Feature-Based Opinion Mining of Crowd-Sourced Video Game Reviews

Robert Jeffrey (u6352710)

A report submitted for the course
COMP3770 Individual Research Project
Supervised by Dr. Penny Kyburz
The Australian National University

May 2019

Except where otherwise indicated, this report is my own original work.

Robert Jeffrey
31 May 2019

Acknowledgments

I would like to thank my supervisor Dr Penny Kyburz for her support throughout my project, and for keeping me on the right path.

Abstract

In this report, we built a baseline for the task of review summarisation within the domain of video game reviews. We developed a script to download reviews from the Steam video game store, and implemented an existing algorithm to tests its efficacy. We found that the algorithm worked to a reasonable extent, but suffered from irrelevant or noisy features. We suggest multiple research ideas for future work to expand upon the ideas we lay out.

Contents

List of Tables

Chapter 1

Introduction

With the explosion of e-commerce over the past two decades, we have seen product reviews transform from being the domain of dedicated journalists to that of the customers. With more sites providing easy-to-use review features, any individual is now able to share their opinions of a product to assist others in making their choices. While this wealth of data provides a great opportunity in allowing individuals to make informed decisions regarding their consumption, it also presents a challenge: how can an individual draw meaningful conclusions when presented with unfeasibly large amounts of text to read?

The answer to this question is straight-forward, though certainly easier said than done: we need to summarise the content of the reviews into a human-readable format. The task of generating summaries of review data has multiple names, but is usually called "opinion-mining" [Eirinaki et al., 2012] or "feature-based summarisation" [Hu and Liu, 2004]. It is considered to be distinct from other forms of automatic summarisation as the summaries are focusing on different information - we specifically are interested in presenting the opinions expressed in the reviews rather than presenting a general summary.

One widely-used alternative to this is to have reviewers give numerical reviews, which can then be simply summarised by averaging the results. However, this can be overly simplistic - if a product has both positives and negatives, a numerical review cannot present that information to a reader. In the case of a review which is predicated on a circumstance of the reviewer, the opinion may not be as useful for a reader who does not share the same situation. Consider, for example, the review "if you have young children, DO NOT BUY THIS! 1/5". In this example, if a reader does not have young children they may not find the numerical review helpful. However, if the summarised text was presented as well, the reader can make an informed decision.

The task of feature-based summarisation has been studied repeatedly in recent years, (discussed in Section 2.2), with research drawing reviews from large review sites. This includes the e-commerce platform Amazon [Hu and Liu, 2004; Bafna and Toshniwal, 2013], product review webiste CNET [Hu and Liu, 2004], travel review website TripAdvisor [Nyaung and Thein, 2015], and business review site Yelp [Fosset, 2017]. These sites provide excellent testing grounds for consumer products and services, this results in a lack of research in other domains.

One of these such domains is the area of video game reviews. With the video game industry reaching $91 billion in revenue in 2016, and being expected to grow at a rate of 6.3% through 2020 [Lin et al., 2019], assisting customers in making purchasing decisions is a lucrative research avenue. In addition, helping the developers of video games understand the content of their product's reviews could help future products address the feedback of their customers. Video games, in comparison to consumer goods & services, have a few traits which make them unique to deal with. One large factor is the increased subjectivity of a consumer's opinion of a video game. While customers will have differing opinions about consumer products (such as electronics), these products often will often have objective features that make reviewing less subjective. In comparison, video games are a from of entertainment, and thus have much greater element of personal taste in a customer's opinion of the product. This is problematic for existing summarisation algorithms [Hu and Liu, 2004], but creates a greater benefit for summarising reviews as customers are able to use the summarised reviews to choose products which appeal to their personal tastes. Additionally, video game reviews are often less formal, and will frequently contain jokes or humour. This is challenging for many natural language techniques to deal with, and adds an extra element of difficulty to this domain.

While research has been conducted in analysing video game reviews [Lin et al., 2019; Eberhard et al., 2018; Zuo, 2018], there is limited research into summarising these reviews. As a result of this lack of research and the benefits given above, we believe there to be a niche for the summarisation of video game reviews.

1.1 Project Scope

The aim of this project is to provide a starting point for research regarding automatic summarisation of video game reviews. We therefore aimed to do four things: first, justify the novelty of summarising video game reviews in comparison to other platforms by discussing the unique aspects of the domain. Second, provide tools and recommendations for building review corpuses. Three, analyse the performance of existing tools & algorithms within this domain. Fourth, provide suggestions for future work into improving the performance of feature-based summarisation tools upon video game reviews.

This project was intentionally scoped to be exploratory in nature. As our goal was primarily to build a framework from which to do further research, we did not aim to produce a useful tool for automatic summarisation. Instead, we aimed to investigate the usefulness of the current state of the art algorithms, and to identify where the current techniques were inadequate for this domain.

1.2 Report Outline

This report is structured according to our goals. We start by providing background, and discussing the unique aspects of video game reviews in Chapter 2. We next dis-

cuss our code for collecting reviews from the Steam platform in Chapter 3. Chapter 4 discusses the algorithm we used for our benchmark, and Chapter 5 discusses our results. Finally, we conclude our project and lay out further work in Chapter 6.

Chapter 2

Background and Related Work

2.1 Background

In automatic summarisation, our core goal is to condense a large corpus of text down into an easily human-readable summary. Often this summary takes the form of a paragraph of text, but in the case of opinion mining of reviews, we are able to be more structured by giving a detailed breakdown of peoples opinions on the product. To do this, we introduce the idea of "feature summaries", first described in Hu and Liu [2004]. Feature summaries are composed of two main elements. First, commonly discussed aspects of the product, called "features", are extracted from the review dataset. Next, each feature is given a "sentiment" composed of the percentage of reviews that discuss the feature which display positive sentiment. To illustrate this concept, we provide an example (hand-made) feature summary in presented in Table 2.1. Note that the final feature is specific to game being reviewed; this is a result of the features being generated from the reviews themselves.

As discussed in Chapter 1, we focus on the area of video game reviews. In order to generate a corpus, we collect reviews from the Steam platform. Developed by Valve Corporation, Steam is a distribution platform for video games and related software. The platform has a strong community, including a large participation in the reviewing of games. Reviews on the site are composed of two main elements: a binary "recommended?" variable, and the review text. Other users are able to rate reviews as either "helpful" or "unhelpful", as well as to tag reviews as "funny". It is not uncommon for applications to have thousands or tens of thousands of reviews,

Table 2.1: Example feature summary.

Feature	Percent Positive	Positive Reviews	Negative Reviews
Graphics	93.7%	373	25
Story	50.7%	246	239
Performance	76.2%	32	10
Beachtown	95.2%	40	2

with some having over one million reviews [Steam Store]. This massive amount of data, in conjunction with its powerful API, makes Steam perfect for our purposes.

2.2 Related work

We have seen several different feature summarisation algorithms be proposed in the past decade. The basis for our work, the "feature summary", is laid out first in Hu and Liu [2004]. In the paper, the authors lay out a baseline algorithm for feature mining using association mining [Agrawal et al., 1994] to detect features and a word similarity technique for classifying sentiment. Abulaish et al. [2009] introduces the idea of using adjectives as indicators of features, with the idea that an important feature of a product will have a strong opinion related to it, which necessitates the use of adjectives. The paper thus parses the dependency trees of the component reviews to find the nouns each adjective refers to, and uses SentiWordNet [Esuli and Sebastiani, 2006] to classify the sentiment. Eirinaki et al. [2012] takes a similar approach, but simplifies the algorithm by heuristically assuming each adjective refers to the nearest noun to remove the requirement to fully parse the dependency tree. Recently, Nyaung and Thein [2015] takes a more statistical approach by using maximum entropy modelling to find the most likely opinion word-feature pairs while also using SentiWordNet for sentiment classification.

We have also seen multiple papers regarding analysis of the steam platform. Zuo [2018] tries to predict the overall sentiment of Steam reviews based on the review text, and Eberhard et al. [2018] attempts to classify reviews as "helpful" or "unhelpful", using the user-assigned "helpfulness" score as a ground truth.

Chapter 3

Data Collection

In order to assess the performance of any tools for summarising reviews, we must first create a dataset of reviews on which to test. In this chapter, we discuss the decisions made in choosing which games to collect reviews from (Section 3.1), and detail the process by which we retrieved the reviews (Section 3.2).

3.1 Game Selection

The first step of building a review corpus was to select which games we would draw our reviews from. We had three main criteria: first, the games should be diverse enough to allow a fair assessment of any algorithm's performance on different types of reviews. Next, we wanted the games to have a large enough number of reviews for our summarisation algorithms to work properly. Finally, we wanted the games to be familiar to the authors so that we could easily assess our results qualitatively.

To best meet these criterion, we decided to select the five games with the highest player count at the time of our experiment, as measured by Steam Charts [2019]. These games, alongside their player counts at the time of retrieval (2019/04/04), are listed in Table 3.1.

Table 3.1: Games selected for feature summarisation.

AppID	Name	Peak Player Count	Review Count
570	Dota 2	1,033,925	6,472
578080	PLAYERUNKNOWN'S BATTLEGROUNDS	931,407	175,151
730	Counter-Strike: Global Offensive	680,071	150,173
359550	Rainbow Six: Siege	134,776	88,848
230410	Warframe	115,102	1,254

Table 3.2: Important & significant review metadata.

Column	Description
review	Review text
voted_up	True if the review made a positive recommendation.
votes_up	Number of users who marked the review as "helpful".
votes_funny	Number of users who marked the review as "funny".
comment_count	Number of comments on the review.

3.2 Data Collection

To collect our data, we use the official Steam API. This provides us the review text alongside metadata about the reviews. A selection of significant/useful metadata values are provided in Table 3.2, with the full list available at the Steamworks documentation [Steam Games]. Through these Steam API calls, we were able to collect approximately 150 reviews per second.

One unusual behaviour is that the amount of reviews accessible through the API seems to be limited for some games. This is most notable for DOTA 2. While the store page lists 391,203 English reviews [Steam Store], we were only able to collect 6,472 before receiving an error normally given when there are no more reviews to collect. The reason behind this is currently unknown.

Chapter 4

Algorithm and Implementation

Same as the last chapter, give the motivation and the high-level picture of this chapter to readers, and introduce the sections in this chapter.

Our summarisation technique is based upon the algorithm described in Eirinaki et al. [2012], with two major modifications that assist in our use. Below we describe conceptually how the algorithm works, with pseudocode being provided in the source paper.

4.1 Summarisation Algorithm

The algorithm can be decomposed into two tasks - feature detection, and sentiment scoring. The first step in creating feature summaries is identifying words and phrases that may potentially be features. The technique we use is called the High Adjective Count (HAC) algorithm. The core idea of the HAC algorithm is that the importance of a feature is related to the amount of opinions expressed about it. To approximate this, the HAC algorithm assumes that each adjective (which we believe represents an opinion) refers to its nearest noun, and then counts how many adjectives refer to each noun in the dataset. From this, we can conclude that nouns with a high adjective count are most likely to be important features in the text.

Our next task is sentiment scoring. This means that we need to detect how the author of a review talks about a feature, i.e. if they treat it in a positive or negative light. For example, the sentence "The food was really bad." talks about the feature "food" with negative sentiment, whereas the sentence "I loved the food!" discusses it positively. Eirinaki et al. [2012] looks at the adjectives used to describe each noun, and matches them against a hand-labelled list of "sentiment scores" representing the polarity of the noun. This, alongside the presence of negation words (e.g. "not") in the two words before the adjective, are used to determine if the noun is being discussed positively or negatively. While this theoretically works well given a good enough set of scores, building such a list is taxing, and can be biased. To avoid this, we instead decided to use the VADER algorithm [Hutto and Gilbert, 2014], as implemented in the NLTK library [Bird et al., 2009]. This algorithm is designed for sentiment analysis of social media posts, and was found to generalise across contexts better than other state-of-the-art algorithms. These two properties combined make

the algorithm favourable to use within our research context.

Our sentiment scoring algorithm therefore works as follows. Similarly to the HAC algorithm, we find each adjective and the noun associated with it, i.e. the nearest noun. We then create a four word "context" composed of the two words before the adjective (if they exist), the adjective itself, and the nearest noun. This context is then scored using the VADER algorithm as either positive (compound score > 0), or negative (compound score < 0), and the counter for the positive or negative uses respectively of the nearest noun is incremented.

We finally determine the features we use by selecting the features with the highest HAC scores. We can then present these features, alongside the scores we find using our sentiment analysers.

We initially planned to use a preexisting implementation upon this data to same time & allow ourselves to work on improvements. However, we quickly found that the current implementations were either strongly machine-learning based and thus not able to be re-purposed easily to a new domain, or were not well enough written to be expanded in the way we wanted. As a result, we instead decided to re-implement the algorithm from scratch.

4.2 Heuristics

In order to reduce the amount of noise in our results, we implemented a small number of heuristics for deleting invalid or "bad" reviews. The most successful of these was simply defining a list of stopword features; that is, potential features that we should skip. We did this to remove terms that were often discussed in the same way as features, but gave no meaning to the reader. The most significant of these was the word "game", which would consistently appear as a top feature. This was due to a large amount of reviews being composed of a variant of "this is a good game". While these reviews may be helpful for a human, the reviews already have "recommend" and "not recommend" tags, which means these generalist features are not useful to analyse when looking for features.

4.3 Summary

Same as the last chapter, summarize what you discussed in this chapter and be a bridge to next chapter.

Chapter 5

Results

In order to evaluate our algorithm, we generated feature summaries of the five games we selected, and qualitatively analysed the top 15 features of each summary. We qualitatively marked each feature as "relevant" or "irrelevant", and computed the percentage of reviews that were "relevant". Note that relevance encompassed both relevance to the game itself, and the property of being specific enough to provide relevant information to a reader. For example, the title of each game was not marked as relevant. These results are presented in Table 5.1

Our results suggest that the existing algorithms are reasonably effective but have faults. The majority of features were useful, both having more generic features such as "gameplay" and "graphics" as well as features specific to the games themselves, such as "weapons" for Warframe. However, there was still a reasonable amount of non-useful terms, both with non-specifc features such as titles genres of the game, as well as difficult to interperate features such as "life" or "operators".

One major flaw is that the sentiment analyser we use is not trained on our dataset, and thus does not have full knowledge of the context. A good example of this issue is the difference in the phrases "too many cheaters" and "too many hackers". While these terms have similar meanings within the context of video game reviews (both referring to players who cheat), the former sentence is rated as a negative sentiment, while the latter is rated as neutral. This resulted in the overall sentiment ranking of the feature "hacker" to be positive in our summary of the game "Counter Strike: Global Offensive", as comments about the prevalence of hackers were not counted as negative.

Table 5.1: Percentage of Relevant Features for Each Game.

App_ID	Name	Percent Relevant Features
570	Dota 2	53.3%
578080	PLAYERUNKNOWN'S BATTLEGROUNDS	60%
730	Counter-Strike: Global Offensive	53.3%
359550	Rainbow Six: Siege	66.7%
230410	Warframe	66.7%

Another issue is the susceptibility of the algorithm to spam due to the occurrence-based HAC algorithm. This can be seen most notably in the "regionlockchina" feature for PLAYERUNKNOWN'S BATTLEGROUNDS. This feature has only 2 positive usages and 5 negative usages, but has a total HAC score of 10254. This appears to be a result of a number of reviews which repeated the phrase "#regionlockchina". While in this case the term is import, and appears to be commonly used, one could imagine a scenario where a small subset of reviewers repeating a phrase composed of an adjective and a noun (e.g. "stupid story") could disproportionately change how the feature is scored. One solution to this would be to limit the number of times a feature can be referred review, or to count the score of a feature by the amount of reviews it is positively or negatively discussed, in rather than the number of occurrences total with that sentiment.

Chapter 6

Conclusion

Our main conclusions are as follows. Firstly, we find that the current state of the art algorithms work to a reasonable extent upon video game reviews, especially once overly-common features are filtered out. However, this is not without concessions. We specifically find that the sentiment analysis component is lacking when analysing contextual ideas such as "hacking". We also find that a large amount of overly general features, such as "money" or "people" are returned, which are hard to analyse as humans. As such, we conclude that the current algorithms are workable, but there is a large amount of improvement possible.

6.1 Future Work

As our project aimed to produce a starting point for future analysis, we propose a variety of research areas that would build upon these ideas presented here. Firstly, the analysis presented within could be expanded to include less popular games, and to determine the minimum number of reviews needed to usefully summarise the data. In terms of the algorithm used, there is much room for improvement. As discussed, the sentiment analyser used lacked contextual knowledge of certain words such as "hacker". One improvement would be to develop or retrain a sentiment analyser to use knowledge about video game reviews. Another improvement would be to use chunking techniques to identify multi-word features such as "frame rate".

One issue with analysing the results of these algorithms is the lack of quantitative data on their success. To do this, we would need to manually classify a large amount of reviews regarding their features. While we did not perform this in our project due to the large amount of human classification required, this would be a good basis for future research.

Chapter 7

Appendices

7.1 Appendix 1: Project Description

The project will involve developing and applying text analysis and/or topic mining algorithms to video game reviews. The student will research existing approaches to text analysis of video game reviews, and develop a program to analyse web-based game reviews. This program will aim to build systems to assist users in making quicker & more accurate decisions regarding the quality of games.

The project will involve reviewing literature, identifying similar NLP models in other domains (e.g. Amazon reviews), developing a prototype algorithm based off previous attempts, evaluating the accuracy of the summarised reviews, and reporting on the results.

7.2 Appendix 2: Independent Study Contract

The contract for this project is presented at the end of this report.

7.3 Appendix 3: Description of Artefacts Produced

The project is composed of two main folders of code.

download_reviews contains our script for downloading Steam data. This code is modification/rework of the download code provided by Zuo [2018]. Links to the original code are provided within the notebook for comparison.

feature_mining contains code relating to the summarisation of the downloaded reviews. The user interface is presented in the main.ipynb file, with preprocess.py, report_features.py, and summarise.py providing the functionality. This work is original, with minor reference to the code presented in github.com/esh-b/Feature-based-opinion-mining.

7.4 Appendix 4: ReadMe

Installation should be performed using Anaconda. Once Anaconda is running, create the environment:

conda env create -f environment.yml

Next, activate the environment:

conda activate COMP3770-Steam-Feature-Reviews

Finally, open jupyter notebook:

jupyter notebook

Bibliography

Abulaish, M.; Doja, M. N.; Ahmad, T.; et al., 2009. Feature and opinion mining for customer review summarization. In *International Conference on Pattern Recognition and Machine Intelligence*, 219–224. Springer. (cited on page 6)

Agrawal, R.; Srikant, R.; et al., 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, 487–499. (cited on page 6)

Bafna, K. and Toshniwal, D., 2013. Feature based summarization of customersâĂŹ reviews of online products. *Procedia Computer Science*, 22 (2013), 142–151. (cited on page 1)

Bird, S.; Klein, E.; and Loper, E., 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.". (cited on page 9)

Eberhard, L.; Kasper, P.; Koncar, P.; and Gütl, C., 2018. Investigating helpfulness of video game reviews on the steam platform. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 43–50. IEEE. (cited on pages 2 and 6)

Eirinaki, M.; Pisal, S.; and Singh, J., 2012. Feature-based opinion mining and ranking. *Journal of Computer and System Sciences*, 78, 4 (2012), 1175–1184. (cited on pages 1, 6, and 9)

Esuli, A. and Sebastiani, F., 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, vol. 6, 417–422. Citeseer. (cited on page 6)

Fosset, J., 2017. Yelp Summarization Miner (YUMM). https://github.com/Fossj117/opinion-mining. (cited on page 1)

Hu, M. and Liu, B., 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. ACM. (cited on pages 1, 2, 5, and 6)

Hutto, C. J. and Gilbert, E., 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*. (cited on page 9)

Lin, D.; Bezemer, C.-P.; Zou, Y.; and Hassan, A. E., 2019. An empirical study of game reviews on the steam platform. *Empirical Software Engineering*, 24, 1 (2019), 170–207. (cited on page 2)

Nyaung, D. E. and Thein, T. L. L., 2015. Feature-based summarizing and ranking from customer reviews. *World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng*, 9, 3 (2015), 734–739. (cited on pages 1 and 6)

Steam Charts, 2019. https://steamcharts.com/. Online; accessed 2019-04-04. (cited on page 7)

Steam Games. User Reviews - Get List. https://partner.steamgames.com/doc/store/getreviews. (cited on page 8)

Steam Store. Dota 2 on Steam. https://store.steampowered.com/app/570/Dota_2/. (cited on pages 6 and 8)

Zuo, Z., 2018. Sentiment analysis of steam review datasets using naive bayes and decision tree classifier. (2018). (cited on pages 2, 6, and 15)

Australian National University

INDEPENDENT STUDY CONTRACT SPECIAL TOPICS

Note: Enrolment is subject to approval by the course convenor

SECTION A (Students and Supervisors)

UniID: u6352710

SURNAME: Jeffrey FIRST NAMES: Robert

TOPIC SUPERVISOR (*may be external*): Dr Penny Kyburz

FORMAL SUPERVISOR (*if different, must be an RSSCS academic*): Dr Penny Kyburz + Prof Tom Gedeon

COURSE CODE, TITLE AND UNITS: COMP3770, Individual Research Project, 6 units.

SEMESTER ☒ S1 ☐ S2 YEAR: 2019

TOPIC TITLE:

Wisdom of the Crowd: Automatic Summarisation & Sentiment Analysis of Online Crowd-Sourced Video Game Reviews

LEARNING OBJECTIVES:

By the end of the project, the student should be able to:

- **Apply natural language processing techniques to short, plentiful text segments;**
- **Build systems capable of aiding human decisions;**
- **Reapply current state-of-the-art techniques to similar but different contexts;**
- **Perform in-depth literature reviews & usefully summarise papers;**
- **Communicate research results effectively.**

DESCRIPTION:

This project will involve developing and applying text analysis and/or topic mining algorithms to video game reviews. The student will research existing approaches to text analysis of video game reviews, and develop a program to analyse web-based game reviews. This program will aim to build systems to assist users in making quicker & more accurate decisions regarding the quality of games.

The project will involve reviewing literature, identifying similar NLP models in other review domains (e.g. Amazon reviews), developing a prototype algorithm based off previous attempts, evaluating the accuracy of the summarised reviews and reporting on the results.

Australian National University

ASSESSMENT (evaluated by the Topic Supervisor, unless stated otherwise here)

Assessed project components:	% of mark	Due date	Evaluated by
Presentations	10%	Late week 11 or early week 12 of the S1 2019.	Dr Weifa Liang
Research Report	60%	5pm 31/5/19	Dr Hanna Suominen
Software	30%	5pm 31/5/19	Dr Penny Kyburz

MEETING DATES (IF KNOWN):

Biweekly on Tuesdays, 4:30pm. First meeting on 26/2.

STUDENT DECLARATION: I agree to fulfil the above defined contract:

RUIST ..... 13/3/19 .....

Signature Date

SECTION B (Supervisor):

I am willing to supervise and support this proposal. I have checked the student's academic record and believe the student can fulfil this contract. If I have nominated an examiner above, I have obtained their consent (via signature below or attached email)

..... T.D. Gedeon 13/3/19 .....

Signature Date

Examiner:
Name: Hanna Suominen Signature .....

REQUIRED DEPARTMENT RESOURCES:

SECTION C (Course convenor approval)

....

Signature Date