DEEPak Chopra: A Text Analysis of Deepak Chopra's Tweets

Saisharan Chimbili
chimbili@wisc.edu

Grant Dakovich
gdakovich@wisc.edu

Nick Vander Heyden
nvanderheyde@wisc.edu

Abstract

This STAT 479 project is a text mimicker that utilizes many functionalities and topics covered throughout the course this semester. More specifically this project aims to mimic the tweets of celebrity and known public speaker Deepak Chopra. Deepak Chopra was chosen because of his unique way of thinking and usage of words. Utilizing python to clean data from his tweets and one of his books, this report has a dataset with over 6000 rows of his sentences. This data is then put through a complex double-model to create unique text that resembles Deepaks word choice and sentence structure. This double model includes both a character level RNN and a parts of speech model made possible thanks to the power of the NLTK python package. These models work in a way that the Char-RNN does the text generation, and the part of speech model verifies if it is a valid English word for the needed part of speech. After training models there were some difficulties realized along the way. These included the fact that the dataset is likely too small for a deep learning project, and most importantly it can be difficult to objectively measure the accuracy of a text mimicker. However in the end we find that the goals of this project were achieved through the resulting generated tweets.

1. Introduction

We will be creating a deep learning model to replicate tweets from Deepak Chopra. Deepak Chopra is an author who promotes mindfulness and alternative medicine. The reason why we chose Deepak Chopra is that his tweets have very obscure topics and unique vocabulary. Through his work, he has coined many unique phrases that grabbed our attention, often tweets tagged with the hashtag #Cosmic-Consciousness. The goal is to train a model on his previous tweets and be able to reproduce phrases that seem like something he might tweet himself, generated bits of cosmic consciousness. Overall, there should be very limited differences between his real tweets and our fake tweets. While this is hard to measure explicitly, due to the subjectiveness of mimicking, we can utilize our peers to determine the models effectiveness.

Here is a link to his twitter account, where one can observe the word soup for themselves. It is easy to find many of his common key words including reality, experience, universe, and awareness.

1.1. Motivation

Language modeling is something that can dramatically change the course of literature in our society. With its power maximized, we can begin to emulate the work of journalists, and in this case, an author of influence. Additionally, as we continue into an age of information, there is value in the ability to provide in-demand content as quickly as possible. Today many of us utilize social media to learn the thoughts and knowledge of those we look up to as instantaneously as possible. Deepak Chopras Twitter account has over three million followers, so it can be implied that his statements carry great influence. The ability to provide thoughts of the same style as Deepak in a greater volume could gain an equal or greater following, and thus have great influence and power. There is also some personal motivation amongst us as we find some of Deepaks tweets and quotes to be very unique especially in their word choice. We are very curious to see if we can create an accurate enough of a model that our peers are unable to determine if tweets are from the mind of Deepak or from a computer. See figure 2 on the next page for a visual representation of an n-gram language model that will be similar to what will be used on this project.

If we were to successfully implement this model, we could potentially mimic other peoples social media and essentially create bots that run various social media websites, and could handle tasks like basic customer Q&A. Additionally, many celebrities are keen about fan outreach but fail to reach out to many of their fans due to lack of time. If we understand how individuals hold conversations, we will be able to create messages based off what they normal reply with. That being said, this can potentially decrease the amount of time celebrities/people spend on small tasks such as social media and still creating connections with their fans/friends, while focusing on their actual work. We find that language modeling can positively impact our society and are looking forward to working with it.

2. Related Work

Towards Data Science implemented a Char-RNN by using Trumps tweets to create the same tone as his tweets. The structure and goal of our Char-RNN and the Trump project is very similar. However, the Trump project does not use a Part of Speech-Recurrent Neural Network like we do to maintain a better sentence structure (reference section 3.2 POS-RNN Model Overview for more information). Additionally, the Trump projects data takes into consideration both capitalization and punctuation while our project does only takes into consideration periods and lowercase letters.

Here is a link to the article.

Additionally, we received assistance for our code by using code from Python Machine Learning by Sebastian Raschka & Vahid Mirjalili for our Char-RNN model. Raschka and Mirjalili implemented a Char-RNN to create Shakespeare passages from his books. Again, the major difference between our model and the authors is the implementation of the POS-RNN with our project. Both materials are cited in the reference section.

3. Proposed Method

3.1. Char-RNN Model Overview

Our model must be able to generate a seemingly original sentence. Sentences are sequential which lend themselves to Recurrent Neural Networks. However, the model must not overfit to the data, or else it will simply copy sentences from the dataset. Additionally, if each element in the model is allowed to observe every element that proceeds in a sequence during training, then the data will just be memorized. This motivates the use of LSTM (Long Short-Term Memory). Since we are inserting text and outputting text, we will be using the Char-RNN. Recurrent Neural Networks (RNN) take into consideration the order of the data being integrated into the model and outputs the data dependent on the order. Additionally, RNNs can potentially recognize patterns in sequences of data. The Char-RNN observes each character when the string is implemented into the model, then outputs character-by-character dependent on the previous character (Figure 1).

While Char-RNN will have a large role in the final model, it is not used exclusively. Char-RNN can develop and capture word choice as well as develop words that do not overfit within a sequence, as described above. However, word choice alone does not define diction. The order of words matters when trying to communicate with language, and a Char-RNN does not account for order unless it overfits. The Char-RNN is especially unable to do this because we do not have much data relative to standard Natural Language Processing models. In order to add a mind for word ordering, we have included another model beyond Char-RNN which will henceforth be referred to as the Part of Speech model (or POS model). These two models will be linked during sampling.

3.2. POS-RNN Model Overview

The POS-Model is a separate RNN model from the Char-RNN. The model is RNN because we want to define a relationship for a sequence but from a sentence structure perspective. The inputs for the POS Model will be an array of tokenized words (from the tweets/sentences used in the Char-RNN), mapped to their parts of speech using the python library, NLTK (reference section 4.2 Software for more information about NLTK). An example of this mapping is in figure 2.

3.3. Char-RNN & POS-RNN Hyperparameters

Furthermore, there are multiple hyperparameters that require tuning to optimize our model. The key hyperparameters we will be focusing on is number of layers, learning rate, keep probability, gradient clipping, and long short term memory (Raschka & Mirjalili). The number of layers allows us to set the amount of hidden layers in the model (Raschka & Mirjalili). The learning rate gives us the opportunity to control how much we are adjusting our weights within our network with respect to our loss gradient (Raschka& Mirjalili). Since the model is making decisions based on the past, the keep probability lets us set the probability of the model keeping the units (Raschka & Mirjalili). Gradient clipping sets a threshold where the gradient will be clipped to avoid the gradient from getting too large. Lastly, the long short term memory (LSTM) will introduce a memory cell that will not be multiplied over time to avoid the potential issues of gradient explosion or gradient vanishing (Raschka & Mirjalili). Figure 3 will explain how the LSTM effectively handles both vanishing and exploding gradient issues.

3.4. POS-RNN Model & Char-RNN Model Description

The two models will be used in conjunction during sampling. The POS will be run first and provide a sequence of parts of speech (Figure 4). These parts of speech can range from noun to adverb and more referenced from the NLTK library. Then, the Char-RNN is used to fill this part of speech structure once the Char-RNN model finishes creating a word. If the Char-RNN result does not match the type of word the POS-RNN is seeking, the Char-RNN will

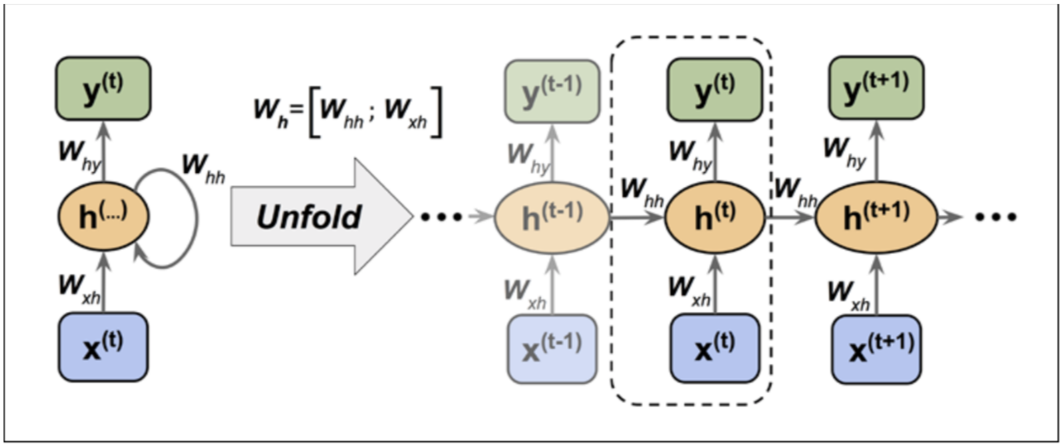
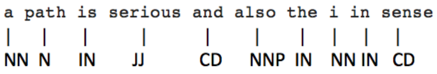


Figure 1. Description of Char-RNN Model Mechanics. (Raschka & Mirjalili, 455)



```
a  path  is  serious  and  also  the  i  in  sense
|   |    |      |      |     |    |   |  |    |
NN  N    IN     JJ     CD    NNP  IN  NN IN   CD
```

Figure 2. This is an example of a mapping using NLTK

regenerate a new word until the POS-RNN accepts the word given. This is represented in figure 4.

4. Experiments

4.1. Dataset

There are many ways to obtain a users tweets in csv file format on the internet. Unfortunately through initial research, it was realized that there were two main barriers to the data. Firstly, Twitter itself only allows 3rd party services to grab at most 3200 of a users most recent tweets. Secondly, a vast majority of these services had a paywall to pass through before they could be used. After some additional research, it was discovered that the website Vicintas has a free tool for obtaining a users tweets in csv file format. The catch was that it couldnt quite obtain all of the allowed 3200 tweets. In this case the service obtained 2423 of Deepak Chopras most recent tweets (as of February 13th, 2019). After some initial work, it was realized that more data was necessary, especially for a project in a deep learning class. Unfortunately even with using the money to get past the paywall of more quality tweet-grabbing services, only roughly 800 more tweets would be obtained. Instead, one of Deepaks books The Book of Secrets: Unlocking the Hidden Dimensions of Your Life was obtained in PDF format. From there it was converted into CSV format, with each sentence representing a row of the new data. These two data sets combined represented all of Deepaks words utilized for this project.

4.2. Software

As mentioned before, this projects first piece of software was Vicintas free tweet-grabbing tool online. Without it, this project may have never happened or couldve had expenses. From there we utilized the popular language Python for our data cleaning script, which allowed us to remove unnecessary information from tweets/sentences including URLs, emojis, excessive character repetition, and unnecessary punctuation (commas, slashes, dollar signs, etc.). From there we began developing our .ipynb file, which was developed through the Google Colab service on Google Drive (for reasons we will discuss in the hardware section). Multiple packages were imported for use in this project including NumPy, TensorFlow, csv, and Pandas. In the future we hope to utilize Twitter as a platform to make a Deepak style chatbot, though our capabilities are not there yet.

The POS-RNN utilized the Python library NLTK (Natural Language Toolkit) giving the capability to tokenize and convert sentences into an array of parts of speech. The website for this package can be found in the references section.

4.3. Hardware

As with many deep learning projects, GPUs are a highly valuable tool to improve the speed of training models. Fortunately through the Google Colab service on Google Drive access to a GPU was obtained. While it was our only hardware need, it was crucial in saving time through this project.

4.4. Hyperparameter Experiments

In our first experiment, our goal was to optimize the number of hidden layers in both models while keeping other hyperparameters constant. Normally, the more data we in-

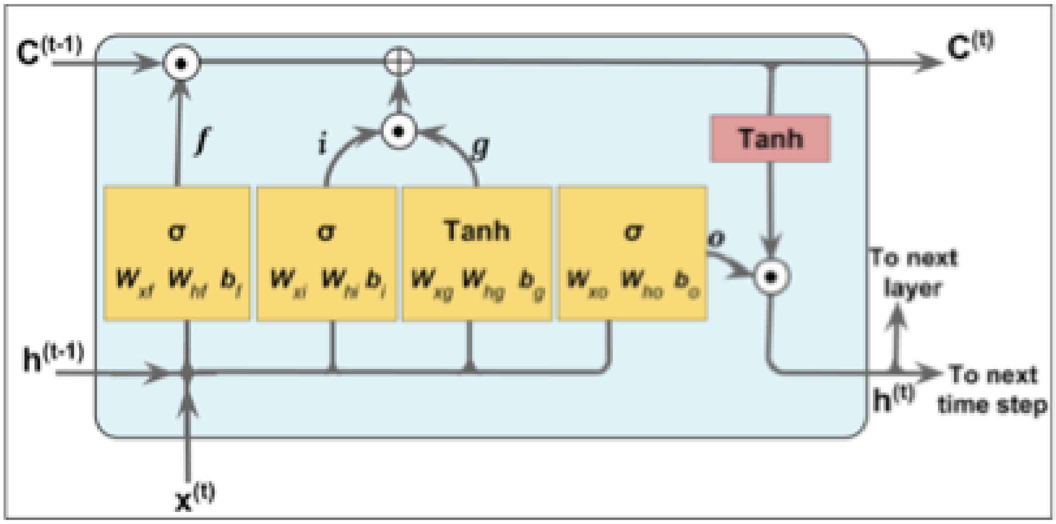
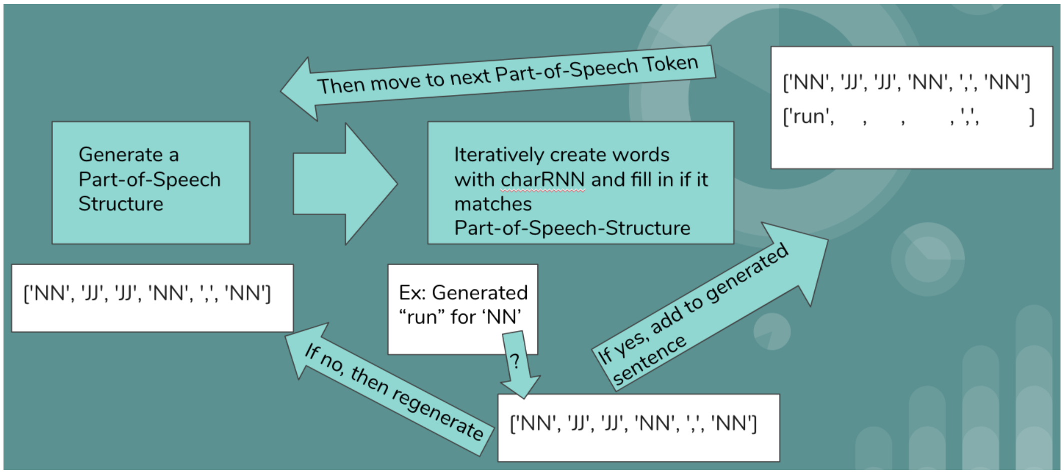


Figure 3. Mechanics Behind LSTM (Raschka & Mirjalili, 458)



Then move to next Part-of-Speech Token

['NN', 'JJ', 'JJ', 'NN', ',', 'NN']
['run', , , ',',]

Generate a Part-of-Speech Structure

Iteratively create words with charRNN and fill in if it matches Part-of-Speech-Structure

['NN', 'JJ', 'JJ', 'NN', ',', 'NN']

Ex: Generated “run” for ‘NN’

?

If yes, add to generated sentence

If no, then regenerate

['NN', 'JJ', 'JJ', 'NN', ',', 'NN']

Figure 4. This figure depicts how to two models are connected in their sampling.

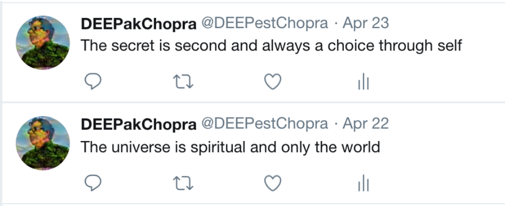
troduce and the more inseparable the data is, the more hidden layers are required for the model to understand the data. At first, we started off with one layer to see if the hidden layer was able to learn from the model without overfitting. Unfortunately, the models were giving us results that did not make complete sense and was not able to establish the differences between the characters as well. Moving forward, we increased the hidden layers to two which gave us stronger results. However, we wanted to see if three hidden layers would be optimal, which ultimately led to the models beginning to memorize and began to overfit. To avoid overfitting, two layers were selected.

In our second experiment, our goal was to find the strongest keep probability for both of our models. For our Char-RNN, we decided to use a low keep probability (0.65), because we wanted our Char-RNN model to be regularized and not essentially memorize from past inputs. On the other hand, we made our keep probability high (0.9) for our POS-RNN due to our goal being to analyze the sentence structure of Deepaks tweets. This is the only hyperparameter that is not set to a similar value between both models.

Lastly, we were experimenting with the learning rate to ensure our model was giving the results we wanted to see. We decided to use a slower learning rate (0.001) to avoid dealing with gradient explosion. Because of this, the models to a longer time to converge to the loss minimum.

5. Results and Discussion

The goal of this project was to develop a model that could generate tweets that are similar to Deepak Chopra in both diction and word choice. Figure 5 shows some exam-



DEEPakChopra @DEEPestChopra · Apr 23
The secret is second and always a choice through self

DEEPakChopra @DEEPestChopra · Apr 22
The universe is spiritual and only the world

Figure 5. Above is a picture of two tweets that resulted from the full model. The tweets were posted to the twitter page, @DEEPestChopra.

ples of output from the model. Correctness of these outputs is difficult to evaluate since we cannot measure our results against a test set. So in order to evaluate our correctness, we showed three subjects six different tweets. The six tweets are shown in figure 6.

The subjects were then given the fact that three of the tweets were real Deepak Chopra tweets and three were made from the model. Then, they were asked to select which tweets were real and which were fake. In total, the tweets generated from the model were identified as real 2 out of the 9 selections. This does not yield a very good evaluation percentage (2/9 is much lower than 50%); however, there were some comments that were quite telling. One of the subjects that correctly evaluated all of the tweets said, If you hadnt told me that there were fake tweets in here, I could have made something (wisdom) out of all of them. Another subject that correctly evaluated all of the tweets said after that, It was quite difficult. Even though our sample size for evaluation is quite small, it seems clear that the tweets are not completely indistinguishable from true Deepak Chopra tweets. However, from the limited feedback given, we can conclude that the tweets generated at least approximate Deepak Chopras diction, which was the goal of the project. Figure 7 shows the validation error for both the POS-RNN and the char-RNN. It is clear that the two graphs converge.

The model is not without weakness. One weakness of the model is the limited amount of data that was used to create it. While a few thousand sentences seems to be sufficient, the model could be better given more data. Also, the data is not necessarily consistent. The data taken from Chopras book seems to be similar to the twitter data in word choice and structure, however this may not be the case in actuality. We also do not handle all types of punctuation; only commas and periods. Lastly, the sentences are not all the same in tone and flow, and our model assumes this is the case.

6. Conclusions

As described in the results, the goals of this project are quite qualitative and cannot really be measured well. However, through the feedback given by peers and our own intuition, we have decided that the model has at least approximated the tone and diction of Deepak Chopra, and the closeness of this approximation is up for debate. However, the results do in fact appear promising. The results were also posted to the Twitter page @DEEPestChopra to allow the public to determine their validity.

The applications of NLP are quite vast; however, this research targeted a very specific type of NLP. The model created is not able to create an understanding of the words being put together. It required the inputs from a subject with a well defined diction with a common pattern of speaking. Instances of this nature do not occur very often in conversation. On the other hand, on a platform such as social media, one could make entire pages that run on models of the type created. Similar places that this model could be applied could be news segments in which a particular voice would like to be attributed to the speaker. These instances are subtle, but could be powerful if combined with other models.

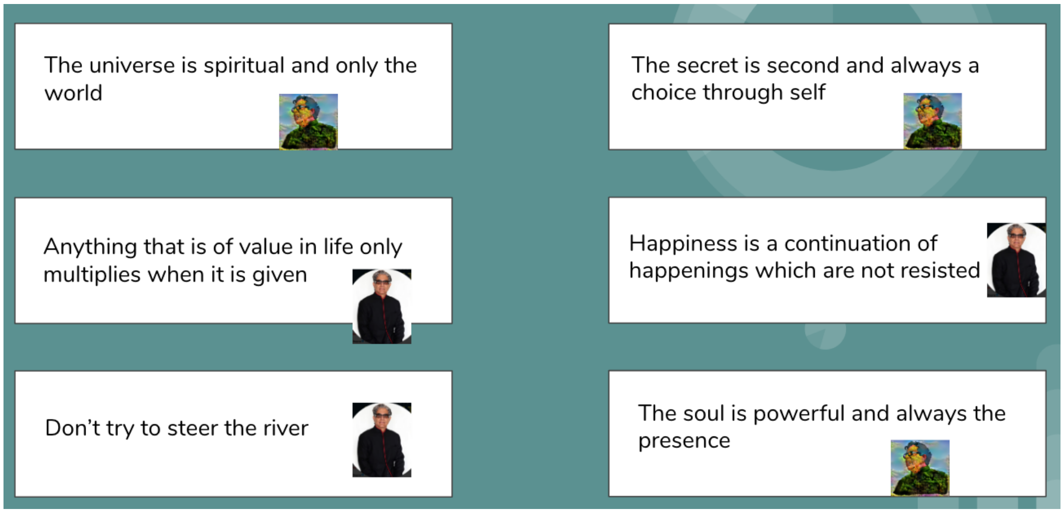
This model is especially good for NLP models that wish to recognize tone and diction. Unique tones are what set certain writers, authors and artists apart from each other. Often times, learning models can sound very robotic. With the addition of a particular tone and diction for a model, it can give the model a sense of humanness and thereby, credibility.

To go further with this project, we hope to link the model to Twitters API. Doing this would allow the model to respond to direct messages and comments. One could envision the model using words from twitter users comments and direct messages in a response, creating results that could both sound like Chopra, and also be tailored to comments from the users. Again, to reiterate from section 1.1 Motivation, our goal is to help celebrities and users on social media to still maintain their brand/image by communicating with their followers on their respective platform without directly using their social media accounts. Ultimately, this will lead to more productivity and satisfied social media followers.

7. Acknowledgements

7.1. Dataset

First of all this text mimicking couldnt have been possible without Deepak Chopra himself. Thanks to his many unique phrases posted on his public Twitter account and his book, this project was able to generate unique but well-mimicked text. An additional thanks to the website Vicintas and their free tweet downloading tool online. Without it the beginnings of this projects dataset may have never been obtained, or at a cost. Furthermore thanks to Grant for obtaining a pdf of Deepaks book The Book of Secrets: Unlocking the Hidden Dimensions of Your Life. With it this project



The universe is spiritual and only the world

The secret is second and always a choice through self

Anything that is of value in life only multiplies when it is given

Happiness is a continuation of happenings which are not resisted

Don’t try to steer the river

The soul is powerful and always the presence

Figure 6. Above are the tweets used to evaluate correctness. The Deep Dream photos represent tweets from the model while true Chopra tweets are indicated with the real photo of Chopra.

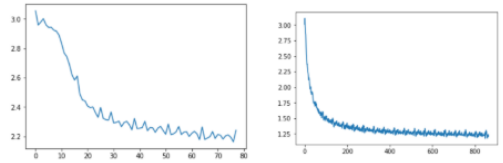


Figure 7. On the left is the validation error of the POS-RNN by epoch and on the right is the validation error of the char-RNN by epoch.

was able to nearly triple their amount of data to improve the quality of this project.

7.2. Char-RNN Code

To begin our project, we first implemented a Char-RNN model that was used in the Python Machine Learning book that the Professor Raschka suggested us to use. We decided to use the code from the book, because we needed a base to start at with our model. Eventually, we changed multiple hyperparameters for better optimization of the model and also added a POS-RNN model to our project.

8. Contributions

8.1. Nick

Nick provided a lot of the work at the beginning of the pipeline so to speak. Nick acquired Deepaks tweets through Vicintas, wrote up a Python data cleaning script to turn the tweets into more speech-normal sentences, and helped convert the book pdf file into a csv file separated by sentences. Additionally he wrote up the very initial Char-RNN with which this project began generating text. Later on in the project he assisted Grant with some troubleshooting related to trying to get a pretrained model to re-generate text.

8.2. Sai

Sai worked on optimizing the Char-RNN with the hyperparameters given in the model. The key goal of his contribution was to ensure the Char-RNN would give optimal output for the POS-RNN model, which Grant implemented. Additionally, his goal was to understand why specific parameters gave the team better results than others and to also understand how the model tuning could be optimized efficiently due to long runtimes.

8.3. Grant

Grant worked on developing and testing sampling methods and network engineering techniques used. He wrote the code to convert and sample from the Part of Speech RNN and from the char RNN and the combination of the two.

9. References

Bird, Steven, et al. Natural Language Toolkit. Natural Language Toolkit, 3.4.1, 17 Apr. 2019, www.nltk.org/.

Raschka, S. (2015). Python Machine Learning: Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analysis. Birmingham: Packt Publishing Limited.

Testud, JC. Yet Another Text Generation Project. Towards Data Science, Towards Data Science, 29 Aug. 2017, towardsdatascience.com/yet-another-text-generation-project-5cfb59b26255.

The Book of Secrets: Unlocking the Hidden Dimensions of Your Life. (n.d.). Retrieved from https://www.amazon.com/Book-Secrets-Unlocking-Hidden-Dimensions/dp/1400098343