Movie Review Generation

Thomas Palmer

Goldsmiths College

University of London

A thesis submitted for the degree of

B. Computer Science

May 15, 2017

Acknowledgements

Personal

Thanks (obviously come back to this when u can think of something that doesnt sound embarassing)

Institutional

Abstract

This project investigates methodologies for generating movie review texts. It provides implementations and evaluation of several systems designed to generate review text for a given film. Several approaches have been taken towards generating these texts - a system based on Markov Models, a template based system, and a Natural Language Generation approach to producing the texts. In order to evaluate these systems, a bot for Twitter has been created, alongside two database-backed PHP websites. One posting reviews under the pretence of being written by a real human, and one for running a comparative test.

Contents

Appendices

List of Figures

List of Abbreviations

NLG . . . . . Natural Language Generation - the field of research dedicated to the computational generation of natural sounding text.

NLP . . . . . Natural Language Processing - the field of research dedicated to the understanding and processing of human language.

1 Introduction

Contents

1.1 Overview of the System

This project attempts to create believable reviews for a movie given a corpus of text (such as reviews of that specific film taken from users of IMdB) which discusses it.

There are a collection of systems I have implemented for this project which range from simple to greater complexity, which attempt to create believable text in review or discursive form.

The first system is a review website created in PHP, using a MySQL database to host content, which host movie reviews generated as well as to be a platform for collecting user data, feedback and analytics. As well as this system - a further PHP website for comparative review testing has been created to supplement the gathering of feedback.

The second system is a bot for the website Twitter, which uses Markov chains to reply to tweets discussing movies and post discussion in the tags for specific films, in order to drive traffic towards the review website as well as explore the ability of Markov chains to generate believable text.

The next system is a collection of methods used in generating reviews for movies. The first being the use of a Markov Chain model on a corpus of movie review text. Next, which aims to be more insightful is a template-based system which mines sentiment from a corpus of text about a specific film as part of speech tagging to create text based on opinions expressed in the corpus of text given, and outputs a structured review. The third system attempts to use a Natural Language Generation methodology in creating a review, following the 5 part methodology proposed by Dale and Reiter.

1.2 Motivation

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.

This project aims to create a system that addresses 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. It would prefer to be somewhat summarative of the

corpus it is given, but due to the natural polarity of movie review text it would be difficult to engineer and assess quite how summarative a piece of work may be.

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. As well as this, it may simply be used for entertainment purposes, to fool or generate conversation about a film in particular.

1.3 Thesis Structure

This thesis is structured as follows:
Chapter 2: This chapter provides background research into generating prose as well as evaluating text and a review of literature in these topics.
It also includes an exploration of past works in similar generative projects as well as human-like text agents such as Twitter bots.
Chapter 3: A specification of the project, including the design and implementation details of the systems discussed in this dissertation.
Chapter 5: Testing of the systems performed are documented in this chapter.
Chapter 6: An evaluation of the systems created.
Chapter 7: A conclusion, discussing the outcomes of the systems implemented.
After this, a bibliography and appendices section are included.

2 Background

2.1 Review of Literature

I have researched a number of ways in which text is processed for information extraction, as well as methods for generating prose and generating summaries.

2.1.1 Markov Chain Text Generation

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[1]. This can be fairly effective in generating random prose but is heavily dependant on how the input text varies and the selection of a good parameter, and does fall apart for larger outputs of text which lose structure and coherence.

The Markov models model text by building lists of n-words (usually 2 or 3), followed by the word that follows them in the text. Then, choosing an arbitrary starting point the next word is chosen randomly based on the frequency of how often the potential following word is found following the particular preceding n-word set. As the text is modelled on a real input, the output should look like it was penned by a human at least at a glance. The text produced is at the least feasible but is very likely to fall apart upon closer inspection or when creating

larger bodies of text. I find the outputs of these models at least interesting as a baseline in generating movie review text, as their implementation requires only a body of text and relatively little processing.

2.1.2 Document Summation

H. P. Luhn discusses a method for the automatic generation of a literature abstract through selecting significant sentences evaluated through word frequency distribution[2]. 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, no understanding of the text is made, and text is not generated - merely sentences taken verbatim from the text. This is not an issue in the context of its use in the research, but for the intentions of creating useful text out of a larger corpus it couldn't be used on its own as a solution.
R. Barzilay and M. Elhadad 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.[3]

The Textrank algorithm is another solution for the problem of document summation. It is graph-based and is used to rank key-words or sentences in a document in order to find the most summarative sentences or key-words[4]. It is based on the Page-Rank algorithm used in Google searches, and builds a graph out of the text. It provides strong summarative solutions as the connective vertexes for each key-word is used to vote for the most significant key-word, meaning the words selected are very likely to be the most representative of the text.
While all of these methodologies for summarising documents suffer from the drawback of not understanding a document, their use could be explored for summarizing a large text such as a film's plot to extract the most relevant sentences in the hope of providing a brief synopsis to a reader.

2.1.3 Part of Speech Tagging

Part of Speech tagging is a technique for automatically assigning and identifying what part of speech a specific word is (such as adjective, adverb, noun), with features for handling word sense disambiguation (eg identifying when can is a noun or a verb). Some of these are rule-based and some of these use machine learning methodologies to identify the part of speech of these words.
Eric Brill proposes a system for rule based PoS tagging, noting that most rule-based taggers have substantially higher error rates than ones that use stochastic methodologies.[5]

Kristina Toutanova and Christopher D. Manning explore a maximum-entropy model for PoS tagging, which uses a machine-learning approach to building a model out of pre-defined tagged text. [6]
PoS tagging could be used within a generative system for extracting valuable adverbs or adjectives referring to elements of a movie (such as performances, actors and direction) for a more precise evaluation of sentiment or choice of word in a generative system.

2.1.4 Wordnet

Wordnet is essentially a thesaurus in the form of a database of English language words grouped by synonymity, where each group refers to an individual concept. Each of these groups of synonyms is known as a synset and are linked to other synsets through lexical relations. [7] The primary relations are synonym, and antonym, but also cover the relations "hypernym" and "meronym". Hypernym meaning a word more specific than a less specific word (eg ewe as a hypernym of sheep). A meronym is a word that makes up a whole (eg leg being a meronym of table).

When it comes to lexical choice, this wordnet could prove invaluable in terms of understanding what words or themes occur in a text, as a frequently common

hypernym could indicate a word that is usefully representative of a corpus of evaluative text.

2.1.5 Sentiment Analysis

There are several approaches taken in order to evaluate sentiment in texts, one of which is using a rule-based system, and others being using supervised and unsupervised machine learning.

VADER is a rule-based model for the sentiment analysis of text[8]. It requires no training data, as it is rule based, and is constructed from a dictionary of words that has been selected manually by humans. It also features heuristics such as exclamation points and all-caps words increasing the intensity of the sentiment conveyed in analysis. It applies noticeably well to social media such as Twitter and other social media.

One area of expansion noticeable with sentiment analysis discussed in these areas is that they only tackle polarity (positive or negative) sentiment. This could be improved upon with the use of a more precise list of key terms with more precise polarity terms, and produce more interesting evaluation which for the purposes of selecting language in a more complicated prose generation system.

2.1.6 Building NLG Systems

The book 'Building Natural Language Generation Systems', by Ehud Reiter and Robert Dale explains the building of NLG systems. It proposes a 5-step process for document generation, one which I intend to follow in the building of a more complex prose generation system. The steps are:
Content determination: The gathering of data which will be included in the document.[9]
Document structuring: Ordering the data in such a way that makes sense in regard to the format of the document - such as describing the most important members of

a cast before someone in a less important role.
Aggregation: Combining similar sentences into larger ones to provide a better flow to the document as well as a more human-like text.
Lexical choice: Putting data to words - such as wording poor sentiment related to an actor as a 'bad performance'.
Generation of Referring Expressions: Creating expressions that call back to aforementioned subjects without using the same term repeatedly, such as 'the director' or simply 'Bloggs' in the generation of a review of a film directed by 'Joe Bloggs'.
Realisation: The final stage, converting all of the processes performed into a text that follows syntactic rules.

The primarily covers the explanation and building of NLG systems which generate texts to sound human-like from data sources that are less interpretation-based than what my project aims to deliver, but it is a useful resource regardless.

2.2 Related Existing Projects

A large amount of inspiration in terms of my methodology has come from currently existing language generation systems and projects.

2.2.1 NLTK

The Natural Language Tool-Kit is a large library for the Python programming language, which provides a large amount of functionality for processing language.[10] It offers Part-of-Speech tagging, a working Wordnet as well as pre-made sentiment analysis algorithms - machine learning and rule-based. This toolkit offers a lot of inspiration in terms of methodologies for developing understanding of language, although it does not seem to touch upon the generation of text. It also has access to VADER and an implementation of Wordnet, mentioned in the literature review.

2.2.2 Mark V Shaney

Mark V. Shaney was a Usenet newsgroup user whose posts were generated through forming Markov chains of other posts on the newsgroup.[11] The posts would often fool people into believing the comments were written by a real person. It is an early example (from 1948) of people using machines to generate prose in order to see how people react. I intend to explore how Markov chains hold up when generating review prose. As my project aims to create passable human-like text this approach to gathering data on how believable my systems are is also appealing as it provides much more interesting data than a more sterile Turing-like test approach.

2.2.3 Twitter Bots

A growing trend on Twitter is the automation of services and behaviours for accounts wishing to increase their outreach and handle having incredibly large amounts of follower engagements.

Archie[12] is a service which offers twitter automation for businesses and individuals, which implements a number of behaviours from targeting people talking about a particular market and engaging with followers and other twitter users. Many content creators wishing to gain more traction will use these automated twitter services to follow, like or message users who have tweeted about topics they have related to.

Some of these bots however offer no purpose other than amusement and fooling people who may believe that they are human. These often generate tweets in a similar way to Mark V Shaney did, with Markov chain models that generate text through choosing the next word out of a corpus of user tweets probabilistically, based on words that precede it in that corpus. For example, there are bots that generate their tweets through applying a Markov Chain model to Donald Trump's tweets[13], and many people have applied this to their own personal twitter accounts.

2.2.4 Parody Generators

There are several projects which exist that generate text which looks believable, but upon closer inspection is clearly nonsense. The post-modern generator uses Recursive Transition Networks (RTNs) in order to produce text, instead of Markov models, noting that the text produced from them tend to be "choppy and incoherent".[14] A RTN is a diagram showing how a task may be performed - essentially a directed graph with no cycles such that following the graph will take you from the start to the end of a task - in this case generating a sentence.
SCIgen[15] is another similar project that generates random Computer Science research papers. These generated papers have notably been submitted to conferences suspected to have low submission standards in order to test how stringent they really are. It uses a "context-free grammar" to generate the texts, which is essentially a set of rules that describe the generation of all the possible sentences used within a generated paper. It consists of sentence templates as well as nouns, verbs, adjectives and adverbs which will be used to fill in the templates. This kind of methodology could prove useful when it comes to generating reviews, as informed generation of subjective text seems like an area which has not been explored to a great degree.

2.2.5 NLG Systems

There are many examples of NLG systems which exist for a multitude of different purposes. STANDUP[16] is a system which creates question and answer style jokes with the purpose of developing language skills in young children and those with disabilities affecting communication. The output jokes are puns, so must be generated with an understanding of word sense, synonyms and phonetic similarity in words. It aims to fill in a surface template for the words to be filled into, and present these puns through a simple GUI.

The STOP system[17] was a system built to create letters encouraging smokers to stop smoking based on their responses to questionnaires about their smoking

habits. It would use the information that they filled in to complete the leaflet, which was posted through the doors of the smokers. It was found that there was no significant effect on quitting smoking between those sent personalised letters or those who were sent regular ones.

I have found it difficult to come across any NLG systems that generate review prose or any subjective prose at all, with the majority of systems being in the domain of reporting facts or wording the results of a single search problem, such as STANDUP's wording of wordplay it has discovered or the STOP system using the data from questionnaire responses. The reporting of a subjective opinion seems to be a topic that has not been touched upon, or at least I have failed to discover anything like this in my research.

3 Design and Implementation

3.1 Aims and Objectives

3.1.1 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.

3.1.2 Objectives

To explore methods for generating text which reviews film. 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.

3.2 The Structure and Content of a Movie Review

From reading movie reviews from various sources and types of publication (such as user-submitted reviews as well as articles in publications or on their websites) attempting to observe their structure, I have found that between different movie reviews the structure typically varies wildly and different reviews focus on many different aspects of a film. Depending on the movie and the reviewer, many different topics can be addressed. Typically, more comprehensive reviews such as those published on news websites include an introduction touching upon the more important or noteworthy details of a film, followed by brief plot synopsis which may or may not be used as a vessel for the review of each element of the film as it becomes relevant within the context of the plot synopsis. After this, reviews typically get more in to detail on the film, commenting on how the cast and crew have performed, noteworthy themes and topics and other evaluative sentiment. To close the articles, they usually make a recommendation of the film, or some kind of statement about its quality.

User submitted reviews typically are shorter, and tend to focus more on elements a user particularly enjoyed, or quite often disliked (I have found user reviews tend to skew towards either glowing or completely negative with usually less cases for a middle ground). These usually focus on the believability of performances, special effects and other things that are much more immediately obvious than some of the topics paid reviewers address.

This, while general, gives me something to work with in terms of determining the kind of content that should go into a generated movie review.

3.3 Planned Order of System Development

Three methods of generating prose are planned. The first, utilising Markov Chains and input of specific movie reviews or tweets. The second, a template-driven system

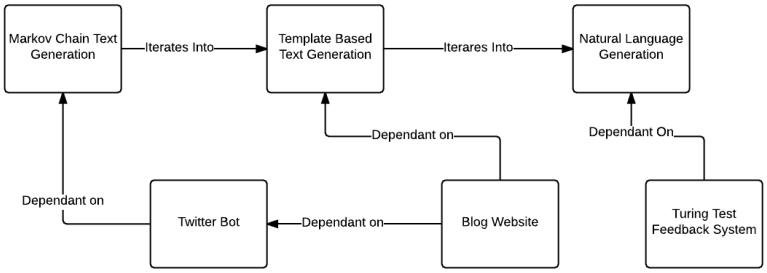


Figure 3.1: Proposed order of development for systems. Dependencies indicate works that be used without the other but does not require the system to be implemented

which will require movie metadata as well as a corpus of text on which to perform sentiment analysis. The final, an NLG system which will require further metadata as well as the corpus of text obtained from the template driven system in order to generate a review.

I planned to implement the Markov chain system first, since it required the least data and its relative ease of implementation. Data is to be obtained through twitter search or from a movie review as an input. The second system to be the template system, as it required the second least input data, and this data gathering can be iterated upon in the NLG system, which is why I chose to implement this last.

As well as this, it is planned that a twitter bot, and blog website be implemented in order to collect data on these systems working in the real world. These are planned to be implemented in tandem with these other systems - the Twitter bot needing to be made live after or at the same time as the blog is developed, as its main purpose is to drive traffic towards the reviews generated.

3.4 Markov Chain for Text Generation

Markov models represent a series of possible events whose occurrences depend only on the last event which happened. The probability of each event occurring is based on how frequently this event occurred in the data it is given. For example,

a Tuesday would have a 100% chance of transitioning into a Wednesday, given an input sequence of a week.

When representing language, these models are built from input corpora, and take a number of n-prefix words to model a point in the chain, and the list of single words (as well as their occurrences) preceding them in the corpus to represent the potential outputs. For example, the sentence "the dog and the hound the cat and the walrus" would produce a chain (with 2 preceding words) in the form of:

the dog	and
dog and	the
and the	hound, walrus
the hound	the
hound the	cat
the cat	and
cat and	the
the walrus	(no value)

This kind of table could then be used to produce an output of text by selecting a starting point, stochastically or arbitrarily (such as at the beginning), and following the chain, choosing its next word based on the occurrence of the preceding word. It could generate "the dog and the walrus" or "the dog and the hound the cat and the hound the cat and the hound the cat and the walrus". The chain could repeat "the hound the cat and the hound..." indefinitely if other ending conditions were not specified, such as a word or character count.

My implementation includes a breaking point of a character limit, as well as taking multiple documents (as opposed to a single long text) to build a chain in order to accommodate for using data sources such as twitter in place of longer texts, depending on what is being generated.

3.5 Template-Based System for Text Generation

The design of this system started with the premise of filling out template sentences for segments of a review in a random fashion until a text of a satisfactory size is

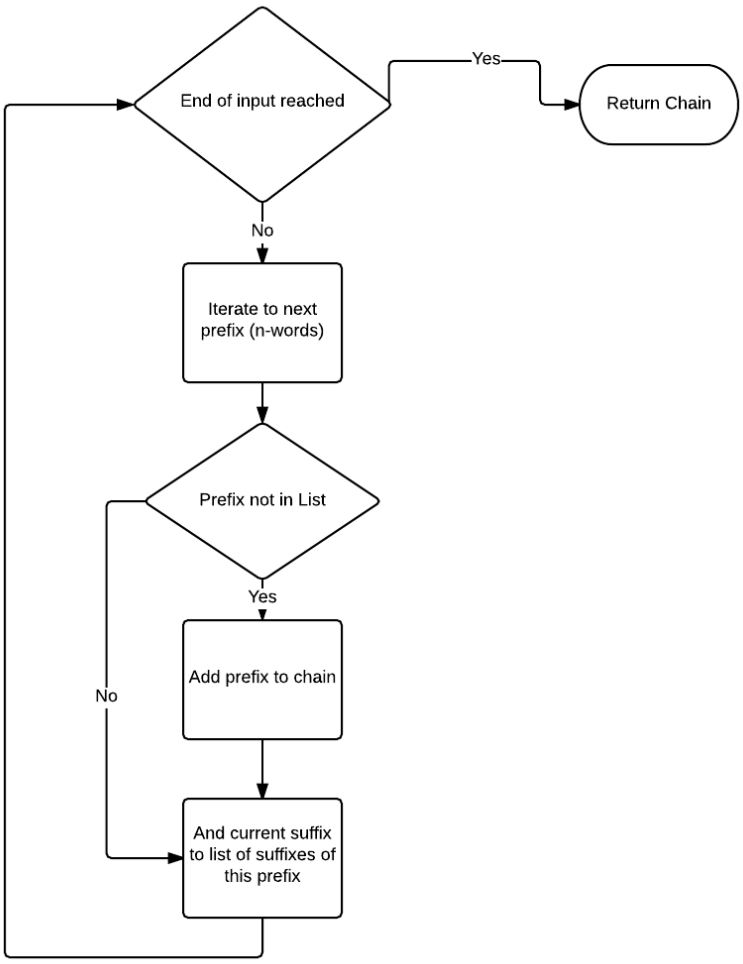


Figure 3.2: Flow chart demonstrating the construction of a Markov model for text input

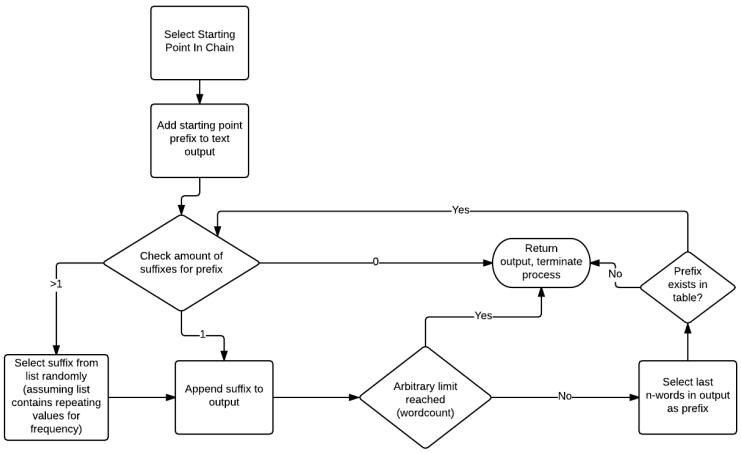


Figure 3.3: Flow chart demonstrating how text can be generated out of a Markov model

produced. The initial idea being that an introduction is filled in, then a brief plot synopsis, some text evaluating the performances of actors and noteworthy crew (such as the director or writers) and then a closing statement regarding whether or not its worth seeing a film.

The first step of the template system is data gathering.

It web scrapes a corpus of movie reviews for a given movie from IMdB. These are user reviews taken from a movie's reviews page, and an amount are taken sorted by their highest rating. Then it web scrapes plot synopsis, also from IMdB this is then summarized using the TextRank algorithm. It finally uses themoviedb's API to gather metadata about a film, which in this case is the cast and crew, their roles, and the genre.

The second step in this process is then forming an understanding of that data. First is the separation each review text into a large list of sentences so that they can be tagged and analyzed for sentiment. Categorise each sentence as about a particular topic or person (cast, crew, director), using the metadata gathered from the OMDb (the Open Movie Database) API to match sentences to these topics. Sentiment on each member of cast and crew as well as the director is then evaluated by the total value of these sentiments (a sentiment tagged sentence has a positive, negative and neutral value, and I measured polarity based on the larger of the two sums of positive or negative values).

The rest of the process is then building a review text out of templated sentences until completion. First, a templated introduction is selected and then filled out. This is a sentence that has missing sections to be filled in with factors such as the director, the name of the film, actors and their roles, and sentiment based adjectives and adverbs. In the templates, they are indicated by being surrounded by square brackets, such as '[director]', or '[genre]' - indicating they need to be filled by the name of the director and the name of the genre.

For the introduction, the aim was to inform the reader of the film's name, the director, some general sentiment, and the name of at least one actor and their role. Next, the TextRank summarized plot summary is inserted into the document, then it goes through several rounds of filling randomly selected templates to create a main body of review text, much in the same way as the introduction. The content of the main body consisted of evaluation of members of the crew, cast and director, providing adjectives and adverbs to add the illusion of informed opinion and sentiment. Finally an outro template is selected and then filled.
The outro template consists of a sentence that recommends the movie based on some factor such as the performance of the director, an actor, general sentiment and liking the genre. It too uses the Part of Speech tagging method, or a general sentiment based word selection.
Adjectives and adverbs in this review were selected in one of two ways, depending on the parameter given to the system. The first was a simple polarity based selection of 'good' or 'bad', 'well' or 'poorly' that fill out the templates appropriately, although in a fairly bland style. The second used a Part-of-Speech tagging system to retain all of the adjectives and adverbs from the sentences related to the subject we wish to add sentiment to, and then these words individually tagged using the VADER sentiment analysis method included in NLTK to return a (hopefully representative) sentiment. In the absence of any adjectives or adverbs in the corpus, the simple polarity words are returned.

3.6 NLG System

The design of the NLG system is in line with the 6 phases outlined by Dale and Reiter in 'Building Natural Language Generation Systems', although it was difficult to design a system which could report opinion.

Content determination:

Content determination is the step in the process which simply involves deciding what content should be included in the generated document. It proved difficult to gather all of the right content that is commonly discussed within the domain of

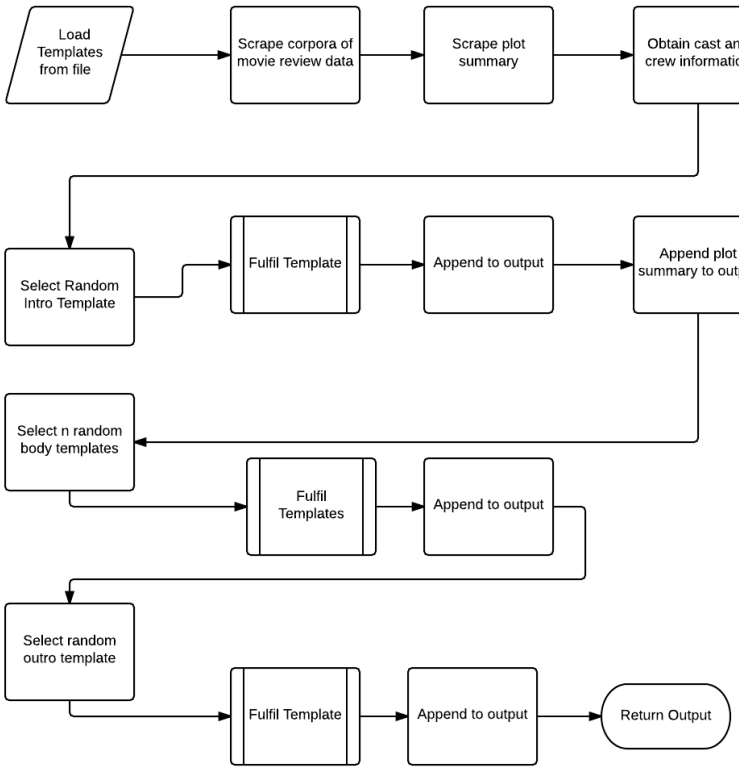
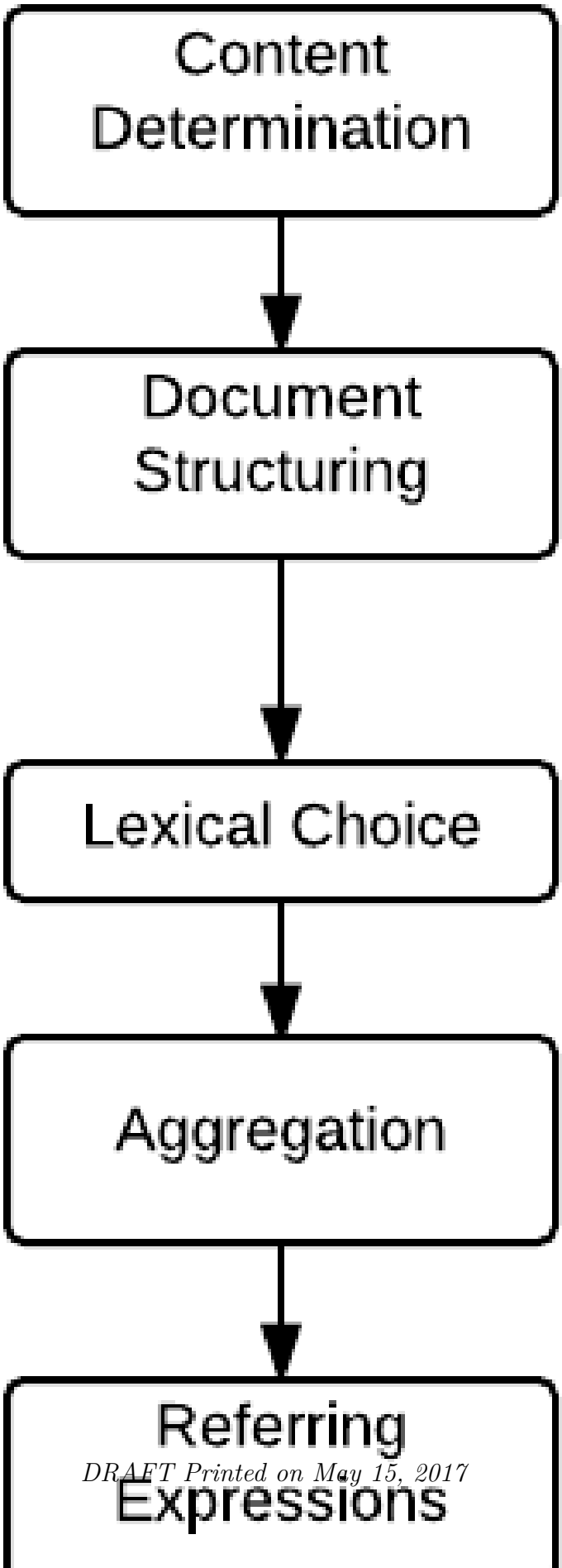


Figure 3.4: Flow chart visualising the process of the template-based system.

a movie review. I limited the range of discussion to the cast, crew, director and their performances, as I did not have time to develop a method for discovering the presence of the discussion of abstract topics such as tone or themes in text.

For my purposes this stage involves data gathering and web-scraping, as well as making decisions as to what of this data is used in the document. Much like in the template-based system, the first step is obtaining all of the data I need for generating a document. I reuse all of the data-gathering methods from this system, but while I am forming an understanding of this data, I also search for the past appearances of the most relevant cast and crew in other movies, as this provides more content to discuss. This uses the TMDb 'Discovery' API, which did require some working around as multiple names in a search would not return films that only included the given cast and crew names.


Content Determination
Document Structuring
Lexical Choice
Aggregation
Referring Expressions

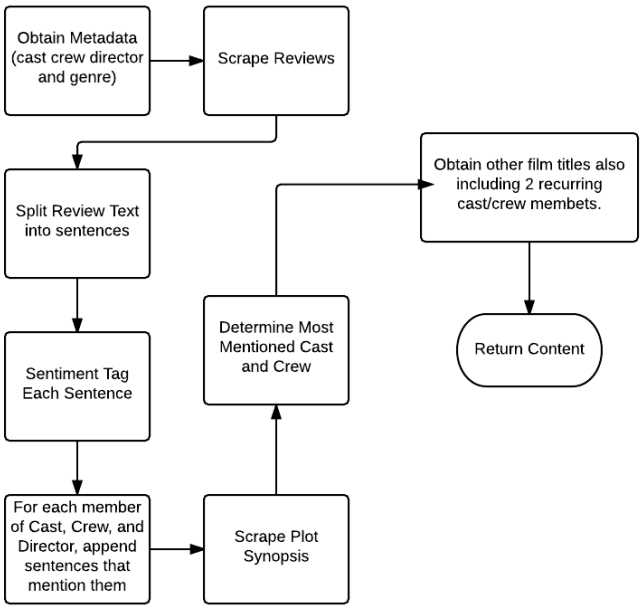


Figure 3.6: Demonstration of the content determination process

As mentioned previously, some determination of the most relevant members of the crew is made. It seemed most intuitive to design this as a simple search for the n-most frequently mentioned members of the cast and crew to put into the film review. Arbitrarily I chose 4 members of the cast, and 3 members of the crew as maximums, and iterated through the list of sentiment-tagged sentences associated with each member of the cast and crew.

With all of this data attained, the content determined is surmised as a list of the most important cast members with all the sentiment tagged sentences which mentioned them, with the same for the crew, and the director. Of course the title of the film is also a part of the content determined, as well as the genre, and a general overall sentiment to help decide on a tone. These elements have been gathered in the same way as they have in the template based system. I did choose to drop the plot synopsis, as I felt it did not provide particularly brilliant shortened forms from the use of TextRank, and copying a plot synopsis verbatim in order to seem more human felt inappropriate as it is a piece of writing that has actually been created by a human and not modified at all.

Document Structuring:

This is the stage which involves deciding the order of the document. For my generation of a movie review, I decided that this stage involved the structuring of an introduction, main body of evaluative text, and an outro consisting of a closing statement and recommendation of the film. I decided that this structure should be built upon the inclusion of specific clauses, which could later be aggregated into full sentences and pieces of content.

The structure of the document is stored as a list of clauses, along with the relevant data that should be used to fulfil that clause.

The introduction and ending statement of the review is structured in a different manner to the main body, which is simply the random selection of relevant topics to include inside of an introduction and outro, and placing each of these topics into a clause alongside the data that fulfils it. Topics included in the introduction are mention of the film's title, the films genre, the director (optionally including how well he performed), an overall sentiment of the film, and mention of the most relevant actor and his role (as well as the second most relevant actor, randomly). The 'outro' handles its structure in the same way as the intro, choosing from the topics of overall sentiment, the most relevant actor, the director, and the genre. It is worth noting that these structures may appear to be very similar, although the clauses are specified to be part of an outro or intro which is intended to change how they are worded.

The main body of the document is somewhat structured in a similar manner, although repetitions of the type of clause are made, rather than being limited to one mention at the most in the case of mentioning different members of the cast or crew. The topics to choose from were decided to be an assessment of the director, assessment of the cast and crew as well as mentioning their roles, additionally whether or not the actors were better or worse in past roles that they had worked together in (such as Quentin Tarantino and Samuel L. Jacksons performances being

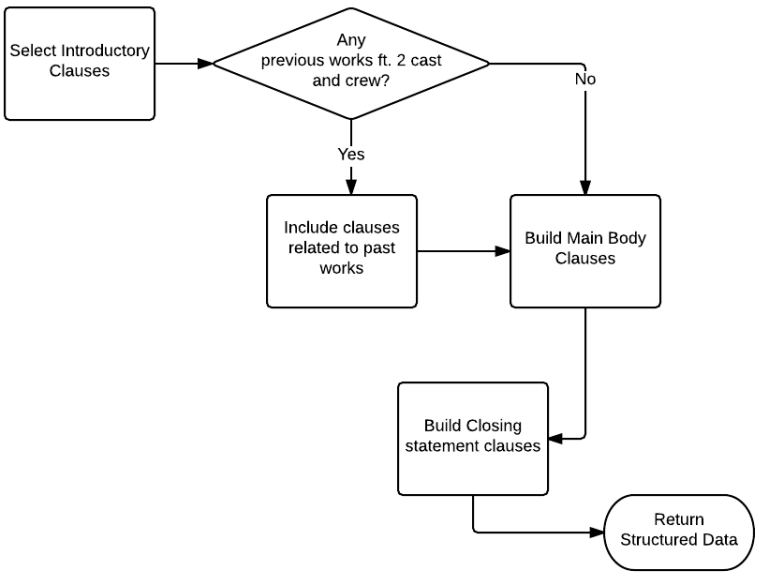


Figure 3.7: Diagram of the general document structuring process

compared between Pulp Fiction and Jackie Brown). If past work had been found, other clauses concerning people involved in this past work were mentioned close to this comparison.

After this, the rest of the cast, crew and director if still not mentioned in evaluation are then given clause pairs concerning their performance and describing their roles. When cast or crew members are mentioned, they are done so with pairs of clauses - one of which gives sentiment and another which describes their role or job. The order of the clauses is random and intended to make the description of the same type of thing multiple times seem more human in its variety.

Aggregation:

This stage involves combing the document structure in order to combine clauses into sentences so that they are more readable and flow more effectively. For example, the combination of two clauses such as "Actor plays the role of Character" and "Actor was believable" being aggregated into "Actor plays the role of Character believably", or "Actor was believable in the role of Character" depending on the order these clauses were determined to be placed in.

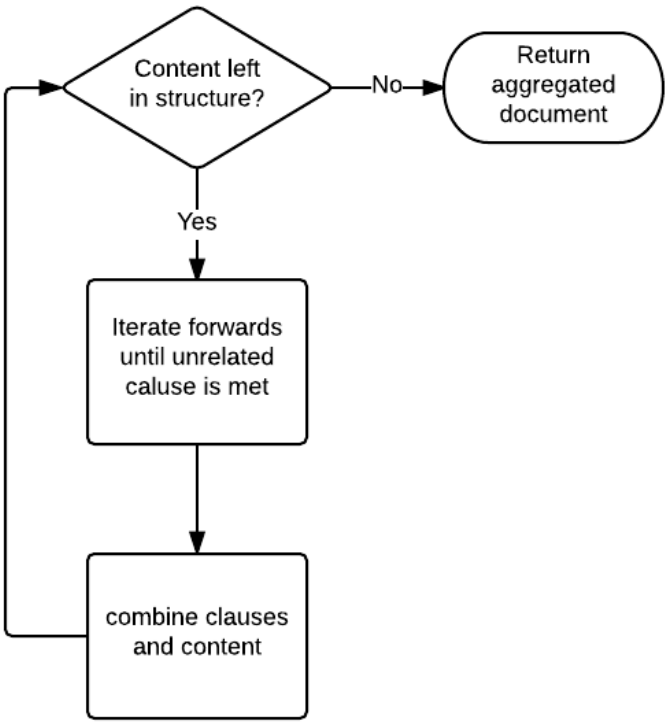


Figure 3.8: Simplified visualisation of document aggregation.

The introductory clause is concatenated into a single sentence concerning all of the topics it hits upon, maintaining the order each item is mentioned. The outro is also handled like this in order to make it more simple to handle during lexical choice, although it is not necessarily intended to be a single sentence.

Lexical Choice:

The choice of how exactly to word each of these sentences was quite a difficult problem to solve. I initially wanted to model the language in subject-verb-object pattern objects but realised that it would be more efficient to fill in patterns that already matched these sentences with pre-determined lexical choices, save for sentiment related adverbs and adjectives which would be chosen from the collection of sentences associated with the topic lexical choice was being made for. It also became clear that this was generating quite bland sentences, even with differing wording, so I included the insertion of sentiment based templates to add further variety to the text.

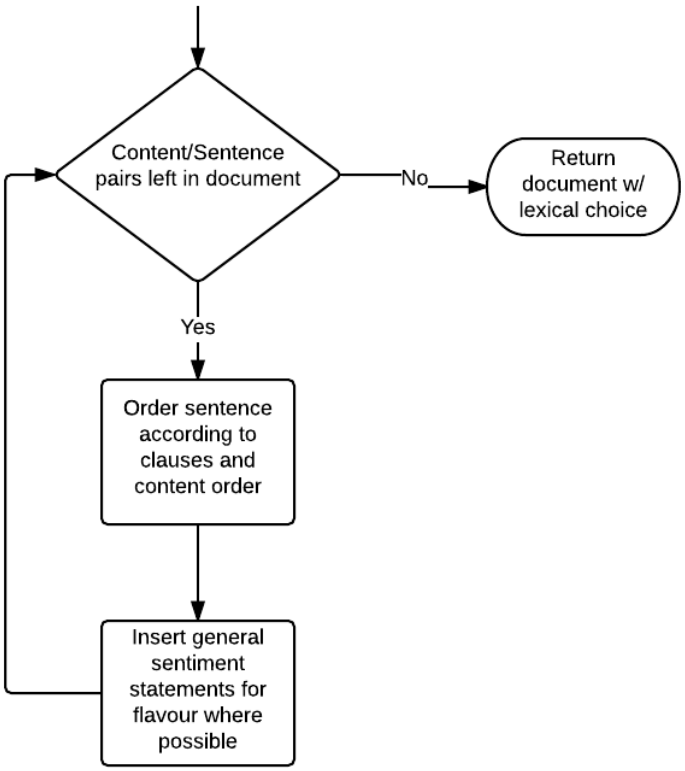


Figure 3.9: Simple model of lexical choice, how these choices are made are heavily dependant on the type of clause.

In order to make these lexical choices, it iterates through each sentence in the document and converts it from a description of the sentence and the required data to fulfil it into an array of words and phrases which can be exploded between sentences to create an actual sentence. These sentences often require a word that conveys sentiment to be fulfilled, which is a fairly large problem in itself.
I solved the issue of selecting relative sentiment through passing the list of sentences related to the topic in to a part of speech tagger, I have used the one included in the NLTK, and retaining only the adjectives or adverbs depending on which is required. If there are none of the required word, a generic word of either positive or negative sentiment is selected to fill the space. Otherwise, the word is chosen from this list of remaining adjectives or adverbs, and inserted into the document.

Referring Expressions:

The generation of referring expressions in NLG refers to resolving anaphora and how to refer to a particular entity within text, in order to improve the naturalness of the text created. In my case, I just needed to avoid the review sounding more robotic by referring to a person with their full name only each time, and make callbacks to their being mentioned earlier on in the text if they are. In order to resolve the issue of overuse of full name, a probability check is made and then if it passes, the full name is replaced with just the surname of the person being iterated through at this point. There is a further probability check made for repeated mentions, which includes a templated referral along the lines of 'bringing them up again'.

Realisation:

Realisation in my case essentially resolved to placing full-stops, commas and spaces in the document and converting it from a list of lists representing sentences into a single string. This was implemented mostly in a rule-based way that involved placing commas and full-stops in places that they are known to be needed in the structure of the document.

3.7 Twitter Bot

The Twitter bot exists in order to drive traffic towards the reviews generated and also act as a test of generating text in shorter formats than review - for example the markov chains discussed earlier are much more believable if you don't let them go on for too long. The limit of 144 characters is certainly suitable for this.
The bot itself uses a twitter API wrapper called Tweepy, which handles Twitter API requests required in order to make posts, search and navigate twitter.
The bot is programmed with two behaviours, which it enacts alternately. The first is simply the automated posting of links to movie reviews generated, with a template message that reads along the lines of "Read my review for #filmname here". The second is to search for tweets about a movie and use that as a corpus for generating

its next tweet. This aims to make the bot look like it is more realistic.

3.8 Blog Website

The design of the blog website is relatively simple. It is a Wordpress-like blogging website written in PHP, wish a MySQL database that stores the reviews, user information and the comments made and analytics for the site.

My own implementation of the website was chosen as I had prior experience making blogs in PHP, as well as the increased control over features (as I could easily modify my own code to enable/disable comments, a need for user accounts should it be targeted by spam, and analytics) seemed something which would benefit me more than the less familiar and potentially less flexible Wordpress. I have chosen MySQL and PHP as they are languages I am familiar with, and have worked using before, as well as their being suitable for the task of a small blogging platform.

The website itself is a front-page which lists paginated results for movie reviews written by the movie review generator, an interface for making/creating posts, and a way to view the reviews in full. It also has user comments for gathering qualitative feedback and page visits and engagements are tracked using Google analytics and a MySQL hit-counter.

Along with this website, an additional site was designed to collect feedback on the reviews generated, as a back up in the event there is not enough interest in the blog. This consists of an introduction page as well as a sequential loading of text of full reviews or sentences obtained from IMdB reviews and my own generated reviews. Feedback options are whether the texts are 'human-like', 'coherent' and an additional feedback text area which users are encouraged to enter into.

4 Testing

4.1 Introduction

Testing each part of this system involves checking primarily that outputs are what they should be (or at least as expected due to the outputs being semi-random in places), as we are handling a lot of data which is scraped from the internet, using APIs and other tools for gathering data from the internet.

4.2 Markov Chain System

4.2.1 Testing

This involved simply testing that the program executed at the correct point and could handle both of the input sources I gave it, as well as that the Markov table is built successfully from the two types of input and the actual obtaining of these inputs also works.

Test Case	Expected	Actual	Status
Twitter corpora obtained	Twitter API returns search data	Twitter API returns search data	Pass
Twitter corpora parsing	All non-reply data is trimmed, only tweet text remains	URLS of images are still retained	Fail
Markov Table Building - Single text source	Table builds successfully	Table builds successfully	Pass
Markov Table Building - Twitter as Text source	Table builds successfully	Table builds successfully	Pass
Text Generates from single review input	Text is output, no runtime error	output obtained	pass
Text generates from Twitter search corpora	Text is output, no runtime error	output obtained	pass

4.2.2 Summary of Changes Made

Images and URLS are still retained when the text is parsed so I introduced a check for hyperlinks and removed those from the text as well, since the reuse of this images and urls is not something I necessarily want.

4.3 Template System

4.3.1 Testing

Test Case	Expected	Actual	Status
IMDB Corpora Scraped	n*10 reviews scraped	n pages of reviews are scraped for a particular review	pass
IMDB Plot Summary Scraped	A plot-summary is scraped from IMdB	Plot summary scraped successfully	pass
Metadata obtained from API	Cast, Director, Crew, Genre all scraped from TMdB	The aforementioned data is all obtained	pass
Cast and Crew sentiment assignation	Cast, crew, director sent all assigned appropriately	Odd repetition and multiple of the same text to one cast member, some members not mentioned and assigned to a sentence anyway	Fail
Adverbs and adjectives searching	An adverb or adjective is obtained from a given corpora	Either adverb or adjective is returned, unless there aren't any one available	fail
Intro templates fulfil successfully	Each pre-defined introduction section is fulfilled when asked	They all were filled succesfully	Pass
Main body templates fulfil successfully	Each pre-defined introduction section is fulfilled when asked	They all were filled succesfully	Pass
Outro templates fulfil successfully	Each pre-defined introduction section is fulfilled when asked	They all were filled succesfully	Pass
Full review text	A full review text is obtained	Full review text is indeed obtained	Pass

4.3.2 Changes Made

Cast and crew being assigned individually tagged sentences that mentioned them was broken in a number of ways. For one, it would search for instances of either name of a cast or crew member by spitting the name obtained from the API into a list based on where spaces occur in that name or their role. This meant that

people with roles such as "The man with the hat" would be overly represented in the sample due to "The" and other common words being in their name. I included a list of names that could be removed from these acceptable names in order to mitigate this.

As well as this, the cast and crew members were being multiplied as it was not correctly identifying when cast and crew members had already had sentences assigned to them, so kept creating instances for them in the list. Instead of a list of actor/cast names, accompanied by a nested list of sentences that mention them with their sentiment polarity, it would mention the actor and cast member every single time, creating a much larger and much less efficient list. This was fixed so the data collection worked as intended.

In cases where adverbs or adjectives weren't present in the corpora provided, there was no method for handling this sentiment request. This was changed so that at the very least a basic adjective or adverb would be returned in the stead of something more interesting. "Good" or "well" would be returned in these cases, which is at least going to fulfil the template.

4.4 NLG System

4.4.1 Testing

Primarily involves testing that the new data obtained works as intended, as well as each stage in the NLG process was producing a correct result that could lead to an output.

Test Case	Expected	Actual	Status
Past Work Mining			
Most Important Cast/Crew			
Intro Structure			
Body Structure			
Outro Structure			
Intro aggregation			
Body aggregation			
Outro Aggregation			
Intro lexchoice			
Body lexchoice			
Outro lexchoice			
Referring Expressions			
Realisation			
Full text	A full text is obtained, from start to beginning	We get a full review	Pass

4.4.2 Changes Made

4.5 Website Testing

4.5.1 Testing of Blog Website

This testing mainly involved testing that aspects of the system that face the users work, as the rest can be handled by the PHPMyAdmin portal for interfacing with the database.

Test Case	Expected	Actual	Status
Create Account Submit	Account submits, database updates, pwd hashes+salts	pwd hash+salt successful	pass
Login Submit (correct details)	Login is successful	Login is successful	pass
Login Submit (incorrect details)	Login is unsuccessful	Login is unsuccessful	pass
Log Out Clicked	Account is logged out	Account is logged out	pass
Review Submit	Details are submitted into the db	Review is submitted successfully	pass
Review Edit Submit	Review is updated in database	Review is updated	Pass
Main Page Loads	Paginated data loads, review loads chronologically	Reviews load	Pass
Individual Review Loads	Review loads, comments section loads	Review and comments section load	Pass
Comments Submit	Comments insert into database for correct entry	Comments insert correctly	Pass
Comments Load	Comments load for correct entry	Comments do load	Pass

4.5.2 Testing of Turing-Like Test Website

Primarily involved checking data entry worked completely.

Test Case	Expected	Actual	Status
Data Fetches	Data for each review loads	It does	Pass
Completion of First Item	SQL executes successfully, update visible on database	Update is successful	Pass
Completion of Last Item	SQL executes successfully, visible on db	Update is successful	Pass

4.6 Twitter Bot

4.6.1 Testing of Twitter Bot

I have mostly decided to test that the Bot runs as expected, although it only has two outputs.

Test Case	Expected	Actual	Status
Generating Promotional Tweet	Posts	Post Successfully	Pass
Gen promo tweet (length intentionally too long)	Doesn't post, code continues to execute	As prior	Pass
Generate Markov Tweet (Acceptable Length)	Posts	Tweet outside of character limit (far too often to be inadmissable)	fail
Gen Markov tweet (length intentionally too long)	Doesn't post, code continues to execute	As prior	Pass

4.6.2 Changes Made

First, the bot would far more often than not attempt to create tweets that were too long to be postable using the Markov model, as it would not check for a character limit to have been exceeded due to this not being needed when simply generating Markov text. I added a parameter to check that things would work.

Then without my noticing, the bot would post tweets that were a single word long due to a mistake checking the tweets were less than the character limit and breaking the chain. It checked that the tweet was longer than 144 characters as a conditional for continuing, which no entry was as it was a single word to begin with, and just returned that first input.

5 Evaluation

5.1 Introduction

A well documented issue with Natural Language Generation and other methods for generating prose is the problems with evaluating such systems. It can be hard to extract statistical data about how natural or believable the language sounds as well to derive performance measures for such a system in a way you could with other systems such as machine learning algorithms with more obvious relevant performance measures such as time and space complexity and measures of accuracy. These are not particularly useful in the case of a NLG system where an output being created in a reasonable amount of time is sufficient.

The majority of my evaluation was planned to be on the review of comments and replies observed on twitter and the blogging website, and a separate scenario where the believability of review generated is asked for in a test environment against real movie reviews gathered from the internet.

5.2 Markov Chain text generation

This was used as my baseline for generating movie prose, although it definitely served to be the least feasible of all the methods used. This was because the chains

mostly produced complete nonsense when the corpora was expanded or the number of words represented as a node in a chain were decreased.
From my own observations, the text generated seemed more likely to be coherent at larger numbers of words per point in the chain, but were much more likely to form linear chains and copy text from the corpora verbatim, instead of producing something new. Smaller word counts would produce more original texts but would lose coherence much more quickly and fall apart much more quickly over larger output texts.

5.3 Engagement With Twitter Bot

The Twitter bot produced posts generated from Markov chains formed from corpora of tweets produced from twitter searches for terms related to certain movies. Every other tweet attempted to promote the blog website's hosted reviews.

5.3.1 Results

While the bot was running between the 17th of April to the 11th of May - a 25 day period - the tweets made generated a total of 13500 impressions. An impression is defined as any time a user has seen a tweet on Twitter.
An average of 551 impressions per day has been achieved while the bot has been running, although this has not amounted to a great amount of qualitative data - which is what I had hoped to obtain. The Twitter bot received very little attention in terms of link-clicks, follows and likes, even with hash tags for popular or controversial movies being used. A total of 34 link-clicks were obtained over this period, which is a very small amount relative to the number of overall engagements or impressions made over the time period. There were no real interactions with the bot outside of being followed by 3 other automated twitter accounts, all but one of which has since unfollowed. Occasionally words would trigger other bots to retweet them, but no human-controlled interactions were observed.

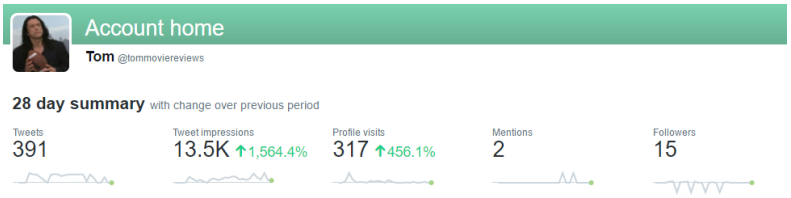


Figure 5.1

Figure 5.2

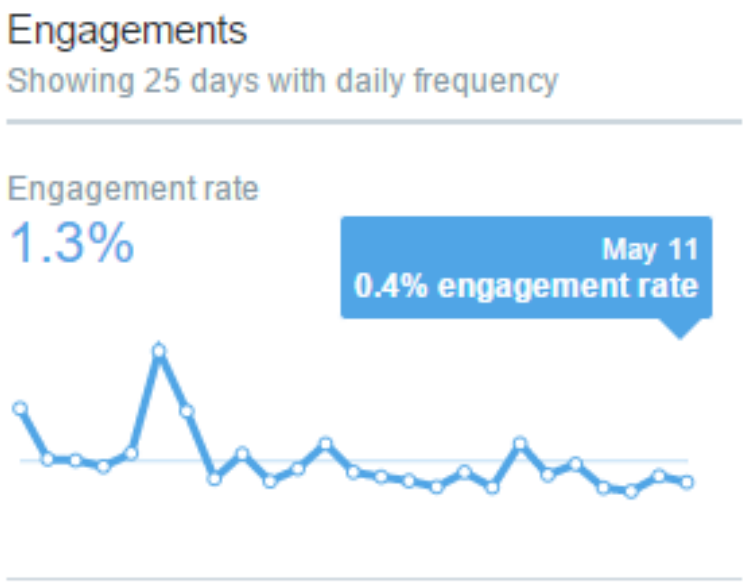


Figure 5.3: Engagements with tweets made by the bot

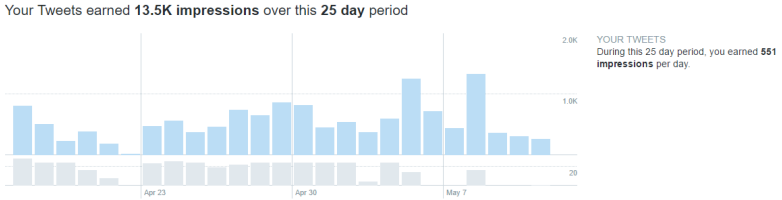


Figure 5.4: Impressions had from the tweets made by the bot over 28 days

5.3.2 Observations

The Twitter bot did not directly interact with any twitter users other than making posts, which may have limited the effect of its outreach. It would be interesting to implement post-liking or replying behaviours in the bot at a later date and see if that has an increased impact on the attention it receives. As well as this, the URL of the blog website may have been offputting towards potential readers and could have had an impact on the number of link clicks I have observed.
I would not say the bot was greatly successful in generating attention for the review website, due to the low number of link clicks had, but it did attract a small amount of attention using these relatively simple generative techniques.

5.4 Comments and Interaction with Blog Website

A number of reviews generated through the template system were hosted on the blog website, in order to see if they would receive any replies or feedback at all. This was later expanded to include reviews from the NLG system as well once that was implemented. It consists of an index page which loads a paginated list of movie reviews with a shortened preview of the article written with a link to the full text. The full text displays with a comments section below it.

5.4.1 Results

Unfortunately, there were no comments were left on the blog website, which is a shame as it would have been an interesting measurement of the believability of the reviews. Google analytics has however observed 53 unique sessions over the month the bot was active, involving 39 users generating 123 pageviews. The average number of pages visited in a session is 2.32 which indicates that users will have attempted to read at least one review, and clicked on to the homepage to view the rest of the site. The average session duration was 57 seconds, which indicates that at least some of the users stuck around long enough to attempt to read a review,

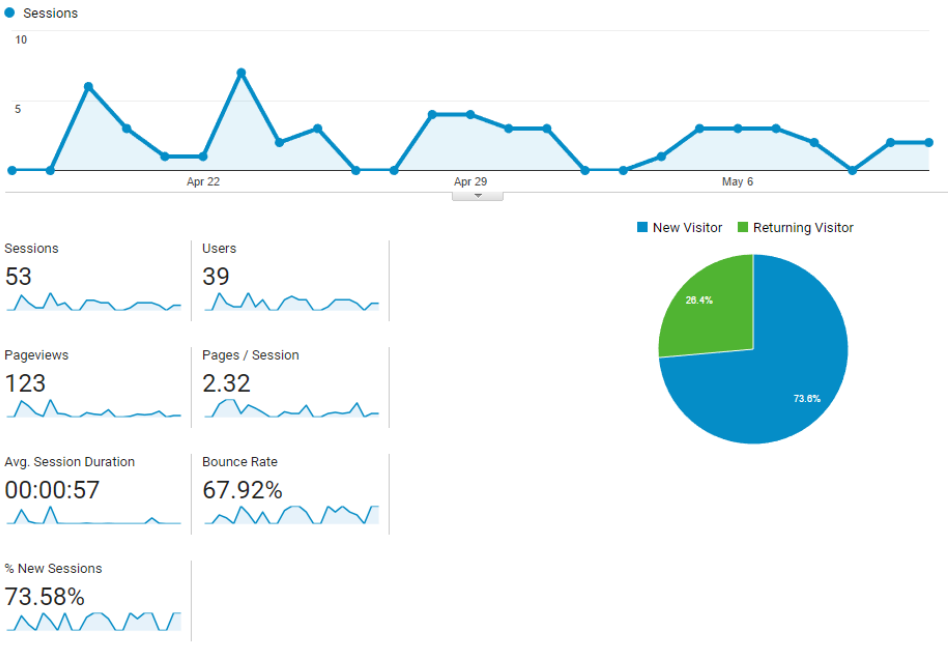


Figure 5.5: Diagram of unique sessions held on the blog website.

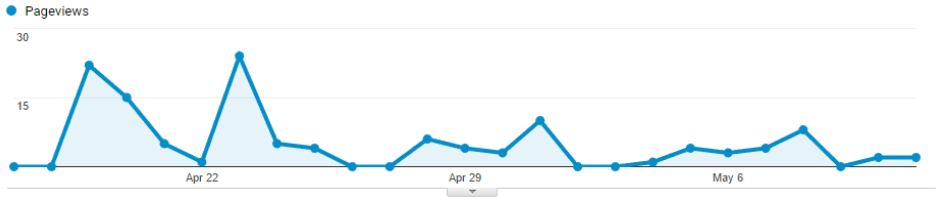


Figure 5.6: Pageviews during the time the blog website ran

although not all of them will have.

5.4.2 Observations

I believe that a small, but not great amount of success was had from the Twitter bot generating this many unique sessions, although a failure of the content to be

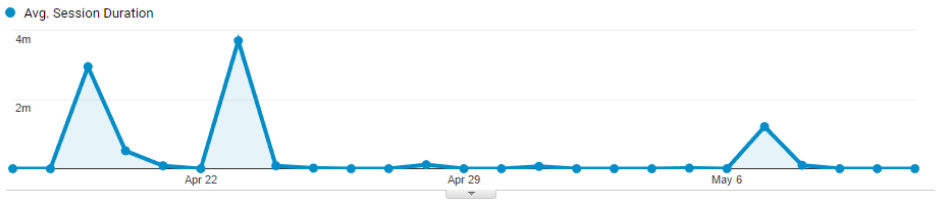


Figure 5.7: Average session duration over the life of the blog website

engaging enough to generate comment and feedback.
It is a valid criticism that the hosting of the website on the Department of Computing servers could throw people off from following the link and interacting with the website, as it is not a tidy URL (doc.gold.ac.uk/ tpalm003/reviews/index.php vs. something like tomsreviews.com) with an easily recognisable domain.
Another criticism is that the design of the blog is fairly simplistic and that its design may not be as convincing as other blog platforms for hosting movie reviews because of this.

5.5 Turing Test Scenario

In order to collect more detailed feedback on the NLG system, I set up a Turing-like test. Users were shown sentences and full reviews which were generated via the NLG system implemented, as well as real sentences and full reviews written by IMdB users. These were ordered randomly, and 4 examples of each case were selected. The users were shown these in a randomized order so there were no contextual clues as to whether or not the review was human or not, although the order that each user is shown the data was the same.

5.5.1 Results

I was able to obtain a good amount of feedback on the NLG system through this system. 11 respondents participated in the test (and 9 provided feedback for each of the examples given) and I managed to obtain interesting data on how the NLG system performs on the sentence and full text levels.
Generally, people found the full review texts that were generated to be less coherent and less human-like than real reviews, although the first review they were shown that was generated was received as coherent by 7 of the 11 respondents, and believably written by a human by 5, this effect wore off as they were shown more reviews and their structure was noticed. The final generated review seen was only thought to be human by 2 of the respondents, although voted coherent by 5 of 9 who made it

to the end of the feedback session.

One user wrote "after you've read a couple of these the pattern becomes very obviously robotic," which demonstrates the point that the text generated follows a structure that is easy to recognise over multiple examples.

On the sentence level, it became very hard for participants to tell the difference between a generated sentence and a sentence taken from a real review, where the difference in scores closed, although some users were able to tell a sentence was generated due to it's pattern being seen in previous examples of full reviews. Generated sentences generally scored 8 out of 11 for how human they were each time, and between 5 and 7 out of 11 for their coherence. In contrast, real sentences scored between 5 and 8 out of 11 for how human they appeared, and 2 and 8 for how coherent they were. It is worth mentioning that I intentionally included sentences from less well written reviews in order to throw users off in terms of the exact way in which the reviews and sentences are generated by my system.

5.5.2 Observations

At the sentence level, it was difficult for readers to identify between the NLG system and real written reviews. However, this fell apart at the full review level when the initial review they read often failed to appear human-like and after more generated reviews were shown to them they also often became able to identify their structure and easily tell that they were generated through this.

The generation of sentences which convey sentiment and opinion seems to have gone well with the NLG system, where users struggle to tell them apart from real sentences. There is an issue with the variety of these sentences produced as some of the patterns and their handling are hard-coded and have flavour added to them from templates, as the data needed to form the actual opinion proved too difficult

Position Seen	Text Type	Human	Coherent	Total Respondents
5	NLG Review	5	7	11
10	NLG Review	6	5	10
13	NLG Review	3	6	9
15	NLG Review	2	5	9
3	NLG Sentence	8	6	11
7	NLG Sentence	8	7	11
11	NLG Sentence	8	5	10
2	Real Paragraph	5	11	11
6	Real Paragraph	2	10	11
9	Real Paragraph	7	7	10
1	Real Sentence	8	2	11
4	Real Sentence	7	5	11
8	Real Sentence	6	7	11
12	Real Sentence	5	9	9
14	Real Sentence	7	8	9
16	Real Sentence	6	4	9

Table 5.1: Table displaying feedback provided on each excerpt, along with its type and order of appearance

for me to extract.

It would appear that a weakness of the NLG system is the rigid document structuring that has been implemented, which seems to quite heavily impact the detection of these reviews over multiple documents. While this is less of a problem for single documents, it is certainly a worthwhile improvement to have more fluid and variable structures when generating documents to give the impression of a human author.

As well as this, another shortcoming of the NLG system is that to discuss a topic, data needs to be gathered on it to begin with. This is difficult in the case of discussing themes, plot synopses, tone and other higher concept topics that are discussed in reviews of art. An interesting area of improvement would be to implement some method for identifying and extracting data on some of these higher concepts from the review text corpora.

6 Conclusion and Future Work

6.1 Review of Aims and Objectives

I would say that this project has been moderately successful. Some feasible movie review text has indeed been generated, although not a lot and not in quite as large quantity or variety as I had aimed.

6.1.1 Aims

The aim of this project was to generate movie review prose that passed as human. This has been partially achieved, at the sentence level, although a movie review generated by any of my systems is not likely to pass as human, mostly due to the falling-apart of coherence outside of reporting facts and sentiments. I would not say this was achieved with great success, as little interaction with any of the systems aiming to pass as human has occurred.

6.1.2 Objectives

To explore methods for generating text which reviews film.

This object has certainly been met. I have explored a number of options for generating prose and implemented several methods for generating prose. I have looked at Markov models, Feed-Forward Neural Networks, Template based systems

using context-free grammars, and Natural Language Generation architectures for generating prose.

To develop a natural language generation system which picks points from multiple review texts and uses them to structure an informed review.

Although I have implemented a system to meet his goal, the outputs generated do not manage to pass as human, and therefore I would not say I have met this objective with great success. As well as this, I made the concession of including templates into the NLG system in an attempt at making the system more human-like. More time to work on this system and I feel I could have met this objective more, although it would be partially template-based rather than strictly NLG.

To develop an online platform to host generated movie reviews which can gather feedback and data on the reception of said review.

I have met this objective successfully, although it had not been used as much as I had hoped for the collection of data. I could have put more time into the development of the site in order to make it more welcoming as well as hosting it on a URL that is potentially less scary than the URL of my system. (doc.gold.ac.uk/ tpalm003/reviews/index.php).

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.

This goal has been met, although its outreach is limited by the behaviours it puts into place. It makes posts and can post with hash-tags in an attempt to reach an audience, but it does not engage with twitter users through other interactions which would arguably make it a more believable bot, and at the very least have it seen by more people.

6.2 Lessons Learned

I have learned a number of things from this project, and had I started it again have a number of things I would do differently. At the programming level, I've learnt a lot about using APIs as well as interracting with XML and JSON objects, as well as parsing text and working around very error-prone systems such as requesting data that might not necessarily exist.

I have had to research language processing and sentiment tagging, although I did opt to use libraries for these in my implementation as there was little point reimplementing code for systems with so many interacting elements.

I'd also not recommend relying on user data for systems that require finding attention in the real world (rather than appealing to users for feedback), as this has proven to be difficult to manage. As well as this I would recommend that people choose less subjective topics to implement NLG systems for if they were to choose NLG as a dissertation topic.

6.3 Future Work

6.3.1 Improvements to current work

An interesting area to explore is expanding the Twitter bot to handle behaviours other than self promotion and Markov chain generated text to reply to others. This would enable a system which intended to draw more attention to itself to operate in a more human-like manner as a real twitter user is more likely to utilize all of the features of the website and have a greater outreach through the use of replies - as notifications are sent directly to the user rather than having to search for a tweet generated by the bot.

It would also be interesting to expand the NLG system to handle a larger range of topics within movie generation for a more believable and comprehensive review of the film, and expand the grammars used to generate the sentiment-driven sentences. This would mostly include much more specific sentiment analysis using lists of

sentiment tagged words for specific topics and themes that occur in cinema.

A further interesting expansion would be to target reviews at different audiences where people actively read for them, such as Youtube comments, reviews for Amazon streaming videos, and other websites that host users movie watchlists and reivews such as IMdB and letterboxd, as they have active communities that use will read these reviews to gage whether or not they want to watch a film, or for other purposes such as validating their own opinions.

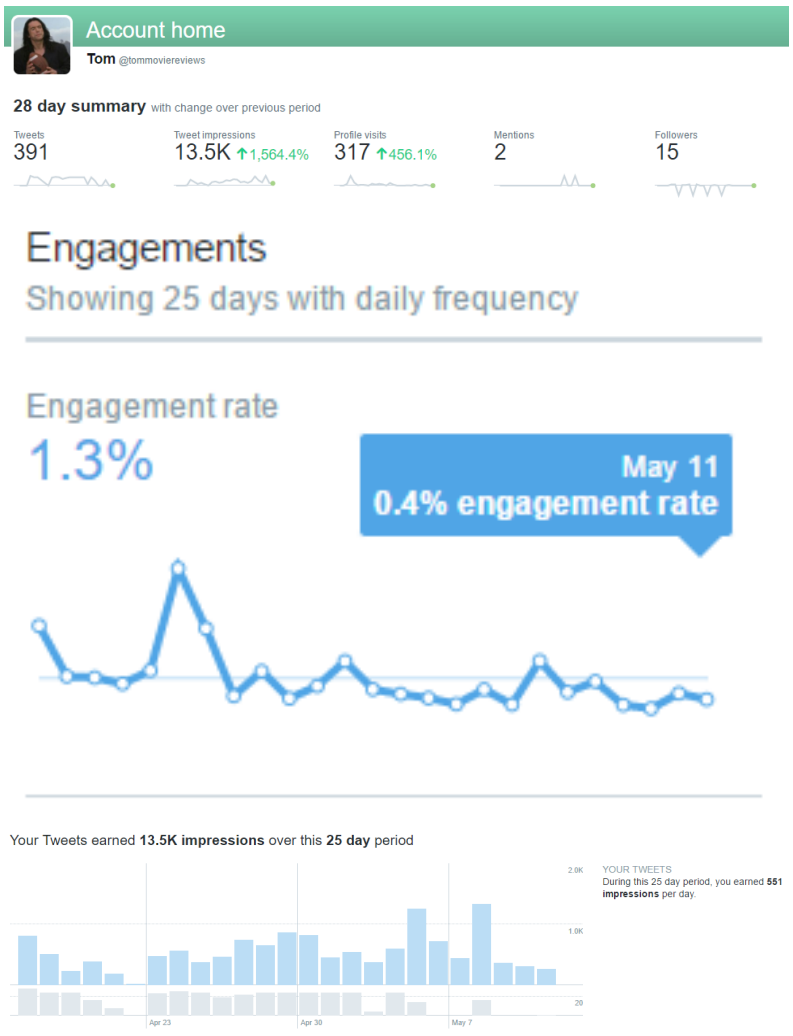
6.4 The Future of Generative Film Reviews

It is definitely safe to assume that the human-written review will not be made obsolete by reviews created from NLG methods for a long while yet. Given that a system like this would need to build its knowledge from some agent with enough insight to identify talking points and provide sentiment for each of these talking points, it is safe to say that without brilliant leaps in computer vision and audio processing, an article penned by a trusted reviewer is not going to be made redundant any time soon.

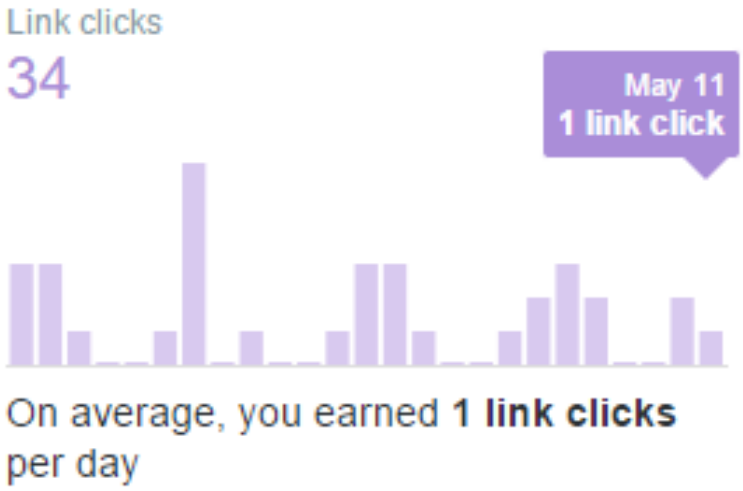
Appendices

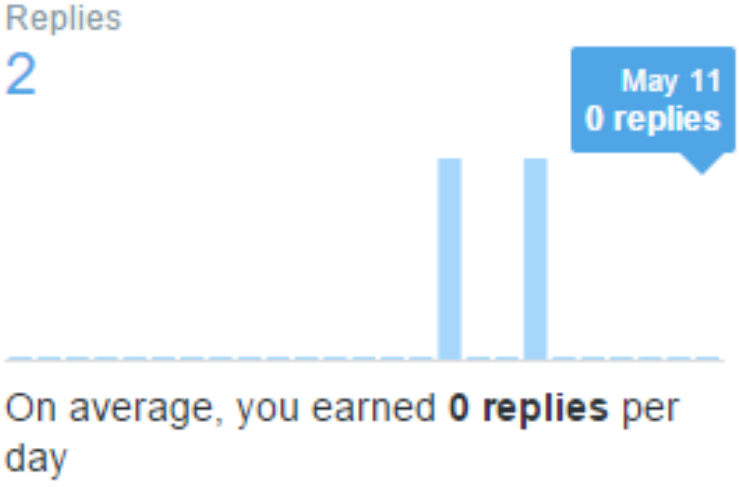
A
Evaluation Data

A.0.1 Twitter Analytics Data

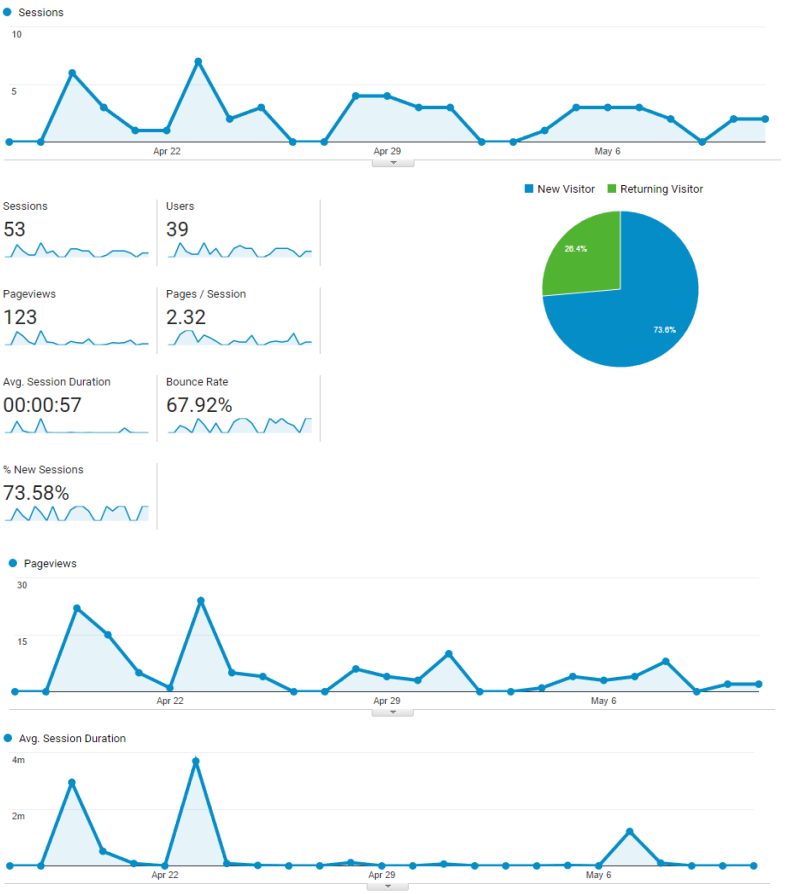



Likes
27
May 11
0 likes
On average, you earned 1 likes per day


Link clicks
34
May 11
1 link click
On average, you earned 1 link clicks per day


Replies
2
May 11
0 replies
On average, you earned 0 replies per day

A.0.2 Google Analytics Data



B

Turing-Like Test Results

Full text	Text Type	Pos	HumanLike	Coherent	Feedback
Argo is a Drama, directed by Ben Affleck as a true project of passion, starring Scott Anthony Leet as The Minotaur and Taylor Schilling. In a much less memorable movie The Hangover Part III features Leet again and John Goodman. Leet plays The Minotaur slightly. Goodman plays John Chambers accurately. Ethan Van der Ryn is Sound Designer not. John H. Samson has been cast as Construction Coordinator incredibly giving a forgettable performance. Has really embarassed theirself in this role John T. Reitz true is Sound Re-Recording Mixer. John T. Cucci plays Foley just and is not worth writing home about. (And it goes without saying) Leet as The Minotaur lightly and is not worth writing home about. As you may have already guessed by the tone of this review Goodman is John Chambers enough.Maybe. Fans of the genre will love this Drama Leet and again gives a standout performance Affleck bringing them up again has created a work of art Argo is worth coming to the cinema for fans of the genre will love this Drama, Leet and again gives a standout performance, Affleck bringing them up again has created a work of art, Argo is worth coming to the cinema for.	nlg paragraph	5			
			0	0	looks like copied and pasted parts from mixed stuff
			1	1	
			1	0	
			0	1	
			1	1	
			1	0	The size of this passage alone made me believe this was human initially, and the first few sentences support that. However the grammar quickly breaks down as the passage continues.
			0	0	
			0	1	
			0	1	
			1	1	
			0	1	
			5/11	7/11	

Get Out is a Thriller, starring Daniel Kaluuya as Chris Washington and Zailand Adams, directed by Jordan Peele and in probably his most striking work, Bradley Whitford plays Dean Armitage. In a much less memorable movie An American Crime features Catherine Keener and Whitford. Keener plays Missy Armitage slavishly. Again Peele is Producer too in one of the worst works of their career. Chris Craine in the role of Art Direction very giving a forgettable performance. Jason Blum psychological in the role of Producer. Quenell Jones is cast as Camera Operator appropriately giving a career defining performance. Bringing them up again Keener plays the part of Missy Armitage very and is not worth writing home about. Whitford plays the part of Dean Armitage proudly. And again Kaluuya really makes this movie worthwhile Get Out is a true classic fans of the genre will love this Thriller Peele as you may have already guessed by the tone of this review has delivered and again Kaluuya really makes this movie worthwhile, Get Out is a true classic, fans of the genre will love this Thriller, Peele as you may have already guessed has delivered.	nlg paragraph	10			
			0	0	repeated parts
			1	0	
			0	0	feels like a lot of clauses just pasted together. lots of words that shouldn't be there/don't make sense - 'missy armitage very and is not worth writing home about' for example. zailand adams as who?
			0	1	
			1	1	
			1	0	This breaks down the same way as before. The way you've generated how different people performed their different roles falls into a very obvious pattern. However how you return to the subject at the end is a very good touch that adds a realism, even though the body of the review feels incredibly tan
			0	1	
			1	1	
			1	1	
			1	0	
			6/10	5/10	

Directed by M. Night Shyamalan, The Happening is a Thriller, starring Mark Wahlberg as Elliot Moore and Zooey Deschanel. In a much better movie than this Stealing Cars features Wahlberg and John Leguizamo. Bringing them up again Wahlberg plays Elliot Moore never. As you may have already guessed by the tone of this review Leguizamo plays Julian poorly. In one of the worst works of their career John Rusk past plays the part of Producer. Has really embarassed theirself in this role John Bernard convincing plays Line Producer. Tak Fujimoto plays Director of Photography excellently selling the role perfectly. Doing an excellent job James Newton Howard new plays the part of Original Music Composer. (And it goes without saying) Wahlberg in the role of Elliot Moore alive.It. Again Leguizamo golden has been cast as Julian. Wahlberg gives an underwhelming performance. Shyamalan deserves recognition for this, fans of the genre might enjoy this Thriller, The Happening is ultimately a bit underwhelming Wahlberg gives an underwhelming performance, Shyamalan and again deserves recognition for this, fans of the genre might enjoy this Thriller, The Happening is ultimately a bit underwhelming.	nlg paragraph	13			
			0	0	repeated again
			0	1	
			0	0	doesn't make sense at all. 'elliot moore alive.It.' lots of clauses sewed together and none of them make sense
			0	1	
			1	1	
			0	0	I just realized I've been using the rating system wrong oops.
			0	1	
			1	1	
			1	1	
			3/9	6/9	

Starring William H. Macy as Jerry Lundegaard and Frances McDormand, Fargo is a Crime, directed by Joel Coen as a true project of passion, McDormand plays Marge Gunderson also. Steve Buscemi plays Carl Showalter perhaps. In a far less competent movie Minnesota Nice features Macy and McDormand. Doing an excellent job, Coen plays the part of Editor. John S. Lyons is Casting too. Roger Deakins in the role of Director of Photography excellently giving a career defining performance. Mary Zophres quirky is cast as Costume Design. As you may have already guessed by the tone of this review McDormand plays the part of Marge Gunderson supposedly and is not worth writing home about. Buscemi plays the part of Carl Showalter together. Coen has created a work of art Macy again gives a standout performance Fargo deserves your attention fans of the genre will love this Crime Coen has created a work of art, Macy again gives a standout performance, Fargo deserves your attention, fans of the genre will love this Crime	nlg paragraph	15			
			0	0	
			0	1	
			0	0	after you've read a couple of these the pattern becomes very obviously robotic
			0	1	
			1	0	
			0	0	
			0	1	
			0	1	
			1	1	
			2/9	5/9	
Directed by M. Night Shyamalan in the defining work of his career, The Happening is a Thriller, starring Mark Wahlberg as Elliot Moore and Zooey Deschanel	nlg sentence	3			
			1	0	bit too short for me to fully be able to tell
			1	0	zooey deschanel as what?
			0	1	
			0	1	
			1	1	

			1	0	"Is" instead of "In" trips this up again. Also in comparison to the previous example in the survey this does seem more likely to be generated purely by simplicity (not that this is any slight against you, because that kind of detail would require intensive work, just think about contextualizing exam
			1	1	
			0	1	
			1	1	
			1	0	
			1	0	
			8/11	6/11	
Pulp Fiction is a Thriller, directed by Quentin Tarantino in the defining work of his career, starring Stephen Hibbert as The Gimp and Samuel L. Jackson	nlg sentence	7			
			1	1	
			1	0	Samuel l jackson as what?
			0	1	
			0	1	
			1	1	
			1	0	Again, "in" instead of "is" trips this up.
			0	0	
			1	0	
			1	1	
			1	1	
			1	1	
			8/11	7/11	
Logan deserves your attention, Jackman really makes this movie worthwhile, fans of the genre will love this Action.	nlg sentence	11			
			1	1	
			1	0	
			1	0	action is awkwardly capitalised
			1	0	
			0	1	
			1	0	The last sentence trips me up because it reads "Action." instead of "Action movie."

			1	1	
			0	1	
			1	1	
			1	0	
			8/10	5/10	
The Happening seemed to have potential from its concept and the idea but the overall execution and writing of the film collapses. A main reason why the film is harshly criticized is because of the acting from the leads Mark Wahlberg and Zooey Deschanel. They seemed miscast and it was hard to take Wahlberg serious as a Science teacher and the chemistry between them two seemed almost fake (at least in the first 45 mins).	real paragraph	2			
			1	1	
			1	1	
			0	1	
			0	1	
			1	1	
			0	1	
			1	1	
			0	1	
			1	1	
			0	1	
			0	1	
			5/11	11/11	
Passengers is an excellent film, great cast, very enjoyable. I don't understand why people, and critics are giving it bad reviews, it deserves better than that, it is certainly a much more satisfying watch than Arrival which leaves you feeling cheated, and yet at the time I write this Arrival has an 8.4 rating, and Passengers 6.6. Chris Pratt and Jenifer Lawrence are a pleasure to watch, if you're a fan of these two, then I don't think you'll be disappointed. The story keeps you interested all the way through, good writing, good directing, great visuals, and a few laughs thrown in too. I found it very moving, and parts of the film brought tears to my eyes, but then I'm a bit of a wuss like that for sad bits in movies!	real paragraph	6			
			1	1	
			0	1	way too many commas where fullstops would have been better. jennifer is spelled wrong
			0	1	

			0	1	
			0	1	
			0	1	This review feels like it was written by someone who isn't a film critic which is interesting to play around with, and framed this way grammatical errors might not be so jarring?
			0	0	
			0	1	
			0	1	
			1	1	
			0	1	
			2/11	10/11	
Don't even bother watching this movie. It is absolutely horrible with terrible acting and even worse script. I anticipated this movie for quite a movie actually but was really disappointed. I thought Mark Wahlberg would have done a better job. It was a waste of time and space on my hard drive. The plot so ridiculous it's almost funny. How could anyone in their right mind think up something like this? I mean come on, it's not even close to being believable. Plants attacking the humans? WOW! Don't bother with this movie! I thought it could be good with the actors and the comeback of the director was I was very wrong. I thought this was a joke when i watched it, honestly.	real paragraph	9			
			1	1	
			1	1	
			1	0	
			0	1	
			1	1	
			1	0	This is probably the hardest example to distinguish. The main points that make me hesitate in my decision are "anticipated this movie for quite a movie" and "The plot so ridiculous it's almost funny." which I think was directly lifted from the previous example? Also again a missing "is" here, which
			0	1	
			1	0	
			1	1	
			0	1	

			7/10	7/10	
This movie really have a strong script, good actors and the net result is amazing.	real sentence	1			
			0	1	
			1	0	should be 'has' not 'have'
			1	0	Almost perfect except for the misuse of the 4th word
			1	0	this movie has
			1	0	
			1	0	The use of "have" instead of "has" is a dead giveaway that this is a generated response.
			0	0	Wrong tense of verb, should be "movie has a strong script". People don't really say net result much.
			1	0	Sounds foreign/broken English
			1	0	the information this sentence is trying to convey is gotten, however there are obviously gramatical errors and there is use of nonstandard language ("net result" in terms of describing a movie's attributes)
			1	0	
			0	1	
			8/11	2/11	
What started out as a brilliant plot line concept, the poor acting quickly destroyed.	real sentence	4			
			1	1	
			1	1	was quickly destroyed by poor acting' would work better
			1	0	
			0	1	
			0	1	
			0	1	
			0	0	
			1	0	
			1	0	
			1	0	sentence appeared constructed
			1	0	

			7/11	5/11	
This is bad. I mean, so bad sometimes that it's almost a comedy.	real sentence	8			
			1	1	
			0	1	
			1	0	
			1	0	
			0	1	
			0	1	
			0	0	
			1	0	
			1	1	
			1	1	
			0	1	
			6/11	7/11	
"Logan" is directed by James Mangold and stars Hugh Jackman for one final go around as Wolverine. With an R Rating secured, something the previous two Wolverine films should have had, "Logan" was bound to be fantastic. It wasn't fantastic. It was phenomenal.	real sentence	12			
			1	1	
			0	1	
			1	1	
			0	1	
			1	1	
			0	1	
			0	1	
			1	1	
			1	1	
			5/9	9/9	
Of course everything stems from a rock solid script, where the plot points are cunningly engineered, and then fleshed out in a disciplined and take no prisoners kind of way.	real sentence	14			
			1	1	
			0	1	
			1	1	

			0	1	
			1	1	
			1	1	
			1	0	
			1	1	
			1	1	
			7/9	8/9	
Written by Jordan Peele. Produced by Jordan Peele. Directed by Jordan Peele. Stars Daniel Kaluuya as Jordan Peele.	real sentence	16			
			0	1	Jordan Peele.
			1	0	
			0	1	
			1	0	
			1	1	
			1	0	
			1	0	
			0	0	
			1	1	
			6/9	4/9	

C
Preliminary Project Report

Preliminary Project Report

Thomas Palmer

February 17, 2017

Contents

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/.

D
Project Logs

Weekly Progress Logs and Meeting Notes

Thomas Palmer

March 24, 2017

1 Week ending 24/3/17

1.1 Progress Log

Wordnet solution added to program code
Change wordnet solution to pick adjective/adverbs with a degree of randomness as absolute average word can be very plain and nondescript
Starting on nlg system
Reading book on creating nlg systems,
Looking in to other nlg systems out there again for inspiration - JAPE, KNIGHT, PIGLET, STOP system

1.2 Meeting notes

Meeting moved to Monday due to issues with public transport

2 Week ending 17/3/17

2.1 Progress Log

Fixing bugs in the template system- a lot of changes had resulted in a lot of errors overlooked
Working on wordnet solution - harder than assumed
Changes to report given new context of reviews to be believable rather than particularly summarative or profound
Writing literature review and implementation sections

2.2 Meeting notes

Plot out the 9 weeks - data gathering for evaluation
Deploy the final form in roughly 3 weeks? to start getting data

3 Week ending 10/3/17

3.1 Progress Log

Adding to report - introduction
Adding to report - literature review
Wrote out test plan for systems implemented currently
Looked into wordnets, aiming to pick the most descriptive adjectives out of sentences and pick the most representative word from the wordnet to see what I get.
Add comment section to blog website
Improve method for detecting sentences

3.2 Meeting notes

How am I going to measure my output criteria
start on nlg system
continue testing
do small jobs if having trouble focusing

4 Week ending 3/3/17

4.1 Progress Log

Scraping movie synopsis off of IMDb, and summarising it using the summa library for now.
Gather a corpus of movie reviews using screen scraping from IMDb
This corpus isn't actually very good for generating review text - it's mostly negative and not really constructive, seems to follow the general adage that people don't go on the internet to sing the praises of films and mainly to slate them.
It gives a strong negative bias towards movie reviews generated because of this.

Set up a mySQL database for blog website, has users, posts and feedback gathering tables.
Set up the website, reviews are manually input through the blog at the moment but it is easy to insert them using mySQL in python

4.2 Meeting Notes

Discuss how to gather feedback
let users comment on reviews
agreeable/disagreeable reviews in order to generate traffic?
testing, minimal tests some degree of randomness
make sure i always get the same text with one seed
set up scenario to fix problem etc
test where bug doesn't happen again kind of scenario

5 Week ending 24/2/17

5.1 Progress Log

I have not been able to do very much work this week as my laptop had something spilled on it on Sunday and have mostly been waiting to get it back.
Added to report introduction, further explanation and content from preliminary report that applies.
Added to report literature review, content from my preliminary report.
More reading on natural language generation:
Got hold of Building Natural Language Generation Systems for further reading.
Exploring web scraping, trying to find a way to gather a corpora of more than one movie review easily. Currently seems easy enough to get one from a specific website but more than one is difficult and might just require a bit of brute-forcing.

5.2 Meeting Notes

The meeting was cancelled this week and rescheduled for during easter so one did not take place.

6 Week ending 10/2/17

6.1 Progress Log

Obtained an API key for themoviedb which allows lookup of movie metadata in a much nicer way than trawling text files from IMDb
Looked at textrank algorithm for summarative sentence generation. Used a library to test it out
programmed a metadata lookup for the template generation system
generate a list of sentences pertaining to actors and staff memebers with their sentiments attached

Generate a movie review based off of the metadata and extracted sentiments from reviews
Not enough is learned from the corpora to talk much more than superficially however - we cannot justify the opinions, only report them.
Reading ch 7 of the nlp with python textbook, I might want relation extraction from the corpora.
Started writing project preliminary report.

6.2 Meeting Notes

Worry about the level of readability - plausability is ok, system doesn't have to be perfect if we are just testing
-Reminder to be realistic about what i can create
having a plan and something to show to participants
send Christophe draft plan - gantt chart w/ fairly small sub-tasks, more tasks - for wednesday 4pmish probably
Think about whether or not i am experimental about this or want to examine the effect on a group of users (methodology)
-entology of concepts related to film in order to produce realistic text

-preliminary report follows the general structure of the fyp report

-Create a gantt chart ofplanned further work

7 Week ending 3/2/17

7.1 Progress Log

Started programming with the twitter API using Twython to post tweets and read a timeline for a corpus
Text processing for twitter timeline, trying to get markov chain generated for corpus of tweets, experiencing strange bug
Read several articles on keyword extraction, decided using RAKE library for python and SKLearn for sentiment analysis just to start
Used RAKE to extract keyphrases
Installed NLTK because it should be more straightforward to perform sentiment analysis with
Downloaded movie review corpora of pos/neg film reviews
NLTK import error name overridden (Solved)
Used NLTK Vader to classify sentiment of movie reviews

Used markov chains to construct movie review text
Markov chain automated twitter posting is set up (no replies or searching yet)
Twitter search + chain from search

8 Meeting notes

Move on to template system
Markov chain for template generation

9 Week ending 27/1/17

9.1 Progress Log

Read into neural networks for natural language generation. It was interesting and could be worth exploring although articles stipulate that recurrant neural networks (the kind used for modelling sequential data and thus generating text) are difficult to train effectively. This could be too time costly for the space of my project. Other draw-backs are that the neural network isn't going to form truely informed opinion and might not even be particularly coherent.

Read into sentiment mining, markov models and information extraction (ch 6, 9 and 21) https://web.stanford.edu/ jurafsky/slp3/

Found resources for sentiment analysis - movie reviews and sentiment keywords for films

Researched methodologies for natural language key phrase mining - unsupervised vs supervised

Looked in to related work - there are lots of article summarizers out there using keyphrase extraction

Added to the context/motivations section of the introduction
Wrote cursory list of goals and aims for the introduction
Wrote about related works in natural language generation / text generation

9.2 Meeting notes

#icanhazpdf
writing to authors to obtain papers isn't a bad idea

browse the library
senate house library for printed journals
footnotes vs references
internet archive link for online resources
google spreadsheet the plan

to do:
have a look in the library, see if I can find any movie review discourse in the media/language section update plan a bit, have a look at google
do some programming

10 Week ending 20/1/17

10.1 Progress Log

Set up github repository for final year project deliverables
Implementation of Markov chain text generator - the first and most simple NLG tool I will be implementing, forms a table of n-word prefixes mapped to the suffixes that occur in a given corpus of text. Then choosing an arbitrary starting point they follow the chain with some stochastic probability in the case where maps contain more than one suffix value and continue the chain until a stopping point of n words, or the end of the chain is reached.

10.2 Meeting notes

(discussion of project structure)
abstract

context / literature review
- review of text generation and movie reviews

- problem statement , high level aims, individual objectives of the project
fairly broad aims, objectives are measurable

- design / implementation
talk about design of natural language more generally or of my systems

- testing and evaluation chapter
testing of individual components

have i mangaed to drive traffic in any way for twitter bot

turing-like test

- conclusions and further work

- bibliography

context + lit review, design + imp, testing and evaluation should be roughly the same size

can collect things that i will be writing about

bibliography generation tools - endnote (?), latex

show next week the context + lit review skeletal stuff
categorise things i want to write about into text generation, evaluation of text, etc

investigate neural networks for text generation, not necessarily to persue

come back with something that works, some context, problem statement and some objectives

gant chart / some description of milestones and the bits of work that need to be done for those milestones

10 minute summaries of work at end of each session

E
Program Code

E.0.1 Python

E.0.2 PHP

E.0.3 mySQL

Bibliography

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

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

[3] 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.

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

[5] Eric Brill. *A Simple Rule-Based Part of Speech Tagger*. http://luthuli.cs.uiuc.edu/~daf/courses/Signals%20AI/Papers/HMMs/h92-1022.pdf.

[6] Christopher D. Manning Kristina Toutanova. *Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger*. https://nlp.stanford.edu/~manning/papers/emnlp2000.pdf.

[7] C Fellbaum. *Encyclopedia of Language & Linguistics, Second Edition*. http://iaoa.org/isc2012/docs/encycloped.article.pdf.

[8] Eric Gilbert C.J. Hutto. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf.

[9] Robert Dale Ehud Reiter. *Building Natural Language Generation Systems*.

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

[11] Devika Subramanian. *The Curious Case of Mark V. Shaney*. https://www.cs.rice.edu/~devika/comp140/Shaney.pdf/.

[12] *Archie*. https://www.archie.co/.

[13] *@robodonaldtrump*. https://twitter.com/robodonaldtrump.

[14] Andrew C. Bulhak. *On the Simulation of Postmodernism and Mental Debility using Recursive Transition Networks*. http://www.elsewhere.org/journal/wp-content/uploads/2005/11/tr-cs96-264.pdf.

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

[16] Helen Pain Ruli Manurung Graeme Ritchie. *The Construction of a Pun Generator for Language Skills Development*. http://users.sussex.ac.uk/~christ/crs/gc/STANDUP_AAI_revised.pdf.

[17] Roma Robertson Ehud Reiter. *The Architecture of the STOP System*. http://homepages.abdn.ac.uk/e.reiter/pages/papers/rags99.pdf.