Generate Random Sentence using PCFG and Markov Chain

Group Members:
Huili Chen
Xinyi Wang

Relevance
Our goal of project is to improve our automatic English sentence generator, which originally has been constructed using probabilistic context-free grammars (PCFGs) in our project 2, by using Markov chains, a probabilistic automaton that generalizes the non-deterministic finite automaton. .

Effort
Huili Chen: 4 hrs (April 17th), 2 hrs (April 18th), 0.5 hr (April 19th). Total: 6hrs 30mins
Xinyi Wang: 3hr (April 15th), 3hrs (April 16th), 2 hrs(April 18th). Total: 7hrs

Introduction

Hidden Markov Chain

Hidden Markov process is a stochastic process that has the Markov property. Markov process can be categorized into two types, which are discrete time and continuous-time. A Markov process with a finite or countable state-space is often referred as “Markov chain”[1]. A Markov chain is a collection of random variables {x_t where t is an integer index that increments with time}, of which value at the present time is dependent on its value at the previous moment. Since Markov chain is a stochastic process, each state change has a probability [2]. The Markov chain is a probabilistic automaton (PA), which extends the non-deterministic finite automaton (NFA) by adding two probabilities. The first probability is that of a particular state transition taking place, while the second probability is the probability that the automaton has in a given initial state. Additionally, PA includes the probability of a given transition and turns the transition function in NFA into a transition matrix or stochastic matrix [3]. Thus, Markov chain also inherits these properties of PA.

Markov Chain has been widely applied in different domains, such as text generators, genetics, information sciences, and Internet applications [4]. The Markov text generators sometimes are used to enable spam messages to avoid spam filters by injecting “realistic” Markov chain-generated sentences into email and comments so that spam filters cannot detect those spam messages [5].

Our project uses Markov chain to randomly generate “real-looking” English sentences, using words from a source text. The Markov chain in our project works in the following way. The Markov Chain algorithm is built on a hidden assumption that for every pair of words in a given text, there are some set of words that follow those words. Intuitively speaking, our generator joins words together in sequence, and it selects each new word based on how often it follows the previous word in the source text file. In other words, the Markov Chain takes the two current words and then randomly selecting a word to follow from a list of words generated using the original text file [4]. Thus, given a sample document, Markov chain can be used to generate superficially real-looking text.

There are some observations regarding the performance of the Markov chain on generating random texts. As we increase the size of the Markov chain, we have more choices at each transition and thus are more likely to generate a more real-looking text. In addition, the number of words taken into account when choosing the next word would negatively impact the randomness of the generated text, and may generate more realistic text. However, as the number of words used for choosing next word becomes larger, the computational time for generating the next word would also increase accordingly. Increasing the number of words too much would make the usage of a Markov model redundant because choosing too many words would be similar to choosing a whole sentence [6].

Code

The code is built upon CP2 which implemented random sentence generation with PCFG. We added random sentence generation with Hidden Markov Chain, which used Jane Austen's Pride and Prejudice as training text. The file "train.txt" could be changed to any text to imitate the writing style. Here is how the Hidden Markov Chain code structured:

1. Training text is read in and processed by NLTK[7] into word tokens
2. Hidden Markov Chain model is built by processing the tokens
3. For each generation, a random word is chosen as the starting point, and a certain generation length is set to generate the result.

Our program allows user to interactively choose either PCFG or Hidden Markov Chain to generate random sentences.

[1]Everitt,B.S. (2002) *The Cambridge Dictionary of Statistics*. CUP
[2]http://agiliq.com/blog/2009/06/generating-pseudo-random-text-with-markov-chains-u/
[3]https://en.wikipedia.org/wiki/Probabilistic_automaton
[4]https://en.wikipedia.org/wiki/Markov_chain#Markov_text_generators
[5]http://code.activestate.com/recipes/194364-the-markov-chain-algorithm/
[6]https://github.com/hrs/markov-sentence-generator
[7]http://www.nltk.org/index.html