AURYGA: A Recommender System for Game Tagging

Riccardo Rubei[1], Claudio Di Sipio[1]

[1]Università degli studi dell'Aquila, 67100 L'Aquila, Italy

Abstract

Multimedia data have become predominant in several application domains. To cope with this huge amount of data, recommender systems come in handy by delivering valuable items to end-users. In particular, existing PC games stores could be supported by such systems. Even though the user can easily browse such platforms to find desired items, there is still room for improvements, i.e., enhancing available tagging systems, adopting new analysis techniques. With the aim of assisting users in the definition of the tags, we propose AURYGA, an **AU**tomated **R**ecommender s**Y**stem that exploits a well-founded stochastic model to predict **GA**me tags using the textual description of PC games stored on the Steam platform. First, we filter the initial set of tags by using the frequent itemset technique. Afterwards, we use a TF-IDF encoder on textual descriptions to feed the underpinning model to retrieve the list of tags. Due to the lack of a proper baseline, we evaluate AURYGA by means of a 10-fold cross validation on a real-world dataset. The initial results show that the proposed prototype represents a workable solution in the domain.

Keywords

recommender systems, machine learning, videogames classification

1. Introduction

To properly manage information overload yielded by multimedia data [1], recommender systems [2] have been successfully employed in several domains, e.g., e-commerce [3], streaming platforms [4], social media [5, 6], and scientific papers [7, 8]. In particular, the videogame industry benefits from digital stores [9, 10, 11] that allow users to discover and purchase games easily. Among the others, the Steam platform [12], maintains and sells a large number of PC games. Each of them is enriched with useful information for the customer, e.g., description, features, genre, to name a few. To better increase the discoverability of the games, a user-based tagging systems is employed, although it can be improved by exploiting more sources of knowledge.

To this end, we propose AURYGA, a novel recommender system based on a Complement Naive Bayesian (CNB) network to retrieve a set of tags given the textual description of the game. First, we empirically select 7 main game genres and extract frequent sub-tags patterns using the FP-growth mining algorithm. Then, the textual content is encoded by relying on a TF-IDF module. The produced vectors are eventually used by AURYGA to predict a final set of tags. As a preliminary evaluation, we used a Steam dataset composed of PC games with the corresponding tags. We conduct a 10-fold cross evaluation by computing different metrics, i.e., success rate, precision, recall, and F-measure. The initial results show that AURYGA is

The 11th Italian Information Retrieval Workshop (IIR 2021), September 13–14, 2021, Bari, Italy
riccardo.rubei@graduate.univaq.it (R. Rubei); claudio.disipio@graduate.univaq.it (C. Di Sipio)

Figure 1: *Syberia* game tags and the suggested items

capable of recommending the correct tags most of the time. We make available the initial implementation as well as the dataset to replicate the experiments in the corresponding Github repository[1].

2. Motivation and background

In this section, we present an explanatory example in Figure 1 to highlight our contribution. The left side of the picture shows *Syberia*, a point-and-click adventure game [2] and its user-defined tags ranked according to their frequency, i.e., the number of users that define that tag for the game. Although the top-3 tags are correct, i.e., *Adventure*, *Point & Click* and *Puzzle* the game is labeled with ones that represent game's features rather than the actual genre [13]. For example, the tag *Beautiful* expresses a subjective evaluation that could compromise the correct categorization of the game.

Based on this list, Steam suggests additional games to purchase as shown on the right side of the picture, i.e., *More like this* section. By carefully inspecting three of them, namely *Red Dead Redemption II*, *Shadow of the Tomb Raider*, and *Half-Life Alyx*, we discover that the recommended items belong to completely different genres i.e., western, purely action-adventure, and first-person shooter[3] respectively. Thus, inaccurate labeling can compromise the visibility of games as the platform groups them according to the user-based defined tags.

3. Proposed methodology

The proposed approach and its main components are shown in Fig.2. As the first step, we empirically select 7 main tags from the most famous genres [4] by eliciting the ones with higher

[1]https://github.com/Arkanoid01/Auryga
[2]https://fr.flossmanuals.net/creating-point-and-click-games-with-escoria/what-is-point-and-click-games/
[3]https://www.pcmag.com/encyclopedia/term/first-person-shooter
[4]https://www.statista.com/statistics/189592/breakdown-of-us-video-game-sales-2009-by-genre/

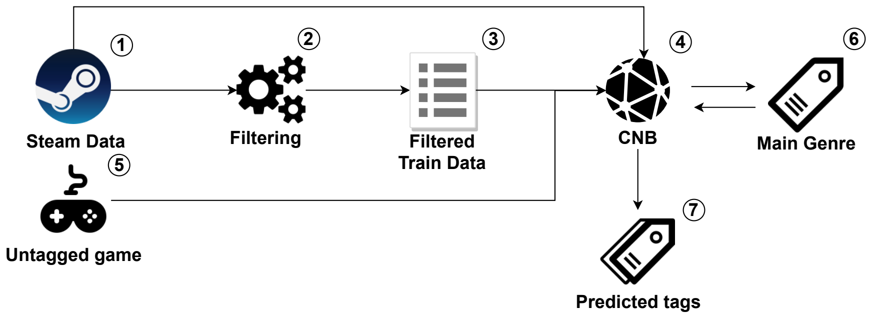

Figure 2: AURYGA's architecture

frequencies in the considered dataset, i.e., *Action*, *Adventure*, *Rpg*, *Platform*, *Sport*, *Strategy*, and *Simulation*. On top of these main tags, the raw `Steam data` (1) labeled with 370 unique game tags is used by the `Filtering` component (2) to retrieve the most frequent sub-patterns using the FP-growth, a well-founded mining algorithm [14] provided by Mlxtend Python library [5]. In such a way, AURYGA extracts a set of `filtered train data` (3) for each main category by selecting the *miminum support*, i.e., a threshold that evaluates the strength of a certain pattern appearing in the whole itemset [15]. By running several experiments, we fix the support value to retrieve around 50 sub-tags related to the main one. Overall, this preprocessing phase reduces the number of unique tags from 370 to 100. The reason is two-fold *i)* low-frequency user tags are dropped and *ii)* the overall accuracy is improved for each main tag. For instance, the pattern *[action, singleplayer, sci_fi]* occurs frequently in the original dataset, meaning that *action* games are tagged also with the mentioned sub-tags most of the time.

Afterwards, AURYGA employs (4) `Complement Naive Bayes (CNB)` network [6], an enhanced version of Multinomial Naive Bayes (MNB) that cope with unbalanced dataset [16]. Differently from the MNB, CNB computes the probability that an item appears in the other classes and normalizes classification weights to properly manage word occurrence dependencies. When an `untagged game` arrives (5), the textual description is encoded using the TF-IDF vectorizer [7]. First, the underpinning model predicts the `main genre` (6) by relying on the original Steam data. Second, the CNB exploits the abovementioned filtered training sets to retrieve sub-tags when the main one is correctly predicted. The system eventually retrieves the `predicted tags` (7) that include the main genre and the most probable sub-tags.

3.1. Preliminary evaluation

To assess the conceived approach, we conduct a 10-fold cross validation [17] on the mentioned Steam dataset composed of 27,334 games gathered in 2019 [8]. Each game is identified by a

[5]http://rasbt.github.io/mlxtend/api_subpackages/mlxtend.frequent_patterns/#fpgrowth

[6]https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.ComplementNB.html#sklearn.naive_bayes.ComplementNB

[7]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[8]https://www.kaggle.com/nikdavis/steam-store-games

unique ID, the title, the description, and the list of tags. For each round, we used 90% of the descriptions as training and 10% as testing to compute *Success rate*, *Precision*, *Recall*, and *F-Measure*, considering the following definitions: a *True positive (Tp)* is a recommended tag that corresponds to the actual game tag. A *false positive (FP)* means that the suggested tag is not present in the actual list while a *false negative (Fn)* represents a tag that should be present in the result set but actually is not. Given a set of N recommended tags, the *Success rate* measures the rate at which the proposed approach can return at least c correct tags for c=1,2,3,4,5.

$$Success\ rate = \frac{count_{n \in N}(Tp > 0)}{|N|} \times 100\% \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F - measure = \frac{2 \times P \times R}{P + R} \tag{4}$$

Precision and *Recall* assess the rate of correct items over the entire set of retrieved items and the ratio of the right topics appearing in the retrieved set respectively. *F-measure* represents the harmonic mean of the two previous metrics. All the mentioned indexes are computed by considering the filtered set of tags, i.e., the ones obtained by the `Filtering` component.

Figure 3 shows the results obtained in terms of success rate. As we can see, AURYGA is capable of suggesting at least the first tag with good results, i.e., it reaches 98% with N=8. As expected, increasing the number of correct tags c impacts negatively on the performances. For instance, the success rate dramatically decreases if we consider c=5, i.e., the system recommends exactly five correct tags only in the 0,3% of the cases.

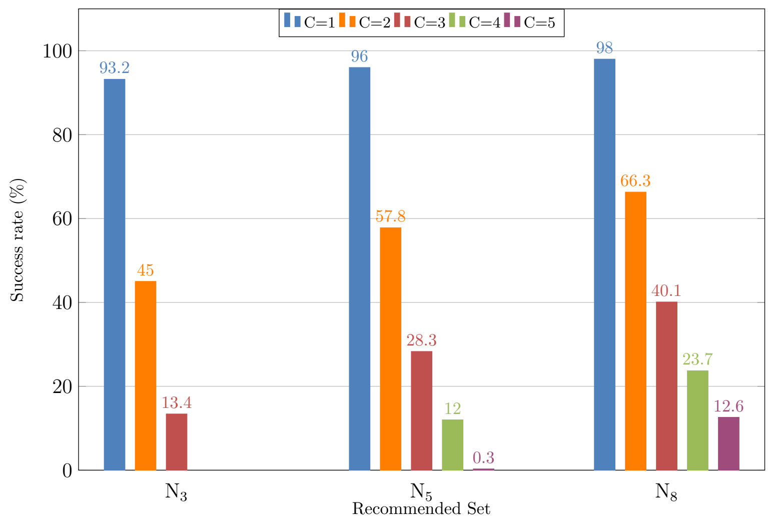

Figure 3: Success Rate for c=1,2,3,4,5

To further study the AURYGA's performances, we compute additional metrics, i.e., precision, recall, and F-measure showed in Table 1. Overall, increasing the number of recommended items leads to low results on average i.e., the maximum F-measure value is reached with N=3. In particular, the precision decrease from 50.01% to 31.2% by increasing the number of retrieved items. Contrariwise, augmenting N yields to better performance in terms of recall, i.e., it achieves 67.2% with N=8. This is quite expected since the system retrieves more tags that should be included. It is worth noting that we aim to recommend 100 different unique tags, considering both the main and sub-tags. It is our strong belief that reducing the number of tags in the training will lead to better accuracy since this strategy has been proved successful in existing studies [18].

Table 1
Average metrics' values for N=3,5,8

N	Precision	Recall	F-measure
3	**50.6**	49.7	**50.1**
5	39.6	58.5	47.2
8	31.2	**67.2**	42.6

4. Related works

The first taxonomy of games was proposed in [19] by manually inspecting their features e.g., genre, the number of players, description. A curated discussion over games' genres has been conducted in [20] by considering several aspects i.e., family resemblances, prototype theory, faceted classification, and appeal factors. Horppu *et al.* [21] make use of the SVM model to classify 2,443 games on the IoS App store by using title and description. Similarly, Amiriparian *et al.* [22] propose an audio-based videogame genre categorization using linear SVM and CNN models on a dataset composed of 1,566 multimedia content including in-game video, audio, and visual elements to categorize games into six categories. A multi-modal deep neural network [23] trained with the cover image and the textual information has been employed to classify 50,000 PC games stored on IGDB.com website by predicting 21 unique tags.

5. Conclusion and future works

Videogame online stores manage hundred of PC games usually labeled with a list of tags. Even though such a tagging system is successfully employed by several platforms, increasing the accuracy of the defined categories is still an open challenge. To fill the gap, we propose AURYGA, a recommender system that automatically retrieves a set of game tags by analyzing the textual description of the content. The system combines a well-founded frequent itemset algorithm and a stochastic model to recommend the main genre and its most frequent sub-tags. The preliminary evaluation shows encouraging results in terms of predicted tags even though some metrics achieve low values.

For future works, we can feed the underpinning algorithm with additional textual data, i.e., game title, developer, reviews. Furthermore, we can enhance the preprocessing phase by tuning the filtering component. Last but not least, we are going to conduct a controlled user study in which the participants evaluate the system qualitatively.

References

[1] Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, Recommender Systems Leveraging Multimedia Content, ACM Computing Surveys 53 (2020) 1–38. URL: https://dl.acm.org/doi/10.1145/3407190. doi:10.1145/3407190, 00020.

[2] F. Ricci, L. Rokach, B. Shapira, Introduction to Recommender Systems Handbook, Springer US, Boston, MA, 2011, pp. 1–35. URL: https://doi.org/10.1007/978-0-387-85820-3_1. doi:10.1007/978-0-387-85820-3_1.

[3] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing 7 (2003) 76–80. URL: http://ieeexplore.ieee.org/document/1167344/. doi:10.1109/MIC.2003.1167344.

[4] C. A. Gomez-Uribe, N. Hunt, The Netflix Recommender System: Algorithms, Business Value, and Innovation, ACM Transactions on Management Information Systems 6 (2016) 1–19. URL: https://dl.acm.org/doi/10.1145/2843948. doi:10.1145/2843948.

[5] L. Posch, C. Wagner, P. Singer, M. Strohmaier, Meaning as collective use: Predicting semantic hashtag categories on twitter, in: Proceedings of the 22nd International Conference on World Wide Web, WWW '13 Companion, Association for Computing Machinery, New York, NY, USA, 2013, p. 621–628. URL: https://doi-org.univaq.clas.cineca.it/10.1145/2487788.2488008. doi:10.1145/2487788.2488008.

[6] B. Sigurbjörnsson, R. van Zwol, Flickr tag recommendation based on collective knowledge, in: Proceedings of the 17th International Conference on World Wide Web, WWW '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 327–336. URL: https://doi-org.univaq.clas.cineca.it/10.1145/1367497.1367542. doi:10.1145/1367497.1367542.

[7] D. Boughareb, A. Khobizi, R. Boughareb, N. Farah, H. Seridi, A graph-based tag recommendation for just abstracted scientific articles tagging, Int. J. Cooperative Inf. Syst. 29 (2020) 2050004:1–2050004:30.

[8] L. Galke, F. Mai, I. Vagliano, A. Scherp, Multi-modal adversarial autoencoders for recommendations of citations and subject labels, in: Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, UMAP '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 197–205. URL: https://doi-org.univaq.clas.cineca.it/10.1145/3209219.3209236. doi:10.1145/3209219.3209236.

[9] Epic Games Store | Official Site, 2021. URL: https://www.epicgames.com/store/en-US/.

[10] Google Play, 2021. URL: https://play.google.com/store/, 00380.

[11] PlayStation™Store, 2021. URL: https://store.playstation.com/.

[12] Steam Store, 2021. URL: https://store.steampowered.com/.

[13] E. Adams, Fundamentals of Game Design, 3rd ed., New Riders Publishing, USA, 2014.

[14] J. Han, J. Pei, Y. Yin, R. Mao, Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach, Data Mining and Knowledge Discovery 8 (2004) 53–87. URL: http://link.springer.com/10.1023/B:DAMI.0000005258.31418.83. doi:10.1023/B:DAMI.0000005258.31418.83.

[15] M. Steinbach, P.-N. Tan, H. Xiong, V. Kumar, Generalizing the notion of support, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, Association for Computing Machinery, New York, NY, USA, 2004, p. 689–694. URL: https://doi-org.univaq.clas.cineca.it/10.1145/1014052.1014141. doi:10.

1145/1014052.1014141.

[16] J. D. M. Rennie, L. Shih, J. Teevan, D. R. Karger, Tackling the poor assumptions of naive bayes text classifiers, in: Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML'03, AAAI Press, 2003, p. 616–623.

[17] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143. URL: http://dl.acm.org/citation.cfm?id=1643031.1643047.

[18] W. Siblini, P. Kuntz, F. Meyer, A review on dimensionality reduction for multi-label classification, IEEE Transactions on Knowledge and Data Engineering 33 (2021) 839–857. doi:10.1109/TKDE.2019.2940014.

[19] K. J. H.G., The gaming landscape: a taxonomy for classifying games and simulations, in: DiGRA &#3903 - Proceedings of the 2003 DiGRA International Conference: Level Up, 2003. URL: http://www.digra.org/wp-content/uploads/digital-library/05163.55012.pdf.

[20] R. I. Clarke, J. H. Lee, N. Clark, Why Video Game Genres Fail: A Classificatory Analysis, Games and Culture 12 (2017) 445–465. URL: http://journals.sagepub.com/doi/10.1177/1555412015591900. doi:10.1177/1555412015591900.

[21] I. Horppu, A. Nikander, E. Buyukcan, J. Mäkiniemi, A. Sorkhei, F. Ayala-Gómez, Automatic classification of games using support vector machine, CoRR abs/2105.05674 (2021). URL: https://arxiv.org/abs/2105.05674. arXiv:2105.05674.

[22] S. Amiriparian, N. Cummins, M. Gerczuk, S. Pugachevskiy, S. Ottl, B. Schuller, "are you playing a shooter again?!" deep representation learning for audio-based video game genre recognition, IEEE Transactions on Games 12 (2020) 145–154. doi:10.1109/TG.2019.2894532.

[23] Y. Jiang, L. Zheng, Deep learning for video game genre classification, arXiv:2011.12143 [cs] (2020). URL: http://arxiv.org/abs/2011.12143, 00001 arXiv: 2011.12143.