Lab 6: Mini Project

Introduction:

Following the success of PageRank, Markov chains have become a ubiquitous tool for data analytics, used in fields as different as genetics, statistics, and sports. One particularly interesting use of Markov Chains is to examine the structure of English sentences. Claude Shannon, the father of information theory, suggested using Markov chains to create a statistical model of a piece of text and then traverse this chain to generate pseudorandom sentences. Our project seeks to implement this Markov-chain based sentence generation and determine the effect that various parameters, such as the text corpus and frame size, have on the readability and grammatical correctness of sentences.

Methodology:

We first scan through the corpus, examining n contiguous words at a time (a sliding window). These words are a key into the dictionary transitions, which represents the transition matrix. Transitions is a dictionary mapping n-tuples of contiguous words that appear in the corpus to a dictionary that maps the word immediately after the n-tuple to the relative frequency at which it appeared after the n-tuple.

To understand this more concretely, consider the following sentence/corpus: “The dog likes the cat, but the cat hates the dog.”

With a sliding window of size 1, the transition dictionary would look like:
‘the’: (dog, .5), (cat, .5)
‘dog’: (likes, 1)
‘likes’: (the, 1)
‘cat’: (but, .5), (hates, .5)
‘but’: (the, 1)
‘hates’: (the, 1)

With a sliding window of size 2, the transition dictionary would look like:
‘the dog’: (likes, 1)
‘dog likes’: (the, 1)
‘likes the’: (cat, 1)
‘the cat’: (but, .5), (hates, .5)
‘cat but’: (the, 1)
‘but the’: (cat, 1)
‘cat hates’: (the, 1)
‘hates the’: (dog, 1)

If we started in the state ‘the dog’, our first transition would be from state ‘the dog’ to the state ‘dog likes’. Our algorithm picks a random state to start in (biased towards states with more successor states) and continues to walk on this Markov chain according to the given transition probabilities until it hits a recurrent state/dead end, at which point it randomly restarts.

The corpuses of text used to generate sentences were fairly large and diverse (e.g. IMDB Movie Quotes, Manually Annotated Sub-Corpus (MASC)).

Results:

Sample sentences:

IMDB Movie Quotes(window size 1):

- *“aaron as that. mrs bucket!”*
- *“payroll currency dispenser in my nudity. law not yet get away from sonic blaster's stopped being taken a warrant to come between glee isn't sad. we have learned from the door.”*
- *“ben? young sozin began at francie but i don't waste of money and the roof! what were white.*”

IMDB Movie Quotes (window size 2):

- *“debbie and make no pacts with sorcerers! not even my wife! john to aeryn do you know the story that major winchester had to see my team and that's what he's planning.”*
- *“ power. phoebe i'm phoebe mr. turner's new wife.*
- *“ perhaps the slowest brain cells. chief inspector you'd been arrested! ted arrested?*
- *“to man and dragon breath. strikemaster that'll work. my emotions give me a lot of alimony.”*
- *“ agent i've got a chat room hermes get ready to grow up so says traditional military doctrine. well we've bought crappy scripts before. peter yeah i think i've got a foreign uniform.”*

MASC Twitter Subset (window size 1):

- *“ever going around and work via twitter. this and t-shirts? trying to get up twitter i think deeply about anything tastes great otherwise”*
- *“changed my annual gigantic birthday post. i'm still putting finishing touches to work. feeling in the update section.”*
- *“it seems to weirdly productive. points for the yuletide season! merry little grillin' and multipoint”*
- *“learn facebook...waiting to get coffee. so far just about gay/straight alliances on way back to url*

MASC Twitter Subset (window size 2):

- *“decided to turn a whole room into a nightmare?”*
- *“testing my twitter capabilities via gtalk for the public..”*

- *“with that but you know a star are u saying that im not blind anymore! chillen in west covina ca getting a coffee drink from starbucks that doesn't taste like sugar order it with my adidas philippines jacket.”*
- *“want to sleep at innapropriate times? watched the secret to dating like a bird that i've set free..it's gone forever.”*

Looking at these sentences from a qualitative standpoint, it seems that the Twitter dataset performed roughly the same, irrespective of the window size. This is likely due to the fact that tweets are characterized by terse, often disjunct thoughts without proper grammar, so a Markov chain with window size 1 can emulate the behavior of the average Twitter user. Dialogue, on the other hand, cannot be as easily replicated by such a Markov chain.

Sentence Length:
One measure of the “quality” of an English sentence is its length. We analyzed the average length of a sentence as a function of the window size. From our randomly selected initial state, we added words to the sentence until the last character in the sentence was in [‘.’, ‘?’, ‘!’], or if we reached a state where there were no more states to go to. The following results were calculated by generating 10,000 random sentences from the MASC Twitter Subset and then calculating the average number of words:

- 1 gram: 11.065 words/sentence
- 2 gram: 11.9841 words/sentence
- 3 grams: 1.983 words/sentence

As we can see, increasing the number of words included in a given state beyond a certain threshold drastically decreases the number of words needed to reach an end state. This is likely caused by a lack of data; there aren’t enough instances of the possible successor states, resulting in many early sentence terminations.

Performance (Time):
Another consideration is the performance of our algorithm as window size varies. Just from the sentence, “The dog likes the cat, but the cat hates the dog,” we can see that the state space increases from 6 to 8 states. We timed how long it takes to initialize the transition dictionary/matrix and generate examples for window sizes 1, 2, and 3 on the file "quotes_1mil.txt".

For a window size of 1, our implementation takes about 10.5 seconds on average. For a window size of 2, our implementation takes about 42 seconds. For a window size of 3, our implementation takes about 45 seconds.

To generate 20 examples, for a window size of 1, our implementation takes about .2 seconds per example. For a window size of 2, our implementation takes about .5 seconds per example. For a window size of 3, our implementation takes about 5 seconds per example. This discrepancy can likely be attributed to the size of the dictionary transitions.

Conclusion:

The ability of Markov chains to replicate English sentences varies greatly depending on the corpus and the window size used. Whereas tweets were easily mimicked by a Markov chain, even with a window of 1, movie dialogue was not as easily reproduced. It is possible that a simple Markov chain model of English may not be sufficient to capture the complexities of grammar and syntax. This application of Markov chains is not merely limited to generating humorous pseudo-random quotes; spammers often use Markov chains to generate “human”-like text to evade spam filters. In the future, using the tools developed for this project, we may generate spam in this manner and test the efficacy of Gmail’s spam filters. Furthermore, we may consider implementing a Markov chain with finer granularity (i.e. the ability to transition to letters); such a chain could be used to auto-complete searches.