LDA

April 18, 2019

Source : https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

This notebook carries out topic modelling on a dataset of news articles using the Gensim implementation of LAtent Dirichlet Allocation (LDA).

Topic Modeling is a technique to extract the hidden topics from large volumes of text. Latent Dirichlet Allocation(LDA) is a popular algorithm for topic modeling with excellent implementations in the Python's Gensim package.

```
In [13]: import nltk; nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/albertstaszak/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
Out[13]: True
```

```
In [14]: import re
         import numpy as np
         import pandas as pd
         from pprint import pprint

         # 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  # 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

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

In [15]:
```
# NLTK Stop words -> We will filter these words out of our dataset so that they do no
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
```

In [16]:
```
# Import Dataset
import csv
import pandas as pd
#We are using the following dataset -> https://www.kaggle.com/snapcrack/all-the-news
csv_file = open('./csv/articles1.csv', 'r')
df = pd.read_csv(csv_file)
articles = df.content
```

In [17]:
```
# Convert to list
data = articles.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])
```

```
['WASHINGTON  Congressional Republicans have a new fear when it comes to '
 'their health care lawsuit against the Obama administration: They might win. '
 'The incoming Trump administration could choose to no longer defend the '
 'executive branch against the suit, which challenges the administrations '
 'authority to spend billions of dollars on health insurance subsidies for and '
 'Americans, handing House Republicans a big victory on issues. But a sudden '
 'loss of the disputed subsidies could conceivably cause the health care '
 'program to implode, leaving millions of people without access to health '
 'insurance before Republicans have prepared a replacement. That could lead to '
 'chaos in the insurance market and spur a political backlash just as '
 'Republicans gain full control of the government. To stave off that outcome, '
 'Republicans could find themselves in the awkward position of appropriating '
 'huge sums to temporarily prop up the Obama health care law, angering '
 'conservative voters who have been demanding an end to the law for years. In '
 'another twist, Donald J. Trumps administration, worried about preserving '
 'executive branch prerogatives, could choose to fight its Republican allies '
 'in the House on some central questions in the dispute. Eager to avoid an '
```

'ugly political pileup, Republicans on Capitol Hill and the Trump transition '
'team are gaming out how to handle the lawsuit, which, after the election, '
'has been put in limbo until at least late February by the United States '
'Court of Appeals for the District of Columbia Circuit. They are not yet '
'ready to divulge their strategy. Given that this pending litigation '
'involves the Obama administration and Congress, it would be inappropriate to '
'comment, said Phillip J. Blando, a spokesman for the Trump transition '
'effort. Upon taking office, the Trump administration will evaluate this '
'case and all related aspects of the Affordable Care Act. In a potentially '
'decision in 2015, Judge Rosemary M. Collyer ruled that House Republicans had '
'the standing to sue the executive branch over a spending dispute and that '
'the Obama administration had been distributing the health insurance '
'subsidies, in violation of the Constitution, without approval from Congress. '
'The Justice Department, confident that Judge Collyers decision would be '
'reversed, quickly appealed, and the subsidies have remained in place during '
'the appeal. In successfully seeking a temporary halt in the proceedings '
'after Mr. Trump won, House Republicans last month told the court that they '
'and the s transition team currently are discussing potential options for '
'resolution of this matter, to take effect after the s inauguration on Jan. '
'20, 2017. The suspension of the case, House lawyers said, will provide '
'the and his future administration time to consider whether to continue '
'prosecuting or to otherwise resolve this appeal. Republican leadership '
'officials in the House acknowledge the possibility of cascading effects if '
'the payments, which have totaled an estimated $13 billion, are suddenly '
'stopped. Insurers that receive the subsidies in exchange for paying costs '
'such as deductibles and for eligible consumers could race to drop coverage '
'since they would be losing money. Over all, the loss of the subsidies could '
'destabilize the entire program and cause a lack of confidence that leads '
'other insurers to seek a quick exit as well. Anticipating that the Trump '
'administration might not be inclined to mount a vigorous fight against the '
'House Republicans given the s dim view of the health care law, a team of '
'lawyers this month sought to intervene in the case on behalf of two '
'participants in the health care program. In their request, the lawyers '
'predicted that a deal between House Republicans and the new administration '
'to dismiss or settle the case will produce devastating consequences for the '
'individuals who receive these reductions, as well as for the nations health '
'insurance and health care systems generally. No matter what happens, House '
'Republicans say, they want to prevail on two overarching concepts: the '
'congressional power of the purse, and the right of Congress to sue the '
'executive branch if it violates the Constitution regarding that spending '
'power. House Republicans contend that Congress never appropriated the money '
'for the subsidies, as required by the Constitution. In the suit, which was '
'initially championed by John A. Boehner, the House speaker at the time, and '
'later in House committee reports, Republicans asserted that the '
'administration, desperate for the funding, had required the Treasury '
'Department to provide it despite widespread internal skepticism that the '
'spending was proper. The White House said that the spending was a permanent '
'part of the law passed in 2010, and that no annual appropriation was '

```
 'required  even though the administration initially sought one. Just as '
 'important to House Republicans, Judge Collyer found that Congress had the '
 'standing to sue the White House on this issue  a ruling that many legal '
 'experts said was flawed  and they want that precedent to be set to restore '
 'congressional leverage over the executive branch. But on spending power and '
 'standing, the Trump administration may come under pressure from advocates of '
 'presidential authority to fight the House no matter their shared views on '
 'health care, since those precedents could have broad repercussions. It is a '
 'complicated set of dynamics illustrating how a quick legal victory for the '
 'House in the Trump era might come with costs that Republicans never '
 'anticipated when they took on the Obama White House.']
```

```
In [18]: # Convert each document into list of individual words, remove punctuation.
         def sent_to_words(sentences):
             for sentence in sentences:
                 yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=Tru

         data_words = list(sent_to_words(data))

         print(data_words[:1])
```

```
[['washington', 'congressional', 'republicans', 'have', 'new', 'fear', 'when', 'it', 'comes',
```

```
In [19]: # We will build bigram and trigrams in order to combine words which often occur toget
         # eg.(White House becomes white-house).

         # Build the bigram and trigram models
         bigram = gensim.models.Phrases(data_words, min_count=5, threshold=100) # higher thres
         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]]])
```

```
['washington', 'congressional', 'republicans', 'have', 'new', 'fear', 'when', 'it', 'comes', '
```

```
In [20]: # 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] 

         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_posta
    return texts_out
```

In [21]:
```
# Remove Stop Words
data_words_nostops = remove_stopwords(data_words)

# Form Bigrams
data_words_bigrams = make_bigrams(data_words_nostops)

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

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

print(data_lemmatized[:1])
```

```
[['washington', 'congressional', 'republican', 'new', 'fear', 'come', 'health_care', 'lawsuit'
```

In [22]:
```
# We will compute the frequency with which each word occurs in a document.

# 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([[(id2word[id], freq) for id, freq in cp] for cp in corpus[:1]])
```

```
[[('access', 1), ('acknowledge', 1), ('act', 1), ('administration', 13), ('advocate', 1), ('af
```

In [23]:
```
# TODO: WE MUST FIND MODEL WHICH MAXIMISES CONERENCE SCORE - configurable params: htt
# CONFIGURE:
# - num_topics (MAIN CONFIGURABLE PARAM)
```

```
    # - iterations
    # - topic threshold
    def createModelAndComputeCoherence(minTopics, maxTopics, passes, chunkSize):
        # Build LDA model
        for topics in range(minTopics, maxTopics):
            print("Model with", topics, "topics,", passes, "passes &", chunkSize, "chunks
            mallet_path = './mallet-2.0.8/bin/mallet' # update this path
            ldamallet = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_
            # Show Topics
            pprint(ldamallet.show_topics(formatted=False))

            # Compute Coherence Score
            coherence_model_ldamallet = CoherenceModel(model=ldamallet, texts=data_lemmati
            coherence_ldamallet = coherence_model_ldamallet.get_coherence()
            print('\nCoherence Score: ', coherence_ldamallet)
            # Visualize the topics
            pyLDAvis.enable_notebook()
            vis = pyLDAvis.gensim.prepare(ldamallet, corpus, id2word)
            vis
    #         lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
    #                                                    id2word=id2word,
    #                                                    num_topics=topics,
    #                                                    random_state=100,
    #                                                    update_every=1,
    #                                                    chunksize=chunkSize,
    #                                                    passes=passes,
    #                                                    alpha='auto',
    #                                                    per_word_topics=True)
    #         # Print the Keyword in the 10 topics
    #         pprint(lda_model.print_topics())
    #         doc_lda = lda_model[corpus]
    #         # Compute Perplexity
    #         print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of h
    #         # Compute Coherence Score
    #         coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized
    #         coherence_lda = coherence_model_lda.get_coherence()
    #         print('\nCoherence Score: ', coherence_lda)
```

In [24]:
```
#createModelAndComputeCoherence(22,23,10,100)
```

In [25]:
```
mallet_path = './mallet-2.0.8/bin/mallet' # update this path
optimal_model = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_topi
# Show Topics
model_topics = optimal_model.show_topics(formatted=False)
pprint(optimal_model.print_topics(num_words=10))

# Compute Coherence Score
coherence_model_ldamallet = CoherenceModel(model=optimal_model, texts=data_lemmatized
```

```python
coherence_ldamallet = coherence_model_ldamallet.get_coherence()
print('\nCoherence Score: ', coherence_ldamallet)
```

```
[(7,
  '0.023*"case" + 0.020*"charge" + 0.015*"court" + 0.010*"prison" + '
  '0.009*"judge" + 0.009*"year" + 0.009*"attorney" + 0.009*"lawyer" + '
  '0.009*"prosecutor" + 0.008*"crime"'),
 (9,
  '0.025*"city" + 0.013*"york" + 0.012*"day" + 0.012*"home" + 0.010*"people" + '
  '0.007*"street" + 0.007*"building" + 0.007*"park" + 0.007*"place" + '
  '0.007*"time"'),
 (23,
  '0.023*"game" + 0.020*"team" + 0.014*"play" + 0.011*"player" + 0.010*"win" + '
  '0.010*"year" + 0.010*"sport" + 0.008*"time" + 0.008*"world" + '
  '0.007*"season"'),
 (22,
  '0.025*"attack" + 0.015*"isis" + 0.014*"group" + 0.014*"terrorist" + '
  '0.014*"islamic" + 0.013*"syria" + 0.012*"state" + 0.012*"military" + '
  '0.012*"kill" + 0.011*"force"'),
 (5,
  '0.010*"book" + 0.008*"work" + 0.008*"time" + 0.007*"write" + 0.007*"world" '
  '+ 0.006*"make" + 0.006*"year" + 0.005*"art" + 0.005*"read" + 0.005*"image"'),
 (15,
  '0.044*"woman" + 0.031*"family" + 0.026*"child" + 0.018*"man" + 0.017*"life" '
  '+ 0.013*"young" + 0.013*"father" + 0.013*"friend" + 0.012*"mother" + '
  '0.011*"year"'),
 (3,
  '0.036*"clinton" + 0.032*"trump" + 0.023*"campaign" + 0.021*"republican" + '
  '0.019*"candidate" + 0.019*"hillary" + 0.019*"voter" + 0.018*"vote" + '
  '0.017*"state" + 0.016*"party"'),
 (0,
  '0.025*"immigration" + 0.020*"country" + 0.019*"texas" + 0.016*"border" + '
  '0.014*"report" + 0.012*"mexico" + 0.011*"breitbart" + 0.011*"refugee" + '
  '0.011*"state" + 0.009*"immigrant"'),
 (1,
  '0.058*"news" + 0.028*"breitbart" + 0.023*"twitter" + 0.021*"medium" + '
  '0.016*"post" + 0.014*"follow" + 0.013*"fox" + 0.012*"facebook" + '
  '0.011*"report" + 0.010*"show"'),
 (2,
  '0.014*"water" + 0.007*"cnn" + 0.006*"year" + 0.006*"area" + 0.005*"plane" + '
  '0.005*"fire" + 0.005*"climate_change" + 0.005*"flight" + 0.005*"people" + '
  '0.005*"air"'),
 (16,
  '0.038*"company" + 0.009*"make" + 0.009*"car" + 0.009*"technology" + '
  '0.009*"year" + 0.008*"apple" + 0.007*"business" + 0.007*"product" + '
  '0.007*"sale" + 0.006*"sell"'),
 (17,
  '0.016*"party" + 0.014*"country" + 0.013*"europe" + 0.012*"leave" + '
```

```
  '0.010*"migrant" + 0.010*"britain" + 0.010*"london" + 0.010*"british" + '
  '0.009*"european" + 0.009*"year"'),
 (20,
  '0.015*"work" + 0.015*"mr" + 0.013*"ms" + 0.012*"member" + 0.011*"chief" + '
  '0.011*"group" + 0.010*"include" + 0.009*"office" + 0.008*"executive" + '
  '0.008*"time"'),
 (8,
  '0.018*"black" + 0.018*"people" + 0.018*"american" + 0.015*"america" + '
  '0.013*"white" + 0.011*"muslim" + 0.010*"country" + 0.009*"community" + '
  '0.008*"group" + 0.007*"world"'),
 (13,
  '0.014*"study" + 0.014*"health" + 0.009*"drug" + 0.009*"medical" + '
  '0.009*"find" + 0.009*"people" + 0.008*"case" + 0.008*"year" + '
  '0.007*"research" + 0.007*"patient"'),
 (6,
  '0.046*"people" + 0.026*"thing" + 0.026*"make" + 0.017*"good" + 0.014*"talk" '
  '+ 0.014*"time" + 0.014*"lot" + 0.013*"happen" + 0.011*"question" + '
  '0.010*"give"'),
 (19,
  '0.026*"state" + 0.025*"law" + 0.022*"student" + 0.021*"school" + '
  '0.018*"university" + 0.013*"rule" + 0.011*"court" + 0.010*"public" + '
  '0.009*"federal" + 0.009*"decision"'),
 (11,
  '0.197*"trump" + 0.067*"president" + 0.033*"donald" + 0.030*"obama" + '
  '0.018*"white_house" + 0.018*"campaign" + 0.010*"presidential" + 0.010*"cnn" '
  '+ 0.009*"administration" + 0.008*"call"'),
 (21,
  '0.033*"republican" + 0.023*"bill" + 0.020*"house" + 0.017*"senate" + '
  '0.016*"democrat" + 0.014*"president" + 0.013*"congress" + 0.013*"vote" + '
  '0.011*"ryan" + 0.011*"senator"'),
 (10,
  '0.020*"clinton" + 0.019*"email" + 0.015*"investigation" + 0.014*"report" + '
  '0.014*"official" + 0.013*"intelligence" + 0.012*"fbi" + 0.012*"russian" + '
  '0.012*"information" + 0.011*"department"')]

Coherence Score:  0.5299364085314472
```

In [30]:
```python
# One of the practical application of topic modeling is to determine what topic a giv
# To find that, we find the topic number that has the highest percentage contribution

def format_topics_sentences(ldamodel=optimal_model, corpus=corpus, texts=data):
    # Init output
    sent_topics_df = pd.DataFrame()

    # Get main topic in each document
    for i, row in enumerate(ldamodel[corpus]):
        row = sorted(row, key=lambda x: (x[1]), reverse=True)
```

```
        # Get the Dominant topic, Perc Contribution and Keywords for each document
        for j, (topic_num, prop_topic) in enumerate(row):
            if j == 0:  # => dominant topic
                wp = ldamodel.show_topic(topic_num)
                topic_keywords = ", ".join([word for word, prop in wp])
                sent_topics_df = sent_topics_df.append(pd.Series([int(topic_num), rou
            else:
                break
    sent_topics_df.columns = ['Dominant_Topic', 'Perc_Contribution', 'Topic_Keywords']

    # Add original text to the end of the output
    contents = pd.Series(texts)
    sent_topics_df = pd.concat([sent_topics_df, contents], axis=1)
    return(sent_topics_df)
df_topic_sents_keywords = format_topics_sentences(ldamodel=optimal_model, corpus=corpu

# Format
df_dominant_topic = df_topic_sents_keywords.reset_index()
df_dominant_topic.columns = ['Document_No', 'Dominant_Topic', 'Topic_Perc_Contrib', 'I

# Show
df_dominant_topic.head(100)
```

```
Out[30]:     Document_No  Dominant_Topic  Topic_Perc_Contrib  \
         0             0            21.0              0.3813
         1             1             4.0              0.3132
         2             2             5.0              0.3845
         3             3            14.0              0.2284
         4             4            12.0              0.5800
         5             5            17.0              0.1584
         6             6            12.0              0.5265
         7             7            13.0              0.5339
         8             8             5.0              0.2403
         9             9            15.0              0.4035
         10           10             2.0              0.1944
         11           11             2.0              0.3749
         12           12             9.0              0.2074
         13           13             6.0              0.2263
         14           14             5.0              0.4139
         15           15             9.0              0.4249
         16           16             7.0              0.4731
         17           17            18.0              0.3368
         18           18            22.0              0.4686
         19           19             2.0              0.3009
         20           20            23.0              0.7245
         21           21            14.0              0.3392
         22           22            22.0              0.2889
         23           23            14.0              0.2959
```

24	24	21.0	0.5267
25	25	21.0	0.5136
26	26	12.0	0.1742
27	27	7.0	0.2251
28	28	22.0	0.5323
29	29	22.0	0.2983
..	...	...	...
70	70	9.0	0.2824
71	71	10.0	0.2871
72	72	18.0	0.2826
73	73	16.0	0.3025
74	74	18.0	0.5885
75	75	20.0	0.2374
76	76	22.0	0.2714
77	77	12.0	0.2196
78	78	16.0	0.3438
79	79	18.0	0.3374
80	80	2.0	0.3778
81	81	18.0	0.3539
82	82	14.0	0.3574
83	83	5.0	0.2364
84	84	14.0	0.2523
85	85	8.0	0.2694
86	86	5.0	0.5291
87	87	10.0	0.4080
88	88	21.0	0.1556
89	89	10.0	0.4425
90	90	7.0	0.2311
91	91	21.0	0.5325
92	92	20.0	0.2306
93	93	21.0	0.2550
94	94	8.0	0.1701
95	95	18.0	0.2686
96	96	16.0	0.4141
97	97	2.0	0.2633
98	98	16.0	0.4347
99	99	16.0	0.3282

	Keywords \
0	republican, bill, house, senate, democrat, pre...
1	police, officer, man, gun, kill, shoot, report...
2	book, work, time, write, world, make, year, ar...
3	show, film, star, play, year, good, movie, ser...
4	china, country, president, russia, iran, israe...
5	party, country, europe, leave, migrant, britai...
6	china, country, president, russia, iran, israe...
7	study, health, drug, medical, find, people, ca...
8	book, work, time, write, world, make, year, ar...

```
9   woman, family, child, man, life, young, father...
10  water, cnn, year, area, plane, fire, climate_c...
11  water, cnn, year, area, plane, fire, climate_c...
12  city, york, day, home, people, street, buildin...
13  people, thing, make, good, talk, time, lot, ha...
14  book, work, time, write, world, make, year, ar...
15  city, york, day, home, people, street, buildin...
16  case, charge, court, prison, judge, year, atto...
17  year, percent, pay, money, job, american, busi...
18  attack, isis, group, terrorist, islamic, syria...
19  water, cnn, year, area, plane, fire, climate_c...
20  game, team, play, player, win, year, sport, ti...
21  show, film, star, play, year, good, movie, ser...
22  attack, isis, group, terrorist, islamic, syria...
23  show, film, star, play, year, good, movie, ser...
24  republican, bill, house, senate, democrat, pre...
25  republican, bill, house, senate, democrat, pre...
26  china, country, president, russia, iran, israe...
27  case, charge, court, prison, judge, year, atto...
28  attack, isis, group, terrorist, islamic, syria...
29  attack, isis, group, terrorist, islamic, syria...
..                                                ...
70  city, york, day, home, people, street, buildin...
71  clinton, email, investigation, report, officia...
72  year, percent, pay, money, job, american, busi...
73  company, make, car, technology, year, apple, b...
74  year, percent, pay, money, job, american, busi...
75  work, mr, ms, member, chief, group, include, o...
76  attack, isis, group, terrorist, islamic, syria...
77  china, country, president, russia, iran, israe...
78  company, make, car, technology, year, apple, b...
79  year, percent, pay, money, job, american, busi...
80  water, cnn, year, area, plane, fire, climate_c...
81  year, percent, pay, money, job, american, busi...
82  show, film, star, play, year, good, movie, ser...
83  book, work, time, write, world, make, year, ar...
84  show, film, star, play, year, good, movie, ser...
85  black, people, american, america, white, musli...
86  book, work, time, write, world, make, year, ar...
87  clinton, email, investigation, report, officia...
88  republican, bill, house, senate, democrat, pre...
89  clinton, email, investigation, report, officia...
90  case, charge, court, prison, judge, year, atto...
91  republican, bill, house, senate, democrat, pre...
92  work, mr, ms, member, chief, group, include, o...
93  republican, bill, house, senate, democrat, pre...
94  black, people, american, america, white, musli...
95  year, percent, pay, money, job, american, busi...
```

```
96  company, make, car, technology, year, apple, b...
97  water, cnn, year, area, plane, fire, climate_c...
98  company, make, car, technology, year, apple, b...
99  company, make, car, technology, year, apple, b...

                                                 Text
0   WASHINGTON  Congressional Republicans have a ...
1   After the bullet shells get counted, the blood...
2   When Walt Disneys Bambi opened in 1942, cri...
3   Death may be the great equalizer, but it isnt...
4   SEOUL, South Korea  North Koreas leader, Kim...
5   LONDON  Queen Elizabeth II, who has been batt...
6   BEIJING  President Tsai of Taiwan sharply cri...
7   Danny Cahill stood, slightly dazed, in a blizz...
8   Just how is Hillary Kerr, the founder of a dig...
9   Angels are everywhere in the Muñiz familys ap...
10  With Donald J. Trump about to take control of ...
11  THOMPSONS, Tex.  Can one of the most promisin...
12  WEST PALM BEACH, Fla.  When Donald J. Trump r...
13  This article is part of a series aimed at help...
14  Its the season for family travel and photos ...
15  Finally. The Second Avenue subway opened in Ne...
16   pages into the journal found in Dylann S. Roo...
17  MUMBAI, India  It was a bold and risky gamble...
18  BAGHDAD  A suicide bomber detonated a pickup ...
19  SYDNEY, Australia  The annual beach pilgrimag...
20  When the Green Bay Packers lost to the Washing...
21  Mariah Carey suffered through a performance tr...
22  PARIS  When the Islamic State was about to be...
23  Pop music and fashion never met cuter than in ...
24  WASHINGTON  The most powerful and ambitious C...
25  WASHINGTON  Its or time for Republicans. Aft...
26  Good morning. Heres what you need to know:  ...
27  The body of the Iraqi prisoner was found naked...
28  ISTANBUL  The Islamic State on Monday issued ...
29  WASHINGTON  President Obamas advisers wrestl...
..                                                ...
70  Gov. Andrew M. Cuomo of New York said on Wedne...
71  On the morning of May 18, 2014, Violeta Lagune...
72  It hasnt been a great time to be a man withou...
73  Apple, complying with what it said was a reque...
74  WASHINGTON  Federal Reserve officials expect ...
75  Rajiv J. Shah, a trustee of the Rockefeller Fo...
76  MANILA  A manhunt was underway Wednesday for ...
77  BEIJING  Chinas leaders thought they had a s...
78  DORAL, Fla.  Inside a clandestine Carnival Co...
79  The nations consumer watchdog agency on Tuesd...
80  LONDON  Maybe it wasnt just the iceberg. Eve...
```

```
81  When a Wall Street banking institution starts ...
82  Broadway rang out 2016 with a very big bang. T...
83  LECCE, Italy  One of his first students was a...
84  In the first episode of One Day at a Time, N...
85  Your forthcoming book, Tears We Cannot Stop,...
86  Condé Nast Publications might be sitting on a ...
87  WASHINGTON  A united front of top intelligenc...
88  WASHINGTON  Donald J. Trump is expected to ch...
89  WASHINGTON  When Special Agent Adrian Hawkins...
90  When the United Nations top official tried to...
91  WASHINGTON  Vice Mike Pence and the top Repub...
92  WASHINGTON  Donald J. Trumps transition staf...
93  WASHINGTON  Donald J. Trump lashed out at Dem...
94  TALLADEGA, Ala.  For a band at a tiny, histor...
95  After more than five years of investigations a...
96  The question from the analyst on Thursday was ...
97  A Long Island Rail Road train that crashed in ...
98  DETROIT  Unexpectedly strong sales of new veh...
99  Struggling with sagging sales over another cru...

[100 rows x 5 columns]
```

In [27]:

```python
# Sometimes just the topic keywords may not be enough to make sense of what a topic i
# So, to help with understanding the topic, you can find the documents a given topic
# to the most and infer the topic by reading that document.

# Group top 5 sentences under each topic
sent_topics_sorteddf_mallet = pd.DataFrame()

sent_topics_outdf_grpd = df_topic_sents_keywords.groupby('Dominant_Topic')

for i, grp in sent_topics_outdf_grpd:
    sent_topics_sorteddf_mallet = pd.concat([sent_topics_sorteddf_mallet,
                                             grp.sort_values(['Perc_Contribution'], a
                                            axis=0)

# Reset Index
sent_topics_sorteddf_mallet.reset_index(drop=True, inplace=True)

# Format
sent_topics_sorteddf_mallet.columns = ['Topic_Num', "Topic_Perc_Contrib", "Keywords",

# Show
sent_topics_sorteddf_mallet.head()
```

Out[27]:

```
   Topic_Num  Topic_Perc_Contrib  \
0        0.0              0.7716
1        1.0              0.5558
```

```
2        2.0              0.7655
3        3.0              0.7388
4        4.0              0.8178

                                          Keywords  \
0  immigration, country, texas, border, report, m...
1  news, breitbart, twitter, medium, post, follow...
2  water, cnn, year, area, plane, fire, climate_c...
3  clinton, trump, campaign, republican, candidat...
4  police, officer, man, gun, kill, shoot, report...

                                              Text
0  MATAMOROS, Tamaulipas  Los líderes de dos de ...
1  Most of the mainstream media and the tech jour...
2   (CNN) Here is a look at the 2016 Atlantic hur...
3  On Tuesday, Republicans in Idaho, Hawaii, Mich...
4   Police violence against civilians, particu...
```

In [28]:

```
# Finally, we want to understand the volume and distribution
# of topics in order to judge how widely it was discussed.
# The below table exposes that information.

# Number of Documents for Each Topic
topic_counts = df_topic_sents_keywords['Dominant_Topic'].value_counts()

# Percentage of Documents for Each Topic
topic_contribution = round(topic_counts/topic_counts.sum(), 4)

# Topic Number and Keywords
topic_num_keywords = df_topic_sents_keywords[['Dominant_Topic', 'Topic_Keywords']]

# Concatenate Column wise
df_dominant_topics = pd.concat([topic_num_keywords, topic_counts, topic_contribution]

# Change Column names
df_dominant_topics.columns = ['Dominant_Topic', 'Topic_Keywords', 'Num_Documents', 'P

# Show
df_dominant_topics
```

Out[28]:

```
      Dominant_Topic                                     Topic_Keywords  \
0               21.0  republican, bill, house, senate, democrat, pre...
1                4.0  police, officer, man, gun, kill, shoot, report...
2                5.0  book, work, time, write, world, make, year, ar...
3               14.0  show, film, star, play, year, good, movie, ser...
4               12.0  china, country, president, russia, iran, israe...
5               17.0  party, country, europe, leave, migrant, britai...
6               12.0  china, country, president, russia, iran, israe...
```

7	13.0	study, health, drug, medical, find, people, ca...
8	5.0	book, work, time, write, world, make, year, ar...
9	15.0	woman, family, child, man, life, young, father...
10	2.0	water, cnn, year, area, plane, fire, climate_c...
11	2.0	water, cnn, year, area, plane, fire, climate_c...
12	9.0	city, york, day, home, people, street, buildin...
13	6.0	people, thing, make, good, talk, time, lot, ha...
14	5.0	book, work, time, write, world, make, year, ar...
15	9.0	city, york, day, home, people, street, buildin...
16	7.0	case, charge, court, prison, judge, year, atto...
17	18.0	year, percent, pay, money, job, american, busi...
18	22.0	attack, isis, group, terrorist, islamic, syria...
19	2.0	water, cnn, year, area, plane, fire, climate_c...
20	23.0	game, team, play, player, win, year, sport, ti...
21	14.0	show, film, star, play, year, good, movie, ser...
22	22.0	attack, isis, group, terrorist, islamic, syria...
23	14.0	show, film, star, play, year, good, movie, ser...
24	21.0	republican, bill, house, senate, democrat, pre...
25	21.0	republican, bill, house, senate, democrat, pre...
26	12.0	china, country, president, russia, iran, israe...
27	7.0	case, charge, court, prison, judge, year, atto...
28	22.0	attack, isis, group, terrorist, islamic, syria...
29	22.0	attack, isis, group, terrorist, islamic, syria...
...	...	...
49970	8.0	black, people, american, america, white, musli...
49971	10.0	clinton, email, investigation, report, officia...
49972	13.0	study, health, drug, medical, find, people, ca...
49973	10.0	clinton, email, investigation, report, officia...
49974	21.0	republican, bill, house, senate, democrat, pre...
49975	12.0	china, country, president, russia, iran, israe...
49976	19.0	state, law, student, school, university, rule,...
49977	13.0	study, health, drug, medical, find, people, ca...
49978	18.0	year, percent, pay, money, job, american, busi...
49979	10.0	clinton, email, investigation, report, officia...
49980	5.0	book, work, time, write, world, make, year, ar...
49981	17.0	party, country, europe, leave, migrant, britai...
49982	19.0	state, law, student, school, university, rule,...
49983	6.0	people, thing, make, good, talk, time, lot, ha...
49984	13.0	study, health, drug, medical, find, people, ca...
49985	14.0	show, film, star, play, year, good, movie, ser...
49986	8.0	black, people, american, america, white, musli...
49987	2.0	water, cnn, year, area, plane, fire, climate_c...
49988	23.0	game, team, play, player, win, year, sport, ti...
49989	21.0	republican, bill, house, senate, democrat, pre...
49990	10.0	clinton, email, investigation, report, officia...
49991	21.0	republican, bill, house, senate, democrat, pre...
49992	19.0	state, law, student, school, university, rule,...
49993	20.0	work, mr, ms, member, chief, group, include, o...

49994	8.0	black, people, american, america, white, musli...
49995	12.0	china, country, president, russia, iran, israe...
49996	10.0	clinton, email, investigation, report, officia...
49997	20.0	work, mr, ms, member, chief, group, include, o...
49998	19.0	state, law, student, school, university, rule,...
49999	2.0	water, cnn, year, area, plane, fire, climate_c...

	Num_Documents	Perc_Documents
0	1887.0	0.0377
1	2595.0	0.0519
2	2019.0	0.0404
3	4234.0	0.0847
4	3074.0	0.0615
5	1350.0	0.0270
6	1231.0	0.0246
7	1503.0	0.0301
8	2074.0	0.0415
9	1178.0	0.0236
10	2590.0	0.0518
11	2833.0	0.0567
12	2313.0	0.0463
13	1393.0	0.0279
14	2710.0	0.0542
15	1268.0	0.0254
16	2519.0	0.0504
17	1998.0	0.0400
18	2151.0	0.0430
19	1800.0	0.0360
20	610.0	0.0122
21	1983.0	0.0397
22	2549.0	0.0510
23	2138.0	0.0428
24	NaN	NaN
25	NaN	NaN
26	NaN	NaN
27	NaN	NaN
28	NaN	NaN
29	NaN	NaN
...	...	...
49970	NaN	NaN
49971	NaN	NaN
49972	NaN	NaN
49973	NaN	NaN
49974	NaN	NaN
49975	NaN	NaN
49976	NaN	NaN
49977	NaN	NaN
49978	NaN	NaN

```
49979            NaN             NaN
49980            NaN             NaN
49981            NaN             NaN
49982            NaN             NaN
49983            NaN             NaN
49984            NaN             NaN
49985            NaN             NaN
49986            NaN             NaN
49987            NaN             NaN
49988            NaN             NaN
49989            NaN             NaN
49990            NaN             NaN
49991            NaN             NaN
49992            NaN             NaN
49993            NaN             NaN
49994            NaN             NaN
49995            NaN             NaN
49996            NaN             NaN
49997            NaN             NaN
49998            NaN             NaN
49999            NaN             NaN

[50000 rows x 4 columns]
```

In [29]:
```
# Visualize the topics
pyLDAvis.enable_notebook()
model = gensim.models.wrappers.ldamallet.malletmodel2ldamodel(optimal_model)
vis = pyLDAvis.gensim.prepare(model, corpus, id2word)
vis
```

```
---------------------------------------------------------------------------

KeyboardInterrupt                         Traceback (most recent call last)

<ipython-input-29-33e4d4458f49> in <module>
      2 pyLDAvis.enable_notebook()
      3 model = gensim.models.wrappers.ldamallet.malletmodel2ldamodel(optimal_model)
----> 4 vis = pyLDAvis.gensim.prepare(model, corpus, id2word)
      5 vis


~/anaconda3/lib/python3.7/site-packages/pyLDAvis/gensim.py in prepare(topic_model, cor
117     """
118     opts = fp.merge(_extract_data(topic_model, corpus, dictionary, doc_topic_dist)
--> 119     return vis_prepare(**opts)
```

```
    ~/anaconda3/lib/python3.7/site-packages/pyLDAvis/_prepare.py in prepare(topic_term_dis
    396     term_frequency = np.sum(term_topic_freq, axis=0)
    397
--> 398     topic_info         = _topic_info(topic_term_dists, topic_proportion, term_freque
    399     token_table        = _token_table(topic_info, term_topic_freq, vocab, term_frequ
    400     topic_coordinates = _topic_coordinates(mds, topic_term_dists, topic_proportion)


    ~/anaconda3/lib/python3.7/site-packages/pyLDAvis/_prepare.py in _topic_info(topic_term_
    220     # compute the distinctiveness and saliency of the terms:
    221     # this determines the R terms that are displayed when no topic is selected
--> 222     topic_given_term = topic_term_dists / topic_term_dists.sum()
    223     kernel = (topic_given_term * np.log((topic_given_term.T / topic_proportion).T))
    224     distinctiveness = kernel.sum()


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in f(self, other, axis, leve
   2028             return _combine_series_frame(self, other, pass_op,
   2029                                          fill_value=fill_value, axis=axis,
-> 2030                                          level=level)
   2031         else:
   2032             if fill_value is not None:


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in _combine_series_frame(sel
   1928
   1929         # default axis is columns
-> 1930         return self._combine_match_columns(other, func, level=level)
   1931
   1932


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/frame.py in _combine_match_columns
   5114                                 copy=False)
   5115         assert left.columns.equals(right.index)
-> 5116         return ops.dispatch_to_series(left, right, func, axis="columns")
   5117
   5118     def _combine_const(self, other, func):


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in dispatch_to_series(left,
   1155         raise NotImplementedError(right)
   1156
-> 1157     new_data = expressions.evaluate(column_op, str_rep, left, right)
   1158
   1159     result = left._constructor(new_data, index=left.index, copy=False)
```

```
    ~/anaconda3/lib/python3.7/site-packages/pandas/core/computation/expressions.py in evalu
    206     use_numexpr = use_numexpr and _bool_arith_check(op_str, a, b)
    207     if use_numexpr:
--> 208         return _evaluate(op, op_str, a, b, **eval_kwargs)
    209     return _evaluate_standard(op, op_str, a, b)
    210


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/computation/expressions.py in _eval
    121
    122     if result is None:
--> 123         result = _evaluate_standard(op, op_str, a, b)
    124
    125     return result


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/computation/expressions.py in _eval
     66         _store_test_result(False)
     67     with np.errstate(all='ignore'):
---> 68         return op(a, b)
     69
     70


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in column_op(a, b)
   1142         def column_op(a, b):
   1143             return {i: func(a.iloc[:, i], b.iloc[i])
-> 1144                     for i in range(len(a.columns))}
   1145
   1146     elif isinstance(right, ABCSeries):


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in <dictcomp>(.0)
   1142         def column_op(a, b):
   1143             return {i: func(a.iloc[:, i], b.iloc[i])
-> 1144                     for i in range(len(a.columns))}
   1145
   1146     elif isinstance(right, ABCSeries):


    ~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in wrapper(left, right)
   1583         result = safe_na_op(lvalues, rvalues)
   1584         return construct_result(left, result,
-> 1585                                 index=left.index, name=res_name, dtype=None)
   1586
   1587     wrapper.__name__ = op_name
```

```
~/anaconda3/lib/python3.7/site-packages/pandas/core/ops.py in _construct_result(left, 
  1472     not be enough; we still need to override the name attribute.
  1473     """
-> 1474     out = left._constructor(result, index=index, dtype=dtype)
  1475 
  1476     out.name = name


~/anaconda3/lib/python3.7/site-packages/pandas/core/series.py in __init__(self, data, 
   260             else:
   261                 data = sanitize_array(data, index, dtype, copy,
--> 262                                       raise_cast_failure=True)
   263 
   264                 data = SingleBlockManager(data, index, fastpath=True)


~/anaconda3/lib/python3.7/site-packages/pandas/core/internals/construction.py in sanit
   541     dtype if specified.
   542     """
--> 543     if dtype is not None:
   544         dtype = pandas_dtype(dtype)
   545 


KeyboardInterrupt:
```

In [ ]: