dlnd_tv_script_generation

July 23, 2017

1 TV Script Generation

In this project, you'll generate your own Simpsons TV scripts using RNNs. You'll be using part of the Simpsons dataset of scripts from 27 seasons. The Neural Network you'll build will generate a new TV script for a scene at Moe's Tavern. ## Get the Data The data is already provided for you. You'll be using a subset of the original dataset. It consists of only the scenes in Moe's Tavern. This doesn't include other versions of the tavern, like "Moe's Cavern", "Flaming Moe's", "Uncle Moe's Family Feed-Bag", etc..

In [1]:
```
"""
DON'T MODIFY ANYTHING IN THIS CELL
"""
import helper

data_dir = './data/simpsons/moes_tavern_lines.txt'
text = helper.load_data(data_dir)
# Ignore notice, since we don't use it for analysing the data
text = text[81:]
```

1.1 Explore the Data

Play around with `view_sentence_range` to view different parts of the data.

In [2]:
```
view_sentence_range = (0, 10)

"""
DON'T MODIFY ANYTHING IN THIS CELL
"""
import numpy as np

print('Dataset Stats')
print('Roughly the number of unique words: {}'.format(len({word: None for word in text.s
scenes = text.split('\n\n')
print('Number of scenes: {}'.format(len(scenes)))
sentence_count_scene = [scene.count('\n') for scene in scenes]
print('Average number of sentences in each scene: {}'.format(np.average(sentence_count_s
```

```
        sentences = [sentence for scene in scenes for sentence in scene.split('\n')]
        print('Number of lines: {}'.format(len(sentences)))
        word_count_sentence = [len(sentence.split()) for sentence in sentences]
        print('Average number of words in each line: {}'.format(np.average(word_count_sentence))

        print()
        print('The sentences {} to {}:'.format(*view_sentence_range))
        print('\n'.join(text.split('\n')[view_sentence_range[0]:view_sentence_range[1]]))
```

```
Dataset Stats
Roughly the number of unique words: 11492
Number of scenes: 262
Average number of sentences in each scene: 15.248091603053435
Number of lines: 4257
Average number of words in each line: 11.50434578341555

The sentences 0 to 10:
Moe_Szyslak: (INTO PHONE) Moe's Tavern. Where the elite meet to drink.
Bart_Simpson: Eh, yeah, hello, is Mike there? Last name, Rotch.
Moe_Szyslak: (INTO PHONE) Hold on, I'll check. (TO BARFLIES) Mike Rotch. Mike Rotch. Hey, has an
Moe_Szyslak: (INTO PHONE) Listen you little puke. One of these days I'm gonna catch you, and I'm
Moe_Szyslak: What's the matter Homer? You're not your normal effervescent self.
Homer_Simpson: I got my problems, Moe. Give me another one.
Moe_Szyslak: Homer, hey, you should not drink to forget your problems.
Barney_Gumble: Yeah, you should only drink to enhance your social skills.
```

1.2 Implement Preprocessing Functions

The first thing to do to any dataset is preprocessing. Implement the following preprocessing functions below: - Lookup Table - Tokenize Punctuation

1.2.1 Lookup Table

To create a word embedding, you first need to transform the words to ids. In this function, create two dictionaries: - Dictionary to go from the words to an id, we'll call `vocab_to_int` - Dictionary to go from the id to word, we'll call `int_to_vocab`

Return these dictionaries in the following tuple `(vocab_to_int, int_to_vocab)`

```
In [24]: import numpy as np
         import problem_unittests as tests

         def create_lookup_tables(text):
             """
             Create lookup tables for vocabulary
             :param text: The text of tv scripts split into words
```