bayesian_topic_modeling_explore

February 22, 2020

1 Read in data & clean

```
In [52]: import pandas as pd
         import numpy as np
         import string
         import nltk
         import gensim
         import pyLDAvis
         import pyLDAvis.gensim  # don't skip this
         import matplotlib.pyplot as plt
         from nltk.corpus import stopwords
         from nltk.stem.wordnet import WordNetLemmatizer
         from gensim import corpora


         train = pd.read_csv("/Users/wyattmadden/Documents/school/" +
                             "MSU/2020/spring/cs547/group_project/" +
                             "not_for_repo/nlp-getting-started/train.csv")
```

2 following https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/

Get stop words and tweet cleaning function:

```
In [25]: stop = set(stopwords.words('english'))
         exclude = set(string.punctuation)
         lemma = WordNetLemmatizer()

         def clean(doc):
             stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
             punc_free = ''.join(ch for ch in stop_free if ch not in exclude)
             normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
             return normalized
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/wyattmadden/nltk_data...
```

```
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/wyattmadden/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
```

```
In [42]: train['text_clean'] = [clean(i).split() for i in train['text']]
         dictionary = corpora.Dictionary(train['text_clean'])
         doc_term_matrix = [dictionary.doc2bow(i) for i in train['text_clean']]
         Lda = gensim.models.ldamodel.LdaModel
```

```
In [49]: ldamodel = Lda(doc_term_matrix, num_topics=3, id2word = dictionary, passes=30)
```

```
In [62]: print(ldamodel.print_topics(num_topics=3, num_words=10))
```

```
[(0, '0.008*"like" + 0.005*"im" + 0.004*"get" + 0.004*"one" + 0.003*"emergency" + 0.003*"amp" +
```

```
In [59]: pyLDAvis.enable_notebook()
         vis = pyLDAvis.gensim.prepare(ldamodel, dictionary, doc_term_matrix)
```

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

        TypeError                                 Traceback (most recent call last)

        <ipython-input-59-e8f0e9a9d725> in <module>()
          1 pyLDAvis.enable_notebook()
    ----> 2 vis = pyLDAvis.gensim.prepare(ldamodel, dictionary, doc_term_matrix)


        ~/anaconda3/lib/python3.6/site-packages/pyLDAvis/gensim.py in prepare(topic_model, corp
        116     See `pyLDAvis.prepare` for **kwargs.
        117     """
    --> 118     opts = fp.merge(_extract_data(topic_model, corpus, dictionary, doc_topic_dist)
        119     return vis_prepare(**opts)


        ~/anaconda3/lib/python3.6/site-packages/pyLDAvis/gensim.py in _extract_data(topic_model
         18
         19     if not gensim.matutils.ismatrix(corpus):
    ---> 20         corpus_csc = gensim.matutils.corpus2csc(corpus, num_terms=len(dictionary))
         21     else:
         22         corpus_csc = corpus


        ~/anaconda3/lib/python3.6/site-packages/gensim/matutils.py in corpus2csc(corpus, num_te
        141             if printprogress and docno % printprogress == 0:
        142                 logger.info("PROGRESS: at document #%i/%i", docno, num_docs)
```

```
--> 143             posnext = posnow + len(doc)
    144             # zip(*doc) transforms doc to (token_indices, token_counts]
    145             indices[posnow: posnext], data[posnow: posnext] = zip(*doc) if doc else

    TypeError: object of type 'int' has no len()
```