Aspect Based Sentiment Analysis in boardgames reviews

Stefano Zanlucchi Matr.N.22288A stefano.zanlucchi@studenti.unimi.it

Data Science and Economics, University Milano, Milano, Italy.

Abstract

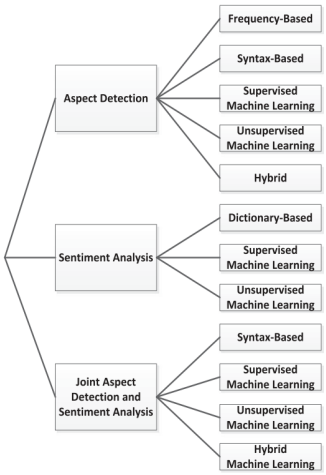
The purpose of this project is to explore how aspect based sentiment analysis can be used in directly real life problems as the investigation of features that are related to board games. The approach followed is quite straightforward and allows to have a nice impression and feeling about the potentiality of this method.

Keywords: ABSA, Word2Vec, Ptyhon, Text Mining, Sentiment Analysis

1 Introduction

This project focuses on the topic of aspect based sentiment analysis (*ABSA*) regarding board games. We can use as data for this research the website *BoardGameGeek*[1] (abbreviated as *BGG*) that is accessible through its API.
In the digital area people share their opinions about products as reviews on the web, and *BGG* website is an example of this type of new web-based contents. As [2] explains there are three main processing steps that can be distinguished when we are performing *ABSA*, as shown in the image on the right:

1. identification: it happens when aspects are detected with various techniques:
   - frequency-bases methods: when a limited set of words is used more than the other remaining word of the vocabulary
   - syntax-based methods: using syntactical relations to find aspects
   - supervised machine learning
   - unsupervised machine learning: as LDA and LSA. As written in [2] the topics that LDA returns are simply too global in the scope to catch the more locally defined aspects



    - hybrid methods
2. classification: it is the part when the sentiment analysis is performed with respect to a specific aspect and it can be achieved with:
    - dictionary-based: as Wordnet that assign a sentiment class to adjectives
    - supervised machine learning: that learn parameters from the data using sentiment lexicons
    - unsupervised machine learning: that look for sentiments phrase in the vicinity
3. aggregation: it is the last step when we present a summary of the collected sentiments about the searched aspects.

As clarified in [3] in social media sentiments can change over time due to the external influence of persons that can affect the opinions of the users.

2 Research question and methodology

The project aims to investigates the board games user reviews with regards to following aspects and their respective sentiments:

- *luck or alea*: refers to elements independent to player intervention
- *bookeeping*: means manual process of the game that causes the need of rulebook as continuous reference
- *downtime*: indicates unproductive waiting time neither doing nor thinking
- *interaction*: refers to the effects of actions on other players
- *bash the leader*: means that there are some behaviors of players that can prevent the victory of a leader for example sacrificing themselves
- *complicated vs complex*: complicated means with a lot of rules and exceptions that need to be known, but the outcome are certain and predictable; complex instead means that action can difficultly predicted and mastered.

The general outline to approach this problem deals wtih these main tasks:

- extraction the user reviews
- identification of the aspects using few different methods
- extraction of the aspects detected in the comments
- classification of the sentiments with respect to the aspects
- presentation of the results with aggregated tables about the aspect sentiments,

3 Experimental results

In this part we can explain better how the full analysis is performed. As briefly introduced in paragraph 2, the reasearch is essentially divided in these fundamental targets:

1. dataset creation based on the extraction of user reviews about board games
2. identification of the aspects, task managed with some possible methods
3. aspects extraction and sentiment analysis performed with 2 approaches and their representation of the outcomes.

3.1 Dataset preparation

The first step is to create the dataset, which can be obtained using the connection with the API of *BGG* to retrieve and store the relevant information as ratings, comments, etc into a python dataframe. To get a significant amount of data we created an algorithm that performs following activities:

1. create a list with the top 15 boardgames of the global ranking available on the website
2. since the API has an upper limit of comments that can be shown for each game in a single xml queried webpage, we can avoid this issue creating a first step that goes through all the games and extracts the total amount of comments so that we compute the number of pages of user comments for each board games
3. iterate for each game over its respective number of comments pages to extract all the available reviews
4. since we are not willing to manage different language we can filter the dataset just with English comments using the python library `langid`; as shown in fig.1 we are considering the majority of user comments selecting just English.

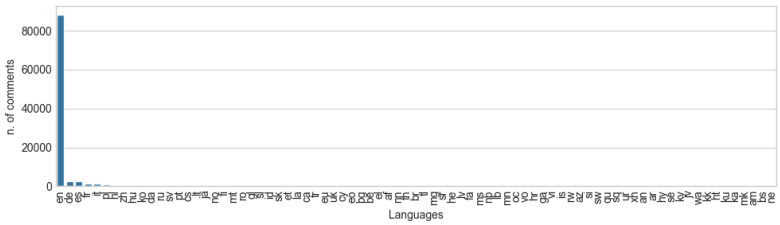


Fig. 1 Bar plot of the aggregated number of comments for language type

At the end of this process we can have two dataframes: the first one is just a simple list with game information, the second one is the collection of all the reviews that will be used in the following steps to extract aspects and classify the sentiments.

3.2 Aspects detection

The first step, before looking at the specific methods for aspect extractions, is to clarify the aspect *bash the leader*. As we can see this is not a single word that can be found in a vocabulary of the various comments, but it rather looks more as a concept or a verbal expression, so we can propose following approaches to get reference to its meaning:

- we can think about some related words to the idea of *bash the leader* as attack, blame, leader, sacrifice, etc
- we can use a pre-trained model to look for other similar nouns to above related words
- based on the most similar words we decide that *sacrifice* could be a good word to get a meaning closed to *bash the leader* when we will analyze the user comments later.

Secondly, we can define a method to find references to the aspects: we decide to create baskets of similar words (e.g. 10 nouns) that are linked to our aspects. We can find this collection of similar words using 3 different approaches:

1. *a pre-trained model*: we can download from Google a collection of news that can be fed to `Word2Vec` with the target to get the meaning of the aspect thanks to surrounding words. In fact in this way we can identify the top 10 most synonymous linked to our target aspects. Using `t-SNE` library of `scikit-learn` and thanks to its non-linear dimensionality reduction, we can nicely obtain a plot into 2 dimensions from the 100 dimensions vector that are given as output of Word2Vec application: this helps us to observe the clusters used for the 6 aspects as shown on the left side of fig.2

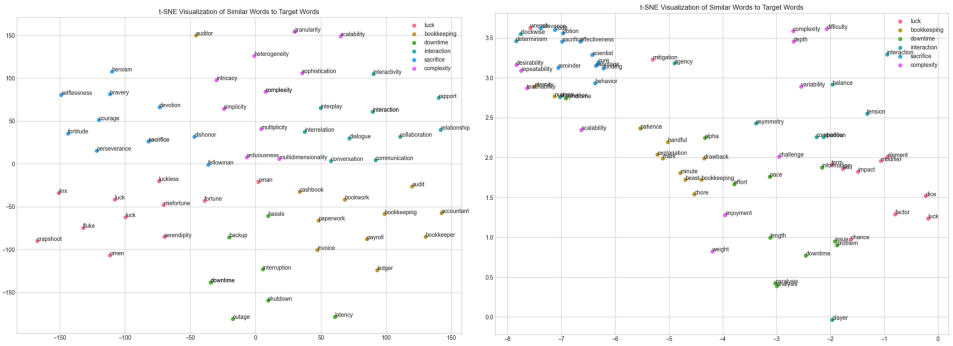


Fig. 2 Reduced overview of similar words with `Word2Vec`: (a) model pre-trained on the left, (b) model trained on dataset of comments on the right right

2. *a trained model on the user comments*: we can give as input to `Word2Vec` model the full vocabulary of the *BGG* comments, that we can get tokenizing each review, and eventually we can retrieve again the top 10 similar nouns for each target aspect; in fact this is represented on the right of fig.2
3. *a rule-based model*: we define a dictionary by our own considering 5/10 synonymous of each aspect, but this approach is somehow disregarded in this project since it is oversimple.

```
Aspect: luck
Similar nouns: ['luck', 'factor', 'skill', 'dice', 'impact', 'mercy', 'mitigation', 'variance', 'determiner', 'chance', 'element']

Aspect: bookkeeping
Similar nouns: ['bookkeeping', 'chore', 'explanation', 'drawback', 'beast', 'minute', 'load', 'repeatability', 'effort', 'playgroup', 'maintenance']

Aspect: downtime
Similar nouns: ['downtime', 'alpha', 'syndrome', 'information', 'length', 'collaboration', 'problem', 'issue', 'analysis', 'complaint', 'hour']

Aspect: interaction
Similar nouns: ['interaction', 'competition', 'player', 'shadow', 'elimination', 'tension', 'agency', 'asymmetry', 'cooperation', 'term', 'balance']

Aspect: sacrifice
Similar nouns: ['sacrifice', 'anxiety', 'potion', 'incentive', 'agony', 'effectiveness', 'reminder', 'item', 'capture', 'pattern', 'cure']

Aspect: complexity
Similar nouns: ['complexity', 'depth', 'variability', 'weight', 'difficulty', 'ease', 'enjoyment', 'variation', 'scalability', 'challenge', 'desirability']
```

Fig. 3 Display of the dictionary in output from the model tuned on the user reviews

The output of the previous three methods is a dictionary as presented in fig.3 that helps us in some ways to wide spreading the aspect meanings. Indeed words that have similar significance have similar representation in the vector dimensions and with its *Skip-gram* architecture, `Word2Vec` allows to get these surrounding words based on the input target.

3.3 Aspect extraction & Sentiment Classification

In this part of the project we are curious to investigate two methods to perform the steps of aspects extraction and sentiment classification which are the following:

- method 1:
  1. the aspect extraction is performed searching exact matches in the comments to the collection of words representing the aspects and saving them into a dictionary with the specific sentence that shows the match
  2. using the `TextBlob` library we can analyze the sentiment linked to the sentence referred to the aspect
- method 2:
  1. we can use a dependency based parsing approach to get adjectives referred to to our target nouns in this case we just look at the lemma and not the exact match
  2. using th `VADER` library we can evaluated the sentiment of the retrieved adjectives.

3.3.1 Method 1

The aspect extraction of this first method is performed according to main following points:

- iterate over each comment of the *BGG* dataset
- tokenize each comment as a sentence using `nltk` library
- for each sentence iterate over all aspects and related keywords of the dictionary and then search for a potential exact match with the words of the sentence
- produce a new column with a dictionary that is storing the aspect and the sentence that matches the keyword (that will be used in the next step of the sentiment classification) as shown in fig.4.

```
# Example of how the columns Aspects_Extracted looks like
df_filtered_1.iloc[6,3]
```

```
{'interaction': ['Fun in terms of hand management and possibility of collaboration with other players, creating moments of "Whoa!"']}
```

Fig. 4 Example of dictionary built with aspect and linked sentence

To extract the sentiment we rely on already built Python libraries as `TextBlob` that takes a sentence and returns a tuple of the sentiment presented with a polarity score (ranging from -1 negative and 1 positive) and a subjectivity rate (ranging between 0 objective and 1 subjective). In detail we process it according to these steps:

- we define a function to get the sentiment of the sentence that refers to an aspect

- we store each sentiment of the aspect into a new dictionary that is placed into a new column of the dataset.

We can test this approach on the two dictionary that we were able to build in the previous paragraph 3.2. In addition we can develop few other functions to perform the analysis about how each board game is evaluated by the users with regards to the aspects:

- one function is used to filter the dataset selecting a specific board game and to prepare a dataframe with the count of the sentiments for each aspect
- another function is used to obtain a multi-plot grid for plotting the aggregated polarities of the aspects

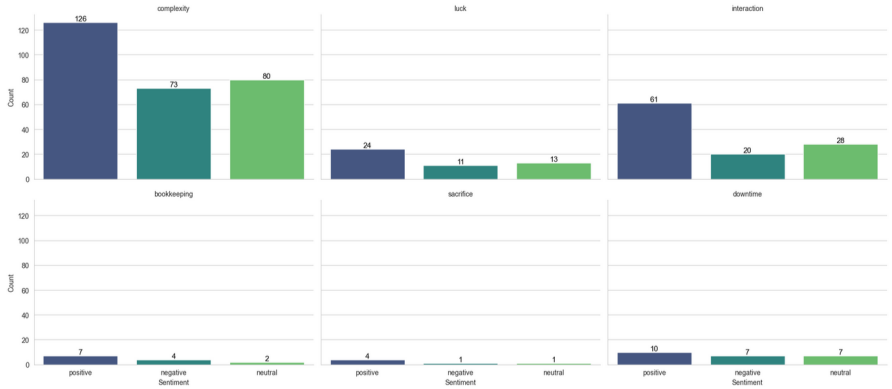


Fig. 5 Example of game Spirit Island (id = 162886): polarity evaluations of the aspects using dictionary synonyms of the pre-trained model

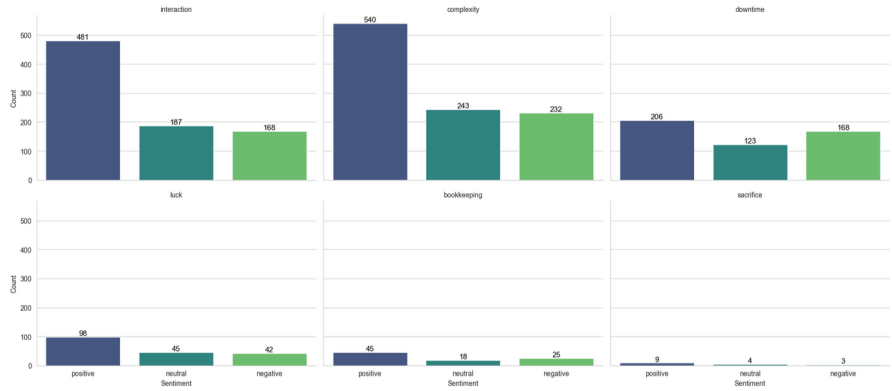


Fig. 6 Example of game Spirit Island (id = 162886): polarity evaluations of the aspects using dictionary synonyms of the comments-tuned model

In this way we can obtain the charts as shown in fig.5 using the similar words obtained by the pre-trained model and in fig.6 applying the words obtained with model tuned

on the user comments.
In both graphs we can see:

- *interaction, complexity, luck, downtime* are better detected as frequent aspects in the comments
- since took the top 15 ranking, we have definitely positive comments and therefore also positive opinions in the sentences referred to the target aspects
- we get higher amount of matched aspects and sentiments using the model trained on the words of user reviews. This is likely since the specific words of the users are more linked to our target aspect and the jargon used in the reviews. One example of this methos is represented in fig.7 that shows a *interaction* opinion is well identified.

```
Aspects Extracted:
{'complexity': ['Gloomhaven toes the threshold between being a challenge and a being an abusive girlfriend or boyfriend.'], 'luck': ['It is
the only part that I can say I genuinely hate: how much trial and error + luck it takes to beat any scenario.']}

Aspect Sentiments:
Aspect: complexity, Sentiment: ['neutral']
Aspect: luck, Sentiment: ['negative']
```

Fig. 7 Example of user comment using the dictionary tuned on user comments and `TextBlob`

3.3.2 Method 2

The second approach develops a parsing method to detect the aspect sentiments exploring the dependencies occurring in the sentences as:

- adjectives linked to nouns
- nouns, verbs and adjectives dependence

We start from the functions known from our course [4] and we can adapt them adding some features:

- we take into account also possible negations present in the dependencies
- instead of looking for and exact match to our similar words, we can search for references to the lemma that should have higher occurrences
- we combine the functions of adjective search and verb dependency into on function that is processing the comments.

Since the dictionary of similar words from the comments-tuned model showed higher matches in the previous paragraph 3.3.1, in this step we just use this case as input to test this method. In general the output of the function `combine_adjectives_verbs_search` is stored into a new columns and it produces as key a match with the lemma and as item a potential adjectives linked to the lemma plus a status whether the dependence is positive or negated based on the presence or not of negative adverbs (e.g. not).
Then the second stage evaluates the sentiment of the adjectives and for this purpose we explore another Python library, that is `VADER`; we proceed with 2 phases:

1. we get the sentiment score of the detected adjectives taking the *compound score*
2. we aggregate the scores and link them to the target aspects of the board games.

In the end we can rely on the already developed functions to present with a dashboard the different aspects ratings as shown in fig.8. We can conclude that:

- this method is getting more neutral evaluations instead of the previous method

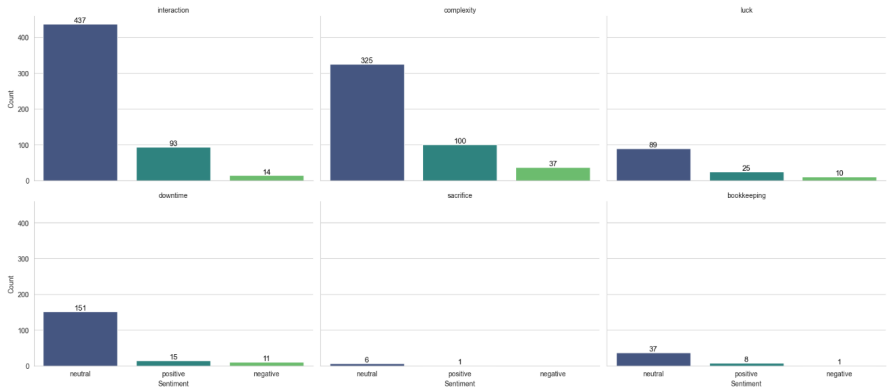


Fig. 8 Example of game Spirit Island (id = 162886): polarity evaluations of the aspects using dictionary synonyms of the comments-tuned model with method 2

- this can be caused due to the fact that noun dependency is finding only the adjacent adjective that in general could express also objective evaluations, in deed a sentiment could be more expressed with the whole sentence meaning.

4 Concluding remarks

In conclusion this project allows us to explore some methods to perform aspect based sentiment analysis on user reviews, in particular we can assess:

- the experimental results are quite interesting with reasonable performances about the correctness of the sentiments
- the approaches depend on the choices as dictionary used for similar referred aspects and on the different methods.

As future work it could be interesting to:

- to prepare a manually labeled dataset that could let us test the real performances of the two developed methods in this project
- investigate how other methods as BERT are able to perform in ABSA
- the dataset itself is more favorable for sentiment evaluation of the user since it has a specific rating given by the user, therefore we could try to find a method that allows as to score from 1 to 10 the comments

References

[1] BGG: Board Game Geek. https://boardgamegeek.com/

[2] Kim Schouten, F.F.: Survey on Aspect-Level Sentiment Analysis (2016)

[3] Ambreen Nazir, L.W. Yuan Rao, Sun, L.: Issues and Challenges of Aspect-based Sentiment Analysis: A Comprehensive Survey (2022)

[4] Ferrara, A.: Text Mining and Sentiment Analysis - repository. https://github.com/afflint/textsent/