RelationshipGuruBot, Tweet Generation Bot, Obama Speech Bot

EE126 , Professor Ramchandran

OCTOBER 22, 2015

UC-BERKELEY
Chris Jeng, Don Han, Giulio Zhou

Introduction

We live in the Information Era, where the amount of data in the internet is too much for any individual to consume. Our project aims to (comically) alleviate this problem by generating summary versions of texts using a Markov Chain. We tackle several domains:

1. RelationshipGuruBot (RGB) - Given a relationship/love related problem/question, forms a sophisticated answer based on the comments of similar questions from Reddit's /r/relationships sub-Reddit.
2. Tweet Generator Bot (TGB) - Spawns a creative, deep, and meaningful sample tweet of the person of your interest.
3. Obama Speech Generator Bot (OSGB) - If you've ever missed on of Barack's acclaimed speeches, you can recreate the experience by reading a summary version.

All three of these bots are powered by 100% recyclable, organic green Markov Chain energy, which applies customized weighting metrics to produce the ideal experience for the user.

Methods:

The key backend feature is the Markov chain, weighted by relevance to the input query and the number of upvotes on the Reddit answer. For non-serious mode, extremely negative answers were given very heavy probability weightings. For each comment, consecutive pairs of words are parsed and saved in the MC. Files are each given a weighting score, and all weightings are normalized to probabilities before generation.

Sentence generation starts on a (uniform) randomly chosen word, and a random walk. When aiming for a target length, we attempt to end on a word that naturally ends in a punctuation. Strings are built using Java's StringBuilder to boost runtime, and HashMaps are used to optimize lookup times. Certain checks are applied to handle strange cases like run-on sentences, capitalization, and special word handling. But runtime boosts are actually insignificant, because our limiting factor is the scraping limit of Reddit (see Experiments section).

As for collecting data, the Tweeter Generation Bot uses Twitter Python API called Tweepy to handle requests. The Obama Speech Generation Bot pre-scraped obamaspeeches.com for all of its offline data to process.

Experiments:

Our foray into the realm of (semi) intelligent question-answering services began with our development of a Reddit web-scraper. As a starting point, we plugged our query into Reddit's own internal search (connections made using Python's urllib2 library) and scraped the resulting pages for content. We encountered a couple of minor issues: Reddit officially limits scraping to a 1 request every 2 seconds, but realistically, it's closer to 1 request every 8 seconds. We morphed

the Python scraping to be more patient to avoid the dreaded 429 error ("Too many requests"). Additionally, we would occasionally encounter NSFW (not safe for work) content, which requires an extra step to proceed (beyond the simple web-scraping abilities of BeautifulSoup). Rather than find a complicated workaround, we (for the user's safety and our sanity) ignore these entries. In the end, our question-answering program analyzes the top 10 posts and examines the top 5 comments of each post by default and also features a relationship-specific mode and non-serious mode.

Results/Analysis:

Here are some excerpts of our outputs of RGB, TGB, and OSGB:

Question: How do I get my girlfriend to love me? Response: *"please stop calling your girl a bmw, and usually it's illegal."* -RGB

"Unacceptable…CNN has very honest commentary." - Donald Trump, TGB

"Every American deserves the castros, not new op-ed de las órdenes ejecutivas, porque es lo." - Hillary Clinton, TGB

"…I supported this administration's pledge to distract us from happening again. I would slaughter innocents in Chicago…" - Barack Obama, OSGB

Limitations:

Quora

Initially, we wanted to scrape Quora and apply PageRank on the user-upvote network to use a better weighting system for our input data. For example, users with highly rated answers, more followers, and (recursively) popular followers would receive higher weighting scores. However, Quora requires authentication to view the posts, a major issue for our Python BeautifulSoup scraper. We instead decided to scrape Reddit because the html format was simple to parse, and the average answer produced more comical results.

NLP

Another major feature we have yet to explore is NLP. Currently, our Markov chain runs blindly on the input data. There is no grammatical thinking or sentence structure correction, and punctuation feels unnatural. If we applied NLP tactics, our answers would make more sense, but we didn't do so in favor of comical results (and less overall work). Additionally, we could have used NLP tactics to better determine the relevance of posts to examine and weight, improving upon Reddit's mediocre-quality search.