# Preliminary Project Report

Thomas Palmer

February 17, 2017

## Contents

**1 Introduction** **2**
1.1 Overview . . . 2
1.2 Relevant modules . . . 2
1.3 Project Supervisor . . . 3

**2 Aims and Objectives** **3**
2.1 Aims . . . 3
2.2 Objectives . . . 3

**3 Methods** **3**

**4 Project Plan** **4**

**5 Progress to Date** **4**
5.1 Research and Review of Literature . . . 4
5.2 Exploration of Related Works . . . 5
5.3 Programming . . . 6
5.4 Comparison to current project plan . . . 7

**6 Planned Work** **7**
6.1 Further Reading . . . 7
6.2 Programming . . . 7

**7 Appendices** **8**
7.1 Appendix A: Gantt Chart of Planned Project Milestones . . . 8

**8 References** **8**

# 1 Introduction

## 1.1 Overview

The Internet is a vast resource for opinion, thoughts and discourse on many topics such as film and other media. There are countless reviews, ratings and comments about any given topic, and this is a valuable resource to mine in order to extract opinion and detail about what is being discussed.

The applications of Natural Language Processing (NLP) and more specifically Natural Language Generation (NLG) are powerful in this domain. Opinion mining and understanding of such a vast field of reviewers and people engaging in discussion can provide interesting data and context on the success of a film. Such methods are able to process and understand a very large corpus of text far faster than one may be able to read through all of the writings on a film manually.

An issue with summarisation and opinion mining of corpora such as these is that they do not necessarily provide representative criticisms or feedback on a film in the user-facing output, often only chunks of text that are deemed the most representative and metrics that are as simple as a positive or negative rating, or a list of keywords that have been extracted.

This project aims to create a system that solves these issues, and generates a review of a movie that is both coherent and insightful, related to the corpus of movie review text it is given.

A system of this kind could be employed in business - for example, reviews and articles about cinema can have a profound effect on their commercial success, and if enough respected reviewers pan a film it may become necessary to understand why - and a tool such as this could aid this process.

It could also be employed at a consumer level in order for a user to quickly evaluate whether or not they wanted to watch a film or buy a product based on a vast amount of review text that exists rather than the opinion of a singular reviewer.

## 1.2 Relevant modules

The Data Mining module is relevant to this project as much of it looks to be about mining text and information extraction, as well as the likely use of Bayesian classifiers which are prolific in Natural Language Processing systems.

The Artificial Intelligence module is relevant as we covered Rule Based Systems and Concept Learning. These may also become a relevant part of my project as the need for a

system tailored for the specific purpose of content determination during generating movie prose will be necessary.

### 1.3 Project Supervisor

My project supervisor is Dr. Christophe Rhodes.

## 2 Aims and Objectives

### 2.1 Aims

The primary aim of this project is to explore methods for text generation and then develop a system through which prose about movies can be generated. This prose should be in the form of a movie review, and should pick the most recurrent points or themes in a corpus of movie reviews discussing a singular movie.

### 2.2 Objectives

To develop a natural language generation system which picks points from multiple review texts and uses them to structure an informed review.
To develop an online platform to host generated movie reviews which can gather feedback and data on the reception of said review.
To develop an autonomous bot that can discuss movies (or at least reply with an opinion of a movie) over twitter. Ideally this would drive traffic towards the movie review blog which hosts reviews made by the bot.

## 3 Methods

I intend to meet the objective of generating movie reviews initially using a template-based system, which will allow the determination of content, and then move on to creating a fully generative system which I can iterate into that will attempt to avoid repetitive text or text ripped straight from another person's review.

I intend to program this mostly in python, with a small amount of PHP and the use of a MySQL database in the web page developed for collecting feedback. Python has been chosen as it has large library support within Natural Langauge Processing such as the Natural Language Tool-kit which I am familiar with, and various other libraries useful for tasks such as Web Scraping and building the corpora I need to generate text. PHP and MySQL have been chosen as I have previous experience using them.

# 4 Project Plan

The majority of the work is to be completed in Python, some of which is using the Natural Language Tool-Kit library[1]. While I aim to implement my final system without a heavy dependency on libraries (as my own sentiment analysis code as a rule-based system specific to film text with more detailed sentiment collection could improve the quality of the text generated), it is a useful tool for testing aspects of the system which I will not be able to implement as quickly before I decide on their use. See Appendix A for a Gantt chart of planned project goals.

# 5 Progress to Date

## 5.1 Research and Review of Literature

I have researched a variety of methods for generating and summarizing text, as well as methods for understanding and extracting information from corpora. These areas intersect somewhat, for example using Part of Speech tagging and computing keyword frequencies both occur frequently in general.

I began my research with the exploration of methods in which text is generated in general.

A very rudimentary methodology for generating prose is using a Markov Chain Model to certain degrees in order to generate a new text out of a given corpus. This can be fairly effective in generating random prose but is heavily dependant on how the input text varies, and does fall apart for larger outputs of text which lose structure and coherence.

Another method of generating text employed is the use of a Recurrent Neural Network (RNN). [2] The outputs of these neural nets do not appear to be particularly coherent but they do seem 'plausible' when a prefix is set for the machine learning algorithm to complete.

A large number of methods for summarisation of documents have been formed, at varying degrees of sophistication. These mostly appear to be applicable for the purposes of generating a movie review out of multiple documents.[3]

H. P. Luhn discusses a method for the automatic generation of a literature abstract through selecting significant sentences evaluated through word frequency distribution[4]. This is a methodology that can potentially be applied to automatic summation of a long plot summary to create a part of a review text. While results from this are feasible, the issue is that no understanding of the text is made, and text is not generated - merely sentences taken verbatim from the text.

Generating text which accurately sums up multiple documents with extraction of information seems a more challenging task.

R. Barzilay and M. Elhadad[5] attempt document summarisation using lexical chains (representing the source text using lexical chains), an improved methodology for generating text summary which takes in to consideration the document's structure and attempts to summarize each section, but again suffering from the same problem of not producing any new text and merely sentence chunks of the input.

Sentiment analysis is a primary part of my research into understanding language, as it is certainly required to be able to evaluate a film through prose about it. There are many methodologies used for this, using supervised or unsupervised machine learning or rule based systems.

One issue with some sentiment analysis is that its output is a baseline polarity rating, rather than a comprehension of why this sentiment is held. This is an area I seek to improve upon in generating my own reviews. [3]

### 5.2 Exploration of Related Works

There are a number of readily available free-online article summary systems which will use algorithms such as TextRank[6] in order to summarise a document by cutting it down into a smaller number of more summative sentences. These systems suffer from the problem mentioned in the previous section - functionally all they do is copy sentences straight out of a document.

Automated Twitter is a popular avenue of exploration for businesses and other users of Twitter who automate direct messages, replies and other functionality in order to provide support or other functionalities. There are also bots such as the account '@Horse_ebooks'[7] that gained a following for its strangely poetic tweets. There was debate at the time as to whether the bot was penned by a human or a bot.

Markov chain models have been used to generate prose[8], and is now employed on twitter in order to generate "tweet mashups" or tweets that sound like the poster, often to comic effect. These models are useful but within this context do not always produce something that is coherent and if it does it is simply copying text verbatim from the input.

A program named SCIgen[9] has been used in the past to generate random papers on the topic of Computer Science. It has in the past been used to generate papers which have been submitted to (and been accepted for) conferences. They use a 'hand written

context-free grammar' to generate likely sounding prose at each section of the paper, and essentially follow a series of rules to create text that could be feasible in a real paper.

### 5.3 Programming

The first piece of work I completed was the creation of a Markov Chain algorithm that takes an input corpus of text and outputs a Markov table, selects a starting point and attempts to construct a piece of text using that table by randomly deciding the next word based on the n-proceeding ones and the probability of the next word to follow it. It can be convincing in small amounts of text and a n-value must be found that creates an interesting chain. I have found that two proceeding words works for most texts but on twitter one proceeding word is more or less the only thing that works without copying text completely verbatim from the source. When generating tweets, I take my corpus from a search for a particular topic, or a user account with the sole purpose of reviewing film. It is easier to search for a film as this then means the tweet generated is more likely to be composed of text specifically about a movie, but taking text from a single user can provide interesting results that seem feasible. See Appendix B for sample outputs from these chains.

The next piece of work was to create a twitter bot that posts these outputs, based on a corpus of tweets. I completed this using the Twython library[10], a wrapper for the twitter API.

After this, I attempted to create a template-based system of generating movie reviews. Given pre-defined sentences with words that need to be filled, and merely the name of a movie, it would be able to generate a movie review that begins with an introduction, contains a plot synopsis, talks about the performance of the cast and crew, and then reports whether or not the movie is worth watching.

First, given the name of a film and using the 'themoviedb' API[11], it looks up the metadata pertaining to the film - cast, crew, director and genre.

Currently it takes a single movie review as an input but the method works the same when expanded to multiple documents. This shall be expanded upon when I have improved my review web scraping script.

I have implemented a method which splits a corpus of review text about a film into sentences and builds a dictionary of all the sentences which mention a member of the cast, crew or their roles. Then, using the Natural Language Tool-Kit's Vader (a rule based system), it tags them with polarity sentiment. I plan on changing this in the future so that sentiment is evaluated by my own Bayesian model or Rule Based System, but it was sufficient at this stage to have something working.

Next, it chooses an introductory sentence from a list of templates I have created, and works to fill this template based on the sentiment and metadata given. It then appends a TextRank[6] summarised plot synopsis scraped from IMDb. Then it selects randomly without repetition evaluative statement templates to fill until an arbitrary amount of text has been generated, and finally appends a recommendation evaluated from sentiment analysis performed of the text as a whole.

The effect of this is somewhat believable but lacks true insight one might find in a thoughtful review that discusses themes and concepts that occur in a movie review. See Appendix C for an example review generated.

### 5.4 Comparison to current project plan

I am slightly behind of the current project plan, with some of the planned implementation details yet to be complete, as well as testing that I had wanted to complete.

## 6 Planned Work

### 6.1 Further Reading

I have to explore more methods for extracting feedback from movie texts as it currently feels feasible to extract relevant feedback from corpora, but the generation of a movie review which also explores the themes and content of a film seems like a much harder challenge for the amount of time I have left. Although I have not yet found much on the subject of generating movie review prose, I intend to continue looking for more on this.

### 6.2 Programming

I need to develop a small blog-style platform with the ability for rating of prose generated in order to collect evaluation of their believability. This will likely be programmed in PHP and using a simple MySQL database for data collection.

The most important of the further work I want to do is create a more "true" natural language generation system, using my own architecture in order to generate movie review text that does not reuse chunks of text from other reviews.

As well as this, systems testing is still required to be completed as well as the implementation of more behaviours such as replies and use of sentence templates rather than Markov Chains to be implemented on the Twitter bot.

# 7 Appendices

## 7.1 Appendix A: Gantt Chart of Planned Project Milestones


## 7.2 Appendix B: Markov Text Generator outputs

A few examples of tweets generated using my Markov Chain method and the corpus of tweets from @FilmReviewIn140 are:
Awful ending. — B-
OSCAR winners drive this stylish, thought provoking & OVER - Gory throwback "DIE HARD meets DEMOLITION MAN...in space!
Easily the half way short. Unsatisfying ending. — B
director and brutally violent, this pedestrian horror in the last hour as Sci-Fi docu-drama. Very cool! — B

An example of a larger piece of prose generated using Markov Chains formed from the corpus of a movie review of Bridge of Spies:

"We are looking at two reflections, one in glass and one in glass and one in glass and one in watercolors. The truth is in the final act, highlighting that this flesh-and-blood man

is shot from behind. We are looking at two reflections, one in watercolors. The truth is in the average man, but he?s also often the smartest and most morally upstanding man in the 1970s, he?s become one of the best films of the piece and its deeply cinematic at the breathtaking attention to detail?the costume design, the sets?and realize how lived-in it all seems."

### 7.3 Appendix C: Generated movie review of Bridge of Spies

Steven Speilberg's Bridge of Spies, starring Tom Hanks as James Donovan, and Mark Rylance as Rudolf Abel, is an excellent film of the Thriller genre.

Donovan is given the report on Abel's case, and Donovan knows what kind of reputation he would gain for defending a suspected spy. Donovan meets with Abel in prison. The two sit together at a bar where Hoffman tries to get Donovan to tell him what Abel is telling Donovan, for the sake of the country, though Donovan refuses to say anything. Abel's trial begins, and nobody is on Donovan's side. The people in court think Abel deserves the death penalty for his supposed crimes, and nobody thinks Donovan can get Abel acquitted. By the end of the trial, Abel is found guilty on all charges, but Donovan convinces the judge to give him a 30 year prison sentence instead of the death penalty. However, Donovan thinks they should get Pryor back as well. The CIA only wants Powers back, but Donovan plans to make a negotiation regardless. Donovan meets three people posing as Abel's family before meeting Vogel. While the CIA thinks they should leave Pryor, Donovan makes a bold move by threatening the East German government. After he is confirmed, the exchange is made, and Abel crosses over to the other side, but not before letting Donovan know that he left him a gift. On the plane ride, Powers tells Donovan that he never told his captors anything, to which Donovan states that none of it matters anymore. He then opens the gift from Abel, which is a painting of Donovan himself.

Tom Hanks performed well in the role of James Donovan. Mark Rylance's performance was strong. Matt Charman has fulfilled the role of writer well.
Steven Speilberg has definitely delivered with his latest film.

## 8 References

## References

[1] *Natural Language Tool-kit.* `http://www.nltk.org`.

[2] Geoffrey Hinton Ilya Sutskever James Martens. *Generating Text with Recurrent Neural Networks*. http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Sutskever_524.pdf.

[3] Lillian Lee Bo Pang. *Opinion Mining and Sentiment Analysis*. http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf.

[4] H. P. Luhn. *The Automatic Creation of Literature Abstracts*. http://courses.ischool.berkeley.edu/i256/f06/papers/luhn58.pdf.

[5] Michael Elhadad Regina Barzilay. *Using Lexical Chains for Text Summarization*. http://scholar.google.co.uk/scholar_url?url=http\%3A\%2F\%2Facademiccommons.columbia.edu\%2Fdownload\%2Ffedora_content\%2Fdownload\%2Fac\%3A160051\%2FCONTENT\%2Fbarzilay_elhadad_97.pdf&hl=en&sa=X&scisig=AAGBfm1-hlclQyAND4sOoh9b_i8tRHRf4A&nossl=1&oi=scholarr&ei=cqieVeqaCcaS7AaX_aSoBw&ved=0CB8QgAMoADAA.

[6] Paul Tarau Rada Mihalcea. *TextRank: Bringing Order into Texts*. https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf.

[7] Horse Ebooks. https://twitter.com/horse_ebooks.

[8] Yisong Yue. *Mark V Shaney*. http://www.yisongyue.com/shaney/.

[9] Dan Aguayo Jeremy Stribling Max Krohn. *SCIGen*. https://pdos.csail.mit.edu/archive/scigen/.

[10] *Twython*. https://twython.readthedocs.io/en/latest/.

[11] https://www.themoviedb.org/.