Professional Videogame Reviews: Information Processing and Retrieval System

Gonçalo de Abreu Matias
Faculty of Engineering - University of Porto
up202108703@up.pt

Henrique Filipe Pereira da Silva Caridade
Faculty of Engineering - University of Porto
up202108817@up.pt

Pedro Miguel Marques da Silva Pinto
Faculty of Engineering - University of Porto
up202108826@up.pt

Abstract

With this project, we are addressing the need for an efficient system that allows users to quickly search and check reviews about potential game they might be interested in buying for example, or only searching for a second opinion about a game they played. With this article, we are documenting the methods we used to implement the Professional Video Games Reviews search engine. We used IGN's professional reviews dataset containing reviews since 1998 until today. We then used Solr for indexing and retrieval. We found some really interesting data during the course of this project, such as the fact that per year, in average 400 reviews are published, a big number considering that many people only play a few number of games. Another interesting fact is that scores haven't stopped increasing, thanks to the progress in the gaming industry.

Keywords

Video Games, Opinions, Gamers, Professional, Reviews, Information Retrieval, Dataset, Data Collection, Data Analysis, Search Engine, Review Aggregation, Professional Criticism, User Experience,Platform Comparison

ACM Reference Format:
Gonçalo de Abreu Matias, Henrique Filipe Pereira da Silva Caridade, and Pedro Miguel Marques da Silva Pinto. 2024. Professional Videogame Reviews: Information Processing and Retrieval System. In *Proceedings of PRI (G65).* ACM, New York, NY, USA, 11 pages.

1 Introduction

In recent years, the video game industry has revolutionized the way we live, and the way we play. In order for people to be able to quickly have a opinion about some game, appeared the professional reviews, that have a big influence and can even impact sales. While growing up, gaming has always been part of our lives, and that is why we think we couldn't choose any other topic for this project. We are aware it is a really rich and diverse industry, with vast amounts of data to explore, and we think it would be a nice addition and a very useful tool for people searching for gaming reviews online. That is why we chose to theme to develop this search engine for the Master's in Informatics and Computing Engineering (PRI - MEIC - FEUP). We are dividing this report into multiple sections, to provide a clear idea of what we are implementing. We start by describing why we chose this dataset, then we describe how we went about getting the dataset and the difficulties we encountered along the way. After that, we show the characteristics of the final dataset and finish by proposing some search scenarios we expect to be able to provide for our users.

G65, October, 2024, Porto, Portugal (FEUP)

ACM ISBN 978-1-4503-XXXX-X/18/06

2 Data Preparation

The first thing we were tasked with was to get a dataset with unstructured data and a pipeline that allowed us to extract more if need be, which we would then use in later stages of the project. This process varies wildly from topic to topic, but in our case, we didn't find any good existing datasets, so we decided to scrape from a website with game reviews.

2.1 Data Selection

After we decided that we wanted to scrape a website for our dataset we had to decide which website it would be. As there are many game review websites we searched through many of them to find one we were happy with but in the end, we landed on IGN [8]. IGN [8] is a well-known website that has vast experience with video game reviews. It is one of the oldest in the business of video game reviews having reviews dating as far back as 1996 so after we found it, it was a no-brainer to have IGN [8] as our source of game reviews.

3 Data Pipeline

The problem with IGN [8] is that it doesn't have an API and doesn't have a place where information about every review is displayed. The only way to get the review's contents is by the review's URL, and to add insult to injury, it doesn't have a place to get all of the reviews' URLs.

The full pipeline as described in more detail ahead is illustrated in Figure 10.

3.1 Selenium Extraction

With that said, the first step in our pipeline is to get as many review URLs as possible. The best way we found to do this is with Selenium [7] because IGN's [8] game review page uses a technique known as Infinite Scrolling which means the user needs to scroll before the website loads more data, Selenium [7] is perfect for that because it allows you to mimic a user and scroll to the bottom of the page.

With Selenium [7] we just let it run for a while but found that either ign[8] or Selenium would fail at around 150th scroll and

would crash the Selenium script. So we used the score filter and started by scraping the reviews with a score of 10 and the reviews with scores between 9 and 9.9 until 0. We found that this way, we could get more URLs by running it with different score ranges and 100 scrolls as the limit (most of the average score ranges got to this limit) and then end it by saving the page HTML source code to a file.

We didn't get all of the scores available on the website but we got a good part of them.

3.2 URL Extraction

The second step is to extract the review URLs from the HTML source code for all the score ranges. For that, we used BeautifulSoup [10].

BeautifulSoup [10] allows for easy traversal of the HTML document tree in Python [12]. With it, we were able to extract URLs embedded in the HTML code for every single review in it.

3.3 Review Extraction

The third and final step is to finally go through every review URL and scrape its contents. For that, we used the requests [9] library for Python [12] and BeautifulSoup [10] again.

After getting the raw HTML with the requests library we use BeautifulSoup to traverse the HTML tree and get the information we want such as the title, subtitle, author, date, the main review content, and the score.

Then using the CSV library in Python we create the final product of the pipeline which is the **ign.csv** file.

4 Data Characterization

We produced numerous documents for data characterization, which helped us analyze the data more easily. We created graphs using Matplotlib [3], Seaborn [4], and Pandas [11].

4.1 Number of Reviews per Year

The first graph we created presents the Number of Reviews per Year, from 1996 until today, 2024. We can see that most reviews were published between 2007 and 2014, which correlates to the time when most video games were released to the public. Many consider this the prime of video game launches.

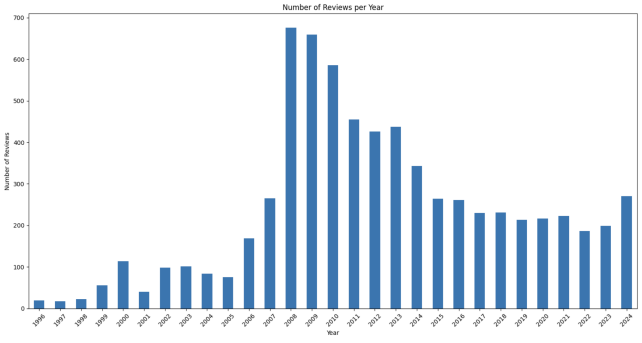

Figure 1: Reviews Per Year

4.2 Number of Reviews per Month

Figure [4] shows the number of reviews published throughout the year. We can clearly see a pattern, where the highest number of reviews publications is between the months of September and December. Normally, this is the time of the year when companies decide to launch their new video games, just before the holiday season.

4.3 Average Score per Year and Grouped Distribution of Review Scores

Figure [5] shows the score evolution over time. Back in the early two-thousands, the scores were relatively low, always in between 1 and 4. Only in 2009, we saw the first major increase in game scores. Until today, scores have not stopped increasing. We even have numerous video games maximum-rated, with a score of 10. This could be explained by the technological advancements, where developers started having the devices to produce better graphics and more immersive experiences for the players, resulting in higher scores. Another reason could be the fact that studios nowadays have way bigger budgets, so they can hire more developers and have more polished and advanced games.

Figure [6] shows almost the same data. This time around we decided to group review scores, we can clearly see the score evolution over time. We can also see the number of reviews published in those years, on top of the histograms.

4.4 Average Score by Top 20 Reviewers and Top 20 Reviewers by Number of Reviews

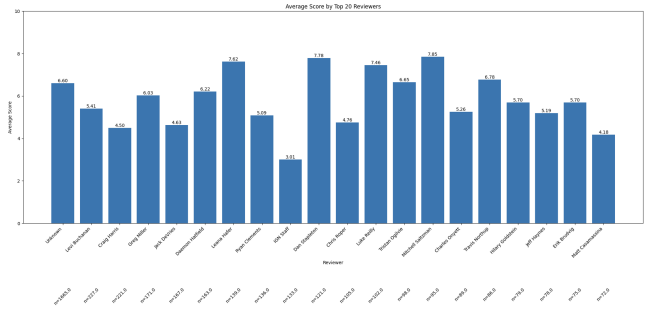

Figure 2: Average Score by Top 20 Reviewers

We also thought it would be interesting to analyze which authors tend to give higher reviews. We see that there are clearly some authors that give higher scores, maybe because some may have personal biases or preferences towards some styles or genres of games.

We also created a graph to see the author with the most published reviews (Figure [7]), and we can conclude that Levi Buchanan has 227 video games reviewed. Not far off is Craig Harris, with an astonishing 221 video games reviewed.

4.5 Word Cloud of Game Titles, Subtitles, and Content

Figure 3: Word Cloud of Game Titles, Subtitles, and Content

Word Cloud is a useful way to see which words stand out the most. In the case of the titles, we can see that the words used more often are in the first place "Review". The dataset consists of reviews, which include this term in all titles. "iPhone" because mobile games are gaining popularity, and IOS is one of the best platforms for mobile gaming. The word "Game" is fundamental and applies to all entries of the dataset.

In Figure [8], we can see that the words "Game", "New", "Fun", "Good Time", and "Best" appear regularly. This repetition shows the excitement surrounding new video game releases.

In Figure [9], we can clearly see that the word that appears the most is "Game", as the central topic of the reviews, followed by "Time", almost certainly associated with the good or bad times we can have while playing a video game, and, at last, the word "Player", that defines the person engaging with the game.

5 Search Scenarios

We identified some possible search scenarios that users would be keen to use when exploring for game reviews. These are, in our opinion, the most common ways a user would interact with our search engine:

- **Titles:** The first and most used search scenario would be to search game reviews by entering specific game titles. We think this would also be the fastest way for each user to find the review they are looking for.
- **Author:** The second possible search scenario would be to search video game reviews based on the author. It is known that some users like to find reviews from specific authors, who follow and trust their opinions.
- **Scores:** The third possible scenario would be to search for a review based on the score. A user could only be interested in games with, for example, a score above 8.
- **Subtitles:** The fourth search scenario would be to search for reviews based on their subtitles, which sometimes contain keywords like we saw thanks to the Word Cloud analysis.
- **Date:** The fifth search scenario consists of searching for a review based on the publish date. A user with a gaming machine from the latest generation would only be interested in finding reviews for games compatible with his device, and not searching for game reviews of games some years old.
- **Genre:** The sixth and last search scenario we think would be realistic is to search for a video game review based on the genres. Normally people tend to like specific types of games, so allowing the user to search by genre would be a very welcome functionality.
- **Games with tight controls:** Responsive controls are very important for gamers. Having input lag is one of the biggest issues in the gaming world. Gamers want precise and responsive mechanics. That is why we chose this search scenario as a possible one.
- **Games good for relaxing:** People interested in relaxing games, that don't want the stress of competitive games, could be keen to search for these types of games. These are the perfect games to play after a long and tiring day.
- **Story narrative games:** Some players would be keen to search for good story narrative games, a style of game really appreciated by many gamers.
- **Technical aspects of a game:** Graphics is a really important topic for gamers, so we imagine many would be interested in searching for a technical performance of a certain game in a certain platform, such as frame rates, loading times or resolution quality.
- **Social gaming experience:** Multiplayer is a key feature in today's games. This is what keeps gamers interest in games after finishing the single player mode. A typical gamer would search for a reliable multiplayer experience, where they can play with seamlessly with their friends.

6 Information Retrieval

We now begin the second part of this project: Information retrieval, which involves extracting relevant information from large datasets. This process focuses on searching and retrieving specific information from documents. We will now explain and present each method we used in this part of the project.

7 Collection and Indexing

7.1 Document Definition

The game reviews we extracted from IGN's website serve as the primary documents. We converted our CSV file to a JSON [1] file, and we added the **Id** attribute to each game review. We then created a subset file, to better improve query search, then marking **Id** as relevant or not.

Each document contains:

- **Title:** Game name and review indicator
- **Subtitle:** Brief tagline
- **Subheader:** Author and Date
- **Content:** Full review text
- **Score:** Numerical rating (0-10)
- **Id:** Individual identification for each review

In regard to the configuration, we used Solr [5] (a search platform from Apache), because it provides text search and real-time indexing capabilities for large document collections, features that we think are essential to handle our large dataset.

The following configurations were implemented for optimal document processing:

- **Text Fields (Title, Content, Subtitle)**
  - Field Type: `game_text`
  - Processing:
    - Standard Tokenization
    - Lowercase Filtering
    - Stop Word Removal
  - Purpose: Enables flexible and advanced text searching and analysis, supporting complex tokenization and stemming
- **Score Field**
  - Field Type: `pfloat`
  - Capability: Supports range queries, numeric operations, and score-based filtering
  - Purpose: Enables score-based filtering and analysis

7.2 Indexing Process

This was one of the most important parts of our project, because it allows fast search and retrieval, as well as relevant ranking, field specific queries, complex queries and full text search.

7.3 Schema Details

We used a schema to index our documents for the search system and so we decided to built a simple schema that is adapted for our retrieval tasks. For example, the *text_simple* as custom field was designed for basic text processing. *pfloat* was used for decimal numbers, more precisely for the score field. The **StandardTokenizer** is used to split text into words. This way, it makes text searchable by individual words. **LowerCaseFilter** converts a text to lowercase, so that we get all text to same case, and so we don't have issues with for example "games" and "Games": both results should be retrieved. **StopFilter** removes words not important, like "the", "and", "is", reducing the index size. All fields were indexed to improve search performance, and all fields were stored so that the original value can be retrieved.

Field	Type	Indexed
Title	text_simple	Yes
Content	text_simple	Yes
Subtitle	text_simple	Yes
Subheader	text_simple	Yes
Score	pfloat	Yes

Table 1: Overview of Important Fields in the Schema

- **Main title of the document** — Corresponds to the field `Title`
- **Full body content of the document** — Corresponds to the field `Content`
- **Subtitle providing additional context** — Corresponds to the field `Subtitle`
- **Secondary header for organizing content** — Corresponds to the field `Subheader`
- **Numeric score used for ranking and sorting** — Corresponds to the field `Score`

8 Retrieval Process

For this part of the project we used our simple schema.

The following table describes the key query parameters used in Solr [6]:

(1) **Query (q)**
Focuses on the most valuable words in the query.
(2) **Query Operator (q.op)**
Utilizes OR or AND for query operations.
(3) **Query Filter (fq)**
Defines a query to restrict the superset of documents.
(4) **Filter List (fl)**
Limits the fields included in a query response.
(5) **Sort Field (sort)**
Specifies the sorting value for the results.
(6) **Start (start)**
Specifies the starting offset for paginated results.
(7) **Rows (rows)**
Limits the number of documents returned in a single query.
(8) **Query Parser (defType)**
Specifies the query parser (e.g., edismax).
(9) **Boost Query (bq)**
Adds boosting to specific query terms to increase their relevance.
(10) **Highlighting (hl)**
Enables highlighting of matched terms in the results.

The query parsers are tools that interpret and process user queries. They essentially convert text to a query format that is then used in the search engine. Query's support boolean operators like **AND, OR, NOT**. They also allow for title search for example. On the other hand, **Dismax** is designed for simple keyword searched, and it searches fields like title and content. It also allows for boosting. There is an improved version of **Dismax**, called **eDismax** [2], that supports more advanced query syntax, and allows for example field-specific boosts.

9 Evaluation Process

Another important part of this project was the evaluation process. We can now identify user search scenarios, clarify types of documents and specify goals of the evaluation process. We used different metrics, such as **Precision-Recall Curves**, **P@**, **MAP** and **AUC**. **Precision Recall Curves** allows us to visualize the trade-off between precision and recall, so, it's effective when we want to check the relevance of retrieved items and the coverage of all relevant items. This way, we ensure a balanced performance for both precision and recall. On the other hand, **P@** is useful to evaluate the precision of retrieved results, in other words, it shows the quality of the top results. The **Mean Average Precision (MAP)**, essentially summarizes precision across queries, and gives more weight to relevant items that were retrieved. Thanks to this, we can see the relevant results across the different queries we implemented during our evaluation process. The **Area Under the Curve (AUC)**, helps us see the performance across all possible classification thresholds. This is how we know if the system can distinguish between relevant and non-relevant items. We think these metrics provide a well rounded evaluation for out retrieval system.

We will now present each evaluation we performed using these metrics.

9.1 Games with tight controls

The first evaluation process we decided to find games with tight controls, so, games with a high degree of responsiveness and precision. This way, players fell a strong sense of control, that results in a natural and engaging game-play. The gamer knows he is in control of everything.

We used keywords like **controls**, **precision**, **responsive**, **accurate**. We think these are the most important words to search in this particular case, that would lead us to the best results.

Table 2: Query Simple

Query	
q	`Content:(*tight* *responsive* *precise*) OR Subtitle:(*tight* *responsive* *precise*)`
qf	`Content^4.0 Subtitle^2.0`
defType	`edismax`
rows	`100`
fl	`id,Title,Content,Score`

Table 3: Query Boosted

Query	
q	`(Title:(*controls* *responsive* *precise* *tight*) OR Content:(*controls* *responsive* *precise* *tight* *accurate* *fluid* *precision*) OR Subtitle:(*controls* *responsive* *precise* *tight*))`
qf	`Title^2.0 Content^3.0 Subtitle^2.5`
pf	`Title^6.0 Content^3.0`
bq	`Subtitle:(*precise*)^5.0 Content:(*fluid* *responsive*)^3.0`
fl	`id,Title,Content,Score`

Table 4: Evaluation Results

Evaluation Metrics	
AVP	0.873
MAP	0.457
AUC	0.493
P@5	1.00
P@10	0.90
P@15	0.80
P@20	0.85

We can see that the boosted query got better results. It used more specific terms, so we think that was key to better target content. The OR condition allowed the boosted query to capture way more content.

9.2 Games good for relaxing

For the second evaluation process we decided to find games good for relaxing, so, perfect games to play after a long day. These games focus on minimal challenges, calm game-plays and smooth mechanics.

We used keywords like **relaxing**, **peaceful**, **calming**, **chill**. We think these are the most important words to search in this particular case that would lead us to the best results.

Table 5: Query Simple

Query	
q	`Title:(*relax* *peaceful* *calm* *chill*) OR Content:(*relax* *peaceful* *calm* *meditat* *chill* *simple* *serene* *enjoy* *soft*) OR Subtitle:(*relax* *peaceful* *calm* *meditat* *chill*)`
qf	`Title^2.0 Content^3.0 Subtitle^2.5`
defType	`edismax`
bf	`Score^2.0`
rows	`100`
fl	`id,Title,Score`

Table 6: Query Boosted

Query	
q	`Title:(*relax* *peaceful* *calm* *chill*) OR Content:(*relax* *peaceful* *calm* *meditat* *chill* *mindful* *simple* *serene* *enjoy* *soft*)`
qf	`Title^4.0 Content^2.0 Subtitle^1.5`
pf	`Title^8.0 Content^4.0`
bq	`Subtitle:(*peaceful*)^2.0 Content(*calm* *relax*)^3.0`
fl	`id,Title,Score`

Table 7: Evaluation Results

Evaluation Metrics	
AVP	0.354
MAP	0.172
AUC	0.178
P@5	0.60
P@10	0.40
P@15	0.33
P@20	0.30

Again, we got better results with the boosted query. It used more specific terms to better target games that focus on relaxation and

calmness. The OR condition allows one more time to capture way more relevant content.

9.3 Story narrative games

For the third evaluation process we decided to find story narrative games, so, games that emphasize strong storytelling, character development and immersive plots. There are games were player choice is prioritized.

We used keywords like **story**, **narrative**, **emotional**, **character**. We think these are the most important words to search in this particular case that would lead us to the best results.

Table 8: Query Simple

Query	
q	`Content:(*story* *narrative*) OR Subtitle:(*story* *narrative*)`
qf	`Content^4.0 Subtitle^2.0`
defType	`edismax`
rows	`100`
fl	`id,Title,Content,Score`

Table 9: Query Boosted

Query	
q	`Title:(*story* *narrative* *emotional* *character*) OR Content:(*story* *narrative* *emotional* *character*) OR Subtitle:(*story* *narrative* *emotional* *character*)`
qf	`Title^2.0 Content^3.0 Subtitle^2.5`
pf	`Title^6.0 Content^3.0`
bq	`Subtitle:(*story*)^5.0 Content:(*character* *emotional*)^4.0`
fl	`id,Title,Content,Score`

Table 10: Evaluation Results

Evaluation Metrics	
AVP	0.748
MAP	0.491
AUC	0.506
P@5	0.80
P@10	0.90
P@15	0.80
P@20	0.75

This time around, we got better results for the simple query. The boosted query maybe had a way more complicated structure, that led to find worse results. The simple query targeted keyword selection, so highly specific words. These terms probably were enough to capture the elements we were looking for. The boosted query additional terms may have introduced unnecessary complexity.

9.4 Visually impressive Games

For the fourth evaluation process we decided to find visually impressive games, so, games that stand out due to their graphics and visual design. There are games that often showcase cutting-edge technology, that offer stunning details for the player.

We used keywords like **visually**, **graphically**, **visuals**, **realistic**. We think these are the most important words to search in this particular case that would lead us to the best results.

Table 11: Query Simple

Query	
q	`Content:(*graphically* *visuals*) OR Subtitle:(*graphically* *visuals*)`
qf	`Content^4.0 Subtitle^2.0`
defType	`edismax`
rows	`100`
fl	`id,Title,Content,Score`

Table 12: Query Boosted

Query	
q	`Title:(*visually* *graphically* *visuals* *realistic*) OR Content:(*visually* *graphically* *visuals* *realistic*) OR Subtitle:(*visually* *graphically* *visuals* *realistic*)`
qf	`Title^2.0 Content^3.0 Subtitle^2.5`
pf	`Title^6.0 Content^3.0`
bq	`Subtitle:(*visuals*)^4.0 Content:(*realistic*)^3.0`
fl	`id,Title,Content,Score`

Table 13: Evaluation Results

Evaluation Metrics	
AVP	0.353
MAP	0.090
AUC	0.096
P@5	0.40
P@10	0.30
P@15	0.33
P@20	0.35

We got really low results in both simple and boosted queries. We think this might due to the fact that both queries didn't find relevant content, because review authors maybe don't focus as much on technical aspects of the game, and so it was difficult for queries to find good results.

9.5 Multiplayer Games

For the fifth and last evaluation process we decided to search for Multiplayer Games. These are games that allow multiple players to play together in the same game. Normally these games offer competitive modes, that make them ideal to player with friends or family.

We used keywords like **online**, **cooperative**, **competitive**, **team**. We think these are the most important words to search in this particular case that would lead us to the best results.

Table 14: Query Simple

Query	
q	`Title:(*multiplayer* *online*) OR Content:(*multiplayer*)`
qf	`Title^3.0 Content^2.0`
defType	`edismax`
rows	`100`
fl	`id,Title,Content,Score`

Table 15: Query Boosted

Query	
q	`Title:(*multiplayer* *online*) OR Content: (*multiplayer* *team*) Subtitle: (*multiplayer* *online* *cooperative* *team*)`
qf	`Title^4.0 Content^2.0 Subtitle^1.5`
fq	`Content:multiplayer`
bq	`Subtitle:(*online*)^3.0 Content:(*multiplayer*)^5.0`
fl	`id,Title,Content,Score`

Table 16: Evaluation Results

Evaluation Metrics	
AVP	0.781
MAP	0.308
AUC	0.303
P@5	0.60
P@10	0.80
P@15	0.80
P@20	0.85

In these two last queries, we got similar results between simple and boosted queries. Both queries used the core "Multiplayer" term, so maybe both queries found similar results. Searching for this type of games is straightforward, so maybe a simple query can get the job done, and in this case both queries did well.

10 Conclusion

Now that Milestone 1 and 2 have finished, we are exited to move on to Milestone 3. In Milestone 2, we collected and indexed documents, performed retrieval tasks and evaluated the different results we obtained from the queries. We are satisfied with the progress we have made so far in these two Milestones, and we think this latest Milestone is a solid foundation for the future, and that is why we can't way for Milestone 3 to start, to finally implement some improvements. We will try to implement a second, more advanced schema, and so have two schemas: a simple and a more enhanced one. We also need to start implementing some query expansion strategies, for example to find synonyms and related terms, in order to try to improve the results of our queries.

References

[1] Douglas Crockford. 2001. JSON: JavaScript Object Notation. https://www.json.org/. Accessed: 2024-11-18.
[2] Elasticsearch. 2024. EDisMax Query Parser: Extended DisMax for Advanced Querying. https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-dis-max-query.html. Accessed: 2024-11-18.
[3] John D. Hunter et al. 2007. Matplotlib: A 2D Graphics Environment. https://matplotlib.org/. Accessed: 2024-10-15.
[4] Michael Waskom et al. 2012. Seaborn: Statistical Data Visualization. https://seaborn.pydata.org/. Accessed: 2024-10-15.
[5] Apache Software Foundation. 2006. Apache Solr: Open Source Enterprise Search Platform. https://solr.apache.org/. Accessed: 2024-11-18.
[6] Apache Software Foundation. 2024. Solr Filters: Token Filters, Query Filters, and More. https://solr.apache.org/guide/. Accessed: 2024-11-18.
[7] Jason Huggins. 2004. Selenium. https://www.selenium.dev
[8] IGN Entertainment, Inc. 1996. IGN: Video Game Reviews, News, and Trailers. https://www.ign.com/. Accessed: 2024-10-15.
[9] Kenneth Reitz. 2011. Requests Library. https://pypi.org/project/requests/
[10] Leonard Richardson. 2004. BeautifulSoup. https://pypi.org/project/beautifulsoup4/
[11] The Pandas Development Team. 2008. Pandas: Python Data Analysis Library. https://pandas.pydata.org/. Accessed: 2024-10-15.
[12] Guido van Rossum. 1991. Python. https://www.python.org

A Annexes

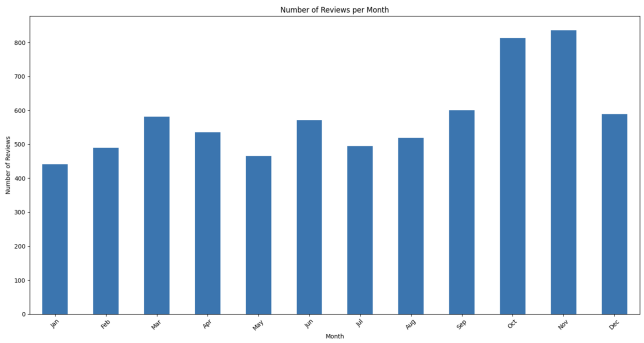

Figure 4: Reviews per Month

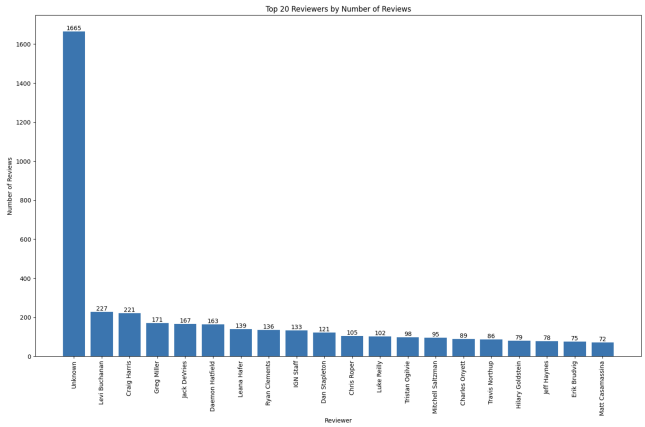

Figure 7: Top20 Reviewers

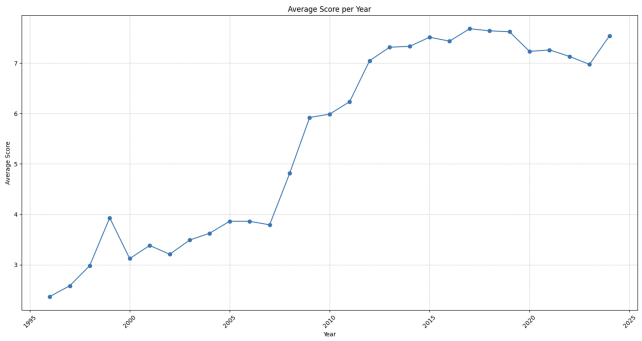

Figure 5: Scores per Year

Figure 8: Word Cloud for Subtitles

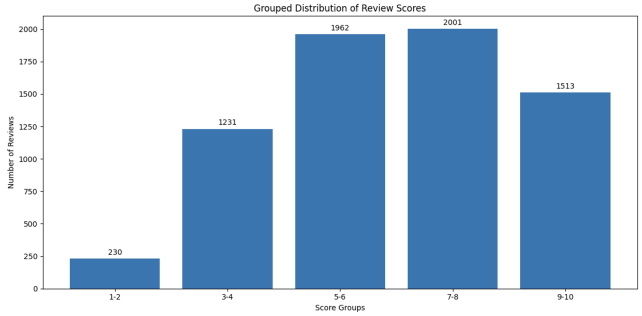

Figure 6: Distribution Review Scores

Figure 9: Word Cloud for Content

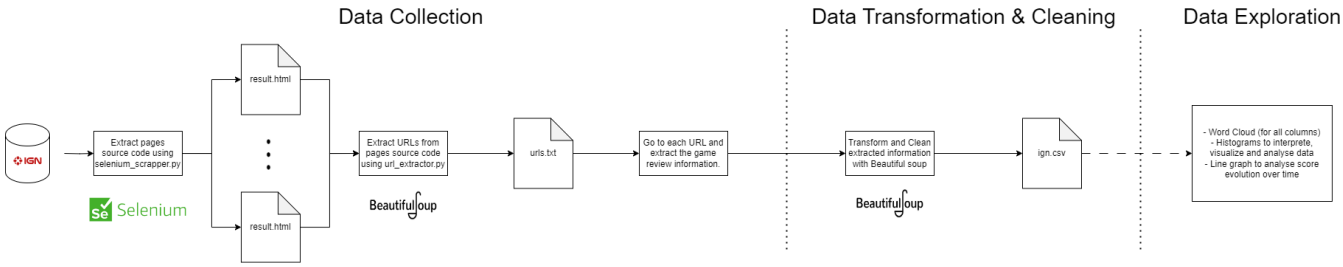

Figure 10: Data Pipeline

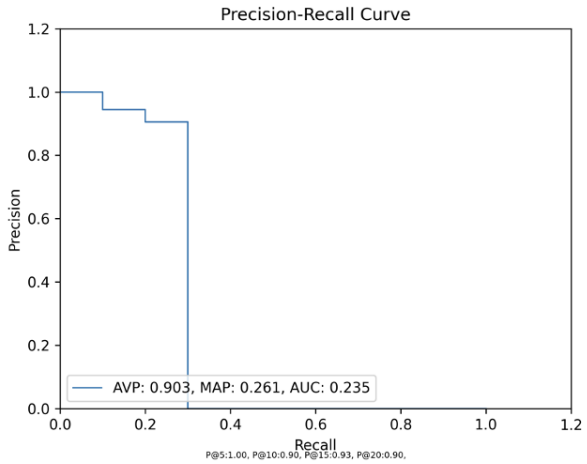

Figure 11: Games with Tight Controls (simple)

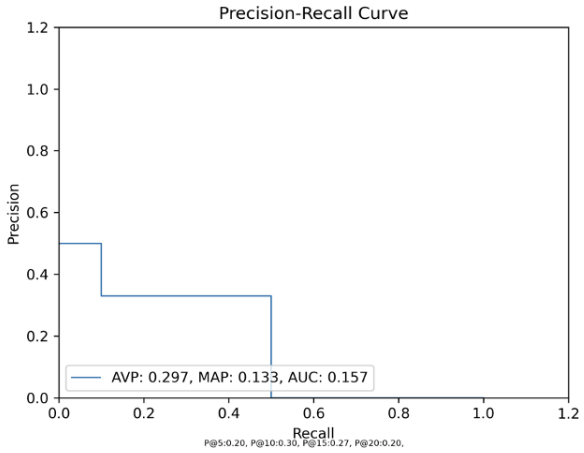

Figure 13: Games good for Relaxing (simple)

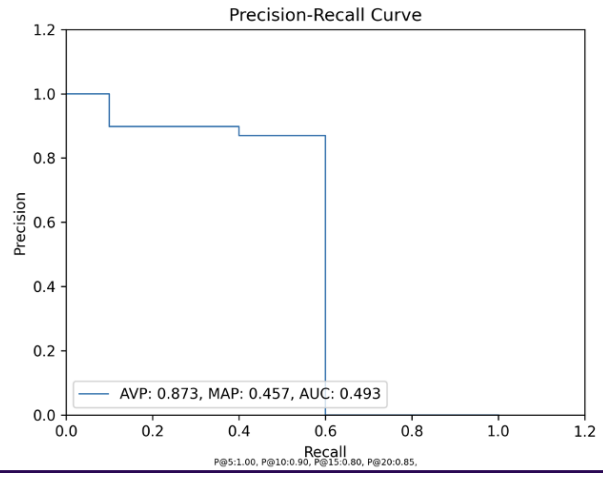

Figure 12: Games with Tight Controls (boosted)

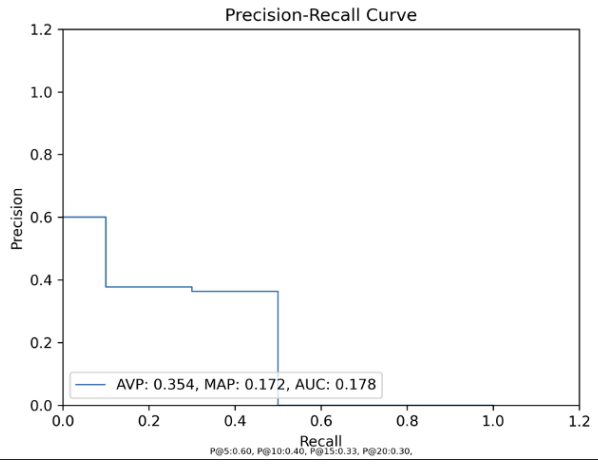

Figure 14: Games good for Relaxing (boosted)

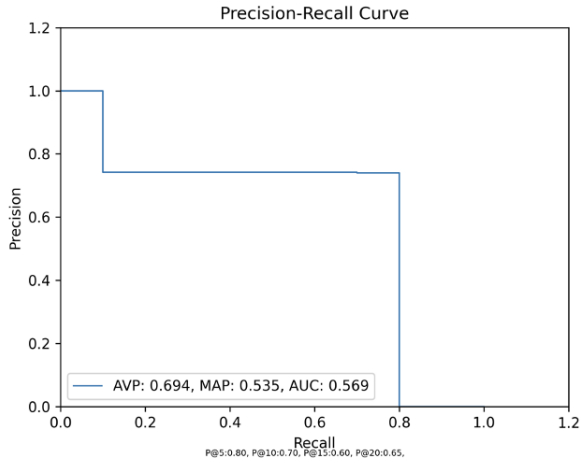

Figure 15: Story Narrative Games (simple)

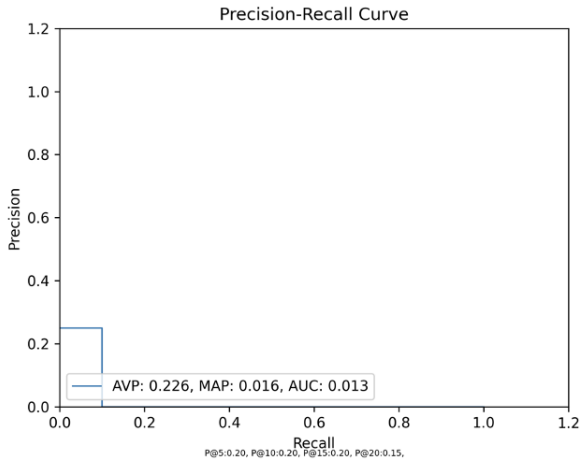

Figure 17: Visually Impressive Games (simple)

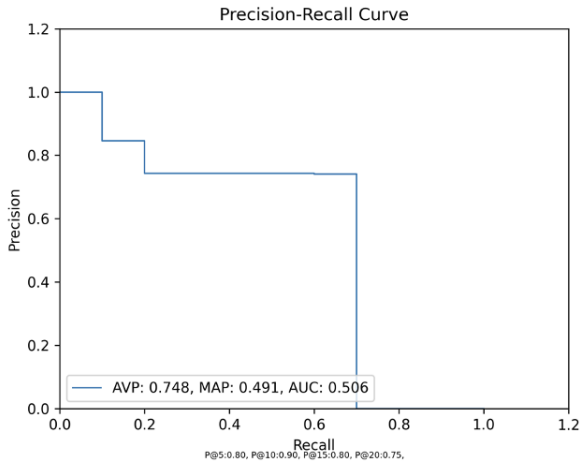

Figure 16: Story Narrative Games (boosted)

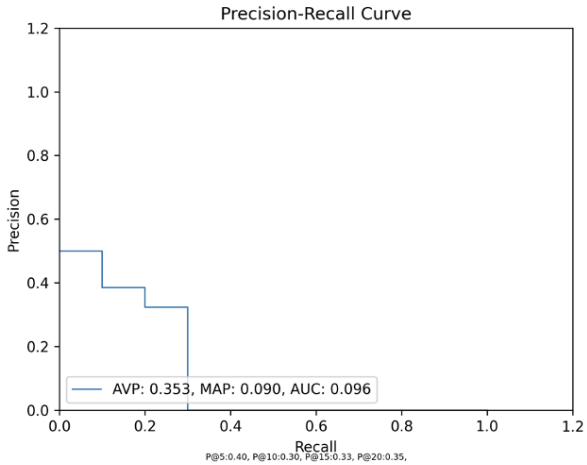

Figure 18: Visually Impressive Games (boosted)

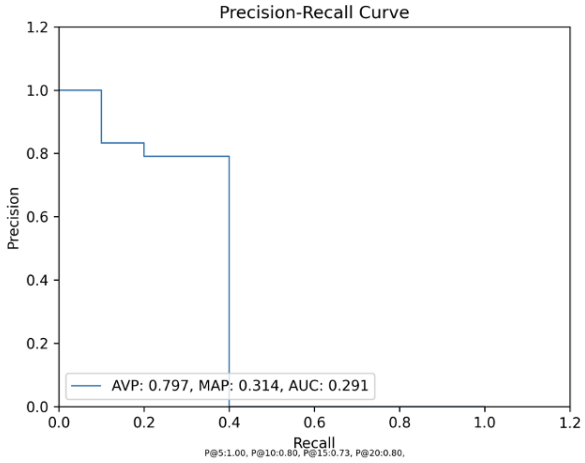

Figure 19: Multiplayer Games (simple)

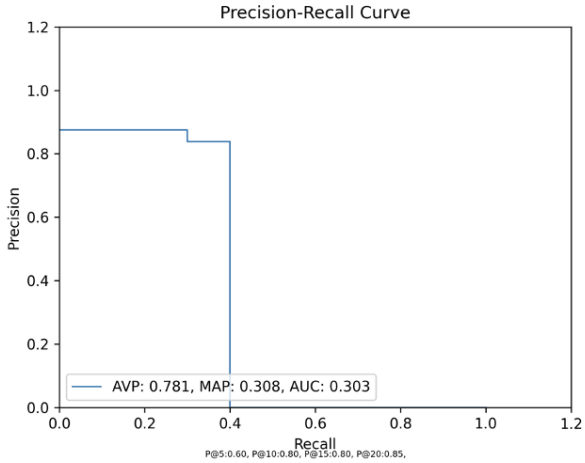

Figure 20: Multiplayer Games (boosted)