What do you like in boardgames study

A natural language processing project

Jacopo Fichera

Contributing authors: jacopo.fichera@studenti.unimi.it;

1 Introduction

Boardgaming has become a really popular hobby and business in the past years.

The field is very broad and the games themselves can be very different. At the core some elements are shared. Identifying some of these, is the goal of this project. By scrapping review from the web platform BoardGameGeek (BGG) to be collected in a corpus, this project aims in identifying the following aspects [1]:

- **Luck:** How much randomization si present in the game. The higher the degree of luck in a game the lower the agency power of a player. Most games involve some form of luck, almost every dice game, while others do not at all or limit it as much as possible (e.g. "Guards of Atlantis 2").
- **Bookkeeping:** Is mostly a negative feature of some games. It is the manual recording of data or execution of automatic/semi-automatic game processes.
- **Downtime:** It is the passive time in which a player has no agency over the game.
- **Interaction:** Degree of influence of one player w.r.t to the others. It can be be direct like trading in "Catan" or indirect like gaining valuables in "Wyrmspan".
- **Bash the leader:** Is a phenomena present in some games where in order to win players have to prevent the victory of whoever is in advantage at the moment. This characteristic is most times exploitable by players by not acting against the current leader and instead trying to get closer to victory themselves. This forces others to sacrifice their possible victory and prioritize bashing. A game that most times features bashing is "Root".
- **Complicated vs Complex** A game is considered complicated if it has a steep learning curve but after learning the rules and the basics it is not difficult to master. The results of ones actions are predictable and immediate. An example for a complicated game is "Zombicide". Complex games on the other hand require critical thinking in order to achieve victory. Those games are hard to master and a difference in skill is easily noticeable. A good example could be "Go".

The proposed problem shares significant similarities with various aspect extractions/ sentiment analysis solutions. This made me believe that the problem could be reducible to the same task in another domain. For this reason for I re-implemented *Attention Based Aspect Extraction* (ABAE)[1] that was proposed for that very problem. Alongside ABAE tweaked versions of the *Latent Dirichlet Allocation* (LDA) were also studied.

Latent Dirichlet Allocation

LDA is a topic modelling method that has also been widely used for aspect extraction under the unsupervised learning framework. It is a probabilistic model in which documents are assumed to be generated by a mixture of topics. LDA does not directly classify documents but assigns a topic distribution to the input. Rather than predefined, these latent topics are inferred from the corpus that is given during the model generation. Words are not bound to a single class and can appear across multiple ones

[1] For a deeper insight into the domain definitions take a look at: Goblinpedia - La tana dei Goblin

with different probability. The topic distribution for each document is drawn from a Dirichlet prior from which it gets its name.

LDA has shown to be quite effective before the arrival of Transformers. A limit to overcome is that aspect extraction is more fine-grained than simple topic modelling. To tackle this problem I decided to explore two possible solutions:

- *Local-LDA*: A commonly used tweak on LDA in aspect extraction.
  We feed the model sentences so that the topic extraction is local.
- *NOUN-LDA*: In the opinion mining research it has been observed that the main holder of information when identifying aspects are nouns.
  I tried to apply this heuristic by generating LDA on a nouns only processed dataset.

Attention Based Aspect Extraction

ABAE is a neural approach model. Its overall structure can be described as:

- **Embeddings**: Generation of embeddings for each word in a *bag of words* input. The embeddings model we use is an implementation of Word2Vec and runs on the default parameters defined by the *gensim* library.
- **Attention**: Weight the embeddings in the sentence using an attention mechanism.
- **Autoencoder**: The weighted embeddings are passed to an auto-encoder that reduces the dimensional space to the target *aspect* size and reconstructs the attention weighted input.

The training objective of the model is to minimize the difference between the decoded sentence reconstruction and the originally calculated sentence embedding.

The trickiest part of the experiment is the lack of ground truth that brings us to an unsupervised learning framework. In order to overcome this problem we used some commonly used clustering metrics to be able to draw some sort of conclusion on which we take a deeper look in the coming section.

1.1 Experimental Setup and Development Environment

All the training procedures and notebooks were ran locally.

To make full use of the GPU power (*NVIDIA RTX 3070Ti*), dedicated drivers were needed. The project ran on CUDA 11.8. A PyTorch backend was used. This choice was pivoted by the fact that the technology is very popular among the research community. Libraries and other references are listed in the GitHub repository[2].

For reproducibility purposes all seeds are set in the code.

2 Research question and methodology

2.1 Dataset and pre-processing

BGG offers a simple yet effective API to scrap data from their platform. It lacks a direct method for listing boardgames, these have been taken from a generated dump.

To tackle possible issues of the raw data, various different pre-processing pipelines were designed by using modular processing components. For data constraints it is desirable to work on one language only but users do not have to explicit the language of their reviews. Thus, all pipelines share a filter on the language of the comments, removing all non english ones.

A good portion of the reviews on BGG do not actually give an insight on the required game aspects. Some focus strictly on the experience and quality of service of the product coming from the popular crowdfunding platforms (e.g. "Kickstarter"). To avoid having these low information records in the dataset the simple heuristic of filtering out reviews containing the related keywords (e.g. "ks", "pledge") was applied.

Reviews were tokenized and lemmatized thanks to a pre-trained POS tagger and processor provided by *spacy*. To further reduce redundant and undesired information the pre-processing pipeline maps game names and numbers to generic tags. Another step pipelines have in common is to filter out short review as we expect to hardly learn anything from them. Besides the "*default*" built pipeline two more were defined as:

- *NOUN*: Takes a spin on the default pipeline by filtering all words that are not recognized as nouns by the used tagger.
- *default-sentence*: Is a slight modification on the default pipeline in which at the start of the process reviews are split on sentences by a sentence splitter (*spacy*). These lines are considered entries of the dataset instead of the full comment.

After running a pipeline duplicates are discarded to avoid increasing bias.

2.2 Metrics

Not having a ground truth to estimate the real performance of the model on makes the pursuit of a strong metric for model evaluation crucial. As proposed when ABAE was presented [1] a metric that has been observed to relate well with human judgement is *topic coherence*, also known as "*umass*" *coherence* [3]:

$$C(t; V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})}$$

.

Where $D(v)$ is the document frequency and $D(v, v')$ the co-document frequency. The values of the metric lie in the interval $(-\infty, 0)$ where values closer to zero represent a better coherence. This is the only metric shared among different model architectures.

2.3 Developed models

The first trained models have been initialized on a "standard" set of hyperparameters. Hyperparameter optimization by random search followed. The best found configuration is then run on the full data and evaluated according to the selected metrics. The top words of each found aspect are extracted and mapped by hand the gold standards where possible. The mapping is then wrapped in a lookup class that maps the model outputs to the required aspects. Values belonging to the same aspect are aggregated together as it becomes more relevant.

2.3.1 LDA

LDA elaboration is not a computation heavy task for modern standards therefore experiments on the different preprocessing pipelines were performed with ease. During hypermarket tuning focus went to the most important one: the number of topics K. For LDA K-fold CV on each seen configuration could be applied.

Configurations were chosen based on the average coherence for some top-k words and the overall *perplexity*. The final model was trained on the full data.

2.3.2 ABAE

To train ABAE, as proposed by the original paper, the *contrastive max-margin* loss was used. During the training phase a sample of sentences that acts opposed to the input sentence is needed. This additional set of reviews are used to compute the loss after decoding is performed. We refer to these as *negative samples*. The loss is so defined as a *hinge loss* that maximizes the distance between the decoded embedding (r_s) and the averaged negative samples embeddings (n_i) while also minimizing the distance with the attention weighted sentence embedding (z_s). Being all values from the same *embeddings* space the distance is measured by dot product.

More strictly the *contrastive max-margin* loss:

$$J(\theta) = \sum_{s \in D} \sum_{i=1}^{m} \max(0, 1 - r_s z_s + r_s n_i)$$

The size of the negative samples for the loss computation was fixed to 20.

For the optimizer *adam* was chosen for two main reasons. The first one is that it was the optimizer used in the original ABAE proposal. Secondly, although *SGD* is as effective[4] and less bloated by variables, the gain in convergence speed is considerate.

The first run on the model was performed by using the default setting applied in the ABAE paper. Instead of K-fold CV classic CV was used. The tuned hyperparameters are: *learning rate, epochs, batch size, embedding size and aspect size*. The best found configuration is then run on the full data and evaluated according to our metrics. Comparison between ABAE instances was done on coherence and max-margin loss.

Inferred Aspect	Top relative words	Gold Aspect
Strategy/Depth	*strategy, time, mechanic, depth*	
Target/Difficulty	*puzzle, engine, adult, child*	Complex/Complicated
-	-	Downtime
Game Mechanics/Rulebook	*rule, placement,worker, rulebook, system*	Bookkeeping
Interaction	*player, interaction, turn, strategy, decision*	Interaction
Player track	*action, turn, player, opponent, victory*	Bash the leader
High/Low luck	*dice, roll, puzzle, euro, luck*	Luck
Various	...	Misc.

Table 1 Gold inferred aspects on the final NOUN-LDA ($K = 13$) model trained on the full data (310k). Interaction seems to overlap with downtime. The various mapped to "Misc" are not reported but can be looked up in the repository.

3 Experimental results

The first performed runs without tuning the hyperparameters gave us a broad idea of the unoptimized solution quality on the evaluated metrics.

Both LDA and ABAE only had a small boost in performance by tuning the hyperparameters. The best identified configurations were then used to make a final evaluation. The different models were compared where possible.

LDA

At first an additional processing pipeline was applied for LDA which splits sentences and filters them to have nouns only. The approach was dropped as the loss of information was too high and the generated model was not on par with the others.

Hyperparameter tuning was therefore done on the two datasets: *NOUN-only* and *sentence* with final best found $K = 7$ for both. As expected by the decreasing complexity of the dataset, the noun models perform better in terms of coherence. The NOUN model was unable to recognize some key requirements this probably given by the limited number of aspects of the final configuration.

The sentence model also resulted under-segmented and did not align with the requirements. For this reason a higher promising value from hyperparameter optimization of $K = 13$ was selected to see if the solution could be improved. The new model outperformed the best expected model in both *topic coherence* and *perplexity*.

For LDA the best processing pipeline in terms of result did not yield the most interpretable model in fact, the sentence one while performing worse on the measured metrics during human inspection it seemed to be more valuable.

ABAE

Experiments on ABAE were performed on both the noun and default generation pipelines. Initially they were done on a small subset of the dataset.

Hyperparameter tuning was performed on 20 different configurations. The best settings were chosen by trading off loss and coherence prioritizing lower coherence.

The best overall configuration was indeed the lowest in loss but, compared to the reduced dataset version, it performed way worse in coherence metrics. To further investigate the "80k" version of the base model was evaluated on the bigger test set.

Inferred Aspect	Top relative words	Gold Aspect
Strategy-Asymmetry Weight	*tactic, layer, tactical, strategic, asymetric weight, playtime, length, long*	Complex/Complicated
Frustration Analysis Paralysis	*tend, frustrating, annyoying, drag, problem decision, choice, planning, paralzsis*	Downtime
Game mechanisms Ruleset	*scenario, progression, app, ai rule, explain, teach, learn, rulset*	Bookkeeping
Cooperation	*cooperative, coop, party, family*	Interaction
Player blocking	*opponent, force, block, avoid*	Bash the leader
Cards/Dice	*card, flip, face, dice, randmolu*	Luck
Various	...	Misc.

Table 2 Gold inferred aspects on the final ABAE model trained on a subsample of the data (80k). The various mapped to "Misc" are not reported but can be looked up in the repository.

Model	$\overline{C}$	$\overline{C}_5$	$\overline{C'}_5$	l	Perplexity
ABAE	-12.62	-10.65	-10.16	3.98	/
ABAE-small	-6.72	-5.49	-3.63	4.06	/
NOUN-LDA($K = 13$)	-3.76	-2.34	-2.39	/	-6.99
sent-LDA($K = 13$)	-4.10	- 3.48	-3.48	/	-8.16

Table 3 Evaluation results. All evalauted on the same test setw ith C' being the coherence only in relevant aspects.

It could be supposed that, not only there was bias for some identified aspects, but that the extended data allowed the model to recognize less prominent patterns.

A last model on the optimal settings but trained on a downsized dataset was run. The mapping between identified aspects and gold ones are reported in table #2. Coherence is overall lower and the aspects seem to be stronger than the one identified by the full data model.

Unlike what expected by the ABAE paper LDA outperforms in coherence ABAE. By looking at the found aspects it seems like the neural model is better at capturing more complex relations. Despite giving a better expected performance the LDA models' mapped aspects were not as convincing during human evaluation and unable. They also had a hard time recognizing some required aspects. This could be related to the loss of contextual information that relates well to some aspects like downtime where most times it is referred as a frustrating activity often associated with a negative adjective value thing that is lost by the NOUN only approach.

Final models evaluation metrics of the dataset are reported in table 3.

4 Concluding remarks

The results do not compare well with initial expectations, as more complex solutions yield worse metric values. Some possible issues in the approach might have during data pre-processing. The pipelines could be too aggressive thus degrading the structure of the information ABAE requires.

If not the processing pipeline the real problem could be the dataset itself. Reviews might be too similar or generally unbalanced in topics. In fact BGG is known to be very biased towards complex games: of the top 20 ranked only 3 have a weight rating below 3 ("Monopoly" for comparison is 1.62)[2]. Based on this, we can expect the topic to gain more focus from the community.

Assuming that all gold aspects can be inferred by the scrapped reviews is a required assumption by the approach, but it could not be the reality. To improve the overall dataset quality additional information given by BGG should have been exploited. The complexity rating could have been a possible filter to retrieve a list of games from which to draw comments that we could suppose to be more likely to cite the "Complicated/Complex" aspect.

Next steps could involte exploiting the found models to filter out reviews that too harshly rely on identified aspects that do not fit the requirements, e.g. "Game components". This way we would be building a more refined dataset that might be used achieve better identification on a completely new model.

A possible note on the low performance of ABAE is the low range of aspect values during the hyperparameter tuning process. The range was set to $[7, 20]$ but unlike what supposed, it seems that the data is way more complex in our project than the one used by other applications involving ABAE. By increasing the number of aspects there a better separation of topics while also increasing overall coherence can be observed. This would also explain why the 80k ABAE performs better: some patterns that are recognizable in the larger dataset are not distinguishable enough in the smaller one.

In the end it cannot be truly stated if the lower coherence in ABAE models results in worse aspect identifications without a real test set.

[2] Ranking was inspected as for 03/2025

References

[1] He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An unsupervised neural attention model for aspect extraction. In: Barzilay, R., Kan, M.-Y. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 388–397. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1036 . https://aclanthology.org/P17-1036/

[2] Fichera, J.: What You Like in Boardgames. https://github.com/Ubriacopo/boardgames-aspect-extraction

[3] Mimno, D., Wallach, H., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Barzilay, R., Johnson, M. (eds.) Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 262–272. Association for Computational Linguistics, Edinburgh, Scotland, UK. (2011). https://aclanthology.org/D11-1024/

[4] Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B.: The Marginal Value of Adaptive Gradient Methods in Machine Learning (2018)