Generating Startup Ideas with Markov Chains

Colton Gyulay cgyulay@college.harvard.edu
Antuan Tran antuantran@college.harvard.edu
Evan Gastman evangastman@college.harvard.edu

December 9, 2015

1 Introduction

Language modeling and text generation are important aspects of the greater field of natural language processing. In this paper, we use a naive language model based on Markovian assumptions to generate novel sentences from a text corpus. A major pain point for engineers and entrepreneurs lies in trying to determine what project or idea to work on. The specific application of text generation we examine in this paper seeks to alleviate this problem: our model creates fake, believable startup ideas using a dataset of company descriptions from the startup database Crunchbase.

Our generative model is primarily based on Markov chains, which are decision processes that are characterized by the Markov property. A decision process is said to have the Markov property when any given state is independent of the preceding and following states. We can express this more formally:

$$P(X_n = x|X_0, \ldots, X_{n-1}) = P(X_n = x|X_{n-1}) \ \forall n \forall x$$

Though the traditional form of the Markov chain was first produced by mathematician Andrey Markov, the research our implementation follows is provided by Alexander Volfovsky, 2007. [2] When generating sentences, the biggest difficulties we had were ensuring syntactical correctness and ensuring a desired level of creativity (i.e., not creating sentences that in large part already appear in the training corpus). We were able to dramatically improve syntax by accounting for parts of speech during training and sentence generation.

2 Background

3 Related Work

For instance, [1]

	Score
Approach 1	
Approach 2	

Table 1: Description of the results.

4 Body 1

A clear specification of the algorithm(s) you used and a description of the main data structures in the implementation. Include a discussion of any details of the algorithm that were not in the published paper(s) that formed the basis of your implementation. A reader should be able to reconstruct and verify your work from reading your paper.

5 Body 2

Algorithm 1 Here is the algorithm.

```
procedure MyAlgorithm(b)
    a ← 10
end procedure
```

6 Experiments

Analysis, evaluation, and critique of the algorithm and your implementation. Include a description of the testing data you used and a discussion of examples that illustrate major features of your system. Testing is a critical part of system construction, and the scope of your testing will be an important component in our evaluation. Discuss what you learned from the implementation.

6.1 Methods and Models

6.2 Results

For algorithm-comparison projects: a section reporting empirical comparison results preferably presented graphically.

6.3 Discussion

A Program Trace

Appendix 1 A trace of the program showing how it handles key examples or some other demonstration of the program in action.

B System Description

Appendix 2 A clear description of how to use your system and how to generate the output you discussed in the write-up and the example transcript in Appendix 1. N.B.: The teaching staff must be able to run your system.

C Group Makeup

Appendix 3 A list of each project participant and that participants contributions to the project. If the division of work varies significantly from the project proposal, provide a brief explanation. Your code should be clearly documented.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Alexander Volfovsky. Markov chains and applications. 2007.