Springboard Data Science Career Track Capstone Project 2:

Developing an Intuition for Deep Neural Networks through Text Generation

By Logan Larson

October 2019

Contents

1 Introduction

Graduation from the Springboard Data Science Career Track requires the completion of two capstone projects. Whereas my first capstone project addressed a problem I ran into during a past work experience, my second project is designed to showcase to future employers my initiative and willingness to expand my problem-solving capabilities. To that end, I embarked on a journey to learn about something that can arguably be considered as the most promising recent advancement in the world of data: neural networks.

Developing an intuition for how these systems work is challenging, but the learning process can be accelerated through a hands-on experience that involves tinkering with various architectures to learn how individual changes impact its output.

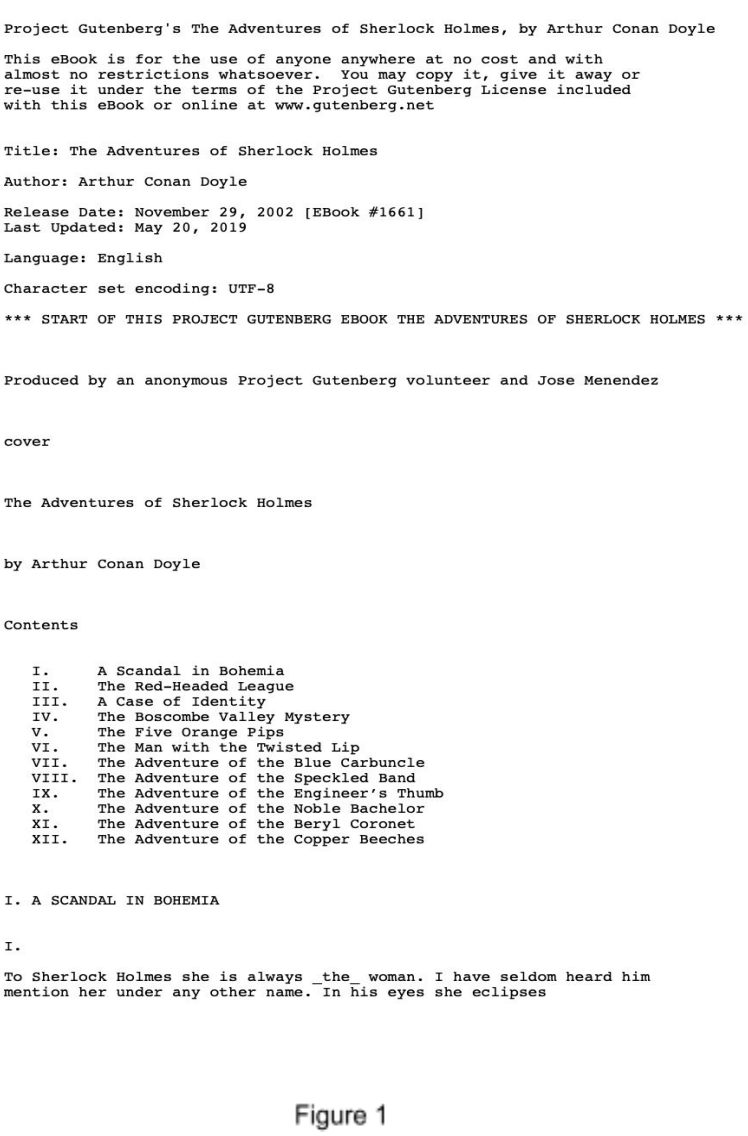
I seek to gain such experience through a project that uses the textual content from some of my favorite childhood stories. More specifically, I will consider the problem of training character-level deep neural nets such that given a randomly-selected string of characters as input, predict the next 100 characters that are most likely to follow, according to the theme of the original text (used to train the neural net.) This is a first approximation to the challenging problem of building deep neural nets that are able to generate cogent textual contents, from a human perspective. In other words, the problem of “teaching” a machine to write “like a human.”

2 Approach

I. Data Acquisition and Wrangling

Many classical works of text that have lost copyright are now available for download from Project Gutenberg, including numerous stories about my favorite fictional detective, Sherlock Holmes. Among them is *The Adventures of Sherlock Holmes*, which contains 12 short stories of significant length. This .txt file was downloaded directly from the site and I used a text editor to manually delete all chapter/story titles plus any added text from the distributor that isn’t part of the stories. In my baseline notebook, I then

wrangled the data by converting all characters to lowercase in order to minimize the size of my vocabulary. Figure 2 exemplifies a portion of the result of trimming superfluous information from a part of the original raw dataset shown in Figure 1.



```
Project Gutenberg's The Adventures of Sherlock Holmes, by Arthur Conan Doyle

This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever.  You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.net


Title: The Adventures of Sherlock Holmes

Author: Arthur Conan Doyle

Release Date: November 29, 2002 [EBook #1661]
Last Updated: May 20, 2019

Language: English

Character set encoding: UTF-8

*** START OF THIS PROJECT GUTENBERG EBOOK THE ADVENTURES OF SHERLOCK HOLMES ***


Produced by an anonymous Project Gutenberg volunteer and Jose Menendez


cover


The Adventures of Sherlock Holmes


by Arthur Conan Doyle


Contents


   I.     A Scandal in Bohemia
   II.    The Red-Headed League
   III.   A Case of Identity
   IV.    The Boscombe Valley Mystery
   V.     The Five Orange Pips
   VI.    The Man with the Twisted Lip
   VII.   The Adventure of the Blue Carbuncle
   VIII.  The Adventure of the Speckled Band
   IX.    The Adventure of the Engineer's Thumb
   X.     The Adventure of the Noble Bachelor
   XI.    The Adventure of the Beryl Coronet
   XII.   The Adventure of the Copper Beeches


I. A SCANDAL IN BOHEMIA


I.

To Sherlock Holmes she is always _the_ woman. I have seldom heard him
mention her under any other name. In his eyes she eclipses
```

Figure 1

```
to sherlock holmes she is always the woman. i have seldom heard
him mention her under any other name.
```

Figure 2

More advanced wrangling techniques like punctuation removal and lemmatization were bypassed since their application isn't necessary in this context. While punctuation removal can be important in cleaning textual data for many other applications, in this project I want to keep punctuation in place - especially the heavy usage of quotations - to learn if I can mimic the theme of the short stories. Likewise, lemmatization would be important in uses that involve understand the meaning of text for classification purposes, but lemmatized words will also change the coherence of the text I'm training my model on and could increase the difficulty of generating text that follows the theme I'm looking for.

II. Storytelling and Inferential Statistics

This collection of short stories titled *The Adventures of Sherlock Holmes* was written by an Englishman named Arthur Conan Doyle and published in 1892. Each of the 12 stories centers around Doyle's fictional detective Sherlock Holmes and each is written in first-person from the perspective of Holmes's assistant, Dr. Watson. Given the author's origins, the text contains understandably frequent usage of a classical English dialect, which most native speakers of modern American English could reasonably consider as dated, but not outdated to the degree where it would be difficult for them to understand it. After all, these writings were originally meant for publication in magazines and, thus, Doyle's writing style was intentionally designed to cater to a wide audience[1].

Altogether, the text (not including any trivial content like titles, chapter headings, etc.) contains 561,209 total characters (53 unique) and 104,340 total words (14,633 unique). The average length of each story (and thus, the average duration of different topics) is roughly 7,000 words and the variation of chapter lengths is relatively stable across each of the 12 stories. The following table shows a summary of these figures.

	Story Title	Total Words	Total Characters	Unique Characters
1	A Scandal In Bohemia	7,301	45,416	47

[1] Bajorski, Jon. "A Comprehensive Look Into the Writing Style of Sir Arthur Conan Doyle." Metamorphosis Literary Agency, December 13, 2018. https://www.metamorphosisliteraryagency.com/single-post/2018/12/13/A-Comprehensive-Look-Into-the-Writing-Style-of-Sir-Arthur-Conan-Doyle.

2	The Red Headed League	7,833	48,200	52
3	A Case Of Identity	6,064	37,147	51
4	The Boscombe Valley Mystery	8,394	50,274	49
5	The Five Orange Pips	6,335	38,589	52
6	The Man With the Twisted Lip	7,999	48,091	51
7	The Adventure of the Blue Carbuncle	6,660	41,151	49
8	The Adventure of the Speckled Band	8,506	51,775	47
9	The Adventure of the Engineer's Thumb	7,201	43,656	53
10	The Adventure of the Noble Bachelor	6,954	43,139	49
11	The Adventure of the Beryl Coronet	8,448	49,900	47
12	The Adventure of the Copper Beeches	8,687	52,015	50

To help confirm whether the text was in fact catered to a general audience, I took a closer examination of the individual words that comprise the aggregated corpus. This, in turn, will provide a better idea of the theme I'm aiming for when I advance to the stage of text generation:

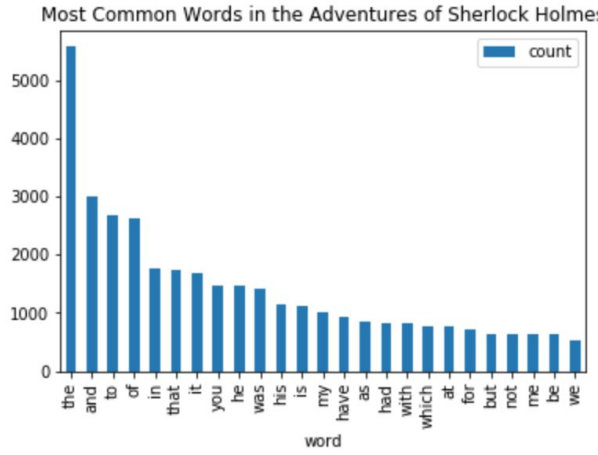


Figure 3

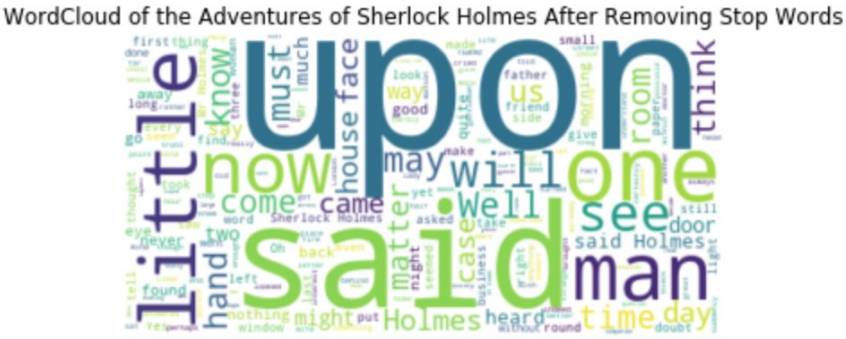


Figure 4

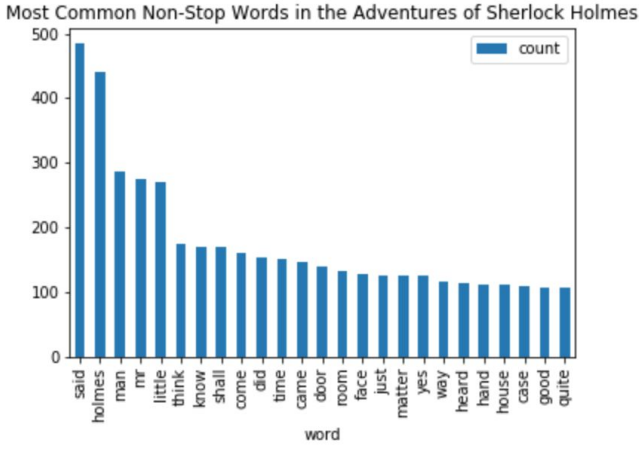


Figure 5

Most common N-grams in the Adventures of Sherlock Holmes

```
----- 20 most common 2-grams -----
"of the" used 703 times
"in the" used 506 times
"it is" used 334 times
"to the" used 304 times
"i have" used 300 times
"it was" used 279 times
"that i" used 257 times
"at the" used 237 times
"and i" used 217 times
"to be" used 199 times
"and the" used 198 times
"upon the" used 196 times
"with a" used 186 times
"i was" used 186 times
"i am" used 182 times
"i had" used 168 times
"of a" used 168 times
"was a" used 160 times
"that he" used 154 times
"he was" used 152 times

----- 20 most common 3-grams -----
"one of the" used 50 times
"it is a" used 46 times
"i think that" used 46 times
"it was a" used 45 times
"that it was" used 38 times
"out of the" used 36 times
"that i have" used 35 times
"i do not" used 34 times
"that it is" used 34 times
"there was a" used 34 times
"that he had" used 30 times
"that he was" used 30 times
"that i was" used 28 times
"lord st simon" used 28 times
"that i had" used 27 times
"i have no" used 27 times
"in front of" used 27 times
"i could not" used 26 times
"think that i" used 26 times
"it would be" used 25 times

----- 20 most common 4-grams -----
"i think that i" used 18 times
"i have no doubt" used 17 times
"have no doubt that" used 14 times
"i do not know" used 13 times
"to me to be" used 11 times
"it seemed to me" used 11 times
"in front of the" used 10 times
"i beg that you" used 10 times
"i do not think" used 10 times
"of a man who" used 9 times
"a man who is" used 9 times
"one of the most" used 9 times
"i am afraid that" used 9 times
"i don t know" used 8 times
"the corner of the" used 8 times
"it is a little" used 8 times
"that it would be" used 8 times
"in the direction of" used 8 times
"the direction of the" used 8 times
"beg that you will" used 8 times

----- 20 most common 5-grams -----
"i have no doubt that" used 13 times
"i beg that you will" used 8 times
"from one to the other" used 7 times
"it seemed to me that" used 7 times
"in the direction of the" used 7 times
"i do not think that" used 7 times
"one to the other of" used 5 times
"to the other of us" used 5 times
"the air of a man" used 5 times
"air of a man who" used 5 times
"i shall be happy to" used 5 times
"i think that it is" used 5 times
"there was no sign of" used 5 times
"i have heard of you" used 5 times
"i do not know what" used 5 times
"in the case of the" used 4 times
"what do you make of" used 4 times
"the door opened and a" used 4 times
"with the air of a" used 4 times
"in his chair with his" used 4 times
```

Figure 6

Referencing Figure 6, the 25 most common words within this corpus are almost entirely conjunctions and prepositions. But if we remove all words that are considered stopwords - words so common they don't add substantial meaning to the text - we still observe that the next-most-common words in this corpus can reasonably be described as stereotypically-detective-like words that relate to the process of deduction and the various senses of perception (like seeing and hearing).

Digging deeper, in Figure 4 we can explore the most common two-, three-, four- and five-word combinations within the corpus. While it's important to keep in mind that these phrases reveal the dialogue-heavy nature of Doyle's first-person writing style, the words that make up these phrases continue to follow a theme of simple and descriptive language. Further confirmation of this is most easily accomplished through a manual reading of a single story from the collection.

The code to view my exploratory data analysis can be viewed in this notebook.

III. Baseline Modeling

In creating a baseline model of the corpus, I started with a simple architecture with the idea of incrementally adding, tweaking, and building upon features as I went on. While any future models then will thus differ in some sense, at a basic level this baseline model and all models that follow will be constructed to learn the dependencies between characters and the conditional probabilities of characters in sequences so that we can in turn generate new and original sequences of characters from a given input[2].

The model I picked is a type of recurrent neural network (RNN), which are commonly used for sequential data because of an ability to remember previous inputs through an internal memory. Specifically, I'm using a Long Short-Term Memory (LSTM) system through Keras and, as a baseline, the layers I used are:

1. An Input Layer
2. A Hidden LSTM Layer
3. A Dropout Layer (Used to randomly set a fraction of the inputs to 0 at each update to help prevent overfitting)
4. An Output Layer with a softmax activation function

[2] Brownlee, Jason. *Deep Learning for Natural Language Processing*. Vol. 1.1. Machine Learning Mastery, 2017.

All code can be seen at this notebook, but the way the network is set up is relatively straightforward. The input layer accepts input sequences of 100 encoded characters each with 53 features (one for each unique character in the corpus). A single LSTM layer equipped with 256 memory cells (picked arbitrarily) then learns the joint probability function of character sequences in terms of the encoded vectors that represent the input sequence. The input then goes through a single dropout layer to deal with overfitting and onto an output layer that produces a single vector with a probability distribution across all 53 characters in the vocabulary.

3 Findings and Analysis

After training the baseline model, a string of 100 characters from the original text is randomly selected and that string is then used as the input to predict the next most-likely character. A second character is then predicted from the final 99 characters in the input sequence plus the first generated character, and the model then uses those total 100 characters to predict the second generated character, and so on until 100 new characters are generated.

I then set up a brief study to assess this model's ability to generate new content. Ultimately, five strings of 100 characters were randomly selected as inputs and the resulting outputs (strings of 100 generated characters) were then graded on 1-5 scale by five human readers to assess readability. The participants were not told the content was computer-generated in order to minimize cognitive biases in assigning grades. The general guidelines for grading were:

1. Text is complete nonsense
2. Text is mostly nonsense, but there are individual words from the English language that can be observed
3. Text is nonsense, but there is at least one group of three words from the English language that could make sense in their observed order
    - Loosening argument to at least *two* words might overstate sensibility of text if those two words are extremely common phrases such as "..to the.."
4. Text is mostly coherent, but spelling and grammatical errors are present
5. Text is perfectly coherent

Results from Baseline Model:

Example #1:

```
Seed:
" ble and waiting for me to come back. so
frank took my wedding-clothes and things and made a bundle o "
f the correrisins of the correrisiss of the correrisiss of the correrisisn of the correrisinnsonn
of the
```

Human #	Grade
1	1
2	1
3	1
4	1
5	1
Average Grade for Example #1: 1	

Example #2:

```
Seed:
"  they are. must i
call you a liar as well as a thief? did i not see you trying to
tear off another p "
occes.

"io is a coaak oo the corrner," said holmes. ""whll you hove the soaee that i have been
ao
```

Human #	Grade
1	3
2	3
3	3
4	3
5	3
Average Grade for Example #2: 3	

Example #3:

```
Seed:
"  hand me over my violin and let us try to forget for
half an hour the miserable weather and the stil "
e of the sooe and sooe ao allartacte of the sooe and sooe ao allar oo the saale which wa
s ao alladri
```

Human #	Grade
1	1
2	1
3	1
4	1
5	1
Average Grade for Example #3: 1	

Example #4:

```
" ue of this
question depended whether i should continue my work at briony
lodge, or turn my attention "
 the saale which was an axpines of the correrssinsion of the correrssiss of the correrssisn
of the corr
```

Human #	Grade
1	1
2	1
3	1
4	1
5	1
Average Grade for Example #4: 1	

Example #5:

```
Seed:
" ut into
the open, that i could think of nothing except of my father. yet
i have a vague impression t "
fat the correrrinn of the saale whs had been soeet to her to the saale.
"i have no toon to to the l
```

Human #	Grade
1	3
2	3
3	3
4	3
5	3
Average Grade for Example #5: 3	

Summary Table of Results:

Example #	Average Grade
1	1
2	3
3	1
4	1
5	3
Average:	1.8

While the majority of examples of the baseline model's generated text were practically gibberish, the second and fifth examples show promise. Specifically, Example #2 is the most exciting since it not only concluded the input string by finishing off the last word of a sentence that seemingly only needed one and exactly one more word (albeit with

incorrect spelling), but it also responded to a quotation with another quotation as a human writer would when authoring a dialogue between characters. Furthermore, this quotation wasn't just of the standard " 'xxxx xxx xxx xxxx xxx xxxx xxxx xxx', he said" format, but it inserted a pause in the form of the non-standard " 'xxx xxx xxx,' he said. 'Xx xxx xxxx." format.

There clearly remains work to be done, though. In four out of the five examples, the nonsensical word "saale" is used and in three out of five there exists some variation of a made-up word beginning with the prefix "corr-". The most concerning instance of this is in Example #4 where it seems as if the model is stuck in some sort of loop:

> "...of the correrssinsion of the correrssiss of the correrssisn of the corr…"

Extended Modeling

There are different things I could change about the setup of my baseline model. Listed below are different options I have to change the network architecture and for each option I described how these changes would impact the performance of the model:

Change	Impact
Corpus size	More data gives the model more to learn from, but training becomes more expensive
Number of layers in the network	Determines depth of network; More layers offer added opportunities for refinement of model weights but will increase training cost
Number of memory units	Determines width of network; More units offer added opportunities for refinement of model weights but will increase training cost
Use word-level input	Rather than character-level input; This significantly increases the size of the vocabulary (and thus computational load) and also reduces flexibility in handling document structure, but perhaps this could remove the issue of spelling and improve the overall coherence of the output message

Length of input	Longer input sequences provide more context for the model to learn the relationship among individual characters, but increasing the input sequence likely also increases training time
Batch size	Indicates the number of samples from the training data that will be used in the estimate of error gradient; Larger values lead to increased training time
Activation functions	There isn't really a hierarchy of superior activation functions as the usage of each depends on the nature of the problem at hand
Loss functions	Like activation function, it is case-dependent
Removing regularization	Removing methods of regularization, like a dropout layer, could hinder the model's ability to generalize
Number of epochs	Refers to the number of times the model will pass through the corpus and is closely related to batch size; More is generally better in terms of the model's ability to learn patterns, but too many could lead to overfitting
Ensemble of methods	Combining the predictions of multiple models could improve overall performance
Usage of embedding layer	Generates a distributed representation of words in vocabulary such that similar words are represented by similar vector representations

My efforts to improve the baseline model started with my greatest curiosity: adding another layer. As mentioned previously, increasing the number of layers increases the depth of the network, where each added layer offer additional opportunities for model improvements but comes at the expense of increased training costs. While this model took over double the amount of time my baseline model took to train, I saw immediate improvements in generated content.

Results from Extended Model #1

Example #1:

```
Seed:
" nked on to
it--that it is the clew which will guide you in the solution of
some mystery and the puni "
ess of the sertent. i have aeen a seruice to the side of the side of the
side of the side of the sid
```

Human #	Grade
1	3
2	3
3	3
4	2
5	3
Average Grade for Example #1: 2.8	

Example #2:

```
Seed:
" hat has
happened."

as he spoke he drew the dog-whip swiftly from the dead man's lap,
and throwing t "
p the street.
"'what is the street."

"i am a care of the sertent."

"i am so she cay in the sime o
```

Human #	Grade
1	4
2	3
3	4
4	3
5	3

Average Grade for Example #2: 3.4

Example #3:

```
Seed:
"  merryweather perched himself upon a crate, with a
very injured expression upon his face, while holm "
es strndd and a serting of the side of the soall past of the side of the
soall past of the side of t
```

Human #	Grade
1	2
2	2
3	2
4	1
5	1
Average Grade for Example #3: 1.6	

Example #4:

```
Seed:
" about the
foresight and the moral retrogression?"

sherlock holmes laughed. "here is the foresight," "
 said he, "i have aeen a sertice to the side of the side of the side of
the side of the side of the
```

Human #	Grade
1	3
2	2
3	2
4	2
5	2
Average Grade for Example #4: 2.2	

Example #5:

```
Seed:
" to a
moral retrogression, which, when taken with the decline of his
fortunes, seems to indicate some "
 searon. i have a conpidte of the sertent of the side of the side of the
side of the side of the sid
```

Human #	Grade
1	2
2	2
3	2
4	2
5	2
Average Grade for Example #5: 2	

Summary Table of Results for Extended Model #1:

Example #	Average Grade
1	2.8
2	3.4
3	1.6
4	2.2
5	2
Average grade of samples:	**2.4**

Based on the feedback from human judges, it appears the deeper network improved the performance of the model. Rising from an average coherence score of 1.8 to 2.4, the improved performance of the model can likely be attributed to the extra memory cells now available to the model. The most encouraging display of this improvement is Example #2, which garnered two individual grades of 4 due to the model's displayed ability to replicate a dialogue between two people.

Having now observed how a deeper network could improve results, my second extension of the baseline model explored my second-greatest curiosity: changing from character-level to word-level input. In theory, doing this will extremely inflate my vocabulary size and likely increase training costs, but the upside is that it could also completely remove the issue of spelling errors in the generated content.

The first model - using almost the exact same architecture as my character-level baseline model - was disappointing since it only produced a string of the word 'the' printed 100 times in a row. However, the silver lining to this outcome is that I could now use this initial word-level model as a new baseline to observe how previously-untried architectural changes would affect model performance. After all, any improvement would be much easier to recognize when compared to a baseline whose performance literally has nowhere to go but up.

The changes I made and the human-graded review of the output is summarized in the table below. Unless noted otherwise, only the specified change was made from the baseline model, allowing each change to be assessed in a vacuum. However, this time I included a second column for the human graders to pick (YES/NO) as to whether any of the five output examples is anything but a recurrent sequence of the word 'the'. Otherwise, it wouldn't be possible to compare outputs that were both labeled as 1 for 'complete nonsense.' These results were ultimately tallied and the fraction of human votes was displayed in the table below.

Notebook Extension #	**Variation**	**Link to Notebook**	**Average Human Grade**	**Differs From Baseline Results?**
2	Word-level inputs	Link	1	-
3	Increase length of input	Link	1	No (4/4)
4	Decreasing batch size	Link	1	No (4/4)
5	Changing activation function	Link	1	No (4/4)

6	Changing loss function	Link	1	No (4/4)
7	Increasing memory units	Link	1.8	Yes (4/4)
8	Add second LSTM layer	Link	1	No (4/4)
9	Add second LSTM layer + Increase memory units	Link	1	No (4/4
10	Increasing # of epochs	Link	1.2	Yes (4/4)
11	Further increasing # of epochs	Link	2.8	Yes (4/4)
12	Add second LSTM layer + Increase # of epochs + Decrease batch size	Link	1	No (4/4)

4 Conclusions and Future Work

The ability for a relatively simple neural network to generate character-level text with non-zero accuracy cannot be ignored, especially given the relatively small corpus on which it was trained. While - based off this project - I can't necessarily make such a claim about a NN's ability to generate word-level content, we need to keep in mind that the character-level model needed to learn just 53 words in its vocabulary compared to 14,633 unique words learned by the word-level model. This computational load was very difficult to overcome given my hardware and time limitations, but looking around at other applications of natural language processing in the real word, there are much more effective models of word-level text generation, like in chatbots and smart home devices.

In hindsight, it would have seemed more logical to start by increasing the number of epochs and then built everything up from there. The only concern I remember having with this during my research process is that, given my available time, doubling or tripling the number of epochs effectively doubles or triples training time.

In any case, probably the most practically important thing I learned through this project is that some changes to the setup of a neural network are motivated by one of two things:

1. Generally improve the model overall
2. Improve the model by better aligning with the business problem

While almost all possible changes I specified on Page 13 belong to Category 1, the architecture of an LSTM model relies heavily upon activation and loss functions. These two parameters belong to Category 2, and which ones used in the model should depend on what sort of data is being used and for what purpose. Meanwhile, the reason we don't maximize all of the Category 1 possibilities is largely due to a lack of computational power, as a wide network with many layers and long input sequences run over many epochs would take a very, very long time to train without a GPU.

Future work includes but is not limited to studying:

- The usage of word embeddings
- The relationship between width and depth of neural networks
- The relationship between total epochs, batch size and the optimization function (which includes a hyper-parameter for learning rate)
- When it's most appropriate to use respective loss and activation functions
- Applying a neural network to my first capstone project

5 Recommendations for Clients

As my own client, I recommend using this project as a learning experience. Not only was this my first hands-on project using neural networks, but I effectively had zero exposure to these systems beforehand. After getting an introductory feel for how they work, it's exciting to think that there are likely unlimited ways for how neural networks could be used to solve future problems. Any future problem I do ultimately work on will likely build off my experience in putting together this project and, with potential future access to more efficient resources like a GPU, it's crazy to think I've barely realized the potential neural networks could have in changing how the world operates.

References

1. Bajorski, Jon. “A Comprehensive Look Into the Writing Style of Sir Arthur Conan Doyle.” Metamorphosis Literary Agency, December 13, 2018. https://www.metamorphosisliteraryagency.com/single-post/2018/12/13/A-Comprehensive-Look-Into-the-Writing-Style-of-Sir-Arthur-Conan-Doyle.
2. Brownlee, Jason. *Deep Learning for Natural Language Processing*. Vol. 1.1. Machine Learning Mastery, 2017.