Steam Game Data Search Engine: Data Preparation

Bruno Mendes
up201906166@edu.fe.up.pt
M.EIC
Faculty of Engineering of the University of Porto
Portugal

Fernando Rego
up201905951@edu.fe.up.pt
M.EIC
Faculty of Engineering of the University of Porto
Portugal

Joel Fernandes
up201904977@edu.fe.up.pt
M.EIC
Faculty of Engineering of the University of Porto
Portugal

ABSTRACT

Steam is a video game digital distribution service and storefront by Valve. It is composed of a large game library from diverse genres and categories, which are often hard to find. This project aims to prepare and process data collected from publicly available *Steam API's* and *steamspy.com*, to fulfill the information needs of a user looking for a new game. For this milestone, we have prepared the dataset for later processing through cleanup and feature engineering tasks, and identified possible information needs that we will be able to fulfill after analyzing the data in hands.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Structured text search**.

KEYWORDS

datasets, information processing, information retrieval, full-text search, steam, search engine

1 INTRODUCTION

According to Steam's website [1], Steam is the ultimate destination for playing, discussing, and creating games. In other terms, it is a large library for discovering and managing games. It is expected that users find new games with ease, even if they do not know their name.

For that purpose, our goal is to develop a search engine starting with data retrieved from the publicly available *Steam's API's* and *steamspy.com* and processing it to the point it is refined, properly indexed and ready to be queried at later stages.

2 DATA SOURCE - IDENTIFICATION AND CHARACTERIZATION

The dataset used in this project was obtained from the Data World [6], where a enormous variety of open data is collected, and includes data from a total of 13357 games in a CSV format file with a size of 52.6 MB. The dataset was generated and made available open-source, with MIT License, by CraigKelly [3] combining public data from Steam API [5] and Steam Spy [4].

Due to the lack of text fields, additional description data from each game queried by title was taken from the Wikipedia API [2] using a web client developed by the team. The textual data was appended (in the form of a new column named *WikiData*) to the *original* csv file.

After an initial phase of data analysis, the team came across 149 duplicated games, fields with high rate of missing values as we show in the table below [Table 1], some fields that can be simplified into one and irrelevant or inconsistent attributes as we show in the graph below [Figure 1].

Field	Missing	%
ReleaseDate	87	0.65
PriceCurrency	2618	19.6
SupportEmail	3518	26.34
SupportEmail	5204	38.96
AboutText	662	4.96
Background	701	5.25
ShortDescrip	1862	13.94
PromotionalDescription	658	4.93
DRMNotice	13272	99.36
ExtUserAcctNotice	13202	98.84
LegalNotice	7812	58.49
Reviews	10043	75.19
SupportedLanguages	35	0.26
Website	3268	24.47
PCMinReqsText	728	5.45
PCRecReqsText	7555	56.56
LinuxMinReqsText	10293	77.06
LinuxRecReqsText	12151	90.97
MacMinReqsText	8725	65.32
MacRecReqsText	11619	86.99

Table 1: Missing values per field in the original dataset.

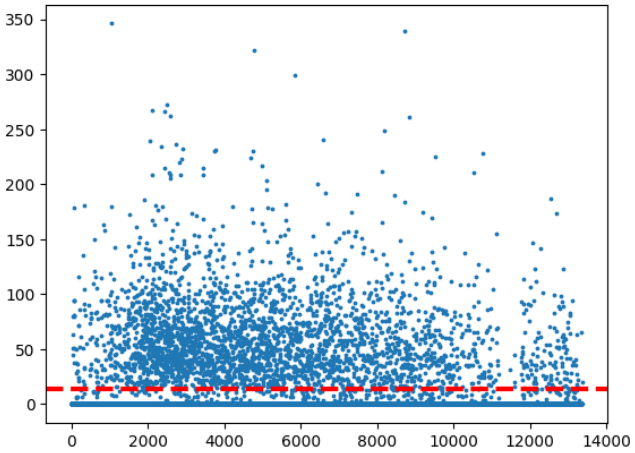

Figure 1: Number of words per review. Average of 14.2 words.

3 DATASET PREPARATION

3.1 Data cleaning

3.1.1 Duplicate removal. The first step of the applied pipeline consists in the removal of duplicates by *ResponseID*, which are unique to each game.

3.1.2 Column removal. Based on the number of missing and inconsistent values analysed before, we took measures:

- We deleted the columns *QueryID* and *QueryName* since they are repeated as *ResponseID* and *ResponseName*. We also deleted repeated rows with the same *ResponseID*.
- We deleted some counters that we found irrelevant to our future engine, such as *AchievementHighlightedCount*, *ScreenshotCount*, *MovieCount* and *PackageCount*. These are very specific information about a game that a user will most likely not know and/or not care about when looking for a new game.
- We deleted some flags, *PCReqsHaveMin*, *PCReqsHaveRec*, *LinuxReqsHaveMin*, *LinuxReqsHaveRec*, *MacReqsHaveMin* and *MacReqsHaveRec*, which are easily derived from the length of the requirements for each of these platforms, which exist a couple of columns to the right.
- Since there were some columns that included links, we deleted those as well - *SupportEmail*, *SupportURL* and *Background*. There were some columns with legal information *indirectly* related to the game itself: *LegalNotice*, *DRMNotice* and *ExtUserAcctNotice*. We got those removed, too.
- We deleted the column that contained the *PriceCurrency* the game was sold at since this value is the same in all rows.
- Redundant data related to the games description and details - *AboutText*, *ShortDescrip* - were also removed since these are overshadowed by the more complete *PromotionalDescription*, which got renamed to *PromotionalDescription* to accomodate the *WikiData* longer description and/or story.
- We removed variance data related to the number of players and owners the game has (*SteamSpyOwnersVariance* and *SteamSpyPlayersVariance*), which would be tailored for a predictive model in a data mining project but not quite for text-based retrievals.
- We removed the *Reviews* column due the complexity required to parse it and extract relevant information from it: most rows had it empty and the ones that did not were messy, with grammar mistakes and low quality input.

3.2 Feature engineering

We started by grouping categories, features and genres. These values where displayed as boolean mask columns in the original dataset, which is not great when it comes to memory usage. Due to this we grouped them and created a new column for each of these topics that includes a list of the value that are True.
Each game has (potentially) different text and voice languages. We converted the *SupportedLanguages* column into two, representing the two means of communication with the end user.

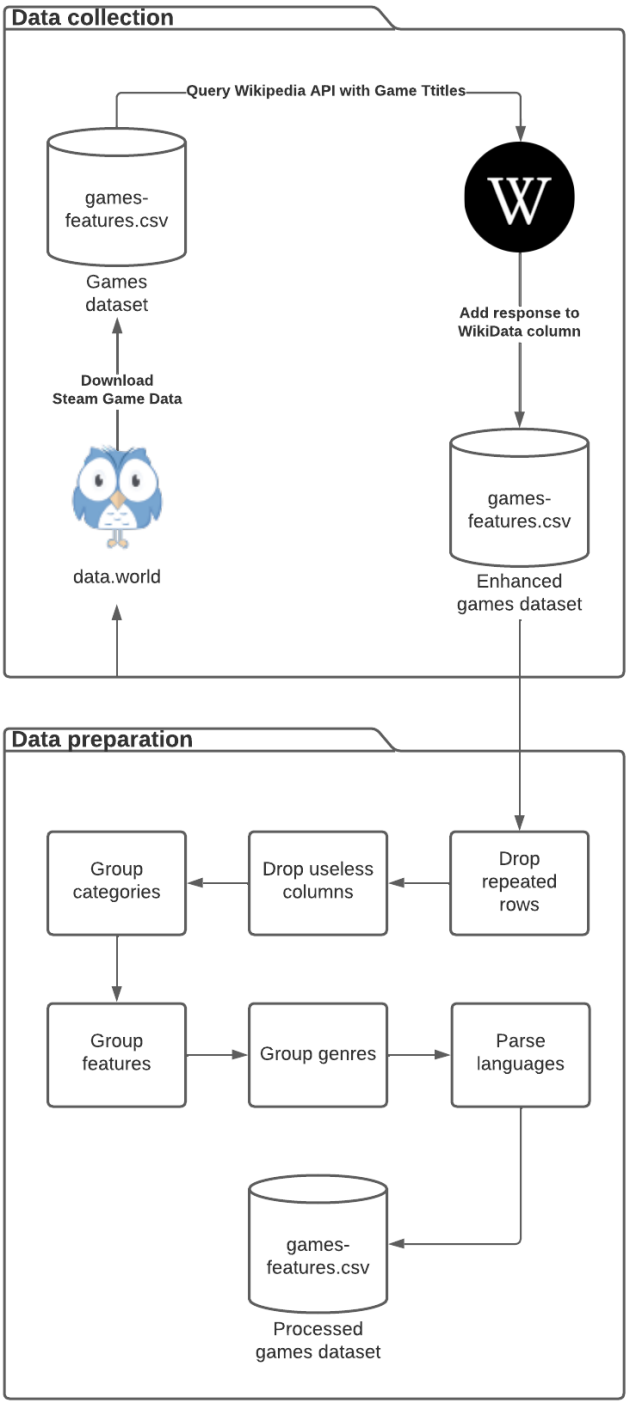

Figure 2: Pipeline datagram.

4 DATASET CHARACTERIZATION

After completing all steps of data preparation, we obtained a dataset `processed.csv` with 13 208 entries and a size of 29.9 MB. It has the fields described in Table 2. The class diagram for the dataset is presented in Figure 3.

Table 2: Fields of the final dataset.

Field	Type	Description
Name	Text	Game name
Release Date	Date	Game release date
Promotional Description	Text	Description of the game
Wiki Data	Text	Wiki data retrieved from the Wikipedia API [2]
Categories	List	List of categories related to the game
Genres	List	List of genres related to the game
Is free	Boolean	True if the game is free, False otherwise
Initial and Final Price	Real	Initial and final price of the game in USD
Free Version Available	Boolean	True if the there is a free version of the game
Platforms	Boolean	Platforms available for the game (Windows, Linux and Mac)
PC Requirements	Text	PC Requirements for each platform (Windows, Linux and Mac)
Required Age	Integer	Minimum required age
Game Players and Owners	Integer	Number of game owners and number of people who have played the game retrieved by Steam Spy [4]
Metacritic	Integer	Game metacritic score
Recommendation Count	Integer	Number of game recommendations
Counters	Integer	Counters of game DLCs, game Demos, developers and publishers
Achievements Count	Integer	Number of game achievements
Controller Support	Boolean	True if the game supports controller, False otherwise
Website	Text	URL to the official website of the game
Header Image	Text	URL to the header image of the game
Supported Languages	Text	Supported game languages and audio languages

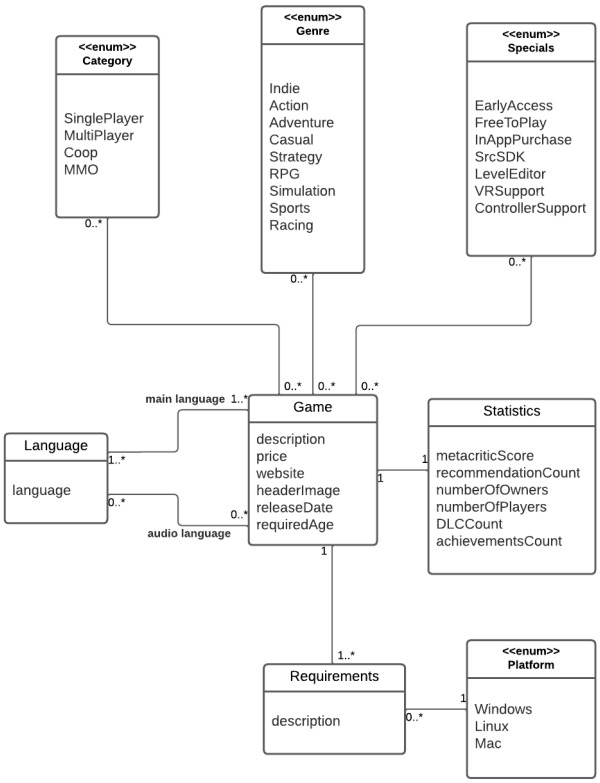

Figure 3: Class diagram for the conceptual model.

5 DATA EXPLORATION

The processed dataset is more consistent and dense when it comes to the presence of data. About the latter, we should have in mind:

- Most games specify the minimum requirements without higher, recommended ones. This is to be expected, since most modern games already set a fairly high minimum spec requirements for smooth operation.
- Linux and Mac requirements are missing much more than the Windows counterparts: this is also expected, since most games are Windows-only.
- We do not find the Wikipedia API result eligible if it is a disambiguation page (we can't really tell what is the matching entry) or if the page is not categorized as a game page (to remove the possibility of a name being titled after a person or any entity with a Wikipedia page of their own). There is also the obvious possibility that Wikipedia does not have information about the queried game: in that case, the search engine should fallback to Steam's promotional description.

Table 3: Missing values per field in the processed dataset.

Field	Missing	%
ReleaseDate	87	0.66
PromotionalDescription	658	4.98
Website	3232	24.47
PCMinReqsText	728	5.51
PCRecReqsText	7444	56.36
LinuxMinReqsText	10157	76.9
LinuxRecReqsText	12008	90.91
MacMinReqsText	8611	65.2
MacRecReqsText	11481	86.92
WikiData	10946	82.87

There are also some interesting observations that will be of great importance later on for deciding on their importance for the documents rankings. We describe them in the following subsections.

5.1 Text analysis

Figure 4: Word cloud for Steam's promotional descriptions.

Words such as *game*, *play* and *player* are, as expected, part of the most used vocabulary for our text fields. Without knowing, at this stage, which algorithms will come into play for keyword matching, we can already prospect that our system should not rank documents too much high based on the appearance of frequent words.

5.2 Release date evolution

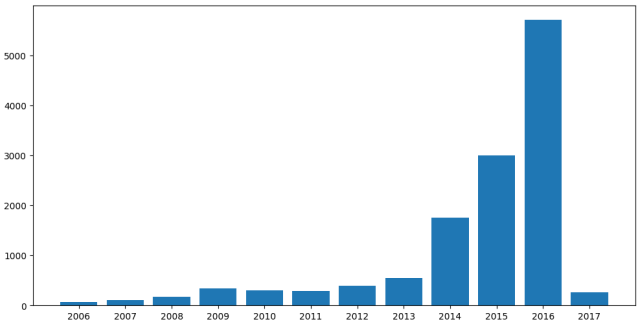

Figure 5: Number of games released per year.

We have noticed that most games in the dataset are released near the date in which the Steam API was queried. That is probably related to cleanups of unpopular old games in Steam's database; it is definitely something to keep in mind when retrieving old games in the future.

5.3 *Metacritic* score evolution

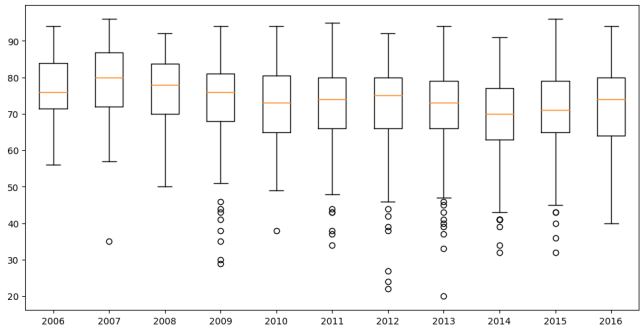

Figure 6: Game critics scores by year.

Most of the games are received in a average way by the critics, which should make finding best games a feasible task. The scores average seems to be lowering slightly over the years, which can be explained by the already explained theory of old unpopular game removal or by a lack of creativity by game creators.

6 INFORMATION NEEDS

At a later stage of the project, we expect to fulfill the need for a certain kind of information, such as:

- Games with most players
- Recently released games
- Highly praised games
- Multiplayer games
- Action games
- Linux-compatible games
- Games from the *FIFA* series
- Games from 2015 with more than 10000 players
- Free games with controller support

7 CONCLUSION

After cleaning up the dataset and extracting the most relevant features, our data processing pipeline returns a robust set of documents with plenty of text and a set of concepts that will allow a later search engine to flourish.

REFERENCES

[1] Valve Corporation. 2022. Steam: About. https://store.steampowered.com/about/
[2] Wikimedia Foundation. 2022. Wikipedia API. https://steamcommunity.com/dev
[3] Craig Kelly. 2016. Steam Game Data. https://data.world/craigkelly/steam-game-data
[4] Steam Spy. 2022. Steam Spy: About. https://steamspy.com/about
[5] Steam. 2022. Steam API. https://steamcommunity.com/dev
[6] Data World. 2016. Data World. https://data.world