Presidential Text RNN

Leon Zhou

Domain

The aim of this project is to create a RNN text generator using a scraped corpus of presidential remarks from the official White House website. I am familiar with text scraping and basic NLP principles from previous work. I have a theoretical understanding of RNNs with respect to text generation, but do not have any experience in implementation.

Additional work or an alternative deliverable could be using clustering methods, topic modeling, etc. to group statements. Extensions would be incorporating the RNN as part of a GAN, or exploring other text generation methods, such as Markov chains.

Data

Data was gathered from the White House Briefing Statements Archive (**https://www.whitehouse.gov/briefings-statements/**). All entries labeled with "Remarks" were scraped. Items not involving President Trump were filtered out in the cleaning process. Where possible, I have tried to use actual transcripts as opposed to "prepared remarks."

As the data was scraped from webpages, all fields are strings. Approximately 420 statements are on record, ranging in length and topic.

Field	Example/Description
Title	Remarks by President Trump at Swearing-In Ceremony of Gina Haspel as Director of the Central Intelligence Agency
URL	Link to page
Text (cleaned)	Well, thank you very much. And good morning. I want to thank all of you and our distinguished guests for joining us today for a ceremony like few will ever have again…

At the end of scraping, approximately 580,000 tokens were generated. Taking the set of this list yielded approximately 17,000 unique words. Should this be insufficient, a database of presidential Tweets may be easily incorporated; my prior work with Tweets means I already have an established workflow for that format.

Known Unknowns

- **RNN Implementation:** I have seen examples for implementing similar networks and GANs using other frameworks, such as using Torch in Lua. Whether I can create a network of similar quality in Python is unknown, but should be doable.

- **Level of Generation:** Training the RNN at the character or the word level. Beyond knowing that these are options, I have not yet researched why one might be preferred over the other.

- **RNN training time:** I know recurrent networks take a long, long time to train. It may be possible to sidestep this with use of AWS. I would need further research into that, as my previous attempts to run substantial things on AWS have been unsuccessful due to difficulty getting all my dependencies in order.