Configuration

```
#Parameters
PROJECT_NAME = 'ML1010_Weekly'
ENABLE_COLAB = True

#Root Machine Learning Directory. Projects appear underneath
GOOGLE_DRIVE_MOUNT = '/content/gdrive'
COLAB_ROOT_DIR = GOOGLE_DRIVE_MOUNT + '/MyDrive/Colab Notebooks'
COLAB_INIT_DIR = COLAB_ROOT_DIR + '/utility_files'

LOCAL_ROOT_DIR = '/home/magni/Documents/ML_Projects'
LOCAL_INIT_DIR = LOCAL_ROOT_DIR + '/utility_files'
```

Bootstrap Environment

```
#add in support for utility file directory and importing
import sys
import os

if ENABLE_COLAB:
  #Need access to drive
  from google.colab import drive
  drive.mount(GOOGLE_DRIVE_MOUNT, force_remount=True)

  #add in utility directory to syspath to import
  INIT_DIR = COLAB_INIT_DIR
  sys.path.append(os.path.abspath(INIT_DIR))

  #Config environment variables
  ROOT_DIR = COLAB_ROOT_DIR

else:
  #add in utility directory to syspath to import
  INIT_DIR = LOCAL_INIT_DIR
  sys.path.append(os.path.abspath(INIT_DIR))

  #Config environment variables
  ROOT_DIR = LOCAL_ROOT_DIR

#Import Utility Support
from jarvis import Jarvis
jarvis = Jarvis(ROOT_DIR, PROJECT_NAME)

import my_python_utils as myutils
```

```
Mounted at /content/gdrive
Wha...where am I?
I am awake now.
I am inspecting the local environment...
Your environment has been configured:
PROJECT_NAME:     ML1010_Weekly
ROOT_DIR: /content/gdrive/MyDrive/Colab Notebooks
ROOT_DATA_DIR: /content/gdrive/MyDrive/Colab Notebooks
DATA_DIR:     /content/gdrive/MyDrive/Colab Notebooks/data/ML1010_Weekly
WORKING_DIR:  /content/gdrive/MyDrive/Colab Notebooks/ML1010_Weekly
UTILITY_DIR:  /content/gdrive/MyDrive/Colab Notebooks/utility_files

Here are all your project work files
[D] /content/gdrive/MyDrive/Colab Notebooks/ML1010_Weekly
---[ipynb]-----> _template_wkX_ML1010_.ipynb (5.41 KB)
---[ipynb]-----> wk0_CSML1010_Day1.ipynb (27.77 KB)
---[ipynb]-----> wk0_Sentiment Analysis Tutorial.ipynb (28.92 KB)
---[ipynb]-----> wk1_ML1010_Code1 (1).ipynb (32.46 KB)
---[ipynb]-----> wk1_ML1010_Code1.ipynb (24.37 KB)
---[  pdf]-----> wk1_ML1010_Code1and2.pdf (1.44 MB)
---[ipynb]-----> wk1_ML1010_Code2.ipynb (1.84 MB)
---[ipynb]-----> wk1_text_classification_rnn.ipynb (16.92 KB)
---[ipynb]-----> wk2_ML1010_Code_FE.ipynb (58.81 KB)
---[ipynb]-----> wk2_ML1010_Flair_tutorial_3.ipynb (170.88 KB)
---[ipynb]-----> wk2_final_bert_long_docs_yay.ipynb (159.04 KB)

[D] /content/gdrive/MyDrive/Colab Notebooks/ML1010_Weekly/wk2_attempts
---[ipynb]-----> wk2_final_bert_long_docs.ipynb (140.11 KB)
---[ipynb]-----> wk2_final_bert_long_docs_v2.ipynb (140.17 KB)
---[ipynb]-----> wk2_final_bert_long_docs_v3.ipynb (138.70 KB)

Here are all your project data files
[D] /content/gdrive/MyDrive/Colab Notebooks/data/ML1010_Weekly
---[   gz][  csv]--> complaints.csv.gz (370.67 MB)
[*][  csv]-----> movie_reviews_cleaned.csv (38.37 MB)
---[   gz][  tsv]--> rspct.tsv.gz (347.13 MB)
---[   gz][  csv]--> subreddit_info.csv.gz (37.80 KB)

[D] /content/gdrive/MyDrive/Colab Notebooks/data/ML1010_Weekly/01_original
-----** No files **

[D] /content/gdrive/MyDrive/Colab Notebooks/data/ML1010_Weekly/02_working
-----** No files **

[D] /content/gdrive/MyDrive/Colab Notebooks/data/ML1010_Weekly/03_train
-----** No files **

[D] /content/gdrive/MyDrive/Colab Notebooks/data/ML1010_Weekly/04_test
-----** No files **

I have set your current working directory to /content/gdrive/MyDrive/Colab Notebooks/ML1
The current time is 10:56
Hello sir. Extra caffeine may help.
```

Setup Runtime Environment

```
if ENABLE_COLAB:
  #!pip install scipy -q
  #!pip install scikit-learn -q
  #!pip install pycaret -q
  #!pip install matplotlib -q
  #!pip install joblib -q
  #!pip install pandasql -q

  print('Google Colab enabled')
else:
  print('Google Colab not enabled')

#Common imports
```

```
Google Colab enabled
```

GenSim Tutorial

```
!pip install pandas==1.3.4 --upgrade
jarvis.getPackageVersion('pandas')
```

```
Requirement already satisfied: pandas==1.3.4 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
pandas version: pandas 1.3.4
```

```
# Run in python console
import nltk
nltk.download('stopwords')

# Run in terminal or command prompt
!python3 -m spacy download en
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Collecting en_core_web_sm==2.2.5
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm
     |████████████████████████████████| 12.0 MB 21.5 MB/s
Requirement already satisfied: spacy>=2.2.2 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.7/dist-pack
```

```
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.7/dist-packages (t
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: importlib-metadata>=0.20 in /usr/local/lib/python3.7/dist
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/dist
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (t
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
✓ Download and installation successful
You can now load the model via spacy.load('en_core_web_sm')
✓ Linking successful
/usr/local/lib/python3.7/dist-packages/en_core_web_sm -->
/usr/local/lib/python3.7/dist-packages/spacy/data/en
You can now load the model via spacy.load('en')
```

```
!pip install pyLDAvis -q
```

```
     |████████████████████████████████| 1.7 MB 27.2 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
    Preparing wheel metadata ... done
     |████████████████████████████████| 15.7 MB 37.5 MB/s
  Building wheel for pyLDAvis (PEP 517) ... done
ERROR: pip's dependency resolver does not currently take into account all the packages t
yellowbrick 1.3.post1 requires numpy<1.20,>=1.16.0, but you have numpy 1.21.4 which is i
google-colab 1.0.0 requires pandas~=1.1.0; python_version >= "3.0", but you have pandas
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is incompatib
albumentations 0.1.12 requires imgaug<0.2.7,>=0.2.5, but you have imgaug 0.2.9 which is
```

```
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

#below gave error, changed to gensim_models (works?)
#import pyLDAvis.gensim
import pyLDAvis.gensim_models  # don't skip this

import matplotlib.pyplot as plt
%matplotlib inline
```

```
/usr/local/lib/python3.7/dist-packages/past/types/oldstr.py:5: DeprecationWarning: Using
  from collections import Iterable
```

```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
stop_words.extend(['from', 'subject', 're', 'edu', 'use'])
```

```
Requirement already satisfied: pandas==1.3.4 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
pandas version: pandas 1.1.5
```

```
# 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'
```

```
# 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])
```

```
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:8: DeprecationWarning: invalid escape sequence \s
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:8: DeprecationWarning: invalid escape sequence \s
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:8: DeprecationWarning: invalid escape sequence \s
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:8: DeprecationWarning: invalid escape sequence \s
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:8: DeprecationWarning: invalid escape sequence \s
<input>:5: DeprecationWarning: invalid escape sequence \S
<input>:8: DeprecationWarning: invalid escape sequence \s
<ipython-input-10-10af9153bd18>:5: DeprecationWarning: invalid escape sequence \S
  data = [re.sub('\S*@\S*\s?', '', sent) for sent in data]
<ipython-input-10-10af9153bd18>:8: DeprecationWarning: invalid escape sequence \s
  data = [re.sub('\s+', ' ', sent) for sent in data]
['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 ---- ']
```

```
def sent_to_words(sentences):
    for sentence in sentences:
        yield(gensim.utils.simple_preprocess(str(sentence), deacc=True))  # deacc=True remove

data_words = list(sent_to_words(data))
```

```
print(data_words[:1])
```

```
[['from', 'wheres', 'my', 'thing', 'subject', 'what', 'car', 'is', 'this', 'nntp', 'post
```

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

```
/usr/local/lib/python3.7/dist-packages/gensim/models/phrases.py:598: UserWarning: For a
  warnings.warn("For a faster implementation, use the gensim.models.phrases.Phraser clas
['from', 'wheres', 'my', 'thing', 'subject', 'what', 'car', 'is', 'this', 'nntp_posting_
```

```
# 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

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


# 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', 'VERB', '

print(data_lemmatized[:1])
```

```
Streaming output truncated to the last 5000 lines.
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
```

```
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
/usr/local/lib/python3.7/dist-packages/catalogue.py:138: DeprecationWarning: Selectab
  for entry_point in AVAILABLE_ENTRY_POINTS.get(self.entry_point_namespace, []):
```

```
# 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, 1), (8, 2), (9, 1), (10, 1
```

```
id2word[0]
```

```
'addition'
```

```
# Human readable format of corpus (term-frequency)
[[(id2word[id], freq) for id, freq in cp] for cp in corpus[:1]]
```

```
[[('addition', 1),
  ('body', 1),
  ('bricklin', 1),
  ('bring', 1),
  ('call', 1),
  ('car', 5),
  ('could', 1),
  ('day', 1),
  ('door', 2),
  ('early', 1),
  ('engine', 1),
  ('enlighten', 1),
  ('funky', 1),
  ('history', 1),
  ('host', 1),
  ('info', 1),
  ('know', 1),
  ('late', 1),
  ('lerxst', 1),
  ('line', 1),
  ('look', 2),
  ('mail', 1),
```

```
        ('make', 1),
        ('model', 1),
        ('name', 1),
        ('neighborhood', 1),
        ('nntp_poste', 1),
        ('park', 1),
        ('production', 1),
        ('really', 1),
        ('rest', 1),
        ('see', 1),
        ('separate', 1),
        ('small', 1),
        ('sport', 1),
        ('tellme', 1),
        ('thank', 1),
        ('thing', 1),
        ('where', 1),
        ('wonder', 1),
        ('year', 1)]]
```

```
# Build LDA model
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)
```

```
Streaming output truncated to the last 5000 lines.
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
```

```
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
```

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

```
[(0,
  '0.051*"report" + 0.027*"black" + 0.020*"fire" + 0.020*"white" + '
  '0.016*"trial" + 0.016*"cover" + 0.015*"medium" + 0.013*"vote" + '
  '0.012*"minor" + 0.012*"title"'),
 (1,
  '0.021*"god" + 0.020*"accept" + 0.016*"member" + 0.015*"man" + '
  '0.014*"israeli" + 0.014*"season" + 0.012*"publish" + 0.012*"lebanese" + '
  '0.012*"jewish" + 0.011*"brain"'),
 (2,
  '0.017*"package" + 0.016*"press" + 0.015*"item" + 0.015*"break" + '
  '0.011*"level" + 0.010*"edge" + 0.009*"hole" + 0.007*"eye" + '
  '0.007*"contribute" + 0.007*"equipment"'),
 (3,
  '0.025*"pc" + 0.022*"contain" + 0.020*"input" + 0.020*"reality" + '
  '0.017*"picture" + 0.016*"object" + 0.016*"level" + 0.015*"box" + '
  '0.015*"quality" + 0.013*"greek"'),
 (4,
```

```
 '0.089*"ax" + 0.076*"max" + 0.032*"space" + 0.021*"launch" + 0.018*"di_di" + '
 '0.017*"orbit" + 0.016*"sphere" + 0.015*"satellite" + 0.014*"plane" + '
 '0.014*"mission"'),
(5,
 '0.019*"people" + 0.017*"kill" + 0.015*"child" + 0.015*"government" + '
 '0.012*"attack" + 0.012*"year" + 0.012*"die" + 0.011*"country" + 0.010*"say" '
 '+ 0.009*"war"'),
(6,
 '0.035*"window" + 0.032*"card" + 0.020*"image" + 0.020*"driver" + '
 '0.020*"problem" + 0.019*"run" + 0.018*"sale" + 0.018*"machine" + '
 '0.017*"color" + 0.016*"screen"'),
(7,
 '0.025*"people" + 0.021*"say" + 0.014*"reason" + 0.014*"believe" + '
 '0.012*"may" + 0.012*"evidence" + 0.010*"make" + 0.010*"think" + '
 '0.009*"many" + 0.009*"mean"'),
(8,
 '0.032*"book" + 0.023*"physical" + 0.021*"science" + 0.017*"choose" + '
 '0.016*"explain" + 0.015*"create" + 0.011*"author" + 0.011*"earth" + '
 '0.010*"study" + 0.010*"nature"'),
(9,
 '0.033*"mail" + 0.028*"file" + 0.027*"send" + 0.026*"program" + '
 '0.025*"thank" + 0.024*"information" + 0.021*"software" + 0.021*"list" + '
 '0.019*"include" + 0.019*"address"'),
(10,
 '0.073*"group" + 0.031*"week" + 0.021*"young" + 0.017*"drug" + 0.015*"watch" '
 '+ 0.013*"nntp_posting" + 0.013*"age" + 0.013*"route" + 0.011*"kid" + '
 '0.010*"capable"'),
(11,
 '0.073*"car" + 0.023*"existence" + 0.022*"model" + 0.020*"engine" + '
 '0.016*"pain" + 0.012*"keyboard" + 0.012*"mile" + 0.011*"should" + '
 '0.011*"price" + 0.011*"insurance"'),
(12,
 '0.070*"drive" + 0.025*"power" + 0.024*"player" + 0.017*"speed" + '
 '0.017*"light" + 0.014*"high" + 0.013*"bus" + 0.012*"university" + '
 '0.012*"fast" + 0.012*"scsi"'),
(13,
 '0.040*"line" + 0.039*"would" + 0.035*"write" + 0.024*"article" + 0.021*"be" '
 '+ 0.020*"get" + 0.020*"know" + 0.020*"go" + 0.014*"good" + 0.014*"think"'),
(14,
 '0.027*"patient" + 0.017*"family" + 0.014*"food" + 0.013*"treatment" + '
 '0.012*"disease" + 0.012*"doctor" + 0.011*"cd" + 0.011*"diagnosis" + '
```

```
# Compute Perplexity
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model

# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2wo
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Streaming output truncated to the last 5000 lines.
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in d
```

```
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
/usr/local/lib/python3.7/dist-packages/gensim/models/ldamodel.py:1077: DeprecationWar
  score += np.sum(cnt * logsumexp(Elogthetad + Elogbeta[:, int(id)]) for id, cnt in do
```

```
# Visualize the topics
pyLDAvis.enable_notebook()
```

```
vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word)
vis
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-23-3af4665cd12f> in <module>()
      1 # Visualize the topics
      2 pyLDAvis.enable_notebook()
----> 3 vis = pyLDAvis.gensim.prepare(lda_model, corpus, id2word)
      4 vis

AttributeError: module 'pyLDAvis' has no attribute 'gensim'
```

SEARCH STACK OVERFLOW

```
# Download File: http://mallet.cs.umass.edu/dist/mallet-2.0.8.zip
mallet_path = '/content/gdrive/MyDrive/Colab Notebooks/utility_files/mallet_latest/bin/mallet
ldamallet = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_topics=20, id2wo
```

```
/usr/local/lib/python3.7/dist-packages/smart_open/smart_open_lib.py:494: DeprecationWarn
  warnings.warn(message, category=DeprecationWarning)
---------------------------------------------------------------------------
CalledProcessError                        Traceback (most recent call last)
<ipython-input-25-d6cdf2e6b3f4> in <module>()
      1 # Download File: http://mallet.cs.umass.edu/dist/mallet-2.0.8.zip
      2 mallet_path = '/content/gdrive/MyDrive/Colab
Notebooks/utility_files/mallet_latest/bin/mallet' # update this path
----> 3 ldamallet = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus,
num_topics=20, id2word=id2word)
```

```
# Show Topics
pprint(ldamallet.show_topics(formatted=False))

# Compute Coherence Score
coherence_model_ldamallet = CoherenceModel(model=ldamallet, texts=data_lemmatized, dictionary
coherence_ldamallet = coherence_model_ldamallet.get_coherence()
print('\nCoherence Score: ', coherence_ldamallet)
```

```
        1881     except KeyboardInterrupt:
```

```
def compute_coherence_values(dictionary, corpus, texts, limit, start=2, step=3):
    """
    Compute c_v coherence for various number of topics

    Parameters:
    ----------
    dictionary : Gensim dictionary
    corpus : Gensim corpus
    texts : List of input texts
    limit : Max num of topics

    Returns:
    -------
    model_list : List of LDA topic models
    coherence_values : Coherence values corresponding to the LDA model with respective number
    """
    coherence_values = []
    model_list = []
    for num_topics in range(start, limit, step):
        model = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_topics=num_t
        model_list.append(model)
        coherencemodel = CoherenceModel(model=model, texts=texts, dictionary=dictionary, cohe
        coherence_values.append(coherencemodel.get_coherence())

    return model_list, coherence_values


# Can take a long time to run.
model_list, coherence_values = compute_coherence_values(dictionary=id2word, corpus=corpus, te


# Show graph
limit=40; start=2; step=6;
```

```
    x = range(start, limit, step)
    plt.plot(x, coherence_values)
    plt.xlabel("Num Topics")
    plt.ylabel("Coherence score")
    plt.legend(("coherence_values"), loc='best')
    plt.show()



    # Print the coherence scores
    for m, cv in zip(x, coherence_values):
        print("Num Topics =", m, " has Coherence Value of", round(cv, 4))


    # Select the model and print the topics
    optimal_model = model_list[3]
    model_topics = optimal_model.show_topics(formatted=False)
    pprint(optimal_model.print_topics(num_words=10))


    def format_topics_sentences(ldamodel=lda_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), round(prop_
                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=corpus, text

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

    # Show
    df_dominant_topic.head(10)
```

```
# 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'], ascending
                                            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", "Text"]

# Show
sent_topics_sorteddf_mallet.head()



# 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], axis=1

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

# Show
df_dominant_topics
```