Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 225–230 DOI: 10.15439/2024F9868 ISSN 2300-5963 ACSIS, Vol. 39

Literature Books Recommender System using Collaborative Filtering and Multi-Source Reviews

Elena-Ruxandra Luţan
0000-0001-5363-9930
Department of Computers and Information Technology
University of Craiova, 200585, Craiova, Romania
Email: elena.ruxandra.lutan@gmail.com

Costin Bădică
0000-0001-8480-9867
Department of Computers and Information Technology
University of Craiova, 200585, Craiova, Romania
Email: costin.badica@edu.ucv.ro

***Abstract*—In this contribution, we present a method for obtaining literature books recommendations using collaborative filtering recommender system technique and emotions extracted from multi-source online reviews. We experimentally validated the proposed system using a book dataset and associated reviews that we collected from *Goodreads* and *Amazon* websites using our customized web scrapers. We show the benefits of using multi-source reviews by proposing a series of recommender system evaluation measures, which include single-source and multi-source recommendations similarity, recommendation algorithm usecases coverage and generated recommendations relevance.**

I. Introduction

RECOMMENDER systems aim to help people choose different aspects of their life by relying on similar peers feedback. Traditional recommender system approaches of user ratings or browsing interaction can be enhanced by features extracted from user reviews, resulting in personalized recommendations results [2], [8].

In this paper, we extend our previous research on the topic of book recommendations considering emotions from social media book reviews ([5], [6]), by investigating the benefits of using multi-source reviews for creating an emotional categorization of literature books and provide book recommendations.

For analysis, we use a set of 1000 books rated on *Goodreads* website as ”Best Books Ever”, which we also used for our previous research [6]. Thus, we want to compare the results and show the improvements resulted by addition of the multi-source reviews.

For the 1000 books, we use two sets of book reviews that we collected from *Goodreads* and *Amazon* website, using our customized web scrapers.

We propose an user-based collaborative filtering recommendations algorithm, which identifies the top 5 most similar users to the user of interest based on the similarity of emotions present in their reviews. Then, a selection of 5 books enjoyed by the most similar users is recommended.

Lastly, we present three performance measures for evaluating our system: *Average Recommendations Similarity* which shows the average similarity between recommendations obtained using single-source reviews and multi-source reviews, *Algorithm Branch Coverage* which quantifies the recommendations algorithm branch used for providing recommendations (i.e. information if the system used preferences of similar users to provide recommendations or suggested random book), *Relevance* which measures the relevance of the recommendations by evaluating the similarity between the reviewed book and recommended book.

For all three considered measures, we observe improvements in the recommendation process resulted from the addition of multi-source reviews information.

Compared to our previous works on literature books recommender systems, this paper includes: (1) comparison between usage of single-source and multi-source reviews for creating emotional book categorization, (2) introduction of a new book reviews dataset, collected from *Amazon* website using our customized we scraper, (3) presentation of recommendations quality performance measures and discussions revealing the benefits of using multi-source reviews.

The paper is structured as follows. In Section II, we present related works. Section III describes our proposed user-based collaborative filtering book recommendation algorithm using emotions from multi-source social media reviews. In Section IV, we provide and insight of the experimental dataset and discuss the experimental results. The last section presents our conclusions and future directions.

II. Related Works

Speciale et al. [10] implement and evaluate two book recommender systems using content-based and collaborative filtering techniques, using implicit user feedback. For experiments, two different data sources are used: a dataset regarding loans in all public libraries in Turin Italy and a dataset from Anobii social network. The authors acknowledge the benefits of using multi-source data, as it allows to include more users (beneficial for collaborative filtering) and to enrich book metadata (beneficial for content-based filtering).

Bouadjenek et al. [1] introduce a distributed collaborative filtering recommendation algorithm, which exploits and combines multiple data sources, aiming to improve the recommendations quality. The system is experimentally validated using two datasets, a bookmarking dataset and a movie dataset, using two training data ratings setups of 80% and 60% to predict the remaining ratings. Their experiments show the effectiveness of the algorithm compared to state-of-the-art recommendation algorithms.

Liu et al. [3] propose a multi-source information approach to improve conversational recommender systems. For experiments they use two datasets with conversations centered around movies. Each movie is represented as an embedding built from the keywords identified in the reviews. During the interaction with the user, the system identifies user preferences based on dialog context, tags, entities, and predicts movies which might interest the user. In case user does not like the recommendation (e.g. user has already seen the movie), the system dynamically updates the knowledge about user preferences and provides new recommendations.

Roy and Ding [7] present a movie recommender system which uses different types of users feedbacks such as likes, comments, tweets, in addition to movie features (title, plot, genre director, actors). Their experiments show that the most accurate results are obtained when all feedback data is combined to represent the movie feature.

Schoinas and Tjortjis [9] propose a product recommendation system based on multi-source implicit feedback. The authors utilize different sources of information in addition to user purchase history, such as the the number of times users viewed an item and added it to the cart, in order to estimate the user preferences for items. The interaction score is computed using specific weights for each observation: viewing products has lowest weight, as it does only indicate that the user was interested to learn more about the product, while by adding it to cart or purchasing it, there are stronger indications that user preferred the product.

Toumy [11] discusses the idea of relying only on single most similar user when making recommendations, considering that by relying also on next most similar users it is likely to overwhelm the customer and loose credibility. Although this is an interesting idea, considering our limited number of users and reviews in the experimental dataset, we decided to use a small set of similar users.

III. System Design

We propose a method for obtaining valuable literature books recommendations using multi-source reviews and collaborative filtering recommendations technique. For analysis, we use a set of books, for which we collected reviews from *Goodreads* and *Amazon* websites using our customized web scrapers.

A book review is a tuple:

$$r = (book, user, date, stars, content)$$

where $book$ represents the id which uniquely identifies a book, $user$ represents the id which uniquely identifies an user, $date$ represents the date when the review was written, $stars$ represents a scaled rating provided by user - expressed as a natural number in interval [1, 5], $content$ represents the content of the review.

At first, the book review shall be preprocessed in order to obtain the input for the recommender system. The input for the book recommender system is a tuple:

$$input = (book, user, emotions)$$

where $book$ represents the id which uniquely identifies a book, $user$ represents the id which uniquely identifies an user, and emotions is the frequency vector of emotions corresponding to the review content.

The emotions are extracted from the review content using the method we proposed in [4] which refers to applying standard NLP text preprocessing techniques (tokenization, lower casing, removal of stop words) to the review text, followed by a word-matching method of determining the emotions. We use an external file composed of adjectives and associated emotions. Following 35 emotions are considered: ’cheated’, ‘singled out’, ‘loved’, ‘attracted’, ‘sad’, ‘fearful’, ‘happy’, ‘angry’, ‘bored’, ‘esteemed’, ‘lustful’, ‘attached’, ‘independent’, ‘embarrassed’, ‘powerless’, ‘surprise’, ‘fearless’, ‘safe’, ‘adequate’, ‘belittled’, ‘hated’, ‘codependent’, ‘average’, ‘apathetic’, ‘obsessed’, ‘entitled’, ‘alone’, ‘focused’, ‘demoralized’, ‘derailed’, ‘anxious’, ‘ecstatic’, ‘free’, ‘lost’, ‘burdened’.

Then the collaborative filtering recommendation algorithm is applied (Algorithm 1).

Algorithm 1 User-Based Collaborative Filtering Recommendation Algorithm

```
Get user input review (book, user, emotions)
Create list of TOP 5 most similar users of user
Identify books enjoyed by similar users
if len(books) == nREC then
   Recommend books
else
   if len(books) > nREC then
      Recommend 5 random books from books
   else
      while len(books) < nREC do
         Add in books random book from the books dataset
      end while
      Recommend books
   end if
end if
```

The algorithm receives as input an user input review $(book, user, emotions)$. Next, the top 5 most similar users are determined, by comparing the emotions of $user$ with the emotions of all users who reviewed $book$.

Two users (A and B) are considered similar, if their emotion for $book$ are matching above a given $threshold$. The similarity is computed using Cosine Similarity measure:

$$sim(A, B) = \frac{emotions_A \cdot emotions_B}{||emotions_A|| \cdot ||emotions_B||}$$

TABLE I
REVIEWS ENTITY DESCRIPTION

Field name	Field Description
Id	Integer number which uniquely identifies the review in our dataset
Book Id	The id which uniquely identifies the reviewed book
Book URL	The *Goodreads* or *Amazon* URL for the book, depending on the review source
Author Id	The id which uniquely identifies the user who wrote the review
Review Stars	The rating provided by the user together with the review - natural number in interval [0, 5]
Review Date	The date when the review was written
Review Content	The text content of the review

where $emotions_A$ and $emotions_B$ are the frequency vectors of emotions corresponding to reviews contents provided by users A and B.

The book preferences of the top 5 most similar users are determined and stored in an array $books$. A book preference is a book that was reviewed by an user with a rating of 4 or 5 stars.

At this stage, depending on the requested number of recommendations to be received ($nREC$), 3 use cases can be identified:

1) $len(books) = nREC$: in this case, all books from $books$ are provided as recommendations.
2) $len(books) > nREC$: in this case, the array of similar users preferences contains more books than the requested number of recommendations to be received, which means that the system shall select $nREC$ books from $books$ and provide them as recommendations. The $nREC$ books are randomly selected from the $books$ array.
3) $len(books) < nREC$: in this case, the array of similar users preferences contains less books than the requested number of recommendations to be received, which means that the list of $books$ shall be completed in order to contain $nREC$ books. The list is completed by selecting random books from the books dataset, which are not already available in the $books$ array.

IV. EXPERIMENTS AND DISCUSSIONS

A. Dataset Overview

For analysis, we use a set of 1000 books rated on *Goodreads* website as ”Best Books Ever”, with the associated 129713 *Goodreads* reviews and 89401 *Amazon* reviews collected using our customized web scrapers.

For each review, we collected several parameters which are stored in a tabular file. The review entity with the collected parameters is presented in Table I.

On *Goodreads* website, we collected the first 6 reviews pages, with 30 reviews available per page, resulting in a maximum of 180 reviews per book, while on *Amazon* websites, we collected the first 10 reviews pages, with 10 reviews available per page, resulting in a maximum of 100 reviews per book.

TABLE II
REVIEWS DATASET STATISTICS

Statistic	Goodreads	Amazon
# of Collected Reviews	129713	89401
# of Reviews per book	[10, 180]	[1, 100]
Average # of Reviews per book	129	89
# of Reviews 5 stars	53305	54507
# of Reviews 4 stars	36547	22855
# of Reviews 3 stars	17559	7748
# of Reviews 2 stars	9572	2447
# of Reviews 1 stars	9183	1844
# of Reviews ’undefined’ stars	3557	-

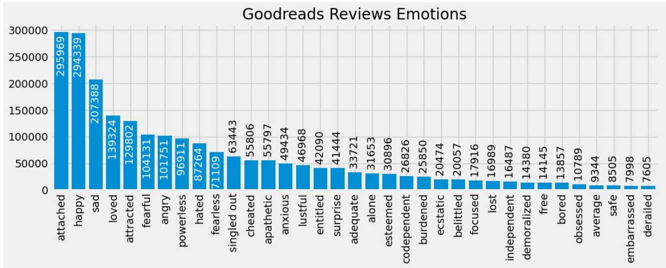

Fig. 1. Distribution of *Goodreads* Reviews Emotions

Each book has in average 129 *Goodreads* reviews (between 10 and 180) and 89 *Amazon* reviews (between 1 and 100) - Table II.

Based on user star ratings, 69% of *Goodreads* reviews are positive (4 or 5 stars), 14% are neutral (3 stars), 14% are negative (1 or 2 stars) and 3% are undefined, meaning that the user has only provided a review content, without assigning a scaled rating. For *Amazon* reviews, 86% are positive, 9% are neutral, 5% are negative.

In general, we observe that the majority of collected reviews are positive. When collecting the reviews, we used the default reviews sorting order, which refers to the fact that at first are displayed the reviews which obtained the most reactions from other users, in forms of likes or comments. This suggests that users tend to react rather on the positive reviews.

We observe a higher amount of emotions in *Goodreads* reviews compared to *Amazon* reviews (Figures 1 and 2) . Although this was anticipated considering the number of reviews collected from each of the websites (1.5 times more

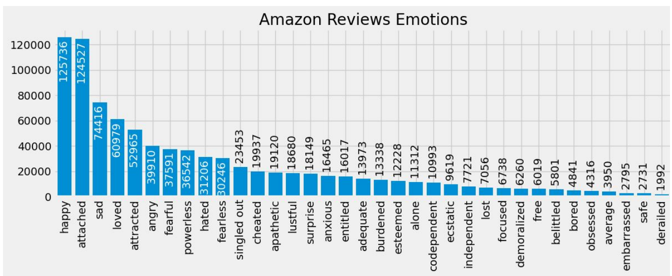

Fig. 2. Distribution of *Amazon* Reviews Emotions

Goodreads reviews than *Amazon* ones), the number of emotions identified in *Goodreads* reviews is 2.5 times greater than the number of emotions identified in *Amazon* reviews. This is justified by the fact that on dedicated books reviews websites (*Goodreads*), users tend to write more emotional elaborate reviews, compared to business oriented websites, such as *Amazon*.

B. Experimental Results and their discussion

We evaluated the performance our our system across 10 iterations. For each iteration, the reviews dataset was split as 80% training and 20% testing using stochastic sampling. Due to the fact that different number of reviews are available for each book (Table II), the training-testing split is done for each book reviews.

The training reviews dataset represents the reviews which are recorded in the system and are used for identifying the users which are similar to the user of interest. We have considered two types of training datasets:

1) *Single Source (SS)* which refers to usage of 80% *Goodreads* reviews for training
2) *Multi Source (MS)* which refers to usage of 80% *Goodreads* reviews and 100% *Amazon* reviews for training

The testing dataset remains the same in both cases (SS and MS) - 20% *Goodreads* reviews, and is used to simulate a population of users writing reviews and seeking for recommendations.

The training SS dataset contains 102854 reviews and training MS dataset contains 192255 reviews, while the testing dataset remains constant as 25955 reviews. This results in a total of 129775 recommendations being made with Recommendation Algorithm 1 in case of training SS and 129775 recommendations in case of training MS, as the system always recommends 5 books.

Let us define the parameters used for rigorous definition of our evaluation measures:

- *Recommendation space R* refers to the total number of possible recommendations, i.e. the total number of books available in the books dataset - in our case 1000.
- *User input space U* refers to the total number of user inputs u.

$$u = (book, user, emotions), where\ book \in R$$

- *Test space* T refers to the subset of the input space $T \subset U$ used for experimental evaluation.
- A *recommendation* $f(u)$ refers to the output recommendations obtained when applying the recommendation algorithm 1. The output is a set of 5 books $r_i \in R$.

$$f\colon T \to R^5$$

$$f(u) = (r_1, r_2, r_3, r_4, r_5)$$

- *Recommendations similarity* s refers to the similarity between recommendations $f_1(u)$ and $f_2(u)$ provided for the same user input u using (1) only information available in *Goodreads* reviews, respectively (2) information available in *Goodreads* and *Amazon* reviews for determining the recommendations.

$$s\colon R^5 \times R^5 \to [0,1]$$

s is determined using Jaccard index.

$$s(f_1(u), f_2(u)) = \frac{|f_1(u) \cap f_2(u)|}{|f_1(u) \cup f_2(u)|}$$

- *Total number of relevant recommendations* $TNRR$ refers to the amount of recommendations which are identified as relevant. A recommendation is considered relevant if the overall emotions of the reviewed book match the emotions of the recommended book $>$ *similarity_threshold*.

$$TNRR = \bigcup_{u \in T} (sim(u, f(u)) > thredhold)$$

$$sim(u, f(u)) \in [0,1]$$

The similarity is computed using cosine similarity measure:

$$sim(u, f(u)) = \frac{emotions_u \cdot emotions_{f(u)}}{||emotions_u|| \cdot ||emotions_{f(u)}||}$$

We propose following performance measures for evaluating our recommendation algorithms:

- *Average Recommendations Similarity* ARS is the average of the similarity between recommendations provided using (1) only information available in *Goodreads* reviews, (2) information available in *Goodreads* and *Amazon*.

$$ARS = \frac{1}{|T|} \sum_{u \in T} s(f_1(u), f_2(u))$$

- *Algorithm Branch Coverage* ABC determines the type of recommendation that was provided, considering following types:
  1) *General (GRL)*: Corresponds the case when the set of books enjoyed by similar users contains at least 5 books
  2) *Random Fill (RF)*: Corresponds to the case when the set of books enjoyed by similar users contains books, but they are less than 5, which means that the list of recommendations has to be completed using a random range of books from the dataset
  3) *Fully Random (FR)*: Corresponds to the case when the set of books enjoyed by similar users is empty, and the user is recommended a completely random list of 5 books from the dataset.
- $Relevance$ shows what proportion of recommendations $f(u)$ are identified as relevant recommendations.

$$Relevance = \frac{TNRR}{(|R|)^5}$$

The ARS value obtained for all iterations are presented in Table III, which shows we obtained an average of 10.84%

TABLE III
AVERAGE RECOMMENDATIONS SIMILARITY MEASURE

Iteration	Identical Recomm.	ARS
I1	14232	10.96
I2	14118	10.80
I3	13989	10.70
I4	14086	10.78
I5	14011	10.71
I6	13966	10.68
I7	14354	10.98
I8	14254	10.90
I9	14214	10.87
I10	14429	11.03

TABLE IV
USER TYPE STATISTICS

Iteration	Reg. user	New user
I1	20534	5421
I2	20698	5257
I3	20741	5214
I4	20731	5224
I5	20637	5318
I6	20701	5254
I7	20727	5228
I8	20735	5220
I9	20741	5214
I10	20756	5199

similarity between the recommendations provided using as basis the SS, respectively the MS training reviews. This is a slightly higher value compared to our previous research [6] (2.43%) where we compared the similarity between the recommendations obtained using collaborative filtering and content based filtering recommendations techniques. The increase is due to the rather similar recommendation techniques used: collaborative filtering method using single-source and multi-source reviews for defining the similar users to the user of interest.

The 25955 testing reviews, were written by an average of 20700 registered users and 5255 new users. The average value is considered for the 10 iterations, and we observe small changes in the number of users in each category from one iteration to another - Table IV. This shows that, although we have a limited number of reviews, they are written by a quite high number of registered users, i.e. users who wrote a review belonging to the testing dataset and have written other reviews before, available in the training dataset.

We appreciate that this is an interesting result, as we have a very high number of users who wrote several reviews in the datasets. When we scraped the *Goodreads* and *Amazon* websites, we collected the top reviews ordered by their relevance and popularity, and, as a result, a high part of the top reviews is written by certain individual users.

The ABC measure (Table V) shows that when using MS training reviews, a higher number of recommendations are provided using the *General (GRL)* method compared to the SS training reviews, more precisely an average of 21254 compared to 20026 (computed for the 10 iterations). On the other hand, the average number of recommendations provided using random book choices decreases, *Random Fill (RF)* from 2151 (Training SS) to 1185 (Training MS) and *Fully Random (RF)* from 3942 (Training SS) to 3681 (Training MS). This is a result of the benefits of using multi-source reviews, as it leads to a higher amount of similar users and books enjoyed by the similar users, which means that more target users can receive proper personalized recommendations.

TABLE V
ALGORITHM BRANCH MEASURE

	Training MS			Training SS		
Iter.	GRL	RF	FR	GRL	RF	FR
I1	21111	1125	3719	19919	2096	3940
I2	21196	1194	3742	19927	2176	4029
I3	21216	1174	3737	19964	2174	3989
I4	21396	1170	3563	20133	2175	3821
I5	21224	1156	3764	20025	2104	4015
I6	21250	1248	3634	19986	2213	3933
I7	21246	1176	3704	20013	2143	3970
I8	21251	1215	3676	20032	2192	3918
I9	21309	1206	3636	20143	2124	3884
I10	21339	1185	3631	20122	2116	3917

For the 25955 training reviews, a total of 199775 recommendations are provided using the proposed book recommendations algorithm. Table VI presents the average values of Relevance measures for the 10 iterations, depending on the *similarity threshold* value.

We observe that considering *similarity threshold* ≥ 0.5, almost all provided recommendations are considered as relevant, as this rather low similarity value permits emotional vectors of reviewed book and recommended book to be quite different. High number of relevant recommendations is observed also for *thresholds* 0.6, 0.7, 0.8, while for 0.9, 54.11% of the recommendations are identified as relevant for Training MS and 49.30% for Training SS.

We remark that, regardless of the chosen *threshold* value, the number of relevant recommendations resulted when using the Training MS reviews is higher than the number of relevant recommendations resulted when using the Training SS reviews. This shows that using the information obtained from multi-source reviews leads to more accurate recommendations, relevant for the preferences of the user of interest.

V. CONCLUSIONS

In this contribution, we presented a method for making valuable book recommendations using emotion information extracted from single-source and multi-source social media in order to identify the similarities between books.

For experiments, we use a set of 1000 books and associated reviews collected from *Goodreads* and *Amazon* websites, using our customized web scrapers.

Our analysis shows the benefits of using book emotional information extracted from multi-source reviews, by considering a series of recommender system performance measures. We observed low similarity between recommendations obtained from single-source and multi source reviews (10.84%), which

TABLE VI
RELEVANCE MEASURE (AVERAGE VALUES PER 10 ITERATIONS)

	Training MS			Training SS		
Threshold	**TNRR**	**not(TNRR)**	**Relevance**	**TNRR**	**not(TNRR)**	**Relevance**
0.5	129729	46	0.9996	129719.9	55.1	0.9995
0.6	129629.6	145.4	0.9988	129524.6	250.4	0.9980
0.7	126462.9	3312.1	0.9744	126013.5	3761.5	0.9710
0.8	113468.7	16306.3	0.8743	111713.2	18061.8	0.8608
0.9	70230.4	59544.6	0.5411	63981.3	65793.7	0.4930

is justified by the fact that approx. 80% of the recommendations were generated using the *General* recommendations approach, i.e. generation of top 5 most similar users preferences and picking a random 5 books selections from their preferences, which leads to different 5 books recommendations.

Approx. 79% of users considered as seeking for recommendations are registered users, which means that the system knows their reading history and preferences expressed through previously written reviews, thus resulting in more accurate, personalized recommendations.

The *Relevance* measure provides an insight of the improvements in recommendations obtained through multi-source reviews, as for high similarity values, 5% more recommendations are identified as relevant.

As future work, we plan to improve the method for computing the similarities between users, by considering that two users are similar if their reviews show other common features such as keywords or tags, in addition to the similar emotions.

Another future direction is in regards of evaluating the received recommendations: the recommendations received by registered users could be compared with user reading history (i.e. the books which user has rated before), and the relevance of the recommendations can be evaluated based on the common features existing between the user's reviews, e.g. emotions, keywords, tags. Considering the fact that we have a quite high ratio of registered users vs. new users (4:1), we appreciate this would give interesting results about user readings preferences.

REFERENCES

[1] M. R. Bouadjenek, E. Pacitti, M. Servajean, F. Masseglia, and A. E. Abbadi. A distributed collaborative filtering algorithm using multiple data sources. In *The Tenth International Conference on Advances in Databases, Knowledge, and Data Applications*, 2018.

[2] E. Hasan, M. Rahman, C. Ding, J. X. Huang, and S. Raza. Review-based recommender systems: A survey of approaches, challenges and future perspectives. *Computer Science*, 2405.05562, 2024.

[3] H. Liu, Q. Cao, X. Huang, F. Liu, C. Zhang, and J. An. Multi-source information contrastive learning collaborative augmented conversational recommender systems. *Complex and Intelligent Systems*, 2024.

[4] E.-R. Luţan and C. Bădică. Emotion-based literature book classification using online reviews. *Electronics*, 11(3412), 2022.

[5] E.-R. Luţan and C. Bădică. Emotion-based literature books recommender systems. In *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, volume 35, pages 275–280, 2023.

[6] E.-R. Luţan and C. Bădică. Experimenting emotion-based book recommender systems with social data. In *Information Technology for Management: Solving Social and Business Problems Through IT*, pages 164–182, 2024.

[7] D. Roy and C. Ding. Multi-source based movie recommendation with ratings and the side information. *Social Network Analysis and Mining*, 11(76), 2021.

[8] D. Roy and F. Shirazi. A review on multiple data source based recommendation systems. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1534–1539, 2021.

[9] I. Schoinas and C. Tjortjis. Musif: A product recommendation system based on multisource implicit feedback. *15th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, pages 660–672, 2019.

[10] A. Speciale, G. Vallero, L. Vassio, and M. Mellia. Recommendation systems in libraries: an application with heterogeneous data sources. In *7th International workshop on Data Analytics solutions for Real-LIfe Applications*, 2023.

[11] H. Toumy. Perfume project. 2019. https://hayatoumy.github.io/recommender_system/ [Accessed: (May 10, 2024)].