What do you like in boardgames

Francesco Torgano 16028A

Department of Computer Science, Università degli Studi di Milano.

Abstract

This project will focus on using sentiment analysis models on comments/reviews about board games to extract informations about which aspects of the game are good and which aren't. This task will be done initially using a pre-defined list of common aspects to all boardgames to gather sentiment about general boardgame aspects and subsequently trying to extract the most liked and hated aspects directly from the comments/reviews to find game-specific aspects and their polarization.

Keywords: Aspect-Based Sentiment Analysis, Aspect Term Extraction, Aspect Polarity Classification, Board games

1 Introduction

Sentiment analysis is a task commonly used to extract an overall sentiment from reviews or opinions, this can be used by companies to check if their new product is well liked or not, but a broad good/bad opinion is not always specific enough. To help with this problem, Aspect Polarity Classification (APC) is often used to check the sentiment of users of a product on some specific features (e.g., the length of a game of a board game). Choosing which aspects to analyze can be complex, omitting some could cause the wrong decision to be taken, so this task is often automated with Aspect Term Extraction (ATE). ATE tries to extract aspects automatically from the text. This ability, paired with APC, allows gathering insights into what makes a product good or bad.

2 Research question and methodology

2.1 Goal

The focus of this project will be on trying to extract additional information from board game reviews. This will be done in two main ways: the first will be to extract sentiment from the comments on a predefined set of general aspects, and the second will be to extract both aspects and sentiment directly from the reviews.

The first task can be useful to further categorize the board games on some specific features or help with recommending games that have some specific aspects more developed than others. An example of this is, given the sentence "Great game, only the luck of the draw knocks it a peg down." and the aspect "luck", the output should be "negative" because the reviewer views the luck aspect of the game negatively.

The second task will be carried out by applying an ATE+APC model to try to find out which are the best and worst aspects for each board game. The information provided can be useful to both board game designers to improve for new versions of the game or while developing new expansions, and to the customers to find out if a feature they hate or love is in the board game they are considering and if it's a positive or negative feature. An example for this task is, given the sentence "Very good game. A little too much luck involved for a 10, but game mechanics are fun", the output should be a series of aspects and the sentiment towards each: ("game", positive), ("luck", negative), ("mechanics", positive).

The first model will be used to evaluate the sentiment towards the predefined set of aspects provided in the project description and obtain an overall sentiment score towards that aspect in that game, meanwhile the second model will be used to generate two lists containing, respectively, the best and worst aspects for each game considered.

2.2 Data

2.2.1 Data source

The board game review data will be downloaded using the BoardGameGeek (BGG) XML API 2 that allows to download the information about a specific game, including comments/reviews, by knowing the board game ID. The board game IDs will be obtained by downloading the complete list of games available on .csv file directly on the website after logging in.

There are more than 150 thousands board games in the BGG website so, to limit the computation time required, only the top 10 games will be considered.

For each game, the full set of reviews/comments will be downloaded and used as the initial dataset. The initial size for the top 10 games is 67.9 thousand reviews.

2.2.2 Data preparation

The downloaded comment will be filtered to try to make the dataset easier to manage and to generate more meaning results.

Since neither of the tasks considered has a test set that can be used to value the performance of the models, the comments will be filtered based on language (using the fasttext library developed by researchers at Facebook [1] [2]) to keep only reviews in English.

The data will also be filtered based on the length of the review, removing all reviews which are shorter than the bottom 25% of reviews. This helps reduce the dataset size and keep texts that are longer, which might contain more information about aspects of the game.

After the last step, there are around 45.1 thousand reviews remaining, each will be used for both sentiment analysis tasks.

2.3 Models

2.3.1 APC

The Aspect Polarity Classification (APC) task will be carried out by using the pre-trained yangheng/deberta-v3-base-absa-v1.1 model available on HuggingFace. This model is based on a novel architecture for Aspect-Based Sentiment Classification that uses a Local Context Focus (LCF) mechanism to focus the attention to the words in the local context and provide better performance for the APC task [3]. The authors of the paper noticed that in analyzing the sentiment towards an aspect, the aspect's local information were particularly important and that in other common techniques for APC, this area was understudied. The LCF was designed to consider the local information more than the global so that the words near the aspect will have more importance. This is done using Semantic-Releative Distance (SRD) which determines if a word is the local context of a specific aspect:

$$SRD_i = |i - P_a| - \lfloor \frac{m}{2} \rfloor$$

where i is the position of the contextual word, P_a is the central position of the aspect and m is the sequence length of the aspect. SRD is then used by Context-features Dynamic Mask (CDM) and Context-features Dynamic Weighting (CDW) layers to weaken the influence of less-semantic-relative contextual word: the CDM layer masks the output representation of words, while the CDW weakens the features.

This architecture allows the model to effectively capture and utilize both local and global context features, leading to improved performance in sentiment classification tasks.

2.3.2 ATE and APC

The ATE+APC will be carried out using the Fast-LCF-ATEPC developed by the Yang et al. [4] available from PyABSA. PyABSA is a framework to allow for a much easier access to complex models for ABSA tasks and subtasks in Python [5]. This model combines the tasks of Aspect Term Extraction and Aspect Polarity Classification in a single multi-task model based on a modified version of the LCF mechanism that was explained earlier. This model uses the global context features obtained from the "global" branch of the architecture with an aspect extractor to determine if a term is an aspect or not.

2.4 Result analysis

2.4.1 No true labels

The main problem of this project is analyzing the results, since the dataset doesn't have labels for neither of the two considered task.

A possible solution could be to label a set of random examples extracted from the dataset, but that can become long and expensive. For the first task, it's hard to generate a small test set that contains all the predefined aspects for each game considered, especially if trying to build a balanced dataset. Building a test set, for the second task, suffers from similar problems.

The solution used here is an empirical one: in addition to the top ten games, two additional games: UNO (3.4k reviews) and Everdell (4.5k reviews) will be considered and will be used as a set of control variables. The additional information provided by these two games won't allow to evaluate the other results in a general sense but will allow checking if the output of the model makes sense in the first place.

2.4.2 Overall sentiment score

Another, smaller, problem is how to obtain a score/metric from a series of Positive, Negative and Neutral labels. Obtaining a score is needed for both our tasks: it's used to evaluate the overall sentiment in the first task and to sort the aspects based on the evaluation of the aspect in the second one.

For each pair (game, aspect) a score is calculated to gauge the polarity of the sentiment towards the aspect for that game, as follows:

$$S(game, aspect) = \frac{\#Positive - \#Negatives}{\#Reviews}$$

A score S close to zero means that the positive and negative classifications balance themselves out so the overall sentiment is neutral, a score closer to the extremes, -1 and 1, means that the overall sentiment is, respectively, mostly negative or mostly positive.

3 Experimental Results

The results will report the output of the various models for each of the top ten board games and the two control games for the different tasks that they will be used for.

3.1 APC — Classifying predefined aspects

The goal of this task is to take the predefined list of aspects that was given in the text of the assignment and try to extract the sentiment towards each aspect from the comment for each game. This output of the model is, given a comment/review and an aspect, a label (positive, negative, neutral) with an associated score.

The scores S for each combination of game and aspect are reported in table 1 and 2.

Table 1 Results obtained from the Aspect Polarity Classification model for the first 4 aspects considered: Luck, Bookkeeping, Downtime, and Interaction

Game	Luck	Bookkeeping	Downtime	Interaction
Brass: Birmingham	0.37	0.26	0.31	0.41
Pandemic Legacy: Season 1	0.34	0.24	0.33	0.39
Gloomhaven	0.29	0.17	0.22	0.3
Ark Nova	0.24	0.2	0.18	0.25
Twilight Imperium: 4th Edition	0.32	0.21	0.2	0.33
Dune: Imperium	0.38	0.31	0.37	0.42
Terraforming Mars	0.28	0.2	0.22	0.29
War of the Ring: 2nd Edition	0.38	0.28	0.32	0.42
Star Wars: Rebellion	0.38	0.29	0.32	0.42
Gloomhaven: Jaws of the Lion	0.35	0.25	0.33	0.4
UNO	0.13	0.036	0.12	0.14
Everdell	0.39	0.32	0.38	0.42

Table 2 Results obtained from the Aspect Polarity Classification model for the last 3 aspects considered: Bash the leader, Complicated and Complex

Game	Bash the leader	Complicated	Complex
Brass: Birmingham	0.24	-0.4	0.095
Pandemic Legacy: Season 1	0.21	-0.33	0.1
Gloomhaven	0.17	-0.42	-0.0082
Ark Nova	0.16	-0.4	0.034
Twilight Imperium: 4th Edition	0.17	-0.44	0.033
Dune: Imperium	0.3	-0.26	0.22
Terraforming Mars	0.17	-0.4	0.063
War of the Ring: 2nd Edition	0.24	-0.42	0.073
Star Wars: Rebellion	0.25	-0.33	0.13
Gloomhaven: Jaws of the Lion	0.23	-0.38	0.038
UNO	-0.068	-0.67	-0.16
Everdell	0.31	-0.24	0.2

The results from the control games show that the results make sense: UNO luck score is the lowest with 0.13 which is to be expected considering the highly random nature of the game, Everdell has the highest downtime score since, and it makes sense since the turns are taken one by one, but it's also important to keep track of what happens since you will then have to change your turn based on what the opponents are doing so the downtime is probably mentioned a lot and positively most of the time.

Most of the results make sense, but for the ones for the aspects complicated and complex. The ”complicated” aspect scores are mostly negative, this is probably caused by the word chosen to represent the aspect, ”complicated” has a negative connotation when used in general language so it might be mostly paired with negative adjectives that could be biasing the model prediction. A similar thing happens with the aspect ”complex” but in this case the results are more neutral.

These results could be probably improved by fine-tuning the model to improve the understanding of each aspect in the specific world of board games.

3.2 ETA+APC — Extracting aspects and classifying them

The goal of this task is to extract from each comment a series of aspect and their polarization (positive, negative, or neutral). This can have various uses: allow a game designer to learn the most liked and disliked elements of their game, allow a website to add automatically generated tags to each game to allow better search, or to give better recommendations to a player based on already played games.

In this specific case, the models will be used to generate a series of aspects for each game, and that data will be used to create a list of the best and worst aspects of each game.

The aspects extracted will be sorted based on the score S calculated as explained earlier in section 2.4.2. The aspect list obtained is filtered to remove some unwanted values like the name of the game or the word ’game’ since the goal is to understand more about the aspects of the game, not the game itself.

The aspect list is also filtered to consider only aspects that have been detected by the model more than a set cutoff value (25 in this case) of times, to avoid having aspects that have been nominated only once and positively go to the top of the chart. Using a higher cutoff value makes the aspects that will be in the list more general, meanwhile having a lower value might lead to more specific aspects being extracted.

The results are reported in table 3 for the best aspects and in table 4 for the worst aspects.

Table 3 Results obtained from the ETA+APC Fast-LCF-ATEPC model from PyABSA combined in a list of the best aspects for each game, the list of aspects is truncated keeping only the top 5 best aspects.

Game	Positive aspects
Brass: Birmingham	’mechanic’, ’interaction’, ’design’, ’player interaction’, ’component’
Pandemic Legacy: Season 1	’gaming’, ’experience’, ’playing’, ’money’, ’narrative’
Gloomhaven	’value’, ’world’, ’content’, ’system’, ’mechanic’
Ark Nova	’theme’, ’mechanic’, ’gameplay’, ’design’, ’play’
Twilight Imperium: 4th Edition	’faction’, ’play’, ’gameplay’, ’expansion’, ’table’
Dune: Imperium	’immortality’, ’mechanic’, ’interaction’, ’play’, ’design’
Terraforming Mars	’replayability’, ’theme’, ’engine building’, ’engine builder’, ’mechanic’
War of the Ring: 2nd Edition	’theme’, ’design’, ’gameplay’, ’component’, ’play’
Star Wars: Rebellion	’theme’, ’component’, ’mechanic’, ’gameplay’, ’design’
Gloomhaven: Jaws of the Lion	’tutorial’, ’book’, ’price’, ’play’, ’dungeon’
UNO	’learn’, ’play’, ’card game’, ’kid’, ’player’
Everdell	’learn’, ’artwork’, ’art’, ’theme’, ’table presence’

The results of the two control board games show that the output of the model make sense. The worst aspects of the game UNO are strategy and luck, both make sense since strategy is almost nonexistent in the game since it’s a mainly luck-based game. The best aspects for UNO instead are: learn and play, highlighting the easy in which everyone can understand and play the game. Meanwhile, in Everdell the worst aspect in the game is text and that makes sense considering that there is a lot of small text on each card and that there are a lot of exceptions to read through in the manual.

Table 4 Results obtained from the ETA+APC Fast-LCF-ATEPC model from PyABSA combined in a list of the worst aspects for each game, the list of aspects is truncated keeping only the top 5 worst aspects.

Game	Negative aspects
Brass: Birmingham	'teach', 'time', 'rule', 'board', 'learn'
Pandemic Legacy: Season 1	'ending', 'replayability', 'card', 'rule', 'difficulty'
Gloomhaven	'set up', 'set', 'setup time', 'setup', 'tear down'
Ark Nova	'randomness', 'luck', 'hour', 'length', 'playtime'
Twilight Imperium: 4th Edition	'hour', 'length', 'playtime', 'player', 'time'
Dune: Imperium	'board', 'card', 'time', 'player', 'component'
Terraforming Mars	'hour', 'downtime', 'component quality', 'randomness', 'material'
War of the Ring: 2nd Edition	'rulebook', 'setup', 'rule', 'time', 'dice'
Star Wars: Rebellion	'hour', 'combat', 'length', 'playtime', 'dice'
Gloomhaven: Jaws of the Lion	'time', 'sticker', 'story', 'card', 'box'
UNO	'strategy', 'luck', 'rule', 'deck', 'card'
Everdell	'text', 'interaction', 'deck', 'player', 'resource'

Comparing the results with the ones obtained for the previous tasks, it's also possible to see that, for example: Brass had one of the highest interaction scores and that feature showed up also as on the top positive aspects of the game, giving more credibility to both results and models.

The initial results (not reported) show that the model does not use a stemmer or a lemmatizer to clean the input. An example of this was the presence of both cards and card as worst aspects of Dune: Imperium. This was done on purpose to allow the underlying BERT model(s) to extrapolate more contextual information. To clean up the results and "normalize" the aspects, the output aspects from the model were lemmatized with the WordNetLemmatizer from nltk.

Overall the results are better than expected, although they can benefit from some improvement and tuning, but each list can give a basic idea of the good and bad general aspects of a game even without fine-tuning the model and could be used to tag a game on boardgamegeek.com for example.

4 Concluding remarks

The results obtained show that both models can obtain apparently reasonable results on both tasks and produce usable results for various tasks related to board games classification, tagging, recommendation and designing.

The output of the models should still be taken with a grain of salt, since there was no test set to properly verify the performance on, but the results of the two control board games feel reasonable and correct.

Some possible expansions/improvements are:

- Expand the number of games and reviews analyzed
- Generate a test set using the comments/reviews to analyze the performance of the model.

- Generate a small training and test set to check the performance of few-shot learning models like SetFit [6] (and SetFitABSA, a variant made for ABSA tasks).
- Fine-tune the models used with domain-specific information to check for performance improvements
- Using the comments/reviews downloaded to generate a custom ABSA dataset to try to evaluate the performance of the model. The authors of the paper of PyABSA[5] also developed ABSADatasets, a tool to make dataset labeling much easier and make them available to use in the PyABSA framework without too many troubles that could be used to make this task more manageable.

References

[1] Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification (2016). https://arxiv.org/abs/1607.01759

[2] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., Mikolov, T.: Fasttext.zip: Compressing text classification models. CoRR **abs/1612.03651** (2016) 1612.03651

[3] Zeng, B., Yang, H., Xu, R., Zhou, W., Han, X.: Lcf: A local context focus mechanism for aspect-based sentiment classification. Applied Sciences **9**(16) (2019) https://doi.org/10.3390/app9163389

[4] Yang, H., Zeng, B., Yang, J., Song, Y., Xu, R.: A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction. CoRR **abs/1912.07976** (2019) 1912.07976

[5] Yang, H., Zhang, C., Li, K.: Pyabsa: A modularized framework for reproducible aspect-based sentiment analysis. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. CIKM '23, pp. 5117–5122. Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3583780.3614752 . https://doi.org/10.1145/3583780.3614752

[6] Tunstall, L., Reimers, N., Jo, U.E.S., Bates, L., Korat, D., Wasserblat, M., Pereg, O.: Efficient Few-Shot Learning Without Prompts (2022). https://arxiv.org/abs/2209.11055