Authorship Identification of the Speeches of US Presidents

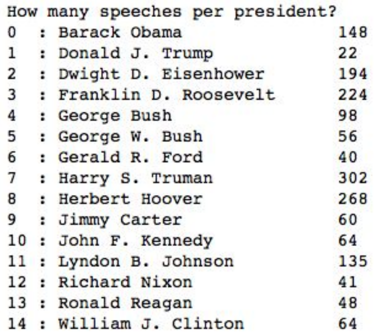
W266: Natural Language Processing with Deep Learning
School of Information, UC Berkeley

Thong Bui, Jason Vantomme
thongnbui@ischool.berkeley.edu, jvantomme@ischool.berkeley.edu

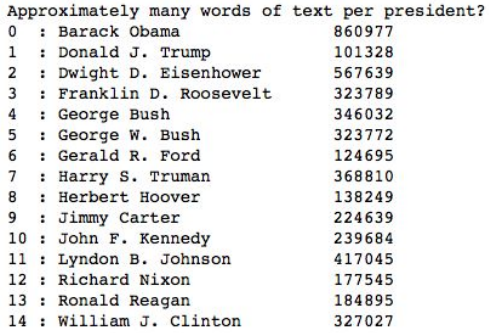
Nirkhi and Dharaska (2013) define authorship identification (also known as authorship attribution) as a process that "determines the likelihood of a piece of writing to be produced by a particular author by examining other writings by that author." In fact, attribution of authorship has been a topic of inquiry since the late 1800's (Wikipedia, 2017) and, relevant to our context, has been applied to the decisions of the US Supreme Court5 as well as the pre-presidential radio addresses of Ronald Reagan (Airoldi et al., 2007). A common focus area of authorship identification in a present-day research is its use in criminal forensics. Furthermore, it is easy to understand in today's era of "fake news" how the verification of statements, policies or quotes emerges as a critical activity. In this project, we attempt to show how authorship identification of the speeches of US Presidents can be implemented using word and character-level models, with specific attention paid to the method described by Bagnall (2015).

Data Sources

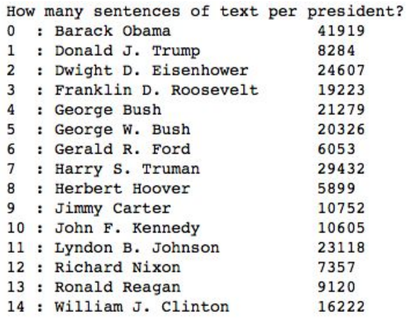
Primary sources for speech text from Presidents Hoover through Trump were obtained from The American Presidency Project @ UC Santa Barbara (https://goo.gl/rCJ8Gv) as well as from the Avalon Project at Yale (https://goo.gl/exPBFm). The majority of the data extraction comes from the JSON files downloaded from this website then cleaned (see: cleanup). Data preprocessing is then used to load the clean speech files and build the list of sentences for each president (see: preprocessing). Here are the statistics of the data of the 15 presidents' speeches:



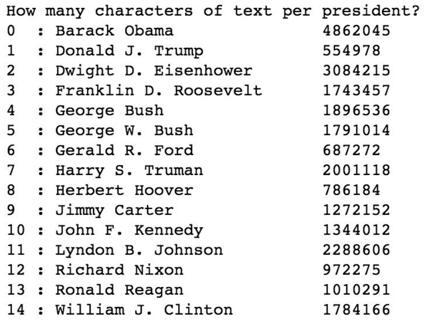
```
How many speeches per president?
0  : Barack Obama              148
1  : Donald J. Trump           22
2  : Dwight D. Eisenhower      194
3  : Franklin D. Roosevelt     224
4  : George Bush               98
5  : George W. Bush            56
6  : Gerald R. Ford            40
7  : Harry S. Truman           302
8  : Herbert Hoover            268
9  : Jimmy Carter              60
10 : John F. Kennedy           64
11 : Lyndon B. Johnson         135
12 : Richard Nixon             41
13 : Ronald Reagan             48
14 : William J. Clinton        64
```



```
Approximately many words of text per president?
0  : Barack Obama              860977
1  : Donald J. Trump           101328
2  : Dwight D. Eisenhower      567639
3  : Franklin D. Roosevelt     323789
4  : George Bush               346032
5  : George W. Bush            323772
6  : Gerald R. Ford            124695
7  : Harry S. Truman           368810
8  : Herbert Hoover            138249
9  : Jimmy Carter              224639
10 : John F. Kennedy           239684
11 : Lyndon B. Johnson         417045
12 : Richard Nixon             177545
13 : Ronald Reagan             184895
14 : William J. Clinton        327027
```



```
How many sentences of text per president?
0  : Barack Obama              41919
1  : Donald J. Trump           8284
2  : Dwight D. Eisenhower      24607
3  : Franklin D. Roosevelt     19223
4  : George Bush               21279
5  : George W. Bush            20326
6  : Gerald R. Ford            6053
7  : Harry S. Truman           29432
8  : Herbert Hoover            5899
9  : Jimmy Carter              10752
10 : John F. Kennedy           10605
11 : Lyndon B. Johnson         23118
12 : Richard Nixon             7357
13 : Ronald Reagan             9120
14 : William J. Clinton        16222
```



```
How many characters of text per president?
0  : Barack Obama              4862045
1  : Donald J. Trump           554978
2  : Dwight D. Eisenhower      3084215
3  : Franklin D. Roosevelt     1743457
4  : George Bush               1896536
5  : George W. Bush            1791014
6  : Gerald R. Ford            687272
7  : Harry S. Truman           2001118
8  : Herbert Hoover            786184
9  : Jimmy Carter              1272152
10 : John F. Kennedy           1344012
11 : Lyndon B. Johnson         2288606
12 : Richard Nixon             972275
13 : Ronald Reagan             1010291
14 : William J. Clinton        1784166
```

Design and Implementation

In order to explore the ability to classify a given text to one of the US Presidents between Hoover and Trump, we implemented four models in Keras with a Tensorflow backend:

- Word-level RNN model
- Word-level LSTM model
- Character-level Simple RNN model
- Character-level, multi-headed softmax model

Word-level Models

The word-level RNN employed 1 embedding layer, 1 RNN hidden layer, 1 dropout layer and an output layer with softmax activation; optimization was done with the Adagrad algorithm with a loss function of categorical cross entropy. The word-level LSTM model employed an identical graph structure except that the RNN hidden layer was replaced with an LSTM layer. In both cases, for each input sequence there will be 15 prediction probabilities (one per president) produced by the softmax layer.

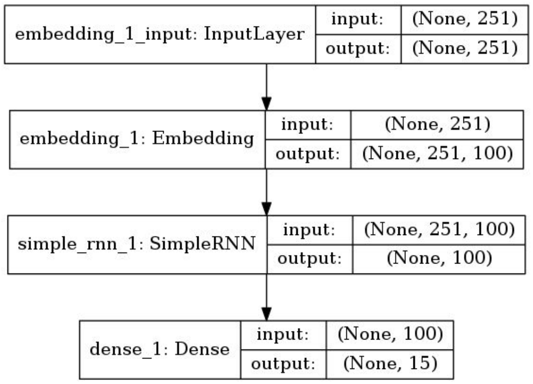


Fig. 1: Word-level RNN

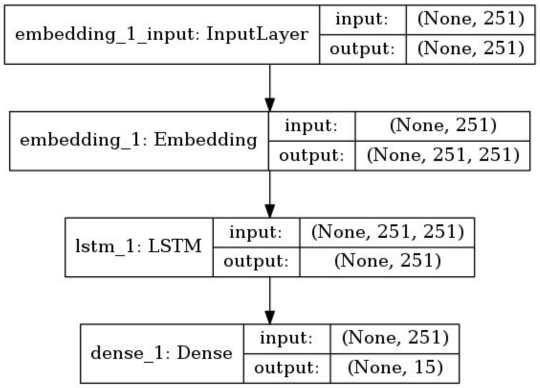


Fig. 2: Word-level LSTM

For each of the president, the 80% of the sentences will be put in train dataset to be trained by the models, 20% will go into test dataset to test and predict which president a test sentence is from. A max length of sentence is determined and will be used as the input length for the model

During training of the word-level models, we have observed that large differences in both training and test loss and accuracy measures which indicated overfitting. After increasing the dropout from 0.2 to 0.5 the deltas between the training and tests sets decreased while increasing test accuracy; the character-level models also used a 0.5 drop-out rate.

For the RNN model, rates when broken down by president are generally proportionate with the size of the data for corresponding presidents even though class size was accounted for with class weighting; see Figure 3. However, there is a notable exception: Even with very low volume of sentences, accuracy for President Trump is much higher than presidents with more data such as Reagan, Carter, Kennedy, and Clinton. One might conclude that this is due to Trump's distinctive speech style even in speeches that are scripted.

For the word-level LSTM, the same patterns of accuracy are similar to the RNN model though accuracy rates generally exhibit an increase for most presidents, even the ones with smallest set of data such as Ford, Nixon, and Reagan; see Figure 4.

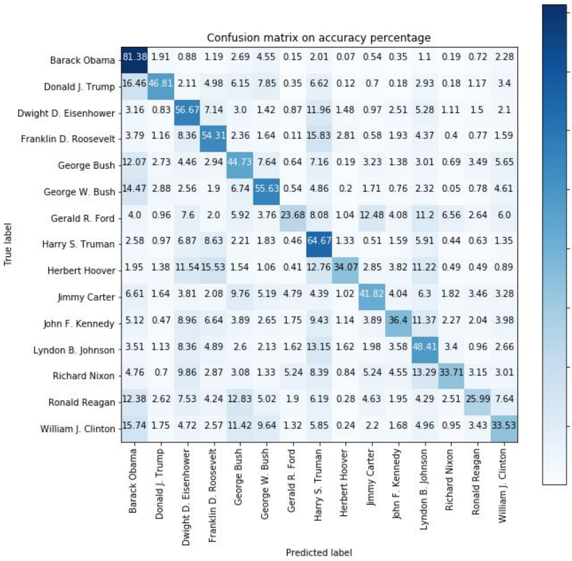


Fig. 3: Confusion Matrix for Word-level RNN

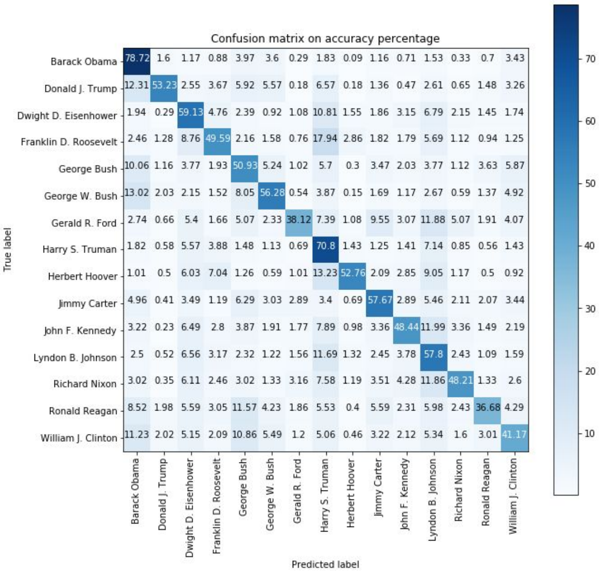


Fig. 4: Confusion Matrix for Word-level LSTM

Specific examples of the confusion in the word-level models are as follows:

Predicted	**Correct**	**Sentence Text**
Bush	Nixon	i think it is a question of timeliness
Truman	Eisenhower	there was no offer made to the peruvians
Johnson	Roosevelt	they cannot get credit and private capital won't give them credit i am not saying improperly
Johnson	Carter	we are increasing every year the allocation of funds for that purpose
Carter	Ford	in order to get new york city to restore their credibility in the money markets they have taken these steps which have eliminated 3 95 billion cash deficit
Johnson	Truman	second the smaller amount of new obligational authority which i am recommending indicates the substantial portion of the financial requirements for our military buildup that has been met in the appropriations already made by the congress
Bush	Carter	and i would hope that from now on after this news conference that we could leave out references to allegations that anybody thinks that i'm a racist or that any of the other candidates in the race for president are racists
Obama	Trump	just dont worry about it

Character-level Models

Similar to the word-level RNN, the character-level RNN employed 1 input layer, 1 RNN hidden layer, an output layer with softmax activation; optimization was also done with the Adagrad algorithm with a loss function of categorical cross entropy. The RNN hidden layer was wrapped in a function (Bidirectional) that ensure the RNN processes a sequence both forward and backward; this was shown to improve accuracy; see Figure 5.

Bagnall (2015) details an approach to the authorship attribution task at the CLEF 15 conference that, while simple in its notion, outperformed all other approaches in 3 of 4 languages tested (Stamatatos et al., 2015). In his approach, Bagnall employs a character-model RNN with a softmax output layer *per author*; the result: “Each softmax group is trained predominantly on one author’s corpus, causing the recurrent layer to model a combination of all the author’s texts, approximating the language as a whole. The output groups learn to weight the recurrent values in a way that best reflects their author’s tendencies.” (Bagnall 2015). This model proved very difficult to recreate in Keras due to the lack of dynamic graph execution which was critical to making the Bagnall (2015) method work.

Our attempt described in Figure 6 is clearly imperfect but allowed us to explore the possibility of other potential configurations as well as their inherent strengths, weaknesses and flaws. (We were able to explore other alternatives, but with no greater test accuracy.)

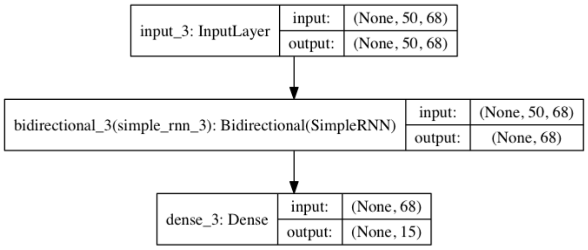


Fig. 5: Character-level Simple RNN

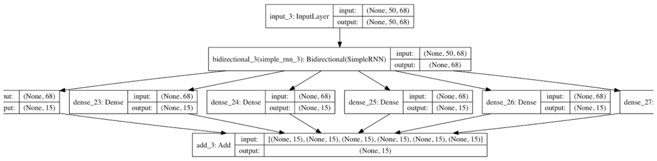
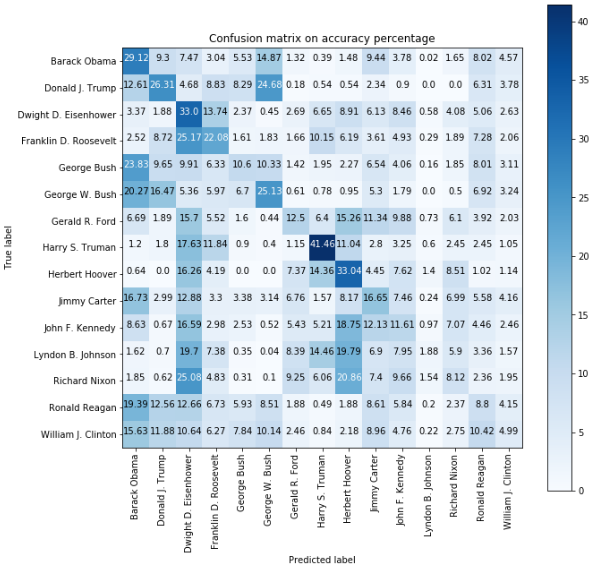
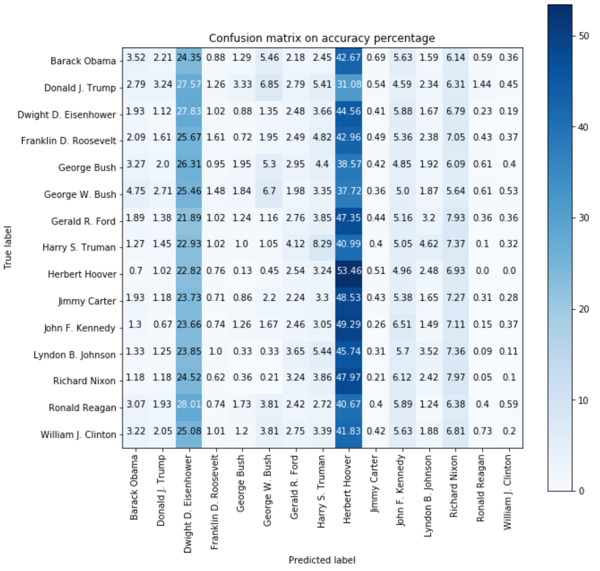


Fig. 6: Character-level Multi-softmax RNN

As can be seen in the confusion matrices for both the simple and multi-softmax models (Figs 7 & 8), the performance of these models is nowhere near the accuracy of the word-based models. It is notable that the most prominent "bands" of inaccurate labels in the multi-softmax case (e.g., Hoover in Fig. 8) does not obviously correlate to a president with the greatest number of characters. Eisenhower does show some over-attribution in the simple RNN model (Fig. 7) and is the president with the second-highest number of characters though class weighting should have accounted for volume differences.



	Barack Obama	Donald J. Trump	Dwight D. Eisenhower	Franklin D. Roosevelt	George Bush	George W. Bush	Gerald R. Ford	Harry S. Truman	Herbert Hoover	Jimmy Carter	John F. Kennedy	Lyndon B. Johnson	Richard Nixon	Ronald Reagan	William J. Clinton
Barack Obama	29.12	9.3	7.47	3.04	5.53	14.87	1.32	0.39	1.48	9.44	3.78	0.02	1.65	8.02	4.57
Donald J. Trump	12.61	26.31	4.68	8.83	8.29	24.68	0.18	0.54	0.54	2.34	0.9	0.0	0.0	6.31	3.78
Dwight D. Eisenhower	3.37	1.88	33.0	13.74	2.37	0.45	2.69	6.65	8.91	6.13	8.46	0.58	4.08	5.06	2.63
Franklin D. Roosevelt	2.52	8.72	25.17	22.08	1.61	1.83	1.66	10.15	6.19	3.61	4.93	0.29	1.89	7.28	2.06
George Bush	23.83	9.65	9.91	6.33	10.6	10.33	1.42	1.95	2.27	6.54	4.06	0.16	1.85	8.01	3.11
George W. Bush	20.27	16.47	5.36	5.97	6.7	25.13	0.61	0.78	0.95	5.3	1.79	0.0	0.5	6.92	3.24
Gerald R. Ford	6.69	1.89	15.7	5.52	1.6	0.44	12.5	6.4	15.26	11.34	9.88	0.73	6.1	3.92	2.03
Harry S. Truman	1.2	1.8	17.63	11.84	0.9	0.4	1.15	41.46	11.04	2.8	3.25	0.6	2.45	2.45	1.05
Herbert Hoover	0.64	0.0	16.26	4.19	0.0	0.0	7.37	14.36	33.04	4.45	7.62	1.4	8.51	1.02	1.14
Jimmy Carter	16.73	2.99	12.88	3.3	3.38	3.14	6.76	1.57	8.17	16.65	7.46	0.24	6.99	5.58	4.16
John F. Kennedy	8.63	0.67	16.59	2.98	2.53	0.52	5.43	5.21	18.75	12.13	11.61	0.97	7.07	4.46	2.46
Lyndon B. Johnson	1.62	0.7	19.7	7.38	0.35	0.04	8.39	14.46	19.79	6.9	7.95	1.88	5.9	3.36	1.57
Richard Nixon	1.85	0.62	25.08	4.83	0.31	0.1	9.25	6.06	20.86	7.4	9.66	1.54	8.12	2.36	1.95
Ronald Reagan	19.39	12.56	12.66	6.73	5.93	8.51	1.88	0.49	1.88	8.61	5.84	0.2	2.37	8.8	4.15
William J. Clinton	15.63	11.88	10.64	6.27	7.84	10.14	2.46	0.84	2.18	8.96	4.76	0.22	2.75	10.42	4.99



	Barack Obama	Donald J. Trump	Dwight D. Eisenhower	Franklin D. Roosevelt	George Bush	George W. Bush	Gerald R. Ford	Harry S. Truman	Herbert Hoover	Jimmy Carter	John F. Kennedy	Lyndon B. Johnson	Richard Nixon	Ronald Reagan	William J. Clinton
Barack Obama	3.52	2.21	24.35	0.88	1.29	5.46	2.18	2.45	42.67	0.69	5.63	1.59	6.14	0.59	0.36
Donald J. Trump	2.79	3.24	27.57	1.26	3.33	6.85	2.79	5.41	31.08	0.54	4.59	2.34	6.31	1.44	0.45
Dwight D. Eisenhower	1.93	1.12	27.83	1.02	0.88	1.35	2.48	3.66	44.56	0.41	5.88	1.67	6.79	0.23	0.19
Franklin D. Roosevelt	2.09	1.61	25.67	1.61	0.72	1.95	2.49	4.82	42.96	0.49	5.36	2.38	7.05	0.43	0.37
George Bush	3.27	2.0	26.31	0.95	1.95	5.3	2.95	4.4	38.57	0.42	4.85	1.92	6.09	0.61	0.4
George W. Bush	4.75	2.71	25.46	1.48	1.84	6.7	1.98	3.35	37.72	0.36	5.0	1.87	5.64	0.61	0.53
Gerald R. Ford	1.89	1.38	21.89	1.02	1.24	1.16	2.76	3.85	47.35	0.44	5.16	3.2	7.93	0.36	0.36
Harry S. Truman	1.27	1.45	22.93	1.02	1.0	1.05	4.12	8.29	40.99	0.4	5.05	4.62	7.37	0.1	0.32
Herbert Hoover	0.7	1.02	22.82	0.76	0.13	0.45	2.54	3.24	53.46	0.51	4.96	2.48	6.93	0.0	0.0
Jimmy Carter	1.93	1.18	23.73	0.71	0.86	2.2	2.24	3.3	48.53	0.43	5.38	1.65	7.27	0.31	0.28
John F. Kennedy	1.3	0.67	23.66	0.74	1.26	1.67	2.46	3.05	49.29	0.26	6.51	1.49	7.11	0.15	0.37
Lyndon B. Johnson	1.33	1.25	23.85	1.0	0.33	0.33	3.65	5.44	45.74	0.31	5.7	3.52	7.36	0.09	0.11
Richard Nixon	1.18	1.18	24.52	0.62	0.36	0.21	3.24	3.86	47.97	0.21	6.12	2.42	7.97	0.05	0.1
Ronald Reagan	3.07	1.93	28.01	0.74	1.73	3.81	2.42	2.72	40.67	0.4	5.89	1.24	6.38	0.4	0.59
William J. Clinton	3.22	2.05	25.08	1.01	1.2	3.81	2.75	3.39	41.83	0.42	5.63	1.88	6.81	0.73	0.2

Fig. 7: Confusion Matrix for Character-level Simple RNN

Fig. 8: Confusion Matrix for Character-level Multi-softmax RNN

Given the lacking performance on even a simple RNN, one might expect that there is something more fundamentally challenged in our basic assumptions regarding character-level NN set-up; this is an obvious area of exploration to consider.

Comparison of Models

As can be seen in the results in Figure 9, the two word-based models significantly outperformed character-based models in predicting classifications for our current data set; in particular, LSTM model is the best in identifying the authors of these president speeches. The relatively strong performance of the word-based models with remarkably little tuning was very encouraging.

Model	Train loss	Test loss	Train accuracy	Test accuracy
Word-based RNN	0.6221	1.8764	0.8165	0.5204
Word-based LTSM	0.7485	1.5055	0.7591	0.5831
Char-based Simple RNN	2.2708	2.3008	0.2190	0.2093
Char-based Multi RNN	2.6327	2.6901	0.1385	0.0804

Figure 9: Comparison of models in the classification task

Conclusion

We were challenged in implementing our target character-based approach described by Bagnall (2015), including using the ReSQRT activation function described in the paper which we were never able to utilize with stable results. Unfortunately, Bangall (2015) was light on details code available online was a bespoke solution written in C with little-to-no documentation.

Over the course of the project, we learned that simple character-based architectures may not be effective and, in fact, state-of-the-art character models are vastly more complicated and employ CNNs, highway networks, RNN+LSTM layers and other innovations and often many of these methods used together. These more complex models provide a path to continue exploring this problem.

Despite the challenges we encountered implementing the target character model, we were able to learn a significant amount about how to apply each of these methods for problems we will encounter in the future.

Bibliography

Airoldi, E., S. Fienberg, and K. Skinner. “Whose Ideas? Whose Words? Authorship of Ronald Reagan’s Radio Addresses.” *PS: Political Science & Politics* 40, no. 3 (n.d.): 501–6. doi:10.1017/S1049096507070874.

Bagnall, Douglas. “Author Identification Using Multi-Headed Recurrent Neural Network.” In *CLEF 2015 Labs and Workshops, Notebook Papers; CEUR Workshop Proceedings*. September 8-11, 2015, 2015. http://ceur-ws.org/Vol-1391/150-CR.pdf.

———. “Authorship Clustering Using Multi-Headed Recurrent Neural Networks.” In *CLEF 2016 Labs and Workshops, Notebook Papers; CEUR Workshop Proceedings*, 791–804. Évora, Portugal, 2016. http://ceur-ws.org/Vol-1609/16090791.pdf.

Graves, Alex. “Generating Sequences With Recurrent Neural Networks.” *arXiv:1308.0850 [cs]*, August 4, 2013. http://arxiv.org/abs/1308.0850.

Nirkhi, Smita, and R. V. Dharaskar. “Comparative Study of Authorship Identification Techniques for Cyber Forensics Analysis.” *arXiv:1401.6118 [cs]*, December 24, 2013. http://arxiv.org/abs/1401.6118.

Pascual, Santiago, and Antonio Bonafonte. “Multi-Output RNN-LSTM for Multiple Speaker Speech Synthesis and Adaptation.” In *Signal Processing Conference (EUSIPCO), 2016 24th European*, 2325–29. IEEE, 2016. http://ieeexplore.ieee.org/abstract/document/7760664/.

Rosenthal, Jeffrey, and Albert H Yoon. “Detecting Multiple Authorship Of United States Supreme Court Legal Decisions Using Function Words.” *The Annals of Applied Statistics* 5, no. 1 (2011): 283–308. doi:10.1214/10-AOAS378.

Stamatatos, Efstathios, Walter Daelemans, Ben Verhoeven, Patrick Juola, Aurelio López-López, Martin Potthast, and Benno Stein. “Overview of the Author Identification Task at PAN 2015.” In *CLEF 2015 Labs and Workshops, Notebook Papers; CEUR Workshop Proceedings*. Toulouse, France, 2015. http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-papers-final/pan15-authorship-verification/stamatatos15-overview.pdf.

“Stylometry.” *Wikipedia*, April 27, 2017. https://en.wikipedia.org/w/index.php?title=Stylometry&oldid=777553870.