Movie Review Generation

Thomas Palmer

Goldsmiths College

University of London

A thesis submitted for the degree of

B. Computer Science

March 31, 2017

Acknowledgements

Personal

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

Institutional

Abstract

An implementation and evaluation of a system designed to generate a review for a film given a corpora of movie review text discussing that movie.

Contents

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.

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 Contribution

1.4 Thesis Structure

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. 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 precedes them in the text. Then, choosing an arbitrary starting point the next word is chosen randomly based on the frequency of how often the n-preceding words are found in the text. 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.

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. 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.

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. 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.

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.

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. 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 another being using supervised and unsupervised machine learning.

VADER is a rule-based model for the sentiment analysis of text. 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 NLG system.

2.1.6 Building NLG Systems

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. 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 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. The posts would often fool people into believing the comments were written by a real person. It is an early example of people using machines to generate prose in order to see how people react.

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 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.

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.

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". 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 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.

2.2.5 NLG Systems

There are many examples of NLG systems which exist for a multitude of different purposes. STANDUP 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 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.

3 Design

3.1 Introduction

3.2 Aims and Objectives

3.2.1 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.

3.2.2 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.

3.3 Movie Review Generation

3.3.1 Markov Chain for Text Generation

3.3.2 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 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, such as cast crew and 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 API to match sentences to these topics.

The final step is then filling in the reviews templates until completion. Builds an introduction based on a template, a plot summary after that, a body of review text, and then a closing statement based on the template. Templates are filled with words determined by sentiment, and adjectives and adverbs mined from part of speech tagging sentences regarding the topic discussed in the template sentence.

3.3.3 NLG System

3.4 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.

There are a number of behaviours programmed for the bot to gather attention and direct users towards the website. 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 next behaviour is replying to posts which have been identified to be about the movie in question with comments generated from a markov chain of other tweets about the movie.

3.5 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.

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.

4
Implementation

4.1 Introduction

4.2 Movie Review Generation

4.3 Blog Website

4.4 Twitter Bot

5 Testing

5.1 Introduction

Testing each part of this system involves checking primarily that outputs are what they should be, but also checking that the data scraped from the internet is handled correctly in edge-cases and that errors are handled correctly and not ignored or allowed to halt the program code. Because of this, I have to create my own example data that looks like what is gathered from web scraping and API calls which mirror edge-cases.

5.2 Movie Review Generation

5.2.1 Testing of Review Generation

This involved a lot of testing of individual modules and functions to make sure that they handle all the data passed in to them. As the completed system merely takes an input which is the name of a film and several on/off parameters, there is not much in the user space that can change with regards to testing.

5.2.2 Changes Made

5.3 Blog Website

5.3.1 Testing of Blog Website

Most of this testing work is making sure that the SQL CRUD functions work, and making sure that the website displays the information that is intended, and checking that my information gathering here is accurate.

5.3.2 Changes Made

5.4 Twitter Bot

5.4.1 Testing of Twitter Bot

I have decided to test the Twitter Bot with a series of items which I have saved to use for testing, as well as some live tests taken from Twitter at runtime.

5.4.2 Changes Made

6
Evaluation

6.1 Methodology

A well documented issue with Natural Language Generation 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.

Most of my evaluation is done on the review of comments and replies gathered from users of the system, 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.

6.2 Engagement With Twitter Bot

6.3 Comments and Interaction with Blog Website

6.4 Turing Test Scenario

7
Conclusion and Future Work

7.1 Conclusion

7.2 Future 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.

Appendices