Classification based on aspect extraction

Lukas Hirsch[1,2*]

Corresponding author(s). E-mail(s): lukas.hirsch@st.oth-regensburg.de;

Abstract

The abstract serves both as a general introduction to the topic and as a brief, non-technical summary of the main results and their implications. Authors are advised to check the author instructions for the journal they are submitting to for word limits and if structural elements like subheadings, citations, or equations are permitted.

1 Introduction

In the field of *natural language processing (NLP)*, the ability to parse and understand text at a semantic level is critical for a wide range of applications, from sentiment analysis to common sense reasoning. Semantic parsing, the process of decomposing text into multi-word concepts, has emerged as a key technique for improving the understanding and interpretation of textual data. It can be performed by exploiting phrase structure grammars or statistically using training-based algorithms [1]. Training-based algorithms use machine learning techniques to model and parse text based on statistical patterns learned from large data sets. These algorithms can range from classical machine learning methods to advanced deep learning architectures such as recently transformer models like BERT (Bidirectional Encoder Representations from Transformers) [2].
This project aims to use semantic parsing to classify comments on board games into predefined categories, using a graph-based technique in combination with the embedding of the multilingual knowledge graph *ConceptNet* [3]. To evaluate the effectiveness of the proposed approach, a comparison with the latest model of OpenAI GPT-4o is made.

2 Dataset

The dataset used is provided by *BoardGameGeek (BGG)* [4], a comprehensive database of board games and the global community of gamers. Users contribute data, including statistics and ratings, which assess the popularity of each game based on several criteria. These criteria include an overall rating, the number of users who voted for the game, and the community's opinion of the game's playability with different numbers of players. In addition, BGG allows access to user comments on games, which is the only part used for this project. Data was retrieved using the BGG API.

3 Methodology

The primary goal of this project is to classify comments on board games into predefined categories using semantic parsing techniques. The predefined categories are based on the Italian definition of Goblinpedia:

1. **luck or alea:** all those game elements independent of player intervention, introduced by game mechanics outside the control of the players.
2. **bookkeeping**: manual recording of data and potentially automatic or semi-automatic game processes, including also the need of continuosly accessing the rulebook for reference.
3. **downtime**: unproductive waiting time between one player turn and the next. By unproductive we mean not only having nothing (or little) to do, but also nothing (or little) to think about.
4. **interaction**: the degree of influence that one player's actions have on the actions of the other participants.
5. **bash the leader**: when, to prevent the victory of whoever is first, the players are forced to take actions against him, often to the detriment of their own advantage or in any case without gaining anything directly. At the table, the unfortunate situation can arise whereby one or more must "sacrifice" themselves to curb the leader and let the others benefit from this conduct.
6. **complicated** vs complex: A game is complicated the more the rules are quantitatively many and qualitatively equipped with exceptions. Once you understand and learn all the variables, a game (that is only) complicated is not difficult to master. In a complicated game, solving a problem leads to immediate, certain and predictable results.
   A game is as complex as the repercussions of one's actions are difficult to predict and master. Even once you understand and learn all the variables, a complex game is still difficult to master. In a complex game, solving one problem leads to other problems.

To achieve this goal, the process begins by parsing the comments to identify their structure, including the different sentences, which are then further parsed to identify the verbs and nouns. Verbs are essential because they typically denote actions or events. Each verb is analyzed in its base form, known as lemmatization, which helps to understand the core action being described. Noun phrases within the sentence are also identified. These phrases often serve as the subject or object associated

with the verbs. By linking these noun phrases to their corresponding verbs, potential event concepts are formed. In addition, any adjectives within the noun phrases are included to provide more context for the concept. In cases where noun phrases are related to auxiliary verbs, which are verbs used in conjunction with a main verb to express tense, mood, or voice, the code accounts for this by forming concepts based on the structure of the auxiliary verb. The overall goal is to create a list of unique concepts that encapsulate the events described in the sentence. These concepts are then compiled into a comprehensive list after processing each sentence individually. In this way, the slightly modified method from [1], ensures that all potential event concepts are captured and organized effectively, providing a structured representation of the information contained in the text.

To further simplify the extracted concepts, pre-trained word embeddings from *ConceptNet* are used. These embeddings provide a numerical representation of words, capturing semantic similarities based on their contextual usage in a large corpus of text. To represent complex concepts, which are often multi-word expressions, an embedding is computed for each concept by averaging the embeddings of the individual words that make up the concept. This process ensures that each concept is represented as a single, coherent vector in the embedding space. Concepts without valid embeddings (i.e., those not found in the pre-trained embeddings) are excluded from further analysis to maintain the integrity of the representation. The next step is to cluster these concept embeddings to identify groups of semantically related concepts. This is done using the *K-Medoids* algorithm [5], a clustering technique particularly suited to scenarios where representative examples (medoids) from the dataset are preferable to mean-based representatives (centroids). The number of clusters is five, with a dynamic determination when the input number of concepts is less than five, ensuring a meaningful and manageable grouping. The clustered concepts are then used for further processing.

To classify the concepts into the given classes, an embedding is created by averaging the embeddings of the words that make up the concept. Similarly, each class is represented by an embedding computed from the keywords associated with that class. To determine the similarity between a concept and a class, cosine similarity is used. If the embedding of a concept is sufficiently similar to the embedding of a class (exceeding a specified threshold), the class is considered a potential match for the concept. The classes are then ranked based on their similarity to the concept, and the top matches are identified. For each concept, the class with the highest similarity score is selected, ensuring that each concept is assigned to the class that best represents its meaning.

4 Results

5 Conclusion

References

[1] Rajagopal, D., Cambria, E., Olsher, D., Kwok, K.: A graph-based approach to commonsense concept extraction and semantic similarity detection. In: Proceedings of the 22nd International Conference on World Wide Web. WWW '13 Companion, pp. 565–570. Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2487788.2487995 . https://doi.org/10.1145/2487788.2487995

[2] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018) 1810.04805

[3] Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. CoRR **abs/1612.03975** (2016) 1612.03975

[4] BoardGameGeek: BoardGameGeek: The BGG API. Accessed: 2024-07-21 (2000). https://www.boardgamegeek.com

[5] Park, H.-S., Jun, C.-H.: A simple and fast algorithm for k-medoids clustering. Expert Systems with Applications **36**(2, Part 2), 3336–3341 (2009) https://doi.org/10.1016/j.eswa.2008.01.039