CSE354 Sp20 - Assignment 2 Description

Natural Language Processing
Stony Brook University
CSE354 - Spring 2020

Assignment 2
Assigned: 3/8/2020; Due: ~~3/25/2020~~ 4/4/2020 11:59pm

Overview.

Goals.

- Create a basic language model.
- Work with word and phrase counts.
- Implement smooth probability models.
- Build on your existing work for a new domain.
- Generate natural language of a particular genre.

Task Summary. Here you will attempt to build language models that can generate lyrics to songs based on a specified mood (adjective). You will begin by processing a corpus of song lyrics, find lyrics associated with adjectives, and then build language models for some of the more common adjectives. Finally, you will use your language models to generate lyrics associated with the given adjectives.

Submission. All code must be placed in a single .py file, submitted to blackboard, and titled:

```
mood_lyric_generator_lastname_id.py
```

Where lastname_id is your lastname and id. If blackboard rejects your file because “it may contain a security threat, then (and only then) you can place it in a zip with nothing else and submit the .zip file.

Requirements. You must use Python version 3.5 or later. The only data science, machine learning, statistics, or nlp libraries that you may import are:

```
re, numpy as np, sklearn, scipy.stats, pandas, csv
```

For this assignment you will not be provided with a template of code but rather you must keep your code for all stages of the assignment in a single python file (though you are encouraged to use multiple.

Code should be clearly organized and commented: every loop conditional or variable assignment should at least have a brief comment description of it. Every step from 1.2 onward should have a clear comment where it begins “##STEP X.X Brief description of step”. Output will provide hints to issues, grading will consider the content of your code to verify all steps have been implemented in accordance with the concepts in the class and book.

Academic Honesty. Copying chunks of code from other students, online or other resources is prohibited. You are responsible for both (1) not copying others work, and (2) making sure your work is not accessible to others. Assignments will be extensively checked for copying of others’ work. Please see the syllabus for additional policies.

Stage 1: Get song lyric corpus and tokenize it

Step 1.1: Download data from https://www.kaggle.com/mousehead/songlyrics#songdata.csv
- To get the file, you will need to sign up for a Kaggle account (or use an existing account).
- Make sure to unzip and save the file as "songdata.csv".

(This is not a step for your code but something to do separately)

Step 1.2: Read the csv into memory.
- Give each row a unique id based on the *"artist-song"* replacing spaces with underscores and lowercasing. For example, for "ABBA" song "Burning My Bridges" give it the id: *"abba-burning_my_bridges"*
- Make sure to store it such that you can easily access both the song lyrics and song.

(From here forward all steps should be represented in your code: make sure to mark them "## STEP 1.2: Read the csv into memory")

Step 1.3: Tokenize the song *titles*.
- Use the tokenizer you developed for assignment 1 (copy the method code into this file).
- Treat "<word>", "</word>", and "[word]" as a single token. (take everything between the open and closed angled brackets or square brackets, and include the brackets as part of the word).
- You can update the tokenizer to produce better output
  (hint: don't spend a lot of time improving your tokenizer until you finish the other stages).

Step 1.4: Tokenize the song *lyrics*.
(lyrics are contained in the "text" column of the dataset.)
- Add special tokens "<s>" at the beginning of lyrics, and "</s>" at the end
- Replace newlines with the token "<newline>"
- Use the same tokenizer method as above.

You can choose whether to use a dictionary, list, pandas dataframe, or any combination of these to store the data, but make sure each song record has the following elements:
artist-song, title_tokenized, lyrics_tokenized

Stage 1 Checkpoint: Print the tokenized title and lyrics for the follow artist-songs:
- *abba-burning_my_bridges*
- *beach_boys-do_you_remember?*
- *avril_lavigne-5,_4,_3,_2,_1_(countdown)*
- *michael_buble-l-o-v-e*

=== Sample Output ===

['Burning', 'My', 'Bridges']
['<s>', 'Well', ',', 'you', 'hoot', 'and', 'you', 'holler', 'and', 'you', 'make', 'me', 'mad', '<newline>', 'And', "I've", 'always', 'been', 'under', 'your', 'heel', '<newline>', 'Holy', 'christ', 'what', 'a', 'lousy', 'deal', '<newline>', 'Now', "I'm", 'sick', 'and', 'tired', 'of', 'your', 'tedious', 'ways', '<newline>', 'And', 'I', "ain't", 'gonna', 'take', 'it', 'no', 'more', '<newline>', 'Oh', 'no', 'no', '-', 'walkin', 'out', 'that', 'door', '<newline>', '<newline>', 'Burning', 'my', 'bridges', ',', 'cutting', 'my', 'tie', '<newline>', 'Once', 'again', 'I', 'wanna', 'look', 'into', 'the', 'eye', '<newline>', 'Being', 'myself', '<newline>', 'Counting', 'my', 'pride', '<newline>', 'No', 'un-right', "neighbour's", 'gonna', 'take', 'me', 'for', 'a', 'ride', '<newline>', 'Burning', 'my', 'bridges', '<newline>', 'Moving', 'at', 'last', '<newline>', 'Girl', "I'm", 'leaving', 'and', "I'm", 'burying', 'the', 'past', '<newline>', 'Gonna', 'have', 'peace', 'now', '<newline>', 'You', 'can', 'be', 'free', '<newline>', 'No', 'one', 'here', 'will', 'make', 'a', 'sucker', 'out', 'of', 'me', '<newline>', '<newline>', '</s>']

['Do', 'You', 'Remember', '?']
['<s>', 'Little', 'Richard', 'sang', 'it', 'and', 'Dick', 'Clark', 'brought', 'it', 'to', 'life', '<newline>', 'Danny', 'And', 'The', 'Juniors', 'hit', 'a', 'groove', ',', 'stuck', 'as', 'sharp', 'as', 'a', 'knife', '<newline>', 'Well', 'now', 'do', 'you', 'remember', 'all', 'the', 'guys', 'that', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', '<newline>', 'Chuck', "Berry's", 'gotta', 'be', 'the', 'greatest', 'thing', "that's", 'come', 'along', '<newline>', 'hum', 'diddy', 'waddy', ',', 'hum', 'diddy', 'wadda', '<newline>', 'He', 'made', 'the', 'guitar', 'beats', 'and', 'wrote', 'the', 'all-time', 'greatest', 'song', '<newline>', 'hum', 'diddy', 'waddy', ',', 'hum', 'diddy', 'wadda', '<newline>', 'Well', 'now', 'do', 'you', 'remember', 'all', 'the', 'guys', 'that', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'hum', 'diddy', 'waddy', 'doo', '<newline>', '<newline>', 'Elvis', 'Presley', 'is', 'the', 'king', '<newline>', "He's", 'the', 'giant', 'of', 'the', 'day', '<newline>', 'Paved', 'the', 'way', 'for', 'the', 'rock', 'and', 'roll', 'stars', '<newline>', 'Yeah', 'the', 'critics', 'kept', 'a', 'knockin', '<newline>', 'But', 'the', 'stars', 'kept', 'a', 'rockin', '<newline>', 'And', 'the', 'choppin', "didn't", 'get', 'very', 'far', '<newline>', '<newline>', 'Goodness', 'gracious', 'great', 'balls', 'of', 'fire', '<newline>', "Nothin's", 'really', 'movin', 'till', 'the', "saxophone's", 'ready', 'to', 'blow', '<newline>', 'do', 'you', 'remember', ',', 'do', 'you', 'remember', '<newline>', 'And', 'the', "beat's", 'not', 'jumpin', 'till', 'the', 'drummer', 'says', "he's", 'ready', 'to', 'go', '<newline>', 'do', 'you', 'remember', ',', 'do', 'you', 'remember', '<newline>', 'Well', 'now', 'do', 'you', 'remember', 'all', 'the', 'guys', 'that', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'do', 'you', 'remember', '<newline>', '<newline>', "Let's", 'hear', 'the', 'high', 'voice', 'wail', 'ooooooooo', '<newline>', 'And', 'hear', 'the', 'voice', 'down', 'low', 'wah-ah', 'ah-ah', '<newline>', "Let's", 'hear', 'the', 'background', '<newline>', 'Um', 'diddy', 'wadda', ',', 'um', 'diddy', 'wadda', '<newline>', 'Um', 'diddy', 'wadda', ',', 'um', 'diddy', 'wadda', '<newline>', 'They', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'Um', 'diddy', 'wadda', ',', 'um', 'diddy', 'wadda', '<newline>', 'They', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'Um', 'diddy', 'wadda', ',', 'um', 'diddy', 'wadda', '<newline>', 'They', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', '<newline>', '</s>']

['5', ',', '4', ',', '3', ',', '2', ',', '1', 'Countdown']
['<s>', 'Verse', '<newline>', 'Countin', 'down', ',', "it's", 'new', 'years', 'eve', ',', '<newline>', 'You', 'come', 'on', 'over', ',', 'then', 'start', ',', 'askin', 'me', ',', '<newline>', 'Hey', 'girl', ',', 'you', 'wanna', 'dance', '?', '<newline>', 'I', 'try', 'to', 'say', 'no', ',', 'but', 'it', 'came', 'out', 'yes', 'and', ',', '<newline>', 'Here', 'it', 'goes', ',', 'middle', 'of', 'the', 'dance', 'floor', ',', '<newline>', "We're", 'both', ',', 'pretty', 'nervous', ',', 'I', 'can', 'tell', 'by', 'look', 'in', 'his', 'eyes', ',', '<newline>', 'Then', 'came', 'the', 'count', 'down', ',', '<newline>', '<newline>', 'Chorus', '<newline>', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', 'New', 'years', '!', '<newline>', 'Everybody', 'shouts', ',', 'and', 'here', 'it', 'goes', ',', '<newline>', 'I', 'wanna', 'see', 'it', 'flow', ',', '<newline>', 'We', ',', 'look', 'at', 'the', 'screen', ',', '<newline>', 'The', 'camera', 'was', 'focused', 'on', 'us', ',', '<newline>', 'You', 'blushed', 'and', 'turned', 'away', ',', '<newline>', 'I', 'kissed', 'you', 'on', 'the', 'cheek', 'and', 'ran', 'away', ',', '<newline>', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', '<newline>', 'Verse', '2', '<newline>', 'My', 'birthday', ',', "everybody's", 'here', ',', '<newline>', 'All', 'except', 'for', 'one', ',', '<newline>', 'At', 'the', 'sleepover', ',', 'you', 'showed', 'right', 'then', ',', '<newline>', 'All', 'the', 'girls', 'scream', ',', 'except', 'for', 'me', ',', '<newline>', 'You', 'have', ',', 'roses', 'in', 'your', 'hand', ',', '<newline>', 'Gave', 'em', 'to', 'me', 'and', 'whispered', ',', '<newline>', 'Payin', 'you', 'back', ',', '<newline>', 'Then', ',', 'all', 'the', 'girls', '<newline>', 'Started', 'to', 'get', 'that', 'feeling', '.', '.', '.', '<newline>', '<newline>', 'Chorus', '<newline>', '<newline>', 'Verse', '3', '<newline>', 'Then', 'they', 'shouted', '<newline>', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', 'Here', 'it', 'goes', ',', '<newline>', 'Then', 'he', 'kissed', 'me', 'on', 'the', 'cheek', 'and', 'said', ',', '<newline>', 'Just', 'remember', 'this', 'day', ',', 'Later', 'on', 'in', 'your', 'life', ',', '<newline>', "You'll", 'recall', 'this', 'day', ',', 'and', 'wonder', 'why', ',', '<newline>', 'I', "didn't", 'count', 'down', ',', '<newline>', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', 'Then', 'I', 'got', 'that', 'feeling', 'when', 'you', 'left', ',', '<newline>', 'It', 'was', 'the', 'count', 'down', ',', '<newline>', 'I', 'got', 'that', 'feeling', '<newline>', 'It', 'was', 'a', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', 'Once', 'again', '<newline>', '<newline>', '</s>']

['L-O-V-E']
['<s>', 'L', 'is', 'for', 'the', 'way', 'you', 'look', 'at', 'me', '<newline>', 'O', 'is', 'for', 'the', 'only', 'one', 'I', 'see', '<newline>', 'V', 'is', 'very', ',', 'very', 'extraordinary', '<newline>', 'E', 'is', 'even', 'more', 'than', 'anyone', 'that', 'you', 'adore', '<newline>', '<newline>', 'And', 'love', 'is', 'all', 'that', 'I', 'can', 'give', 'to', 'you', '<newline>', 'Love', 'is', 'more', 'than', 'just', 'a', 'game', 'for', 'two', '<newline>', 'Two', 'in', 'love', 'can', 'make', 'it', '<newline>', 'Take', 'my', 'heart', 'but', 'please', "don't", 'break', 'it', '<newline>', 'Love', 'was', 'made', 'for', 'me', 'and', 'you', '<newline>', '<newline>', 'L', 'is', 'for', 'the', 'way', 'you', 'look', 'at', 'me', '<newline>', 'O', 'is', 'for', 'the', 'only', 'one', 'I', 'see', '<newline>', 'V', 'is', 'very', ',', 'very', 'extraordinary', '<newline>', 'E', 'is', 'even', 'more', 'than', 'anyone', 'that', 'you', 'adore', '<newline>', '<newline>', 'And', 'love', 'is', 'all', 'that', 'I', 'can', 'give', 'to', 'you', '<newline>', 'Love', ',', 'love', ',', 'love', 'is', 'more', 'than', 'just', 'a', 'game', 'for', 'two', '<newline>', 'Two', 'in', 'love', 'can', 'make', 'it', '<newline>', 'Take', 'my', 'heart', 'but', 'please', "don't", 'break', 'it', '<newline>', 'Cause', 'love', 'was', 'made', 'for', 'me', 'and', 'you', '<newline>', 'I', 'said', 'love', 'was', 'made', 'for', 'me', 'and', 'you', '<newline>', 'You', 'know', 'that', 'love', 'was', 'made', 'for', 'me', 'and', 'you', '<newline>', '<newline>', '</s>']

#Case Insensitive:
```
['Burning', 'My', 'Bridges']
['<s>', 'well', ',', 'you', 'hoot', 'and', 'you', 'holler', 'and', 'you', 'make',
'me', 'mad', '<newline>', 'and', "i've", 'always', 'been', 'under', 'your', 'heel',
```

'<newline>', 'holy', 'christ', 'what', 'a', 'lousy', 'deal', '<newline>', 'now',
"i'm", 'sick', 'and', 'tired', 'of', 'your', 'tedious', 'ways', '<newline>', 'and',
'i', "ain't", 'gonna', 'take', 'it', 'no', 'more', '<newline>', 'oh', 'no', 'no', '-',
'walkin', 'out', 'that', 'door', '<newline>', '<newline>', 'burning', 'my', 'bridges',
',', 'cutting', 'my', 'tie', '<newline>', 'once', 'again', 'i', 'wanna', 'look',
'into', 'the', 'eye', '<newline>', 'being', 'myself', '<newline>', 'counting', 'my',
'pride', '<newline>', 'no', 'un-right', "neighbour's", 'gonna', 'take', 'me', 'for',
'a', 'ride', '<newline>', 'burning', 'my', 'bridges', '<newline>', 'moving', 'at',
'last', '<newline>', 'girl', "i'm", 'leaving', 'and', "i'm", 'burying', 'the', 'past',
'<newline>', 'gonna', 'have', 'peace', 'now', '<newline>', 'you', 'can', 'be', 'free',
'<newline>', 'no', 'one', 'here', 'will', 'make', 'a', 'sucker', 'out', 'of', 'me',
'<newline>', '<newline>', '</s>']

['Do', 'You', 'Remember', '?']
['<s>', 'little', 'richard', 'sang', 'it', 'and', 'dick', 'clark', 'brought', 'it',
'to', 'life', '<newline>', 'danny', 'and', 'the', 'juniors', 'hit', 'a', 'groove',
',', 'stuck', 'as', 'sharp', 'as', 'a', 'knife', '<newline>', 'well', 'now', 'do',
'you', 'remember', 'all', 'the', 'guys', 'that', 'gave', 'us', 'rock', 'and', 'roll',
'<newline>', '<newline>', 'chuck', "berry's", 'gotta', 'be', 'the', 'greatest',
'thing', "that's", 'come', 'along', '<newline>', 'hum', 'diddy', 'waddy', ',', 'hum',
'diddy', 'wadda', '<newline>', 'he', 'made', 'the', 'guitar', 'beats', 'and', 'wrote',
'the', 'all-time', 'greatest', 'song', '<newline>', 'hum', 'diddy', 'waddy', ',',
'hum', 'diddy', 'wadda', '<newline>', 'well', 'now', 'do', 'you', 'remember', 'all',
'the', 'guys', 'that', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'hum',
'diddy', 'waddy', 'doo', '<newline>', '<newline>', 'elvis', 'presley', 'is', 'the',
'king', '<newline>', "he's", 'the', 'giant', 'of', 'the', 'day', '<newline>', 'paved',
'the', 'way', 'for', 'the', 'rock', 'and', 'roll', 'stars', '<newline>', 'yeah',
'the', 'critics', 'kept', 'a', 'knockin', '<newline>', 'but', 'the', 'stars', 'kept',
'a', 'rockin', '<newline>', 'and', 'the', 'choppin', "didn't", 'get', 'very', 'far',
'<newline>', '<newline>', 'goodness', 'gracious', 'great', 'balls', 'of', 'fire',
'<newline>', "nothin's", 'really', 'movin', 'till', 'the', "saxophone's", 'ready',
'to', 'blow', '<newline>', 'do', 'you', 'remember', ',', 'do', 'you', 'remember',
'<newline>', 'and', 'the', "beat's", 'not', 'jumpin', 'till', 'the', 'drummer',
'says', "he's", 'ready', 'to', 'go', '<newline>', 'do', 'you', 'remember', ',', 'do',
'you', 'remember', '<newline>', 'well', 'now', 'do', 'you', 'remember', 'all', 'the',
'guys', 'that', 'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'do', 'you',
'remember', '<newline>', '<newline>', "let's", 'hear', 'the', 'high', 'voice', 'wail',
'ooooooooooo', '<newline>', 'and', 'hear', 'the', 'voice', 'down', 'low', 'wah-ah',
'ah-ah', '<newline>', "let's", 'hear', 'the', 'background', '<newline>', 'um',
'diddy', 'wadda', ',', 'um', 'diddy', 'wadda', '<newline>', 'um', 'diddy', 'wadda',
',', 'um', 'diddy', 'wadda', '<newline>', 'they', 'gave', 'us', 'rock', 'and', 'roll',
'<newline>', 'um', 'diddy', 'wadda', ',', 'um', 'diddy', 'wadda', '<newline>', 'they',
'gave', 'us', 'rock', 'and', 'roll', '<newline>', 'um', 'diddy', 'wadda', ',', 'um',
'diddy', 'wadda', '<newline>', 'they', 'gave', 'us', 'rock', 'and', 'roll',
'<newline>', '<newline>', '</s>']

['5', ',', '4', ',', '3', ',', '2', ',', '1', 'Countdown']
['<s>', 'verse', '<newline>', 'countin', 'down', ',', "it's", 'new', 'years', 'eve',
',', '<newline>', 'you', 'come', 'on', 'over', ',', 'then', 'start', ',', 'askin',
'me', ',', '<newline>', 'hey', 'girl', ',', 'you', 'wanna', 'dance', '?', '<newline>',
'i', 'try', 'to', 'say', 'no', ',', 'but', 'it', 'came', 'out', 'yes', 'and', ',',
'<newline>', 'here', 'it', 'goes', ',', 'middle', 'of', 'the', 'dance', 'floor', ',',
'<newline>', "we're", 'both', ',', 'pretty', 'nervous', ',', 'i', 'can', 'tell', 'by',
'look', 'in', 'his', 'eyes', ',', '<newline>', 'then', 'came', 'the', 'count', 'down',
',', '<newline>', '<newline>', 'chorus', '<newline>', '5', ',', '4', ',', '3', ',',
'2', ',', '1', '!', '<newline>', 'new', 'years', '!', '<newline>', 'everybody',
'shouts', ',', 'and', 'here', 'it', 'goes', ',', '<newline>', 'i', 'wanna', 'see',

```
'it', 'flow', ',', '<newline>', 'we', ',', 'look', 'at', 'the', 'screen', ',',
'<newline>', 'the', 'camera', 'was', 'focused', 'on', 'us', ',', '<newline>', 'you',
'blushed', 'and', 'turned', 'away', ',', '<newline>', 'i', 'kissed', 'you', 'on',
'the', 'cheek', 'and', 'ran', 'away', ',', '<newline>', '5', ',', '4', ',', '3', ',',
'2', ',', '1', '!', '<newline>', '<newline>', 'verse', '2', '<newline>', 'my',
'birthday', ',', "everybody's", 'here', ',', '<newline>', 'all', 'except', 'for',
'one', ',', '<newline>', 'at', 'the', 'sleepover', ',', 'you', 'showed', 'right',
'then', ',', '<newline>', 'all', 'the', 'girls', 'scream', ',', 'except', 'for', 'me',
',', '<newline>', 'you', 'have', ',', 'roses', 'in', 'your', 'hand', ',', '<newline>',
'gave', 'em', 'to', 'me', 'and', 'whispered', ',', '<newline>', 'payin', 'you',
'back', ',', '<newline>', 'then', ',', 'all', 'the', 'girls', '<newline>', 'started',
'to', 'get', 'that', 'feeling', '.', '.', '.', '<newline>', '<newline>', 'chorus',
'<newline>', '<newline>', 'verse', '3', '<newline>', 'then', 'they', 'shouted',
'<newline>', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', 'here',
'it', 'goes', ',', '<newline>', 'then', 'he', 'kissed', 'me', 'on', 'the', 'cheek',
'and', 'said', ',', '<newline>', 'just', 'remember', 'this', 'day', ',', 'later',
'on', 'in', 'your', 'life', ',', '<newline>', "you'll", 'recall', 'this', 'day', ',',
'and', 'wonder', 'why', ',', '<newline>', 'i', "didn't", 'count', 'down', ',',
'<newline>', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>', 'then',
'i', 'got', 'that', 'feeling', 'when', 'you', 'left', ',', '<newline>', 'it', 'was',
'the', 'count', 'down', ',', '<newline>', 'i', 'got', 'that', 'feeling', '<newline>',
'it', 'was', 'a', '5', ',', '4', ',', '3', ',', '2', ',', '1', '!', '<newline>',
'once', 'again', '<newline>', '<newline>', '</s>']

['L-O-V-E']
['<s>', 'l', 'is', 'for', 'the', 'way', 'you', 'look', 'at', 'me', '<newline>', 'o',
'is', 'for', 'the', 'only', 'one', 'i', 'see', '<newline>', 'v', 'is', 'very', ',',
'very', 'extraordinary', '<newline>', 'e', 'is', 'even', 'more', 'than', 'anyone',
'that', 'you', 'adore', '<newline>', '<newline>', 'and', 'love', 'is', 'all', 'that',
'i', 'can', 'give', 'to', 'you', '<newline>', 'love', 'is', 'more', 'than', 'just',
'a', 'game', 'for', 'two', '<newline>', 'two', 'in', 'love', 'can', 'make', 'it',
'<newline>', 'take', 'my', 'heart', 'but', 'please', "don't", 'break', 'it',
'<newline>', 'love', 'was', 'made', 'for', 'me', 'and', 'you', '<newline>',
'<newline>', 'l', 'is', 'for', 'the', 'way', 'you', 'look', 'at', 'me', '<newline>',
'o', 'is', 'for', 'the', 'only', 'one', 'i', 'see', '<newline>', 'v', 'is', 'very',
',', 'very', 'extraordinary', '<newline>', 'e', 'is', 'even', 'more', 'than',
'anyone', 'that', 'you', 'adore', '<newline>', '<newline>', 'and', 'love', 'is',
'all', 'that', 'i', 'can', 'give', 'to', 'you', '<newline>', 'love', ',', 'love', ',',
'love', 'is', 'more', 'than', 'just', 'a', 'game', 'for', 'two', '<newline>', 'two',
'in', 'love', 'can', 'make', 'it', '<newline>', 'take', 'my', 'heart', 'but',
'please', "don't", 'break', 'it', '<newline>', 'cause', 'love', 'was', 'made', 'for',
'me', 'and', 'you', '<newline>', 'i', 'said', 'love', 'was', 'made', 'for', 'me',
'and', 'you', '<newline>', 'you', 'know', 'that', 'love', 'was', 'made', 'for', 'me',
'and', 'you', '<newline>', '<newline>', '</s>']
```

Stage 2: Code an add1-trigram language model method

Note: Develop this method by using a subset of ~1,000 lyrics. Otherwise, it will likely run very slow or even run out of memory. Then, test it on the first 5,000 lyrics.

Step 2.1: Create a vocabulary of words from lyrics:

- Count how often every word appears, case insensitive
- Keep those words appearing more than 2 times (other words will be OOV).

- Create a dictionary of word=>count

Step 2.2: Create a bigram matrix (rows as previous word; columns as current word)
- Consider a word "<OOV>" if it is not in the vocabulary from 2.1.
- Count how frequent each pair of words occur (including OOVs).

Hint: It is useful to store this matrix as a double dictionary: bigramCounts[word1][word2] = count

Step 2.3: Create a trigram matrix (rows as previous bigram; columns as current word)
- Count how frequent each word occurs after a bigram

Hint: It is useful to store this matrix as a double dictionary: trigramCounts[(word1,word2)][word3] = count

Step 2.4: Create a method to calculate the probability of all possible current words wi given either a single previous word ($wi\text{-}1$ -- a bigram model) or two previous words ($wi\text{-}1$ and $wi\text{-}2$ -- a trigram model).
- Start with making a pool of potential wi to generate: Restrict to only consider generating those potential wi that ever appeared after $wi\text{-}1$. (i.e. just those words where there are bigrams counts). Also restrict to no "<OOV>" in the pool of possible words (i.e. but it's ok if it's in $wi\text{-}1$ or $wi\text{-}2$).
  - If no words are available based on $wi\text{-}1$, then backoff to a non-smoothed unigram model (i.e. just take a draw from all word probabilities of occuring by themselves; the only time this should happen is when only "<OOV>" follow $wi\text{-}1$). Ignore next steps.
- If only a single previous word is provided, then use add-one smoothing on the bigram model.
- If two previous words are provided then calculate the probability of the next word based on both the bigram and trigram model:
  - $P(wi \mid wi\text{-}1, wi\text{-}2) = (P(wi \mid wi\text{-}1) + P(wi \mid wi\text{-}1, wi\text{-}2)) / 2$
  - This is called "interpolating" the models which tends to produce more robust probabilities in practice.
  - You will need to appropriately calculate $P(wi \mid wi\text{-}1, wi\text{-}2)$ even if the particular trigram was never observed.
- Make sure to use add-one smoothing for both the trigram and bigram probabilities. Note: because we are not returning probabilities for all words in the vocabulary, with add1 smoothing, the sum of the word probabilities returned will usually be less than 1 (because the probabilities for additional vocabulary words are not being returned).

Note: It is best to store the raw counts for steps 2-1 through 2.3. Then, within this method the smoothing is actually done when calculating the probabilities. This way you only have to store integer counts for those that exist. The method will need to be able to handle when it doesn't find any bigram or trigram occurrences (i.e. it will just count as 1 occurrence). These steps are essentially what is needed to "train" the language model. Step 2.4 is what is needed to apply the language model.

Hint: Trigram probabilities can easily be computed on the fly, even with add-one smoothing:
p(w3 | w1, w2) = trigramCounts[(w1, w2)][w3] + 1 / bigramCounts[(w1, w2)] + VOCAB_SIZE

Hint: The following will randomly select a word from a list given probabilities for each:
np.random.choice(['word1', 'word2', 'word3'], 1, p=[.6, .3, .1]). Since you are not passing all probabilities to the word choice function, you will need to re-normalize the probabilities that you do pass: divide them all by the sum of the probabilities before passing to random.choice.

Stage 2 Checkpoint: Based on just the first 5,000 lyrics, print the following (add-1 smoothed) probabilities:
- p(wi = "you"| wi-2 = "I", wi-1 = "love".)
- p(wi="special"| wi-1="midnight")
- p(wi="special"| wi-1="very")
- p(wi="special"| wi-2="something ",wi-1="very")
- p(wi="funny"| wi-2="something ",wi-1="very")

=== Sample Output ===
#note: due to variance in tokenization, differences are expected. Being within +- 15% is fine.

```
p( you | ('i', 'love') ) = 0.05453
p( special | ('midnight',) ) = 0.00050
p( special | ('very',) ) = 0.00164
p( special | ('something', 'very') ) = 0.00092
p( funny | ('something', 'very') ) = 0.00015
```

Stage 3: Create adjective-specific language models

Step 3.1: Train your model based on your assignment 1 code

- Call training on "daily547.conll" assuming it is also available.
- Welcome to add features to the model to improve it
- Save the model to a variable

Note: Recommend waiting to add features to the model until you have completed all steps in this stage.

Step 3.2: Extract features from titles for adjective classifier

- Convert the titles to lower case instead of title case so the adjective classifier will work better.
- Use your code from assignment 1

Note: the adjective classifier has a separate vocabulary than the language model. The language model is applied to lyrics and thus it's vocabulary (as described in stage 2) is from lyrics. The adjective classifier is trained on the data for assignment 1 (daily.conll) and thus its vocabulary is from that.

Step 3.3: Find adjectives in each title

- For this, use the full set of song records
- Apply your saved model to the features extracted from each title and, for each unique adjective (non-case sensitive: lowercase), track the songs (i.e. the unique *artist-title*) which contained the adjective in the title.
- For each adjective that occurs in more than 10 artist-titles, store all the lyrics of those artist-titles

Note: Recommended to store songs associated with adjectives as a dictionary: 'adjective':[lyric1, lyric2…] or 'adjective':[artist-title1, artist-title2, ….]
Note: This step uses *title*; not lyrics

Step 3.4: Build a separate language model for the lyrics associated with each adjective from the previous step.

- Use your methods from stage 2 to create a separate language model for each adjective. (Building the language model consists of recording the counts for unigrams, bigrams, and trigrams, which is what you do in stage 2).
- Use the lyrics of each song rather than the title itself for this.

Note: This step uses the song *lyrics* (which contains many more words than titles) for the language model.

Step 3.5: Create a method to generate lyrics given an adjective (i.e. the language that was trained on lyrics for titles with the given adjective).

- A basic method to generate language using the language models is discussed in the second paragraph of SLP 3.3 and visualized in p.39 of the SLP3 chapter 3 slides. Here are more details:
  - Start with generating the next word from the bigram model given only "<s>" as the first word.

- After choosing the first word, continue with the next but using the previous two words (e.g. if “walking” was generated from after “<s>” now you would query the model with “<s>” and “first_word”). The method from step 2.4 is already restricting possible next words.
- Stop once generating “</s>” or if reaching max_length = 32 words.

Hint: The following will randomly select a word from a list given probabilities for each: np.random.choice(['word1', 'word2', 'word3'], 1, p=[.6, .3, .1]). Since you are not passing all probabilities to the word choice function, you will need to re-normalize the probabilities that you do pass: divide them all by the sum of the probabilities before passing to random.choice.

Final Checkpoints:

- Print all artist-songs associated with the following adjectives:
  - good
  - happy
  - afraid
  - red
  - blue

  #note: blue is not in the part of speech training corpus as an adjective; however, it can be captured as an adjective thanks to the features around it as long as you handle OOC correctly in the POS tagger. This test is meant to motivate improving your adjective classifier (again, that is suggested as the last thing you do after getting everything else working).
- Print 3 generated lyrics for the the same adjectives listed about

Make sure the main for your final code, when turned in, runs through the check points for all three stages.

=== Sample Output ===
#note: checking for meaningful trigrams; .

```
Good: <s> learning don't rush one in life this is you <newline> this
make heart of want you to believe <newline> i good don't it's rush
gonna make to believe </s>
```

Published by Google Drive – Report Abuse – Updated automatically every 5 minutes