Information Extraction for Users Reviews from the Steam Gaming Platform

Thach Jean-Pierre **Wong Leo**
Department of Computer Science and Operations Research, University of Montreal

Abstract

In this project, we explored various information extraction approaches to tackle different tasks.We used sentiment analysis tool to measure the users' sentiment regarding their experience with the games. We then designed regular expression rules to automate part of the game title extraction process, which was then followed by manual inspection. POS tags were also used to identify characteristics from the reviews. Combining several of these approaches, we also tried to identify sequences maximizing connotative polarity on each extreme. Game types and opinions were also extracted by using open-source packages. Finally, we used a NER approach in SpaCy to try to identify video game related entities.

1. Introduction

Product and service reviews have become a ubiquitous aspect of every major business corporation nowadays. Companies allow their clients to provide feedback and ratings, which contain invaluable information that could be beneficial. Extraction of relevant insight is often a complex task relying on various tools and techniques. In this project, we explore numerous information extraction approaches with the endeavor of solving different problems.

2. Approaches

2.1 Sentiment analysis

In order to identify the social sentiment of users towards games in their reviews, we've used VADER, a lexicon and rule-based sentiment analysis tool. It features analysis on negations, punctuation emphasis and flooding, capitalization, emoticons and emojis, slang words, acronyms, etc. The compound score is calculated based on the summation of the valence (polarity and intensity) of each word in the lexicon, adjusted according to its rules, and normalized between -1 (most extreme negative) and +1 (most extreme positive). Moreover, to obtain the sentiment of a review, the analyzer's *polarity_scores* function measures the compound score and allows us to classify a given sentence based on a set of threshold values for each sentiment: positive (compound score greater than or equal to 0.05), negative (compound score less than or equal to -0.05) and neutral (compound score between 0.05 and -0.05 inclusively).

2.2 Extracting game titles

Video game titles can be extracted by implementing a series of regular expressions based on various sequences of words usually seen before or after. For instance, searching the reviews for all occurrences of "a copy of" would potentially allow us to identify the names of games. However, due to the amount of noise in the reviews, this approach would also fetch

irrelevant information. Therefore, a manual inspection is required in order to identify impossible game titles.

2.3 Extracting elements' characteristics of games

Our approach in extracting elements' characteristics of games involves Part-of-speech tags provided by the NLTK library. To this end, we've used simple patterns such as adjectives (JJ), comparatives adjectives (JJR) and superlative adjectives (JJS), and also pairs of tags, for instance, adjectives followed by nouns. Combining this with the sentiment analysis approach, reviews were analyzed and clustered together depending on their connotative polarity and the *recommend* attribute.

2.4 Identifying game types

Furthermore, to identify game tags such as *action, adventure, RPG, etc.*, our approach revolves around the frequency of the most popular official Steam tags for products.

2.5 Identifying characteristic sequences

Based on the preceding game characteristic extraction approach, identifying sequences that would theoretically maximize connotative polarity on each extreme consists of analyzing reviews in which were recommended by users and resulted in positive sentiment, and similarly for unrecommended reviews with negative sentiment. Then, based on the most frequent sequences and their frequencies over recommended and unrecommended reviews, we've captured potential features.

2.6 Extracting game opinions

With the goal of extracting game opinions, we've tackled this issue by using the Stanford OpenIE python wrapper. This system splits each sentence into a set of entailed clauses through which the clauses are shortened to produce a set entailed shorter sentence fragments. Thereafter, the system outputs OpenIE triples from segmenting these fragments. Thus, we obtain structured relations of an unspecified schema.

2.7 Named-entity recognition

It also piqued our curiosity to see if or how well NER would extract named entities from the reviews, most notably game titles in the current context. Leveraging pre-trained models from SpaCy, the corpus can be annotated on a token-level in order to describe entity boundaries. Given that NER is an adequate technique to extract people's names or dates among other categories, we hypothesized that it would also be capable of capturing and identifying video game companies (ie. *EA*, *Activision*, *Ubisoft*, etc.) or titles (ie. *Minecraft*, *GTA*, etc.).

2. Methodology

2.1 Identifying game titles

The extraction of game names is broken down into two steps. The first one consists of generating a list of substrings of 1 to 3 words long, by matching custom patterns in the reviews. The second step is to manually go through the outputs to weed out improbably game titles (ie. 'this', 'is', 'is the best', etc.).

2.2 Identifying game types

The requests library is used to perform a get method on the Steam popular tags store URL. Together with the Beautiful Soup Python library, which pulls data out of HTML files, we are able to proceed by finding all the *div* class names text equal to *tag_browse_tag*. This resulted in desired output where these game tags can be

used afterwards to determine whether users reviews contain information regarding game types. Given that there are more than 400 game types retrieved, we use a threshold of at least 25 counts to produce *Figure 1*. Counts are incremented by 1 once for each game type occurrence in each review.

2.3 Extracting game opinions

With the limitation of maximum length allowed of 100000 characters per client request on the local StanfordCoreNLP server, and containing 12849892 characters in the extracted reviews, we opted to focus on a subsample of recommended and unrecommended reviews.

3. Results

2.1 Sentiment analysis

From the *Table 1*, we can see false positive and true negative reviews consist of 4.187% of the total number of reviews extracted. This indicates that the VADER analyzer may incorrectly determine the overall sentiment of reviews or that reviews may be contradictory with the *recommend* attribute inputted by the user.

Table 1 - Number of reviews using the VADER sentiment analyzer

Reviews		**Recommended**	
		True	False
Sentiment	Positive	9286	1068
	Neutral	41998	5570
	Negative	1189	194

By examining on false positive reviews, the occurrences of the top ranking adjectives comprised of positive sequences, for instance, *good* (158)*, great* (108)*, AWESOME* (81)*, best* (71)*, Great* (51).

Furthermore, there are 964 reviews out of the 1068 false positive reviews that are determined as majoritarily positive sentiment; probability over 50%, with an absence of negative sentiment; probability of 0%.

This shows that the latter hypothesis of contradictory recommendation and review sentiment is generally accurate.

Moreover, there are 766 reviews out of the 1189 true negative reviews with nonexistent positive sentiment; probability of 0% and a negative sentiment probability over 50%. Negative sequences that ranks highest in terms of frequency are the following: *bad* (878)*, i* (83)*, good* (31)*, dead* (22)*, sick* (19).

Given that the sequences *i* and *good* appears in the ranking, this suggests that the VADER analyzer has more trouble determining with certainty that reviews are truly negative.

2.2 Extracting game titles

Parsing the corpus through multiple regular expressions, we compile a list of potential game titles along with their number of occurrences (in the file *game_titles_raw*). The automated process generating the list of possible names was not accurate, since it wasn't able to filter a lot of the noise. For instance, when we search for words preceding the string 'is amazing', we also received irrelevant information like 'this game' or 'the story'. Analyzing the top 30 entries returned in the output, only 1 game title was correctly predicted.

When we manually go through the output and discard any tokens that are obviously not game

titles (ie. ‘this’, ‘playing’, ‘with friends’), we are able to obtain a list of legitimate titles (in the file *game_titles*). Looking at the records with 10 occurrences or more, we were able to find 48 game titles. The game that was mentioned the most appeared to be *Team Fortress 2* with over one hundred counts, followed closely by *Call of Duty* and *Minecraft*.

2.3 Extracting elements’ characteristics of games

Sequences of adjectives and nouns ordered based on the most occurring pairs mainly informed commonly used and generic adjectives to describe the word *game* on either connotative polarity and recommendation. As such, we investigated and extracted the 500 least common unique pair of sequences and removed uninteresting sequences based on our judgment.

Table 2 - Number of unique pair of sequences (adjectives followed by a noun)

Pair of Sequences		**Recommended**	
		True	False
Sentiment	Positive	314/500	266/330
	Neutral	359/500	295/500
	Negative	185/263	33/46

Manual selection of informative and sequences related to its connotative polarity produced these precision values where more than 59% of sequences are kept for each case. This indicates that our methodology needs to be improved due to high discard rate. We’ve notably noticed from the extracted pairs of sequences, many irrelevant information in relation to its connotative polarity. For instance, the sequences *‘just play* and *u shld* in recommended and positive sentiment or *great port* and *good game* in unrecommended and negative sentiment reviews.

In addition, to extract sequences of one single adjective, we adopted a similar methodology formerly described, but with the 100 most common unique sequences. We get the following table:

Table 3 - Number of unique sequences (an adjective)

Sequences		**Recommended**	
		True	False
Sentiment	Positive	73/100	79/100
	Neutral	75/100	82/100
	Negative	52/100	31/59

We can notice that we have higher sequence discard ratio with negative sentiment than with either positive or neutral sentiment. This can be explained by the noise contained in true negative reviews, but also the VADER analyzer’s capability in determining with certainty, negative sentiment.

For the previous sequences manually selected from a subsample of sequences, they can be found in the provided files with the *recommend* prefix, respectively suffixed with *pairs* for adjectives followed by nouns or *adj* for a sequence of one adjective.

2.4 Identifying game types

We’ve identified 270 game types over the reviews out of 405. As shown in *Figure 1*, game types were taken into account if their counts are above the threshold set at 25. We can observe a considerable disparity between the top 5 game types (*FPS, RPG, War, Mod* and *Multiplayer*)

and other game types with respect to their frequency.

2.5 Identifying characteristic sequences

By combining the top 5 most frequent and unique sequences from positive and negative labelled sentiments, we obtain the following table:

Table 4 - Number of occurrences of sequences in recommended and unrecommended reviews

Sequence	**Number of occurrences in 52512 recommended reviews**	**Number of occurrences in 6836 unrecommended reviews**
good	9597	1124
great	8585	484
best	5696	223
awesome	3544	75
fun	9560	808
free	2138	332
bad	1424	1439
dead	599	161
sick	203	21
terrible	197	270
worst	140	275
boring	464	360
crappy	55	34

*__Note__: Green cells signify sequences that occurs more in either recommended or unrecommended reviews; calculated based on their ratios.

By exploring the top 5 most frequent sequences of adjectives followed by a noun, we found that adjectives contained in the pair of sequences were already present in *Table 4.* For example, *good game* already contains the sequence *good*. Thus, it wouldn't theoretically be a relevant feature to consider.

2.6 Extracting game opinions

From the OpenIE triples yielded from a subsample of recommended reviews, 325 out of 2512 were judged insightful on users opinion pertaining a particular game. Some interesting examples of game opinions extracted from these reviews are related to the leveling system, superb music, awesome light effects, game characters driven, rich layer of gameplay types (survival), business model (game ruined by microtransaction, free-to-play degrading the playerbase, pay to win being detrimental), rich story, good community and playerbase, wonderful graphics, gametypes (need of multiplayer option, fun player vs player, recommended for the player vs environment experience), balance issues, etc. The retained triples can be found in the *triples-OpenIE-recommended-true* file.

Furthermore, triples produced from a subsample of unrecommended review, 384 out of 1994 were determined interesting such as the environment being too vibrant and hollow, fighting combat system being worse due to parries for instance, heavy controls, instability and problems with connection issues, rubber banding, game full of bugs, items quickly becoming outdated in terms of stats, game repetitiveness, dissatisfaction with the storyline or ending, game crashing or undoing progress, broken AI, glitchiness, server lag, unplayable multiplayer due to host migration, terrible looting, pointless side missions, poor game

optimization, interface designed for console, terrible voice acting, game being a clone of *Call of Duty*, game ruined by hackers, unbalanced game, errors with lock-on system, problems with rolling mechanic, unchallenging DLC, models re-textured from early ARMA games, uneven playing field, grinding game, frame per second drop issues, etc. The triples were stored in the *triples-OpenIE-recommended-false* file.

2.7 Named-entity recognition

Using the pre-trained model *en_core_web_lg* from SpaCy, the following categories were extracted from the reviews:

Counter({'CARDINAL': 1584, 'PERSON': 1568, 'ORG': 1294, 'DATE': 468, 'PRODUCT': 379, 'TIME': 360, 'GPE': 329, 'ORDINAL': 276, 'NORP': 217, 'FAC': 199, 'MONEY': 193, 'LOC': 119, 'PERCENT': 96, 'QUANTITY': 77, 'WORK_OF_ART': 76, 'EVENT': 48, 'LANGUAGE': 17, 'LAW': 13})

Shown in *Figure 2*, we can see visually how the different named entities were labeled. Game companies and titles were mostly tagged with the label *ORG*. Extracting the top 50 most frequent tokens (in the file *ner_top50*), 7 were game titles, 4 were ratings and the rest were miscellaneous entries unrelated to video games.

4. References

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, *June 2014*.
https://github.com/cjhutto/vaderSentiment

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D. Manning. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the Association of Computational Linguistics (ACL)*, *2015*.
https://github.com/philipperemy/Stanford-OpenIE-Python

5. Appendix

Figure 1 - Game types frequency in reviews

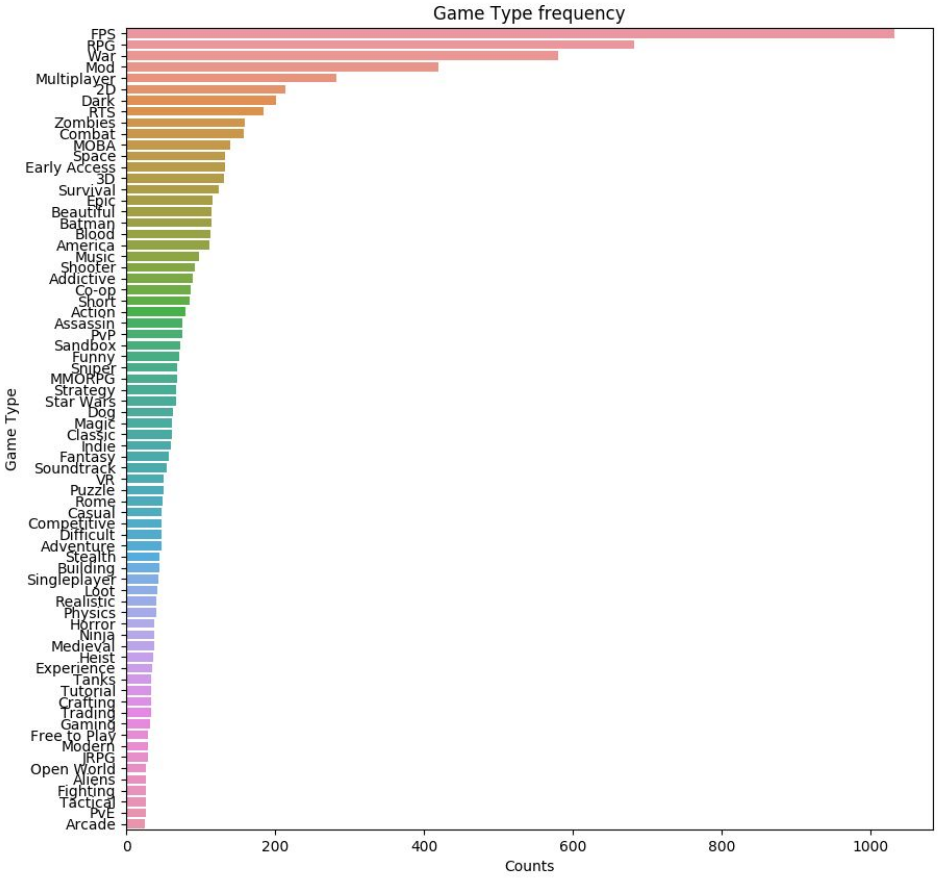

Figure 2 - Snippet of NER annotation by pre-trained model in SpaCy

my opinion a masterpiece, it is an amazing game that is what games should truly be. games like this strike fear into the hearts of large parasitic companies like activision ORG , ea GPE , ubisoft ORG , etc that only care about profit and don't care about their players or their games. games like this show that small groups or even individual developers can take a dream, mix it with determination and sprinkle some pride on top to create an amazing game and even then by a tiny, tiny bit. must buy. best bf PERSON . must buy. laggy, boring and bland. awesome game so cool like diablo 3 PRODUCT it will not launchplease PERSON fix this this is the best city building game i have ever played. it does everything that simcity 2013 ORG failed to do and more. it has good graphics. amazing simulation. suprisingly realistic water. and where do i start with the mods? there is a mod for everything!!!!!!!! and they are easy to install using the steam workshop. (if only minecraft ORG mods were this simple.the only annoying thing is it doens't have natural disasters.