Exercise 3.1

2020-11-24

The goals for today

At the end of these exercises, you'll be able to:

1. Run and evaluate a topic models
2. Analyze the output of topic models to say something about cultural changes

When finished, hand in the code + answers (preferably in a Jupyter Notebook) on Blackboard.

The manual for the exercises is available at:
https://jveerbeek.gitlab.io/dm-manual/

Today, we are going to explore trends and developments within song lyrics over the last sixty years or so. To do that, we will use an external dataset of 160.856 song lyrics from the following Github page:

https://github.com/hiteshyalamanchili/SongGenreClassification/blob/master/dataset/

Download `english_cleaned_lyrics.zip`, and unpack the zip file on your computer. You'll see a file called `english_cleaned_lyrics.csv`. Drag that file to your working directory.

Because the years of songs in the dataset are horribly incorrect (if we would have to believe this data set, 2006 is the year most Beatles songs were written), I've made a correction to the years for each song. [Some technical details: I used to Spotify API to get the Spotify entry of the song, which is still not always the earliest date, but better assuming than The Beatles and Britney Spears (no judgement) are contemporaries. Another advantage: we now also have a Spotify ID, which could be interesting for your final project.].

You can find the corrected dates in `indx2newdate.p`. And here's the code for

getting the corrected data set (copy-pastable code here!):

```
import pickle
import pandas as pd

PATH_DF = 'path/to/english_cleaned_lyrics.csv'
PATH_CORRECTION = 'path/to/indx2newdate.p'

def load_dataset(data_path, path_correction):
  df = pd.read_csv(data_path)
  indx2newdate = pickle.load(open(PATH_CORRECTION, 'rb'))
  df['year'] = df['index'].apply(lambda x: int(indx2newdate[x
    ][0][:4]) if indx2newdate[x][0] != '' else 0)
  return df[df.year > 1960][['song', 'year', 'artist', 'genre', '
    lyrics']]

dataset = load_dataset(PATH_DF, PATH_CORRECTION)
```

Don't forget to change `path/to/english_cleaned_lyrics.csv` to the directory your files are located! (Or just use `english_cleaned_lyrics.csv` if your files are located in the same directory as your notebook is working from.)

Furthermore, in this exercise, we're going to work with Gensim (https://radimrehurek.com/gensim/) and MALLET (http://mallet.cs.umass.edu/topics.php). Make sure you install/download both.

1. Count how many songs of each genre are in the data set, and pick a genre that 1) you think is interesting to explore and 2) has over 5.000 songs. Make a subset of the data set only containing songs of that genre; this is the data set you work with for the rest of these exercises.
2. Inspect the number of songs for each year, either using a data frame or using a visualization. Do you think you have enough songs for each year (at least more than fifty)? If not, filter out the years that do not contain enough songs.
3. Process the texts of your genre (and *only* your genre!) using Spacy. Extract the lemmatized tokens for each song, and remove stopwords.
4. Create a dictionary and filter out the words that occur less than three times, and all words that occur in over 85% of the documents.
5. Train a topic model with 50 topics and inspect the output, both using the ten most relevant words for each topics, and using pyLDAvis. Now also run a topic model with 20 topics, and one with 100 topics. Be sure to save the models using `lda.save('folder/to/save')` What number of topics does result in the "best" topics? [Note: how you operationalize "best" is up to you]
6. Make a change in the preprocessing stage and run the topic model again. This could be: not removing stop words, only selecting nouns

(or only nouns, adjectives and verbs – something I do quite often when topic modeling), not using lemmas but tokens, etc. Inspect the output. Name one benefit and one downside of the change you selected on the preprocessing stage for finding useful topics.

7. Choose your best topic model. What are the most prominent topics in your corpus? Try to give a label to these topics.
8. Look at the topics that have declined, and the topics that have increased. Think of a label for these topics. Visualize the results and write a (very) short report on the results of this analysis.