TAL Project Report
Review classification

Tristan HERMANT Shankar SIVAGNA Bryan VIGEE

May 6th 2017

Contents

Chapter 1

Introduction

1.1 Objectives of the project

The purpose of this project was to gather data on how much Steam community players appreciated a specific game thanks to their reviews. According to their content and other criteria, a global appreciation can stand out from those reviews.

The data extraction will be organised by a wide range of rules which enable to parse all the comments and get the indications on the reviewer's feeling. The technical details from each review should pertain to Video Games and Software/Hardware areas.

1.2 Motivation

As Video Game players, it is interesting to get feedback from other people about how they think when playing a game. Furthermore, it might be possible to obtain the general rating players tend to evaluate for a game and on which arguments they lean on to do so. This general rating can be different according to Gamer communities. Thus, it might be possible to distinguish the criteria that are believed to be important for them in order to make a enjoyable game on technical and artistic characteristics.

Moreover, extracting data from those evaluations could be useful for Video Game studios as it provides clear directions for developers to improve their future games. Indeed, sorting information on the different aspects of the game narrows down the key issues to deal with. Thus, developers can come up with new development strategies promptly.

Eventually, it would identify for each Gamer communities their likings as well as their preferences for each Video Game genre. Then, a system could be set so that it gives recommendations or suggestions to gamers that would match their likings. Or perhaps it would provide comprehensive briefings on a game, since most reviews have been written with a subjective point of view.

Chapter 2

Code Organisation

2.1 Algorithms

2.1.1 Main ideas

The program reads different reviews files from the folder. It extracts all the necessary information about the reviewer before parsing his/her text. This prevents from giving too much credits for a review if it is not relevant to take it account. Once the review was parsedn normalised and tokenised, all terms defining the player's feeling (positive or negative) are checked. Adding them up gives a value representing the reviewer's overall feeling.

2.1.2 List of files

Here is a description of each file and with a short description of their content:

- **parseReview :**
  This file gathers functions in order to parse a review and extract a '"value"' which gives an idea whether the reviewer enjoyed playing the game.

  A review is a list (of strings) containing the name of the game reviewed, the status of the review (this help us know if this review is reliable or not), the number of reviews written by the player, the number of games played and the review itself.

- **connotedDictionary :**
  It gathers functions that enable to create a dictionary. It contains the ratio of each term found in all the reviews for a game (positive ones - negative ones).

- **readInFileFunctions and writeInFileFunctions :**
  Utility functions that enable to read and write in files.

- **treatments :**
  Utility functions from the previous tutorials in class which enable to parse sentences and tokenise them. The function **getSubSent** parse a sentence into "'sub sentences"' according to the number of its clauses.

- **Negative.txt, Positive.txt, Conjonction.text, auxiliary_pos.txt :**
  Negative.txt and Positive.txt gather together respectively derogatory and positive terms. They have an essential part in evaluating the review. Conjonction.txt has coordinating conjunctions which allow to parse a complex sentence. Eventually, auxiliary_pos.txt contains most of the modals.

2.2 Dispatching the tasks

Here is the contribution of the three members on this project :

Tristan : Conception of reviews and determining statistics, wrote parseReview, connotatedDictionary.
Bryan : Made most of the conception for sentence-parsing and review analysis and wrote their functions, also wrote in parseReview, included positive, negative and modal terms.
Shankar : conception of review analysis, included reviews and some terms in text files.
Everyone took part in writing this report.

2.3 Analysis algorithms

The analysis process starts with the review's separation in sentences. Then the sentences are split in sub-sentences with the conjunction, coordinating or subordinating. This split done, the two different analysis begin.

- **The token analysis.**
  The first analysis is a token analysis. Each token is compared to a list of words, positive, negative and auxiliaries contained in 3 different dictionaries. If the token is a positive word, the sub-sentence value is increased by one. If the token is a negative word, the sub-sentence value is decreased by one. If the word is an auxiliary, the value of the sentence is not changed, but if the word is a negative form of an auxiliary, the sub-sentence value becomes the same time -1. The value of each sub-sentence is summed with the other to create the value of the review. if this value is more than 0, the review is considered positive, if it is less than 0 it is considered a negative review. If the value is 0, the review is considered a neutral review.

  This analysis can be wrong because there a numerous neutral words, and sometimes the first word of a sentence is a word used to determine if a auxiliary is at a negative form or not, the sub-sentence value is then 0 and unchanged because 0*-1 is still 0. That is why a second review analysis was implemented.

- **The sub-sentence analysis.**
  This analysis checks the presence of each word, positive or negative, in the sub sentences, and only after all these words are found, it checks the auxiliaries to determine if it is a positive or a negative sub-sentence. Finding a positive or negative word has the same effect in both analysis methods, the same process is used for the auxiliaries research.

- **Statistics.**
  When a reviewed was parsed with all its terms found, the algorithm assigns for each one of them the number of occurrence and a ratio which tells if the term was used mostly whether in a positive or in a negative way.

Chapter 3

Overview of the project

3.1 Difficulties encountered

Our project only used reviews which are mainly positive. This is the reason why our analysis cannot be reliable enough to determine the overall information about a game. Moreover, some of the reviewers' sentences are seen as negative despite they are not, thus the result does not describe properly what the reviewer wrote (if though their review is well structured).

3.2 Suggestions and enhancements

First, it could have possible to implement a classification according to specific/technical details that would have made our analysis algorithms more efficient and interesting. It was not done because the analysis cannot determine properly the right function of a word in a sentence. Thus, it was difficult to associate a technical detail with a annotated term, not only because they are presented in the same sentence. This is why the implementation of syntax grammar may improve our analysis algorithm and make it efficient to sort the feedback about the different aspects of the game.

In addition, recognising sarcasm or irony might have been a great improvement as our algorithm could cover a variety of reviews and it would return trustworthy analysis. Moreover, our text files were all handed written, which do not gather enough terms to handle all the qualifying adjectives. Thus the program misses important information. Using a huge amount of data for qualifying adjective and terms may reveal the actual performance of the algorithm (and at the same make the dictionary trustworthy). And increasing the number of reviews would also be efficient since the occurrences would have had a "'neutral"' meaning (it is the sum of the positive terms used minus the negative ones). Then, the other parameters such as the number of reviews written or the number of hours played would have a more important part in order to distinguish the best reviews.

Finally, with these improvements, it can be possible to set up a trust scale for each review and select only those which are the most interesting for data analysis. Using the NLTK libraries will make it possible.