lib\site-packages (from gensim->pyLDAvis) (0.29.28)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\student\anaconda3\lib\site-packages (from scikit-learn->pyLDAvis)
Requirement already satisfied: six>=1.5 in c:\users\student\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas>=1.2
Requirement already up-to-date: gensim in c:\users\student\anaconda3\lib\site-packages (4.2.0)
Requirement already satisfied, skipping upgrade: smart-open>=1.8.1 in c:\users\student\anaconda3\lib\site-packages (from gensim)
Requirement already satisfied, skipping upgrade: numpy>=1.17.0 in c:\users\student\anaconda3\lib\site-packages (from gensim) (1.21
Requirement already satisfied, skipping upgrade: Cython==0.29.28 in c:\users\student\anaconda3\lib\site-packages (from gensim) (0
Requirement already satisfied, skipping upgrade: scipy>=0.18.1 in c:\users\student\anaconda3\lib\site-packages (from gensim) (1.4
```

In [5]:
```
import re
import numpy as np
import pandas as pd
import nltk
import sklearn
from pprint import pprint
from sklearn.feature_extraction.text import CountVectorizer

# Gensim
import gensim
import gensim.corpora as corpora
from gensim.utils import simple_preprocess
from gensim.models import CoherenceModel

# spacy for lemmatization
import spacy

# Plotting tools
import pyLDAvis
import pyLDAvis.gensim_models as plt_gensim # don't skip this
import matplotlib.pyplot as plt
%matplotlib inline

# Enable logging for gensim - optional
import logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.ERROR)

import warnings
warnings.filterwarnings("ignore",category=DeprecationWarning)
```

```
C:\Users\student\anaconda3\lib\site-packages\nltk\decorators.py:68: DeprecationWarning: `formatargspec` is deprecated since Python
bject directly
  regargs, varargs, varkwargs, defaults, formatvalue=lambda value: ""
C:\Users\student\anaconda3\lib\site-packages\nltk\lm\counter.py:15: DeprecationWarning: Using or importing the ABCs from 'collecti
eprecated since Python 3.3,and in 3.9 it will stop working
  from collections import Sequence, defaultdict
C:\Users\student\anaconda3\lib\site-packages\nltk\lm\vocabulary.py:13: DeprecationWarning: Using or importing the ABCs from 'colle
s deprecated since Python 3.3,and in 3.9 it will stop working
  from collections import Counter, Iterable
C:\Users\student\anaconda3\lib\site-packages\scipy\io\matlab\mio5.py:98: DeprecationWarning: `np.bool` is a deprecated alias for t
use `bool` by itself. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  from .mio5_utils import VarReader5
```

In [7]:
```
# NLTK Stop words
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\student\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [8]:
```
# Import Dataset
df = pd.read_json('https://raw.githubusercontent.com/selva86/datasets/master/newsgroups.json')
print(df.target_names.unique())
df.head()
```

```
['rec.autos' 'comp.sys.mac.hardware' 'comp.graphics' 'sci.space'
 'talk.politics.guns' 'sci.med' 'comp.sys.ibm.pc.hardware'
 'comp.os.ms-windows.misc' 'rec.motorcycles' 'talk.religion.misc'
 'misc.forsale' 'alt.atheism' 'sci.electronics' 'comp.windows.x'
 'rec.sport.hockey' 'rec.sport.baseball' 'soc.religion.christian'
 'talk.politics.mideast' 'talk.politics.misc' 'sci.crypt']
```

Out[8]:

	content	target	target_names
0	From: lerxst@wam.umd.edu (where's my thing)\nS...	7	rec.autos
1	From: guykuo@carson.u.washington.edu (Guy Kuo)...	4	comp.sys.mac.hardware
2	From: twillis@ec.ecn.purdue.edu (Thomas E Will...	4	comp.sys.mac.hardware
3	From: jgreen@amber (Joe Green)\nSubject: Re: W...	1	comp.graphics
4	From: jcm@head-cfa.harvard.edu (Jonathan McDow...	14	sci.space

In [9]:
```
# Convert to list
data = df.content.values.tolist()

# Remove Emails
data = [re.sub('\S*@\S*\s?', '', sent) for sent in data]

# Remove new line characters
data = [re.sub('\s+', ' ', sent) for sent in data]

# Remove distracting single quotes
data = [re.sub("\'", "", sent) for sent in data]

pprint(data[:1])
```

```
['From: (wheres my thing) Subject: WHAT car is this!? Nntp-Posting-Host: '
 'rac3.wam.umd.edu Organization: University of Maryland, College Park Lines: '
 '15 I was wondering if anyone out there could enlighten me on this car I saw '
 'the other day. It was a 2-door sports car, looked to be from the late 60s/ '
 'early 70s. It was called a Bricklin. The doors were really small. In '
 'addition, the front bumper was separate from the rest of the body. This is '
 'all I know. If anyone can tellme a model name, engine specs, years of '
 'production, where this car is made, history, or whatever info you have on '
 'this funky looking car, please e-mail. Thanks, - IL ---- brought to you by '
 'your neighborhood Lerxst ---- ']
```

In [10]:
```
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True removes punctuations

data_words = list(sent_to_words(data))

print(data_words[:1])
```

```
[['from', 'wheres', 'my', 'thing', 'subject', 'what', 'car', 'is', 'this', 'nntp', 'posting', 'host', 'rac', 'wam', 'umd', 'edu',
nd', 'college', 'park', 'lines', 'was', 'wondering', 'if', 'anyone', 'out', 'there', 'could', 'enlighten', 'me', 'on', 'this', 'ca
s', 'door', 'sports', 'car', 'looked', 'to', 'be', 'from', 'the', 'late', 'early', 'it', 'was', 'called', 'bricklin', 'the', 'door
ion', 'the', 'front', 'bumper', 'was', 'separate', 'from', 'the', 'rest', 'of', 'the', 'body', 'this', 'is', 'all', 'know', 'if',
'engine', 'specs', 'years', 'of', 'production', 'where', 'this', 'car', 'is', 'made', 'history', 'or', 'whatever', 'info', 'you',
'car', 'please', 'mail', 'thanks', 'il', 'brought', 'to', 'you', 'by', 'your', 'neighborhood', 'lerxst']]
```

In [11]:
```
# Build the bigram and trigram models
bigram = gensim.models.Phrases(data_words, min_count=5, threshold=100) # higher threshold fewer phrases.
trigram = gensim.models.Phrases(bigram[data_words], threshold=100)

# Faster way to get a sentence clubbed as a trigram/bigram
bigram_mod = gensim.models.phrases.Phraser(bigram)
trigram_mod = gensim.models.phrases.Phraser(trigram)

# See trigram example
print(trigram_mod[bigram_mod[data_words[0]]])
```

```
['from', 'wheres', 'my', 'thing', 'subject', 'what', 'car', 'is', 'this', 'nntp_posting_host', 'rac_wam_umd_edu', 'organization',
k', 'lines', 'was', 'wondering', 'if', 'anyone', 'out', 'there', 'could', 'enlighten', 'me', 'on', 'this', 'car', 'saw', 'the', 'c
s', 'car', 'looked', 'to', 'be', 'from', 'the', 'late', 'early', 'it', 'was', 'called', 'bricklin', 'the', 'doors', 'were', 'reall
t_bumper', 'was', 'separate', 'from', 'the', 'rest', 'of', 'the', 'body', 'this', 'is', 'all', 'know', 'if', 'anyone', 'can', 'tel
'years', 'of', 'production', 'where', 'this', 'car', 'is', 'made', 'history', 'or', 'whatever', 'info', 'you', 'have', 'on', 'this
il', 'thanks', 'il', 'brought', 'to', 'you', 'by', 'your', 'neighborhood', 'lerxst']
```

In [12]:
```
# Define functions for stopwords, bigrams, trigrams and lemmatization
def remove_stopwords(texts):
    return [[word for word in simple_preprocess(str(doc)) if word not in stop_words] for doc in texts]

def make_bigrams(texts):
    return [bigram_mod[doc] for doc in texts]

def make_trigrams(texts):
    return [trigram_mod[bigram_mod[doc]] for doc in texts]

def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']):
    """https://spacy.io/api/annotation"""
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append([token.lemma_ for token in doc if token.pos_ in allowed_postags])
    return texts_out
```

In [14]:
```
# Remove Stop Words
data_words_nostops = remove_stopwords(data_words)

# Form Bigrams
data_words_bigrams = make_bigrams(data_words_nostops)

# Initialize spacy 'en_core_web_sm' model, keeping only tagger component (for efficiency)
# python3 -m spacy download en
nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])

# Do lemmatization keeping only noun, adj, vb, adv
data_lemmatized = lemmatization(data_words_bigrams, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV'])

print(data_lemmatized[:1])
```

```
[['s', 'thing', 'car', 'nntp_poste', 'host', 'umd', 'organization', 'park', 'line', 'wonder', 'enlighten', 'car', 'see', 'day', 'c
y', 'call', 'bricklin', 'door', 'really', 'small', 'addition', 'separate', 'rest', 'body', 'know', 'tellme', 'model', 'name', 'eng
'make', 'history', 'info', 'funky', 'look', 'car', 'mail', 'thank', 'bring', 'neighborhood', 'lerxst']]
```

In [15]:
```
# Create Dictionary
id2word = corpora.Dictionary(data_lemmatized)

# Create Corpus
texts = data_lemmatized

# Term Document Frequency
corpus = [id2word.doc2bow(text) for text in texts]

# View
print(corpus[:1])
```

```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 5), (6, 1), (7, 2), (8, 1), (9, 1), (10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15
(20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1), (29, 1), (30, 1), (31, 1), (32, 1), (33, 1), (34
9, 1), (40, 1), (41, 1), (42, 1)]]
```

In [16]:
```
# Build LDA model for 10 topics
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=10,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

In [17]:
```
# Print the Keyword in the 10 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.643*"ax" + 0.013*"nhl" + 0.011*"gateway" + 0.007*"slightly" + '
  '0.006*"sector" + 0.005*"daughter" + 0.005*"human_being" + 0.004*"frequency" '
  '+ 0.004*"stream" + 0.004*"landing"'),
 (1,
  '0.035*"key" + 0.028*"_" + 0.015*"physical" + 0.012*"chip" + 0.012*"ripem" + '
  '0.012*"system" + 0.012*"public" + 0.010*"encryption" + 0.010*"use" + '
  '0.009*"tape"'),
 (2,
  '0.018*"say" + 0.015*"people" + 0.012*"think" + 0.011*"write" + 0.010*"make" '
  '+ 0.009*"know" + 0.008*"go" + 0.007*"see" + 0.007*"believe" + '
  '0.007*"reason"'),
 (3,
  '0.033*"team" + 0.031*"game" + 0.023*"year" + 0.022*"play" + 0.018*"win" + '
  '0.015*"player" + 0.011*"season" + 0.010*"go" + 0.009*"good" + 0.008*"fan"'),
 (4,
  '0.029*"sex" + 0.024*"rlk" + 0.023*"marriage" + 0.021*"moral" + '
  '0.020*"morality" + 0.013*"immoral" + 0.012*"solely" + 0.011*"evolution" + '
  '0.010*"confusion" + 0.010*"sexual"'),
 (5,
  '0.044*"gun" + 0.021*"fire" + 0.018*"drug" + 0.016*"kill" + 0.016*"crime" + '
  '0.014*"weapon" + 0.013*"tv" + 0.012*"safety" + 0.011*"child" + '
  '0.011*"firearm"'),
 (6,
  '0.013*"government" + 0.010*"year" + 0.010*"space" + 0.010*"state" + '
  '0.006*"war" + 0.006*"attack" + 0.006*"israeli" + 0.006*"kill" + '
  '0.006*"greek" + 0.006*"soldier"'),
 (7,
  '0.017*"hockey" + 0.016*"study" + 0.015*"discussion" + 0.014*"science" + '
  '0.010*"wing" + 0.009*"eat" + 0.009*"scientific" + 0.008*"material" + '
  '0.008*"food" + 0.008*"treatment"'),
 (8,
  '0.029*"line" + 0.016*"organization" + 0.015*"get" + 0.015*"write" + '
  '0.012*"nntp_poste" + 0.011*"article" + 0.011*"host" + 0.007*"m" + '
  '0.007*"work" + 0.007*"know"'),
 (9,
  '0.025*"program" + 0.024*"file" + 0.019*"window" + 0.013*"software" + '
  '0.012*"include" + 0.012*"copy" + 0.011*"image" + 0.011*"version" + '
  '0.011*"available" + 0.010*"entry"')]
```

In [18]:
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

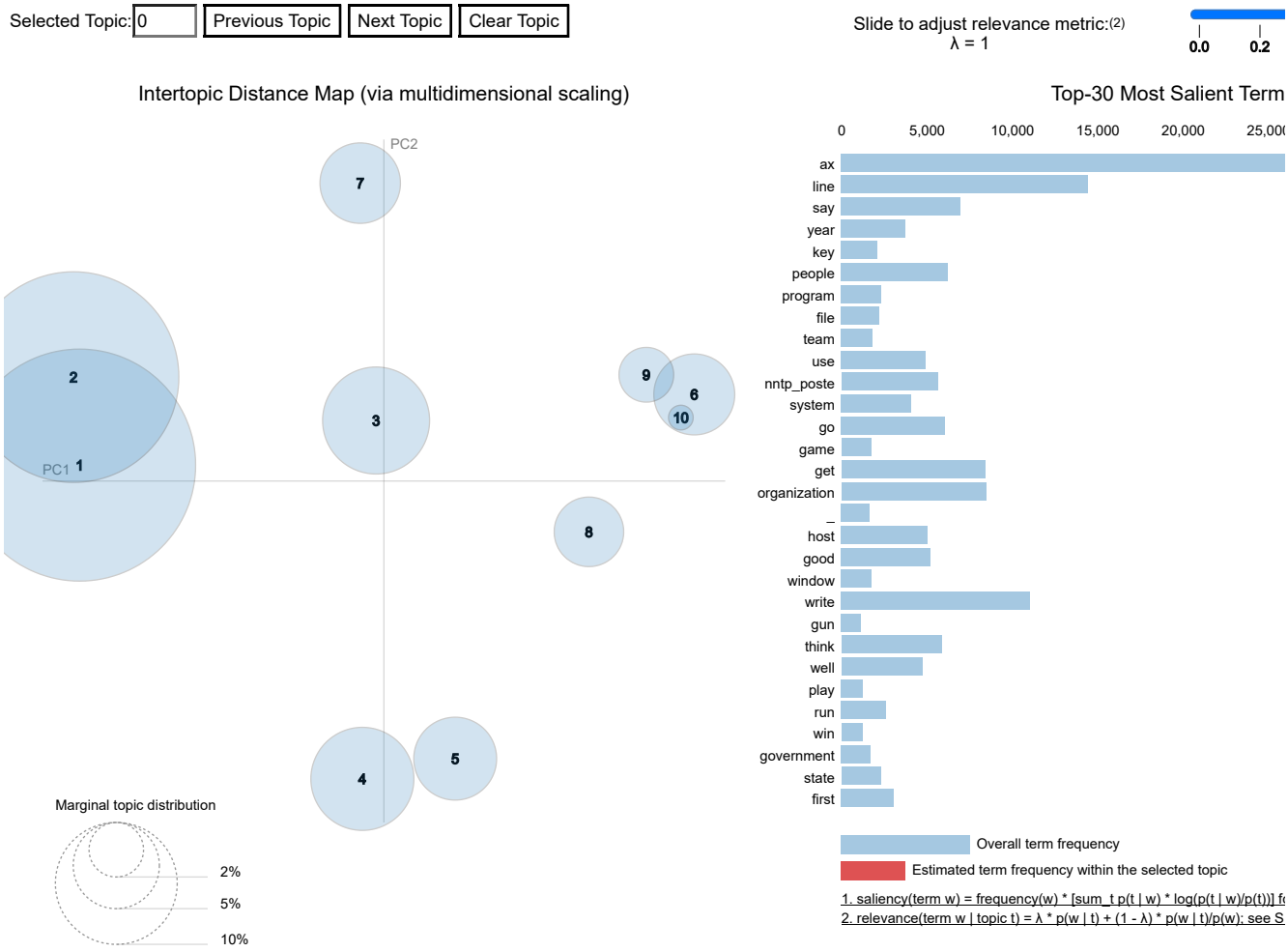
```
Perplexity:  -8.784448166327094

Coherence Score:  0.5490559003581944
```

Q1. How do the measures of perplexity and coherence change as we change the number of topics from 5
topic model based on your judgement and assign topic labels

In [20]:
```
# Visualize the topics for 10 topics
pyLDAvis.enable_notebook()
vis = plt_gensim.prepare(lda_model, corpus, id2word)
vis
```

```
C:\Users\student\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:247: FutureWarning: In a future version of pandas all arguments
'labels' will be keyword-only
  by='saliency', ascending=False).head(R).drop('saliency', 1)
```

Out[20]:



In [21]:
```
# Build LDA model for 5 topics
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=5,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

In [22]:
```
# Print the Keyword in the 5 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.019*"space" + 0.007*"research" + 0.007*"dn" + 0.007*"earth" + '
  '0.007*"launch" + 0.006*"orbit" + 0.006*"science" + 0.006*"mission" + '
  '0.005*"moon" + 0.005*"year"'),
 (1,
  '0.017*"line" + 0.009*"use" + 0.008*"organization" + 0.008*"system" + '
  '0.007*"get" + 0.007*"nntp_poste" + 0.007*"write" + 0.007*"host" + '
  '0.006*"need" + 0.006*"drive"'),
 (2,
  '0.011*"people" + 0.010*"say" + 0.009*"write" + 0.007*"think" + 0.006*"make" '
  '+ 0.006*"article" + 0.006*"know" + 0.006*"line" + 0.005*"believe" + '
  '0.005*"reason"'),
 (3,
  '0.015*"line" + 0.014*"go" + 0.013*"get" + 0.012*"write" + 0.010*"article" + '
  '0.010*"organization" + 0.008*"year" + 0.008*"good" + 0.008*"nntp_poste" + '
  '0.007*"think"'),
 (4,
  '0.488*"ax" + 0.040*"_" + 0.015*"c" + 0.008*"cx" + 0.006*"rlk" + 0.004*"m" + '
  '0.003*"mf" + 0.003*"nei" + 0.002*"mu" + 0.002*"r"')]
```

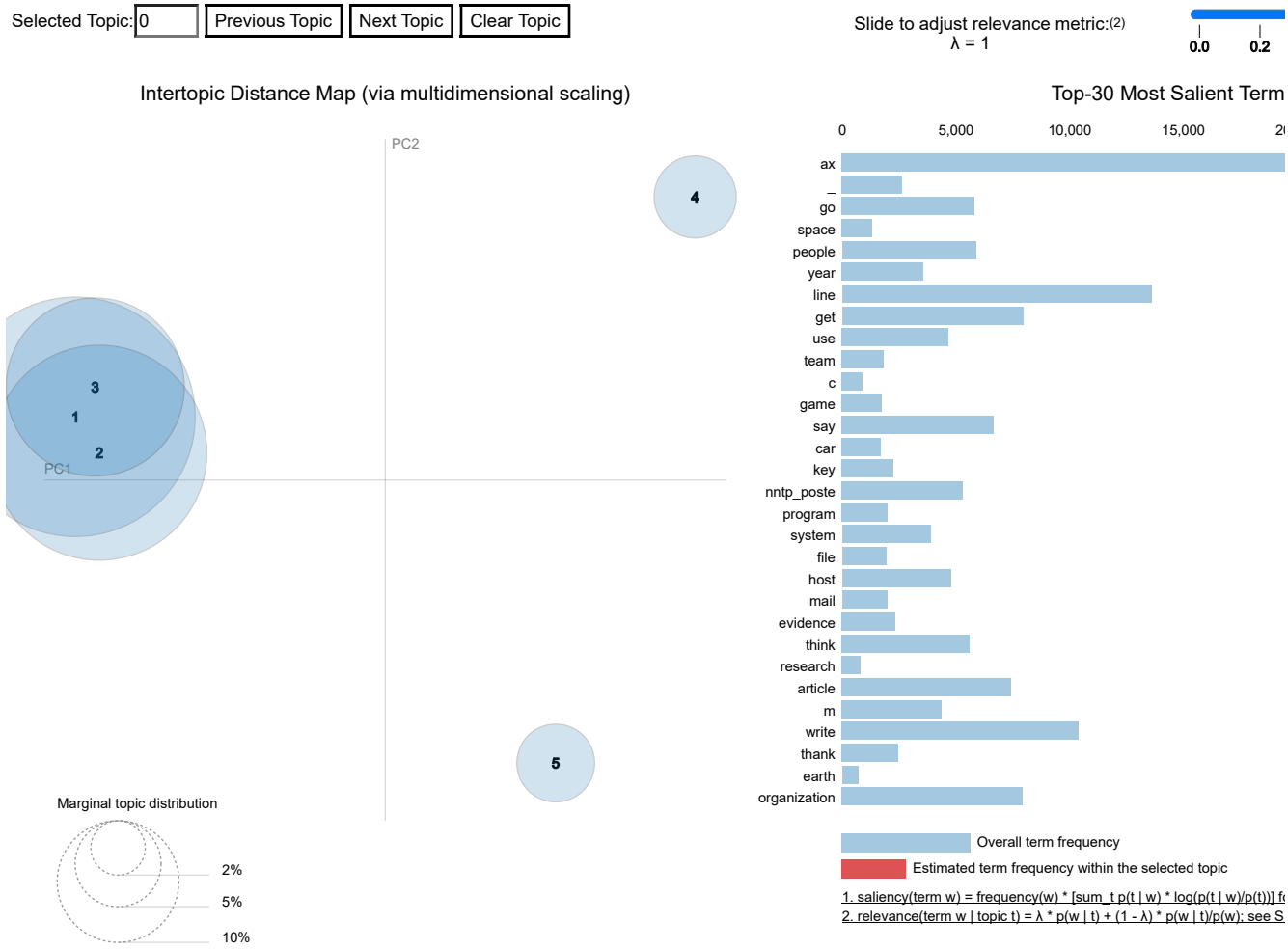
In [23]:
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -8.10490910962846

Coherence Score:  0.5405226390869472
```

In [24]:
```
# Visualize the topics for 5 topics
pyLDAvis.enable_notebook()
vis = plt_gensim.prepare(lda_model, corpus, id2word)
vis
```

```
C:\Users\student\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:247: FutureWarning: In a future version of pandas all arguments
'labels' will be keyword-only
  by='saliency', ascending=False).head(R).drop('saliency', 1)
```

Out[24]:



In [25]:
```
# Build LDA model for 15 topics
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=15,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

In [26]:
```
# Print the Keyword in the 15 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.806*"ax" + 0.005*"mf" + 0.003*"micro" + 0.003*"wm" + 0.003*"sister" + '
  '0.002*"tail" + 0.002*"openwindow" + 0.002*"iron" + 0.001*"mw" + 0.001*"m"'),
 (1,
  '0.050*"key" + 0.029*"public" + 0.027*"gun" + 0.018*"law" + '
  '0.017*"government" + 0.015*"encryption" + 0.013*"security" + 0.013*"tape" + '
  '0.012*"private" + 0.012*"use"'),
 (2,
  '0.091*"_" + 0.047*"test" + 0.034*"graphic" + 0.030*"server" + 0.029*"c" + '
  '0.022*"mark" + 0.019*"clearly" + 0.019*"motif" + 0.019*"corporation" + '
  '0.018*"telnet"'),
 (3,
  '0.058*"team" + 0.056*"game" + 0.040*"play" + 0.039*"win" + 0.036*"year" + '
  '0.029*"physical" + 0.027*"player" + 0.023*"hockey" + 0.019*"season" + '
  '0.015*"chance"'),
 (4,
  '0.065*"headache" + 0.036*"super" + 0.025*"film" + 0.022*"brand" + '
  '0.008*"grip" + 0.007*"readily" + 0.006*"thumb" + 0.005*"mailing" + '
  '0.004*"nm" + 0.004*"infinity"'),
 (5,
  '0.049*"plane" + 0.046*"ok" + 0.036*"cambridge" + 0.033*"bug" + 0.031*"warn" '
  '+ 0.025*"trivial" + 0.017*"plot" + 0.015*"angle" + 0.012*"diameter" + '
  '0.008*"rat"'),
 (6,
  '0.024*"state" + 0.021*"government" + 0.013*"war" + 0.013*"report" + '
  '0.013*"israeli" + 0.012*"year" + 0.012*"city" + 0.011*"force" + '
  '0.010*"member" + 0.010*"attack"'),
 (7,
  '0.042*"treatment" + 0.039*"trust" + 0.037*"doctor" + 0.035*"vote" + '
  '0.031*"circuit" + 0.031*"ground" + 0.028*"militia" + 0.024*"zone" + '
  '0.022*"cap" + 0.021*"band"'),
 (8,
  '0.033*"line" + 0.019*"organization" + 0.015*"get" + 0.015*"write" + '
  '0.013*"nntp_poste" + 0.012*"host" + 0.011*"article" + 0.009*"work" + '
  '0.008*"use" + 0.008*"need"'),
 (9,
  '0.041*"program" + 0.039*"file" + 0.031*"window" + 0.021*"software" + '
  '0.018*"card" + 0.018*"image" + 0.018*"version" + 0.017*"driver" + '
  '0.016*"entry" + 0.016*"available"'),
 (10,
  '0.032*"christian" + 0.026*"faith" + 0.024*"reason" + 0.022*"exist" + '
  '0.020*"sense" + 0.020*"believe" + 0.020*"god" + 0.019*"religion" + '
  '0.017*"law" + 0.015*"claim"'),
 (11,
  '0.026*"say" + 0.022*"people" + 0.021*"think" + 0.017*"go" + 0.017*"write" + '
  '0.016*"make" + 0.015*"know" + 0.014*"see" + 0.012*"come" + 0.011*"article"'),
 (12,
  '0.066*"wire" + 0.043*"nhl" + 0.039*"blank" + 0.032*"cable" + '
  '0.031*"warranty" + 0.029*"pro" + 0.021*"quick" + 0.020*"expansion" + '
  '0.020*"quadra" + 0.020*"probe"'),
 (13,
  '0.103*"space" + 0.066*"science" + 0.046*"church" + 0.041*"earth" + '
  '0.029*"launch" + 0.027*"wing" + 0.027*"scientific" + 0.026*"orbit" + '
  '0.025*"mission" + 0.024*"moon"'),
 (14,
  '0.041*"kill" + 0.036*"woman" + 0.033*"soldier" + 0.032*"greek" + '
  '0.030*"armenian" + 0.029*"village" + 0.024*"murder" + 0.023*"turk" + '
  '0.021*"turkish" + 0.020*"muslim"')]
```

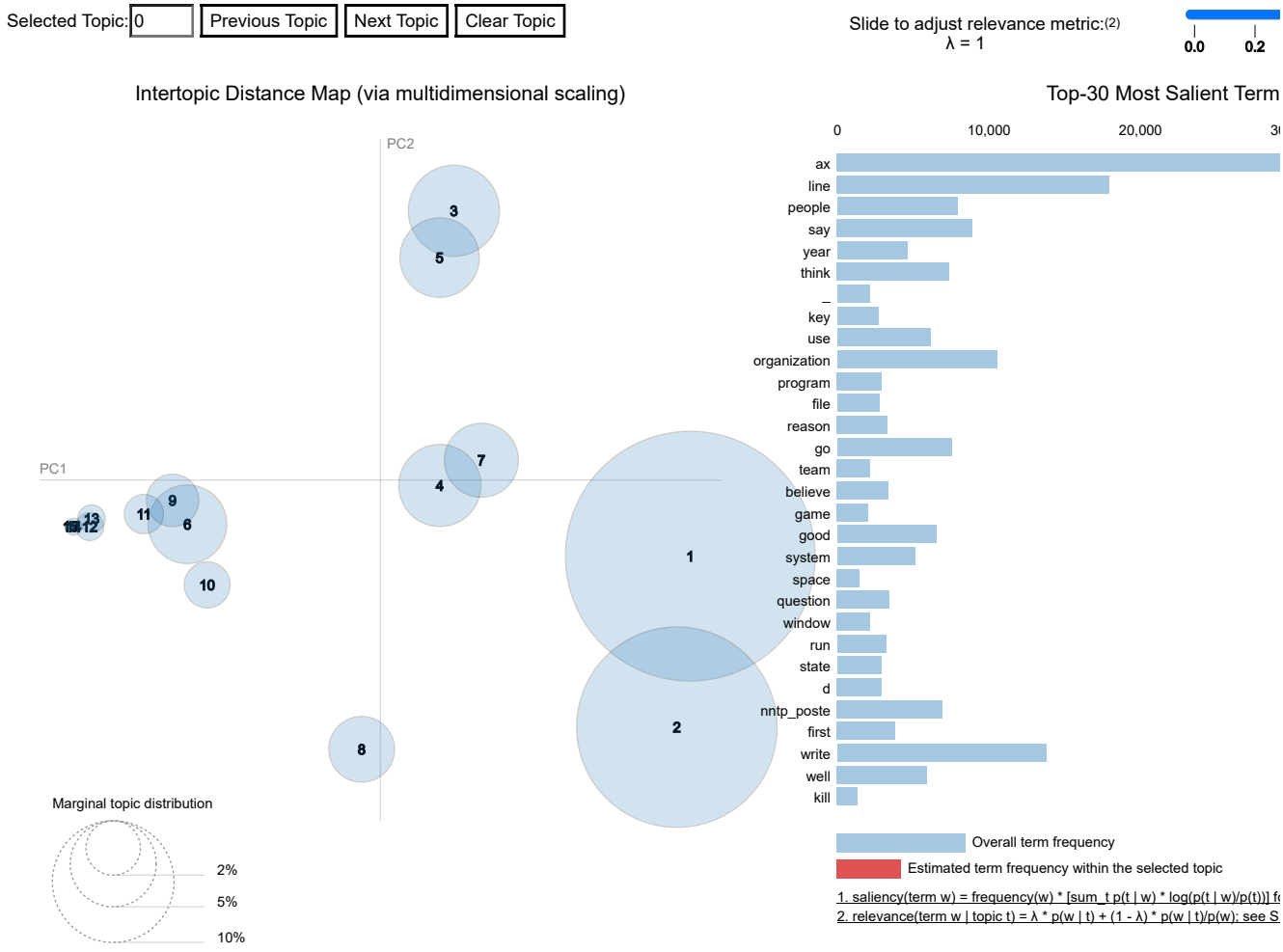
In [27]:
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. Lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -11.986243654509044

Coherence Score:  0.5265270312113319
```

In [28]:
```
# Visualize the topics for 15 topics
pyLDAvis.enable_notebook()
vis = plt_gensim.prepare(lda_model, corpus, id2word)
vis
```

```
C:\Users\student\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:247: FutureWarning: In a future version of pandas all arguments
'labels' will be keyword-only
  by='saliency', ascending=False).head(R).drop('saliency', 1)
```

Out[28]:



In [29]:
```
# Build LDA model for 20 topics
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=20,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

In [30]:
```
# Print the Keyword in the 20 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.064*"nhl" + 0.059*"recommend" + 0.051*"gateway" + 0.039*"flight" + '
  '0.031*"fuel" + 0.027*"floor" + 0.024*"bank" + 0.018*"space_station" + '
  '0.017*"phase" + 0.017*"qualified"'),
 (1,
  '0.095*"key" + 0.040*"physical" + 0.037*"public" + 0.028*"encryption" + '
  '0.027*"chip" + 0.025*"security" + 0.022*"private" + 0.021*"master" + '
  '0.020*"government" + 0.018*"clipper"'),
 (2,
  '0.028*"believe" + 0.025*"evidence" + 0.023*"reason" + 0.018*"say" + '
  '0.017*"claim" + 0.015*"christian" + 0.015*"sense" + 0.013*"exist" + '
  '0.012*"fact" + 0.012*"faith"'),
 (3,
  '0.072*"team" + 0.069*"game" + 0.050*"play" + 0.048*"win" + 0.040*"year" + '
  '0.034*"player" + 0.024*"season" + 0.018*"fan" + 0.018*"goal" + 0.017*"run"'),
 (4,
  '0.093*"ide" + 0.078*"mother" + 0.040*"remind" + 0.019*"ultimate" + '
  '0.015*"winter" + 0.012*"beauty" + 0.011*"absurd" + 0.009*"grip" + '
  '0.004*"credibility" + 0.002*"stall"'),
 (5,
  '0.135*"monitor" + 0.043*"rd" + 0.034*"trivial" + 0.021*"suck" + '
  '0.009*"space_dig" + 0.009*"added_forwarde" + 0.008*"rod" + 0.004*"infinity" '
  '+ 0.004*"caution" + 0.003*"golden"'),
 (6,
  '0.033*"government" + 0.032*"state" + 0.020*"country" + 0.017*"attack" + '
  '0.016*"war" + 0.015*"israeli" + 0.013*"city" + 0.013*"greek" + '
  '0.013*"force" + 0.013*"soldier"'),
 (7,
  '0.057*"people" + 0.023*"group" + 0.020*"man" + 0.020*"say" + 0.018*"book" + '
  '0.016*"issue" + 0.014*"live" + 0.013*"name" + 0.012*"day" + 0.011*"person"'),
 (8,
  '0.021*"get" + 0.016*"go" + 0.016*"make" + 0.015*"write" + 0.015*"know" + '
  '0.014*"think" + 0.013*"good" + 0.013*"time" + 0.012*"well" + 0.012*"see"'),
 (9,
  '0.135*"line" + 0.082*"organization" + 0.061*"nntp_poste" + 0.059*"write" + '
  '0.055*"host" + 0.051*"article" + 0.028*"thank" + 0.025*"reply" + '
  '0.022*"university" + 0.017*"card"'),
 (10,
  '0.128*"graphic" + 0.067*"mount" + 0.054*"convert" + 0.042*"workstation" + '
  '0.030*"capture" + 0.024*"please_respond" + 0.021*"camera" + '
  '0.018*"positively" + 0.011*"creature" + 0.009*"weeks_ago"'),
 (11,
  '0.094*"child" + 0.049*"fire" + 0.048*"drug" + 0.031*"kid" + '
  '0.030*"corporation" + 0.026*"die" + 0.025*"trial" + 0.025*"firearm" + '
  '0.024*"boy" + 0.024*"wife"'),
 (12,
  '0.099*"law" + 0.075*"gun" + 0.029*"crime" + 0.027*"weapon" + 0.026*"murder" '
  '+ 0.025*"revelation" + 0.020*"blank" + 0.017*"death" + 0.017*"hole" + '
  '0.017*"vote"'),
 (13,
  '0.107*"space" + 0.068*"science" + 0.040*"field" + 0.038*"earth" + '
  '0.030*"launch" + 0.028*"scientific" + 0.026*"orbit" + 0.025*"mission" + '
  '0.025*"moon" + 0.021*"satellite"'),
 (14,
  '0.062*"armenian" + 0.048*"turk" + 0.042*"turkish" + 0.041*"muslim" + '
  '0.021*"islamic" + 0.020*"escape" + 0.018*"relation" + 0.017*"proceed" + '
  '0.015*"genocide" + 0.015*"turkey"'),
 (15,
  '0.070*"cop" + 0.065*"clipper_chip" + 0.048*"proposal" + 0.040*"crypto" + '
  '0.040*"export" + 0.021*"revolver" + 0.019*"police" + 0.015*"entitle" + '
  '0.009*"radar" + 0.006*"privately"'),
 (16,
  '0.124*"image" + 0.062*"format" + 0.062*"scan" + 0.056*"family" + '
  '0.049*"scsi" + 0.041*"headache" + 0.023*"quadra" + 0.021*"intel" + '
  '0.020*"utility" + 0.019*"specification"'),
 (17,
  '0.776*"ax" + 0.019*"wing" + 0.016*"direct" + 0.009*"dual" + 0.008*"quick" + '
  '0.007*"trace" + 0.006*"human_being" + 0.004*"partner" + 0.004*"dirty" + '
  '0.004*"quran"'),
 (18,
  '0.048*"drive" + 0.031*"car" + 0.022*"buy" + 0.021*"high" + 0.020*"power" + '
  '0.020*"price" + 0.018*"sell" + 0.018*"sale" + 0.018*"cost" + 0.016*"low"'),
 (19,
  '0.027*"system" + 0.024*"use" + 0.018*"program" + 0.017*"file" + '
  '0.014*"window" + 0.013*"run" + 0.012*"include" + 0.011*"available" + '
  '0.011*"information" + 0.010*"source"')]
```

In [31]:
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -13.368807240519294

Coherence Score:  0.5237564492819576
```

In [33]:
```
# Build LDA model for 25 topics
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=25,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

In [34]:
```
# Print the Keyword in the 25 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(5,
  '0.084*"armenian" + 0.066*"turk" + 0.058*"turkish" + 0.056*"muslim" + '
  '0.025*"proceed" + 0.020*"genocide" + 0.020*"escape" + 0.018*"massacre" + '
  '0.015*"slaughter" + 0.013*"kurd"'),
 (22,
  '0.163*"bike" + 0.112*"ride" + 0.061*"rider" + 0.052*"dog" + 0.029*"road" + '
  '0.013*"mad" + 0.009*"colorado_boulder" + 0.003*"golden" + '
  '0.000*"motorcycle" + 0.000*"leg"'),
 (17,
  '0.054*"dual" + 0.051*"dream" + 0.047*"task" + 0.041*"insert" + '
  '0.040*"widget" + 0.037*"camp" + 0.021*"unusual" + 0.019*"semi" + '
  '0.018*"afford" + 0.018*"stuff_delete"'),
 (14,
  '0.113*"block" + 0.104*"scan" + 0.038*"violence" + 0.038*"islamic" + '
  '0.020*"plot" + 0.019*"quran" + 0.018*"catalog" + 0.017*"mystery" + '
  '0.014*"rebel" + 0.013*"repost"'),
 (13,
  '0.117*"greek" + 0.084*"family" + 0.070*"concept" + 0.048*"militia" + '
  '0.047*"male" + 0.032*"intel" + 0.027*"regularly" + 0.022*"evolution" + '
  '0.010*"comprise" + 0.009*"explode"'),
 (7,
  '0.081*"eat" + 0.071*"treatment" + 0.064*"doctor" + 0.052*"medical" + '
  '0.038*"cap" + 0.029*"spot" + 0.028*"ray" + 0.020*"daily" + 0.020*"diet" + '
  '0.019*"severe"'),
 (4,
  '0.153*"image" + 0.119*"screen" + 0.114*"graphic" + 0.088*"package" + '
  '0.077*"format" + 0.048*"convert" + 0.007*"dimension" + 0.007*"shareware" + '
  '0.006*"utility" + 0.002*"workshop"'),
 (10,
  '0.073*"suggest" + 0.056*"weight" + 0.054*"health" + 0.047*"significant" + '
  '0.040*"dangerous" + 0.039*"associate" + 0.025*"approve" + 0.021*"impact" + '
  '0.021*"packet" + 0.021*"usage"'),
 (23,
  '0.150*"price" + 0.132*"sale" + 0.069*"sell" + 0.048*"digital" + '
  '0.039*"offer" + 0.038*"communication" + 0.029*"double" + 0.023*"expansion" '
  '+ 0.022*"cd" + 0.020*"compute"'),
 (15,
  '0.111*"gun" + 0.040*"shoot" + 0.039*"weapon" + 0.039*"police" + '
  '0.031*"proof" + 0.029*"safety" + 0.027*"carry" + 0.027*"cop" + 0.026*"vote" '
  '+ 0.025*"rate"'),
 (20,
  '0.091*"space" + 0.056*"science" + 0.049*"earth" + 0.025*"launch" + '
  '0.023*"scientific" + 0.022*"orbit" + 0.021*"mission" + 0.020*"moon" + '
  '0.017*"satellite" + 0.015*"plane"'),
 (21,
  '0.052*"fire" + 0.047*"watch" + 0.039*"throw" + 0.037*"corporation" + '
  '0.035*"city" + 0.032*"listen" + 0.028*"house" + 0.028*"night" + '
  '0.027*"past" + 0.026*"building"'),
 (0,
  '0.089*"team" + 0.085*"game" + 0.061*"play" + 0.060*"win" + 0.044*"physical" '
  '+ 0.042*"player" + 0.040*"year" + 0.029*"season" + 0.021*"wing" + '
  '0.021*"score"'),
 (24,
  '0.081*"government" + 0.076*"law" + 0.057*"state" + 0.041*"public" + '
  '0.039*"right" + 0.026*"protect" + 0.023*"crime" + 0.022*"community" + '
  '0.020*"citizen" + 0.018*"country"'),
 (6,
  '0.042*"kill" + 0.031*"attack" + 0.026*"war" + 0.025*"israeli" + '
  '0.024*"wire" + 0.022*"soldier" + 0.020*"brain" + 0.019*"village" + '
  '0.019*"jewish" + 0.018*"murder"'),
 (1,
  '0.093*"key" + 0.080*"file" + 0.033*"entry" + 0.032*"color" + '
  '0.028*"encryption" + 0.025*"tape" + 0.019*"public" + 0.019*"picture" + '
  '0.018*"request" + 0.018*"clipper"'),
 (18,
  '0.027*"group" + 0.017*"case" + 0.016*"part" + 0.014*"report" + '
  '0.014*"number" + 0.013*"order" + 0.013*"issue" + 0.013*"first" + '
  '0.012*"provide" + 0.011*"explain"'),
 (19,
  '0.031*"system" + 0.026*"use" + 0.017*"program" + 0.017*"mail" + 0.013*"run" '
  '+ 0.013*"thank" + 0.013*"window" + 0.013*"card" + 0.013*"computer" + '
  '0.011*"set"'),
 (2,
  '0.035*"say" + 0.031*"people" + 0.016*"think" + 0.014*"believe" + '
  '0.014*"reason" + 0.012*"evidence" + 0.012*"make" + 0.011*"many" + '
  '0.011*"mean" + 0.010*"point"'),
 (8,
  '0.037*"line" + 0.028*"write" + 0.022*"organization" + 0.022*"get" + '
  '0.021*"article" + 0.016*"go" + 0.015*"nntp_poste" + 0.013*"host" + '
  '0.013*"know" + 0.012*"m"')]
```

In [35]:
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -16.12467801122725

Coherence Score:  0.4654780502288751
```

In [37]:
```
# Build LDA model for 30 topics
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                            id2word=id2word,
                                            num_topics=30,
                                            random_state=100,
                                            update_every=1,
                                            chunksize=100,
                                            passes=10,
                                            alpha='auto',
                                            per_word_topics=True)
```

In [38]:
```
# Print the Keyword in the 30 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(22,
  '0.000*"defragmente" + 0.000*"spiderhulk" + 0.000*"eyelet" + 0.000*"slimko" '
  '+ 0.000*"wpfw" + 0.000*"tightness" + 0.000*"punisher" + '
  '0.000*"silver_surfer" + 0.000*"bighelmet" + 0.000*"sleepwalker"'),
 (23,
  '0.262*"entry" + 0.047*"implementation" + 0.039*"terminal" + 0.030*"edit" + '
  '0.015*"login" + 0.000*"sleepwalker" + 0.000*"defragmente" + '
  '0.000*"spiderhulk" + 0.000*"slimko" + 0.000*"silver_surfer"'),
 (16,
  '0.092*"ide" + 0.092*"controller" + 0.091*"scsi" + 0.077*"mb" + '
  '0.076*"headache" + 0.037*"electrical" + 0.025*"ftp_site" + 0.011*"tiff" + '
  '0.010*"fast" + 0.005*"nm"'),
 (13,
  '0.096*"stupid" + 0.058*"unknown" + 0.040*"programmer" + 0.034*"infinite" + '
  '0.024*"fi" + 0.017*"comprise" + 0.015*"explode" + 0.014*"mathematical" + '
  '0.013*"evolve" + 0.013*"atom"'),
 (5,
  '0.137*"normal" + 0.074*"font" + 0.055*"rd" + 0.043*"trivial" + 0.037*"co" + '
  '0.026*"click" + 0.011*"rod" + 0.000*"_" + 0.000*"eyelet" + '
  '0.000*"sleepwalker"'),
 (4,
  '0.134*"choice" + 0.037*"entity" + 0.035*"flow" + 0.029*"plot" + '
  '0.029*"creator" + 0.027*"angel" + 0.020*"computer_science" + 0.020*"winter" '
  '+ 0.018*"criticism" + 0.015*"glory"'),
 (17,
  '0.143*"bus" + 0.054*"quick" + 0.051*"trace" + 0.032*"amp" + 0.029*"dirty" + '
  '0.025*"ins_cwru" + 0.019*"diameter" + 0.017*"tail" + 0.017*"detroit" + '
  '0.014*"western_digital"'),
 (26,
  '0.088*"greek" + 0.076*"armenian" + 0.059*"turk" + 0.052*"turkish" + '
  '0.026*"inhabitant" + 0.024*"russian" + 0.023*"brown" + 0.022*"proceed" + '
  '0.019*"escape" + 0.018*"genocide"'),
 (25,
  '0.096*"motherboard" + 0.079*"fix" + 0.060*"battery" + 0.058*"double" + '
  '0.045*"expansion" + 0.045*"unlikely" + 0.039*"remote" + 0.039*"compute" + '
  '0.020*"volt" + 0.015*"competition"'),
 (14,
  '0.235*"christian" + 0.192*"faith" + 0.054*"proof" + 0.051*"operate" + '
  '0.042*"waste" + 0.040*"scripture" + 0.023*"passage" + 0.016*"biblical" + '
  '0.013*"clh" + 0.009*"commandment"'),
 (12,
  '0.079*"player" + 0.068*"color" + 0.058*"contain" + 0.051*"draw" + '
  '0.042*"picture" + 0.029*"blank" + 0.026*"room" + 0.025*"associate" + '
  '0.024*"pack" + 0.024*"context"'),
 (1,
  '0.074*"price" + 0.067*"sell" + 0.065*"sale" + 0.047*"pin" + 0.046*"model" + '
  '0.032*"master" + 0.032*"offer" + 0.027*"license" + 0.021*"dept" + '
  '0.021*"purchase"'),
 (2,
  '0.114*"believe" + 0.074*"question" + 0.054*"true" + 0.042*"money" + '
  '0.038*"answer" + 0.033*"speak" + 0.032*"argument" + 0.031*"patient" + '
  '0.025*"church" + 0.019*"statement"'),
 (24,
  '0.049*"people" + 0.039*"state" + 0.037*"law" + 0.032*"right" + '
  '0.029*"government" + 0.028*"child" + 0.024*"death" + 0.022*"kill" + '
  '0.021*"religion" + 0.019*"die"'),
 (29,
  '0.045*"exist" + 0.044*"sense" + 0.032*"system" + 0.024*"discussion" + '
  '0.023*"truth" + 0.022*"internet" + 0.022*"purpose" + 0.019*"technology" + '
  '0.019*"section" + 0.016*"goal"'),
 (19,
  '0.031*"system" + 0.024*"use" + 0.024*"program" + 0.023*"file" + '
  '0.018*"window" + 0.017*"card" + 0.017*"run" + 0.014*"available" + '
  '0.013*"set" + 0.013*"software"'),
 (28,
  '0.026*"number" + 0.023*"group" + 0.018*"case" + 0.012*"new" + '
  '0.012*"report" + 0.011*"information" + 0.011*"year" + 0.010*"large" + '
  '0.010*"receive" + 0.010*"include"'),
 (7,
  '0.051*"go" + 0.030*"say" + 0.025*"time" + 0.024*"come" + 0.023*"s" + '
  '0.022*"take" + 0.019*"first" + 0.017*"year" + 0.017*"see" + 0.017*"think"'),
 (18,
  '0.022*"people" + 0.022*"say" + 0.018*"reason" + 0.017*"make" + '
  '0.017*"point" + 0.016*"many" + 0.016*"evidence" + 0.012*"thing" + '
  '0.012*"mean" + 0.011*"even"'),
 (8,
  '0.054*"line" + 0.042*"write" + 0.032*"organization" + 0.030*"article" + '
  '0.025*"get" + 0.021*"nntp_poste" + 0.019*"host" + 0.017*"m" + 0.016*"good" '
  '+ 0.015*"know"')]
```

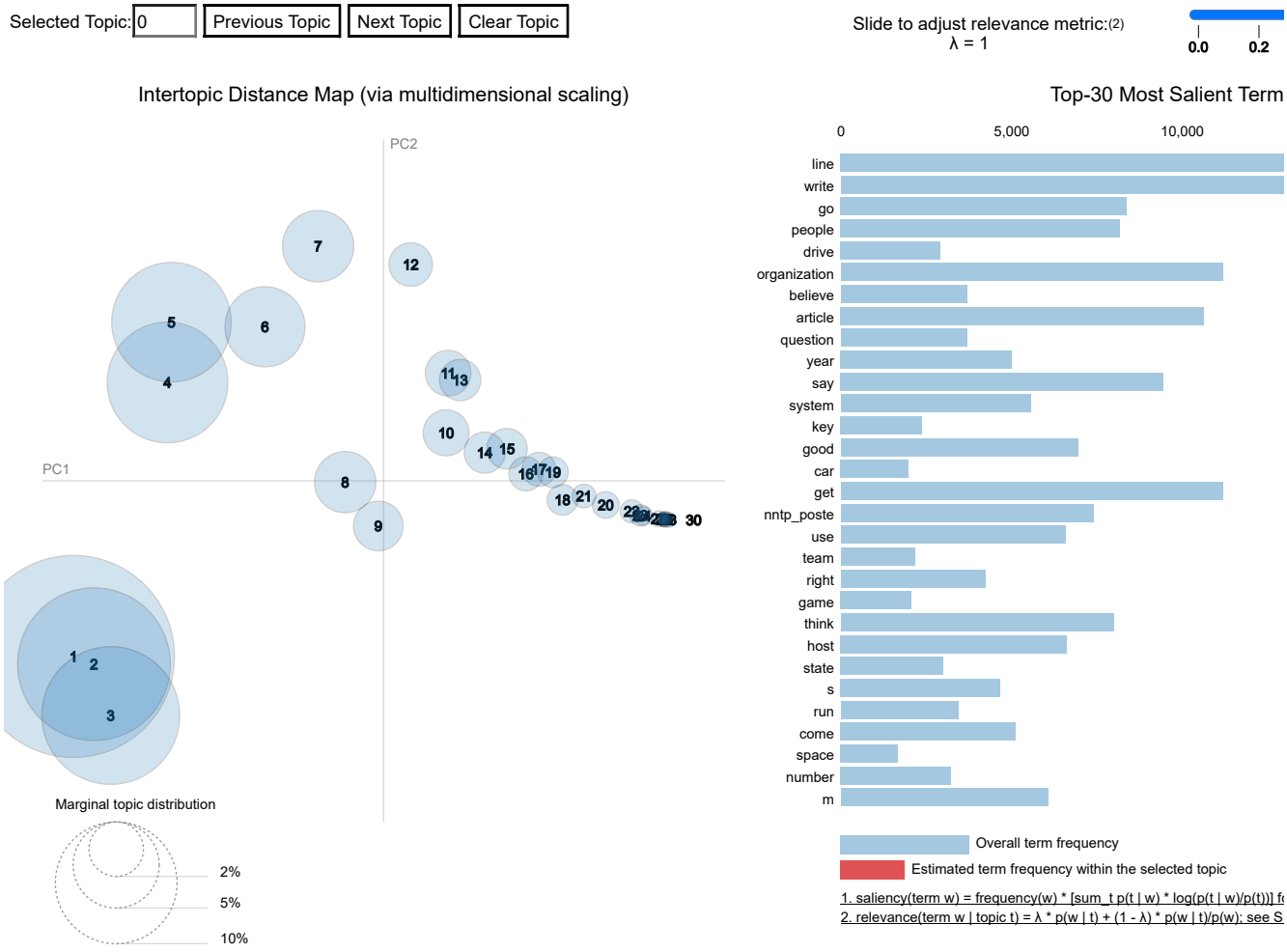
In [39]:
```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Perplexity:  -17.622962615175123

Coherence Score:  0.4791538630347555
```

In [40]:
```
# Visualize the topics for 30 topics
pyLDAvis.enable_notebook()
vis = plt_gensim.prepare(lda_model, corpus, id2word)
vis
```

```
C:\Users\student\anaconda3\lib\site-packages\pyLDAvis\_prepare.py:247: FutureWarning: In a future version of pandas all arguments
'labels' will be keyword-only
  by='saliency', ascending=False).head(R).drop('saliency', 1)
```

Out[40]:



Result

In [ ]:
```
#1. The best topic model is 30 topics in terms of perokexity score, which is -17.623
# we found the perokexity score keep getting better when we increase the number of topics
#2. In terms of coherence score, the best topic model is 10 topics  , which has a score of 0.549
# the score decreases after reaching the peak at 10 topics, but increase a bit at 30 topics
```