

Text generation with Markov chain

Sep 30, 2014 • Category: Programming

[Markov chain](#) is a stateless mathematical model describing a sequence of possible events. Text generation with Markov chains is as old as the model itself. I recently played around with it and it was pretty fun thing to do. I found a Python script that seemed a good start - [shaney.py](#). It probably originates from a experiment called [Mark V Shaney](#), a bot created by Bruce Ellis and Rob Pike. The bot was programmed to post random topics in Usenet newsgroups long time ago. The text was generated from previous topics using Markov chains, often creating quite believable content.



Markov chain is named after [Andrey Markov](#), a Russian mathematician

I modified the code a bit, and most importantly added a check to test generated sentences and parts of the sentences against the whole text to see if there is already a similar sentence or match. Without the check the generated content often seemed repetitive and boring. The code can be seen on GitHub: [markov.py](#). It doesn't do anything too exciting just yet, just generates a finite amount of random content from a sample text.

The size of sample text is an important factor and

should be large enough to generate anything interesting.

To make things more exciting, a text analysis and processing can come in handy. [TextBlob](#) is a text processing library for Python, that was really easy to start working with. It would be possible to generate more meaningful content, especially on the creative side where the norms of text are broader than usual. For example, poetry, haiku, small jokes or even song lyrics could be generated with Markov chains and some text analysis. [Generative art](#) is a whole new field that could be explored more. I think there is a great potential in this, and will try to build something more complex when I have more free time.

