War Game Complexity Comparison
for Coin Series

Junqian Zhang

University of Milan
Data Science and Economics

Abstract. This project aims at comparing the complexity of war games from COIN series by exploring their rule setting. The comparison is based on the features extracted from general war game category, under the assumption that the more complex COIN series game covers more features and has more rules on one feature. The final result is reflected on the complexity index.

Keywords: COIN Series · War Game · Aspect Extraction · Information Retrieval.

1 Introduction

War games are popular throughout the world and it is interesting to study the logic behind. This project starts the exploration with COIN series as target. This series feature Volko Ruhnke's game system presenting guerrilla warfare, and COunterINsurgencies round the world - in both historical and contemporary, so the rule structures under this series are similar and can be an easy beginning for analyzing.

This work tries to compare the complexity of Coin Series games by means of their rule books. The main idea of this work is that the more complex the game, the more features it renders in rule design and the more rules it sets on one single feature. To gain these features and be more clear about the war game rule features, features are extracted from general war game category, assuming that the game rule is composed of settings on different features and all war games, no matter coin series or other war games, share the same rule setting features. What's more, in order to give a clear comparison result, a complex index is employed.

2 Problem Statement and Methodology

The main two questions here are to figure out the main features in general war games and search the corresponding rules in COIN series games. To extract the features, aspect-based analysis is used in the rules of war games to gain the queries in searching. And using semantic meaning for word vectorization, cosine similarity can be adopted for query searching. Finally, a complexity index can be generated from all the information got from the previous steps.

2.1 Text Cleaning

The main analyzed target in this project is official rule books of each game. For the following steps, all the description are cleaned based on sentence-level. All the sentences are tokenized and words in the sentences are executed lemmatization to get uni-grams by **Part-of-Speech** Tagging, but, at the same time, stop words are not abandoned.

2.2 Query Extraction

Under the assumption that the game rule is composed of settings on different features and these features are shared by the games of the same category. This target of this step is to find these features, i.e., aspects. These aspects are also called *queries* here because they are used as key words for retrieving the related rules in the rule books too. This project focus only on nouns and noun phrases appearing in the text document. All the noun phrases are assumed to be composed of two words, i.e., all the noun phrase are bi-grams.

Based on the uni-grams, bi-grams are built at sentence-level to find candidate noun phrases. This project assumes that the candidate noun phrases should be in the pattern of two nouns, or stop word plus noun or adjective plus noun. So, to efficiently get the bi-grams, bi-grams without nouns inside are abandoned. To obtain the valid noun phrases and rank them, I calculate Pointwise Mutual Information (**PMI**) scores for all the bi-grams. PMI values take into account of the correlation between the two words inside the noun phrase, avoiding the case where the noun phrases in fact are the aspects of the aspects. 50 bi-grams with highest PMI scores are selected but the meaningless ones among them are abandoned.

Except these selected noun phrases, all the other bi-grams are splitted into uni-grams again and corpus is re-cleaned the corpus by removing the stop words. With description as documents, I calculate the term frequency–inverse document frequency (**TF-IDF**) for all the nouns and candidate noun phrases. 20 features with highest TF-IDF values are selected as aspects.

2.3 Query Expansion

To improve the information retrieval performance in the searching later, it is necessary to do query expansion, so synonyms and semantically related words or phrases should be used together with queries found in the last step.

Synonyms and related expressions are connected with queries on the basis of context. Thus, **Word2Vec model** is trained to represent all the uni-grams and bi-grams as vectors, and consequently it is possible to compare the context between each other. These vectors can be seen as a description of the context of each element in the vocabulary. **Skip-gram model** is used to get the input

for the word embedding representations to describe the context. Through skip-gram, all the sentences in all the comments are organized into sequences. The length of context window is 3 and each gram concerns the 2 grams surrounded as neighbours.

Since all the words and phrases are in a numeric form, the cosine similarity distance between them can be calculated to see the difference of their contexts. With the word embedding model, for each aspect extracted in the last step, I take the top 10 words or phrases that are closest to it, which are regarded as different aspect expressions. All the expanded queries found in the step 2.3 are replaced by the corresponding query in the sentence.

2.4 Query Retrieval

The logic behind the query retrieval in COIN series is that I calculate the absolute cosine similarity between each word in rule books and the query, and therefore the sentences with words of high cosine similarity are the *related rules*. To get the cosine similarity, a new **Word2Vec model** is retrained to represent all the words and phrases of both COIN series and general war games as vectors and therefore these words and phrases are able to compare to each other.

The related rule sentences to one query here are the ones have at least one word or phrase who have absolute cosine similarity with the query higher than **0.5**.

2.5 Rule Comparison

According to the two criteria, which is covering more features is more complex and setting more rules in one feature is more complex, I count the related rules of each game under each query. Concerning that some games have very small number of rules, nearly no related to one feature, I replace the number smaller than **10** with zero for eliminating the noise and count the queries that one game.

So we now can get two new variables for each game g, which are the number of features one game covers, and the number of rules that relates to all the queries Q. The final complexity is defined as:

$$ComplexityIndex(g) = \frac{\frac{\sum_{q}^{Q} RelatedRules_{q,g}}{100} \times \frac{\sum_{q}^{Q} \mathbb{1}_{RelatedRules_{q,g} \neq 0}}{10}}{1.5} \tag{1}$$

where $RelatedRules_{q,g}$ represents the number the rules of game g related to query q, and 100, 10 and 1.5 are used for limiting the range of index value.

3 Experimental Result

3.1 Data Description

The text employed in this analysis are official rule books of COIN series and war games. Now **10** games from COIN series and their rule books are available on the website of GMT. The general war games are from the top 400 war games list on BoardGameGeek (BGG) and **147** rule books of them are collected.

3.2 Results and Evaluation

After text cleaning, query extraction and query expansion implemented in general war games, the **26** aspects shared by the category and corresponding expanded queries are listed in **Table 1**. From the table, we can see that some aspects are the expanded query of other aspects. This is because the aspects are extracted according the occurrences while the query expansion is based on semantic meanings. So for the purpose of better query retrieval, if an aspect appears in the Expanded query of the other query, then I combine the two aspects into one category. After combination, we have **18** aspects now, which are *area*, *attack*, *attacker*, *command*, *control*, *die*, *end*, *enemy*, *force*, *infantry*, *leader*, *map*, *movement*, *point*, *retreat*, *side*, *supply* and *target*. One thing needs to notice is that the left aspects do not only represent the meaning as the words, but represent a whole category of words.

Searching 18 queries in the COIN series, I get the **Fig.1**, which depicts the number of rules of each game related to the queries. Apparently, *Area*, *Point* and *Movement* are the three features have no relation with each game. According to the complexity criteria, **Pendragon** must be the most complex one while **A Distant Plain** is the simplest one. The left part of the games in the heatmap is manifestly more complex than the right part since they covers more features and have more related rules.

Fig.2 shows both complexity index generated by this work and the complexity from BGG stats. And to get a clear idea of the result in **Fig.2**, **Table 2** presents the ranking list according to both indices. To compare the relationship between two ranking lists, **Rank-biased Overlap** (RBO) is employed to calculate the similarity between the two, giving back a numeric value between zero and one. The similarity returned is **0.76**, indicating a high correlation, since a RBO of one means completely identical.

4 Conclusion Remarks

To explore COIN series, this work firstly extracts the common features of war games from other general war games. And the extracted features are reasonable, and are true aspects that game designers concern during rule setting. But

Query	Expanded Query
Area	hex, location, region, territory, zone, space, country, uncontested, duchy, unoccupied
Army	corp, corps, armies, leven, homeland, longstreet, subordinate, fe, brigade, liège
Attack	tack, df, airstrike, assaulting, defend, combat, fense, attacking, tacking, strafe
Attacker	tacker, defender, the defender, the attacker, fender, defending, loser, bouncer, shifts, forego
Battle	bat, combat, disaster, siege, aftermath, loser, winning, napoleon, audacity, skirmish
Combat	bat, antiaircraft, dmr, bombardment, df, attack, halving, ranged, overruns, airdrops
Command	mand, commanding, plt, subordinate, oc, pw, activation, activate, mands, activated
Control	trol, troll, recapture, controlled, trolled, trols, pied, capture, conquered, ling
Defender	attacker, tacker, the defender, the attacker, defending, fender, defend, loser, dt, beneficial
Die	die roll, dice, roll, unmodified, reroll, modified, a die, sar, rnc, severity
End	conclude, end of, conclusion, begin, finish, strat, earlier, continue, impulses, armistice
Enemy	emy, enemy unit, opposing, detected, friendly, zoi, evading, fzs, an enemy, retreating
Force	tfs, coordinating, merchantman, forced, fleet, nas, army, carrier, expedition, hiryu
Hex	hexes, orchard, uncontested, connecting, containing, hexsides, polder, hexside, area, vacant
Infantry	fantry, infantry unit, cuirassier, bicycle, jaeger, panzergrenadier, mech, inf, cav, paratroope
Leader	brock, mander, samurai, cu, tecumseh, subordinate, cornwallis, hanno, commander, crusader
Map	superimpose, mapboards, grid, compass, board, edge, hexagon, regulate, convenient, mapsheet
Move	reposition, enter, vated, travel, activated, activate, exit, drop, redeploy, moving
Movement	ment, movement phase, redeployment, mp, zoi, mf, entrain, prohibition, exploitation, tegic
Point	bps, xps, ep, dep, unspent, eps, xp, pp, gop, fixed
Retreat	overstacking, overstack, dmr, vacate, aac, elimination, survivor, involuntarily,noverwhelm, overstacked
Roll	unmodified, a die, modified, die, sar, die roll, modifies, consult, dr, rolling
Side	flip, villeneuve, flipped, backside, committed, er, debater, reverse, undetected, asu
Space	region, unbesieged, ravage, duchy, territory, unoccupied, provincial, province, uncontrolled, argo
Supply	ply, overland, isolate, continuous, isolated, communication, oos, tracing, unsupplied, comm
Target	incendiary, firer, the target, firing, targeting, strafe, strafing, detected, sam, targeted

Table 1: Aspects of War Game Rules

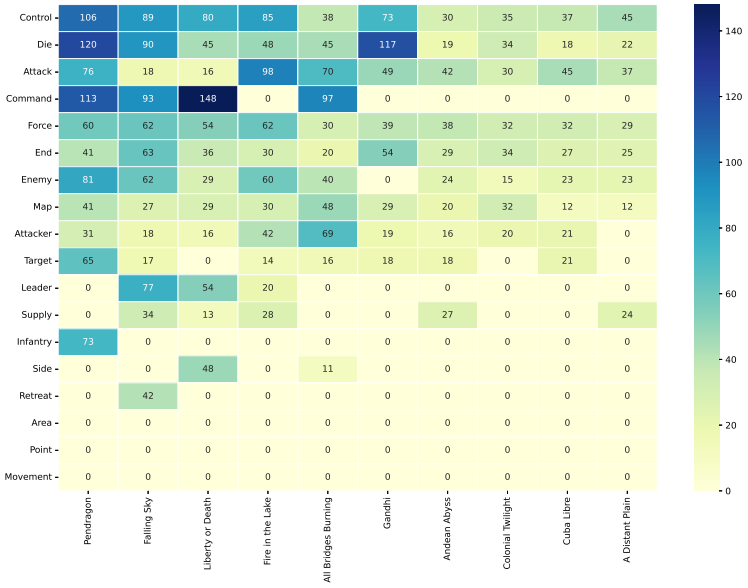

Fig. 1: Game Rule Summary

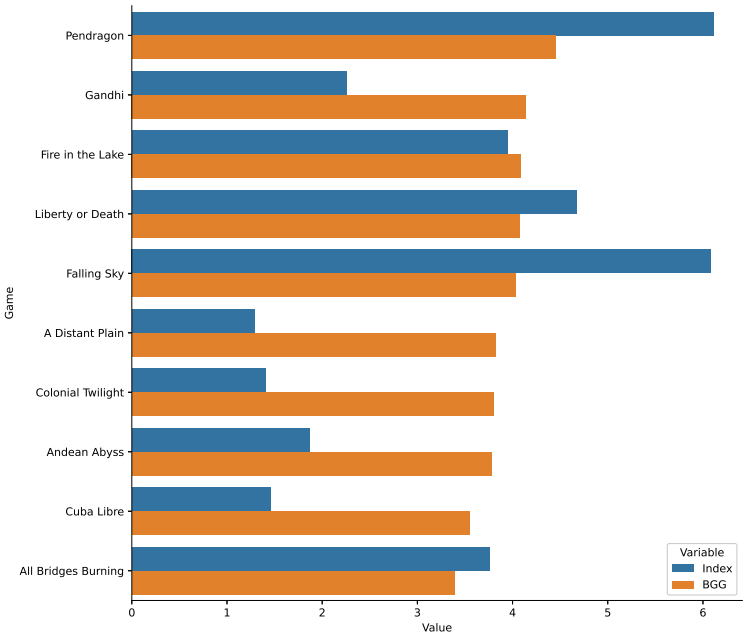

Fig. 2: Comparison between Complexity Index and BGG Complexity Stats

Table 2: Ranking Comparison

	Index Ranking	**BGG Ranking**
Pendragon	1	1
Gandhi	6	2
Fire in the Lake	4	3
Liberty or Death	3	4
Falling Sky	2	5
A Distant Plain	10	6
Colonial Twilight	9	7
Andean Abyss	7	8
Cuba Libre	8	9
All Bridges Burning	5	10

these features are not perfectly the ones in COIN series. Under the assumption that the more complex games will cover more features and set more rules on features, the complexity ranking does correlate to the real feelings from the players.

Although the result is not bad so far, some work has not been finished. This work does not explain the content of related rules and the conflicts described in the rule books, and how these bring the complexity to the games.

References

1. Rahman, N., Borah, B. "Improvement of query-based text summarization using word sense disambiguation". *Complex Intell. Syst.* 6, 75–85 (2020)
2. Webber, William, Alistair Moffat, and Justin Zobel. "A similarity measure for indefinite rankings." ACM Transactions on Information Systems (TOIS) 28.4 (2010): 20