A Hybrid Approach for Online Novel Recommendation

Tien Nguyen
tien.t.nguyen04@sjsu.edu

***Abstract*—Book recommendation systems focus on helping users discover new books based on their previous preferences. However, on novel reading platforms such as Wattpad, where a user finds and reads novels directly on the websites, the interactions between the user and the platform become more complex. The user interactions are not only whether the users choose to find a book and leave a review, a novel reading platform can also keep track of a user's consumption of a specific novel. A user on such a platform usually consumes the novel in multiple sessions. Subsequently, a recommendation system on such a reading platform should be able to recommend the novels that are consumed by users as well as new novels at the same time. Previous work, such as NovelNet [1], shows that, by capturing these behaviors for each user session in a novel reading platform, the authors are able to improve the quality of their recommendation system. However, such recommendation systems might overlook the fact that multiple users can have similar sessions, and Sequence and Time Aware Neighborhood (STAN) [2] recommendation system shows that matching user sessions also produce high-quality recommendations. In our paper, we propose a hybrid model that captures the best of both worlds. Specifically, for each user session, we obtain the ranked recommendations from both NovelNet and STAN, or NovelNet and Association Rule (AR), We use a weighted sum to calculate the final rankings for the recommendations. We experiment with different sets of weights to determine the best weights, and we demonstrate that, in the best case, the hybrid model (NovelNet and STAN) provides users with more accurate and relevant recommendations.**

***Keywords*—** recommendation systems, session-based recommendation, weighted hybrid approach

I. Introduction

A book recommendation system is a tool that suggests books to users based on their past reading preferences and other relevant data such as genres or authors of a book. These systems can assist users discover new content that they may not have found. Historically, popular websites like BookCrossing, Amazon, and GoodReads have all offered assistance with book discovery. However, online novel reading platforms such as Wattpad have the added benefit of allowing users to read novels directly on their platforms. This feature sets them apart from the other book recommendation systems, and the approach for recommending books is different from the approach for recommending online novels. Online novel recommendations must take into account the user's process of reading a novel and predict whether they will continue reading a novel they have already started. An effective online novel recommendation helps users improve their reading experience and keep them engaged with the website.

There is only one paper that has proposed an approach for Online Novel Recommendation [1], which highlights the potential for further research in this area. This project aims to use the paper as a baseline and experiment with a hybrid approach to see if the hybrid recommendation system can provide users with more accurate and relevant recommendations. By using a hybrid approach, the system can combine the strengths of different recommendation methods to provide a more personalized experience for users.

II. Related Work

In recent years, there are a lot of researchers building **book recommendation systems** using different recommendation algorithms. Ekstrand and Kluver investigated how the distribution of authors' gender in the input data affects the performance of collaborative filtering algorithms [3]. Putri et al. built a model to incorporate the Convolutional Neural Network (CNN) algorithm for their book recommendation system [4]. However, the process of a user reading a book and predicting whether the user will continue reading a book they have already read is not adequately modeled in book recommendation systems. This kind of behavior is called **repeat consumption** where an item is consumed multiple times. Repeat consumption has been explored in many domains such as music, TV programs or E-commerce. Ren et al. proposed a model called RepeatNet, which had an encoder-decoder architecture, to handle repeat consumption in session-based recommendation tasks [5]. RepeatNet utilizes a repeat-explore mechanism integrated with Recurrent Neural Network (RNN) to more accurately capture the intention behind repeat or explore recommendations in a session. However, Li et al. have been the first researchers working on the online novel recommendation and so have been the first ones to refer to repeat consumption in online novel reading [1]. The authors proposed NovelNet for an online novel recommendation, which encodes interaction considering fine-grained interaction attributes and uses a pointer network with a pointwise loss to model the user repeat consumption behavior. The setting of NovelNet is similar to the setting of **session-based recommendation**. Session-based recommendation tries to suggest items to users based on their recent interactions. For example, we read Algebra 1, Algebra 2, and Algebra 3 in high school, but now we are reading Computer Science 1, Computer Science 2, and Computer Science 3, then session-based recommendation models will recommend books related to Computer Science. In session-based recommendation, the interaction involves only the item ID [6] whereas in online novel recommendation, the interaction comprises the item ID and additional interaction attributes.

III. Existing Solution

A. Architecture and Metrics

There are four main components that work together to generate novel recommendations [1]. They are Interaction Encoder, Sequence Encoder, Recommend New Novels and Recommend Consumed Novels (Figure 1). The Interaction Encoder takes the attributes of each item in the sequence and converts them into a set of vectors. These vectors are then concatenated to form an interaction vector that represents the sequence of interactions. The Sequence Encoder takes the sequence of interaction vectors as input and uses a bidirectional gated recurrent unit (GRU) to output both forward and backward vectors, which are then concatenated to form the final sequence of vectors. Given the output of the sequence encoder, we will use attention mechanism, activation function, and pointer network to compute scores of consumed and new novels. To evaluate NovelNet's

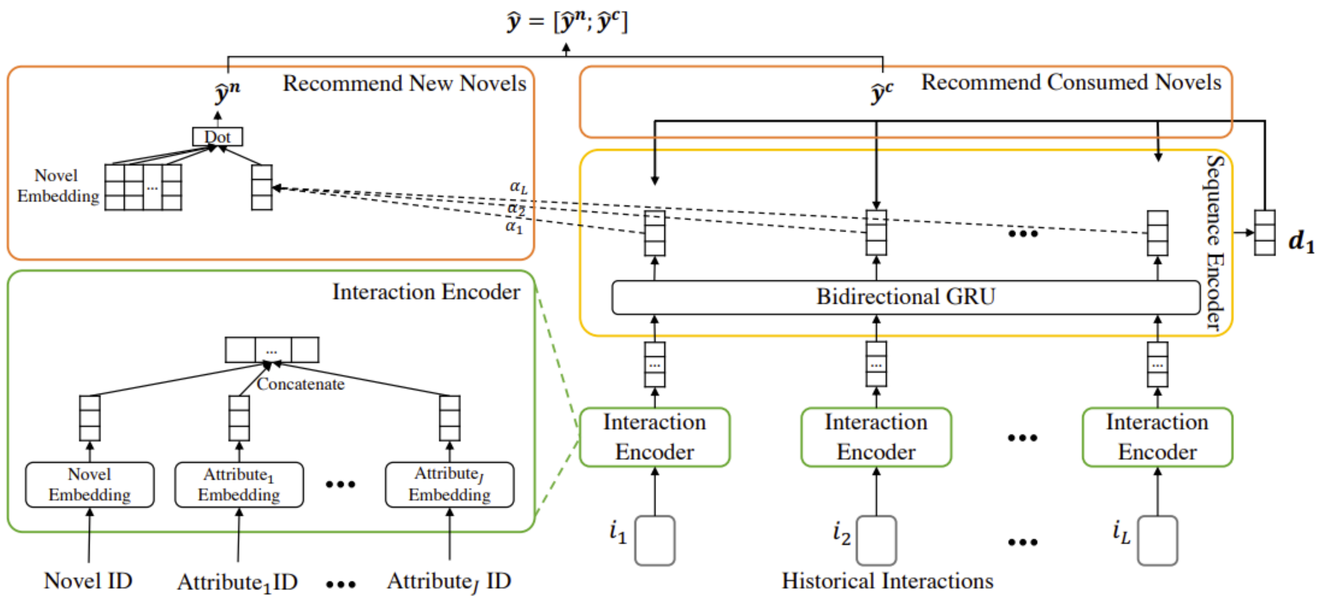

Fig. 1: The architecture of our NovelNet for online novel recommendation [1]

Table 1: A performance comparison between the baseline model and the reproduction of the baseline.

	MRR@k				Recall@k		
	@1	@5	@10	@20	@5	@10	@20
NovelNet (Li et.al)	47.02	51.33	52.00	52.37	58.36	63.43	68.72
NovelNet (Reproduction)	46.35	50.21	50.69	51.09	56.15	59.85	65.53

performance, Li et al. computed MRR@k (Mean Reciprocal Rank) and Recall@k. MRR@k is calculated as the average of the reciprocal ranks of the desired novels [1]. It measures how well the model is able to identify the most relevant item and rank it at the top. Recall@k measures the ability of the model to recommend relevant items within the top k recommendations. The larger k is, the higher the scores are since there is a higher chance that the correct label is covered in recommendations.

B. Reproducing the Baseline

A dataset was downloaded from Dropbox and it was already preprocessed. Library requirements were also installed. To see which interaction features had an effect on the model, Li et al. experimented with several interaction feature combinations. They concluded that their final model included these features: item_intro, item_read, item_real_read, item_count, item_read_duration, item_time_diff, and temporal_gaps [1]. Thus, we only used these features when reproducing the baseline. These authors provided source code for their paper [1]. However, to be able to run the codes, some parts of the codes needed to be modified such as changing library versions, the number of epochs, or file paths.

Recall@1 and MRR@1 are the same, so only MRR@1 is reported. From Table 1, we can see that the results are quite similar. The scores from the baseline model are slightly better than the ones from the reproduction model because the authors ran the model with 10 epochs while the reproduction model only used 3 epochs. The reason is that it took more than 17 hours to train the model with 10 epochs and after that, it crashed during validating and testing. We saved the recommendation list and scores in files, so we don't have to run the codes again. We think the scores are comparable, so we will use this reproduction model to be our baseline for our new approach.

IV. Proposal Approach

Many researchers claim that hybrid approaches improve the performance of their models. For example, Zhang et al. proposed a hybrid recommendation system on the well-known MovieLens dataset and concluded that the hybrid system performed better than the single one [7]. In the paper [1], the authors compare their proposed method with several representative methods in session-based recommendation which are other non-neural-network-based models: sequence time awareness neighborhood (STAN) and Association Rule (AR). We want to try to combine these above single models with NovelNet (hybrid recommendation system) to see if the combination model can give a better result. Since the baseline paper uses MRR@k and Recall@K for their evaluation, the new approach also uses them.

There are several approaches in the hybrid model and we used the weighted approach. In recommendation systems, the weighted approach is a method of combining multiple recommendation algorithms to produce a hybrid recommendation system. The processed data is inputted into individual models. Each of the models outputs two lists of recommended items and associated scores. In a weighted approach, the score of each item is a weighted sum of its score on each list. The recommendation output is the list of the items sorted by the weighted sum. A weighted hybrid approach can be used to improve the performance of session-based recommendation systems by combining the strengths of different algorithms. The diagram of weighted hybrid models is shown in Figure 3.

1) NovelNet + STAN: According to a recent study [8] on session-based recommendation systems, it was found that non-neural-network models offered more precise recommendations compared to other types of neural architectures. Unlike NovelNet, STAN is a non-neural network model that uses a session-based k-nearest neighbors approach to recommend novels. It looks at the previous novels read by the user in the current session and finds the most similar sessions from other users to recommend novels [2]. The ability of STAN to

	Recommendations_NN	Label	Recommendations_Stan
0	[68, 69, 214, 284, 422, 662, 1386, 504, 1106, 2477, 358, 522, 1357, 206, 189, 118, 35, 352, 1233, 609]	[69]	[69, 68, 22381, 9928, 22402, 22401, 22400, 22393, 15357, 13718, 22403, 22376, 10060, 18314, 20250, 20901, 22362, 22361, 21923, 1]
1	[72, 71, 70, 248, 567, 66, 253, 383, 225, 2175, 76, 73, 348, 268, 564, 265, 244, 364, 267, 370]	[73]	[72, 71, 253, 349, 73, 348, 70, 186, 248, 370, 185, 25, 564, 567, 218, 738, 181, 175, 268, 66]
2	[163, 167, 169, 164, 162, 165, 168, 166, 171, 170, 118, 214, 602, 522, 83, 412, 141, 712, 95, 662]	[169]	[164, 162, 163, 165, 306, 36, 609, 358, 206, 1115, 166, 251, 214, 2223, 720, 3856, 662, 3008, 3240, 310]
3	[169, 163, 164, 162, 167, 171, 165, 166, 168, 170, 118, 602, 522, 214, 83, 141, 712, 95, 412, 14]	[167]	[163, 169, 165, 170, 162, 171, 164, 3582, 3436, 12110, 8563, 2969, 2360, 13867, 1976, 609, 168, 3662, 18060, 2167]
4	[200, 214, 456, 243, 274, 260, 201, 2001, 722, 1808, 1830, 269, 219, 952, 953, 184, 650, 1829, 1580, 358]	[200]	[200, 201, 456, 202, 184, 178, 243, 269, 274, 2001, 176, 260, 952, 650, 413, 270, 953, 1830, 1808, 338]
5	[223, 214, 456, 200, 260, 201, 222, 1386, 243, 358, 274, 722, 1808, 202, 2001, 1580, 953, 338, 1830, 219]	[223]	[223, 222, 2076, 2905, 5120, 2299, 649, 3441, 992, 991, 794, 2714, 998, 4266, 2521, 6049, 380, 28769, 3626, 6282]
6	[297, 294, 298, 295, 214, 296, 206, 504, 456, 176, 358, 201, 1829, 25, 260, 274, 1808, 2001, 662, 1580]	[297]	[297, 295, 298, 650, 2306, 176, 178, 200, 1808, 2353, 456, 2113, 294, 1830, 2001, 239, 202, 184, 2179, 296]
7	[127, 351, 214, 456, 353, 201, 260, 352, 243, 358, 274, 354, 1808, 722, 200, 2001, 953, 1386, 952, 1830]	[127]	[127, 352, 351, 241, 353, 354, 197, 201, 207, 720, 736, 8494, 1106, 2315, 581, 126, 25, 358, 214, 5288]
8	[880, 200, 214, 651, 456, 243, 260, 201, 71, 722, 265, 219, 567, 1830, 383, 184, 952, 1808, 274, 2001]	[200]	[200, 880, 274, 456, 184, 178, 201, 269, 243, 2001, 651, 176, 260, 270, 413, 952, 650, 1829, 338, 953]
9	[1001, 71, 218, 214, 456, 243, 567, 383, 260, 248, 274, 201, 2175, 265, 76, 722, 268, 66, 184, 182]	[71]	[71, 1001, 248, 567, 253, 186, 66, 2175, 268, 383, 371, 218, 76, 364, 73, 265, 264, 254, 182, 583]
10	[248, 71, 66, 583, 567, 383, 225, 2175, 76, 268, 73, 265, 255, 564, 348, 267, 364, 254, 414, 244]	[253]	[253, 583, 186, 187, 71, 248, 66, 567, 1237, 73, 383, 268, 76, 348, 2175, 218, 900, 255, 254, 560]
11	[248, 71, 567, 383, 66, 253, 76, 2175, 225, 564, 268, 73, 364, 254, 267, 371, 735, 414, 900, 560]	[348]	[348, 253, 72, 73, 370, 186, 349, 2175, 185, 571, 71, 181, 383, 248, 567, 267, 564, 66, 900, 420]

Fig. 2: Novel lists are recommended by NovelNet and STAN

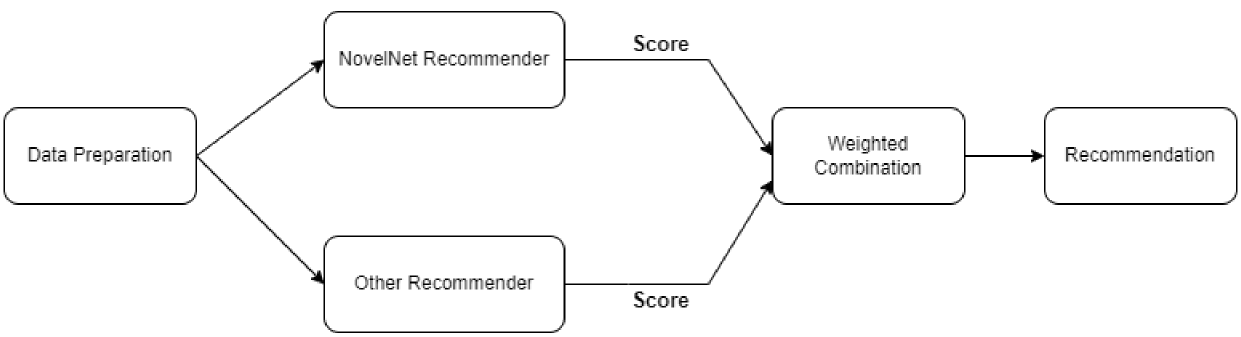

Fig. 3: Weight hybrid approach in online novel recommendation

consider similar sessions from other users can be seen as an advantage over NovelNet. Rather than solely relying on the behavior of the current user, STAN can potentially provide a more diverse set of recommendations by considering the behavior of other users who have similar interests.

2) NovelNet + AR: AR is also a non-neural network model. Sonali Gandhi and Monali Gandhi propose a model that combines collaborative filtering with AR to create a scalable and robust recommendation system [9]. Inspired by their work, we also want to try to combine NovelNet with AR model. The Simple Association Rules (AR) [10] algorithm is an algorithm that identifies how frequently two events occur together, such as "users who read... also read". This algorithm determines the significance of the rules by counting the number of times items i and j are found together in any user's session, allowing the algorithm to learn and identify important associations between these items [10].

V. Experiment

Since online Novel Recommendation was underexplored at the time Li et al. worked on it, there was no dataset available. Thus, the authors have created a dataset for online novel recommendation by collecting data from a popular online novel reading platform. The dataset was split into training, valid and test set. Training set has 91,311 users and 18,487 novels. Valid set has 34,857 users and 10598 novels. Test set has 38,338 users and 10,265 novels. The dataset has several interactive features such as novelID, item_intro, item_read, item_real_read, item_read_duration, item_time_diff, item_count, temporal_gaps, collect, index_gap, quality and item_tag [1]. We ran NovelNet, STAN and AR models separately to get the recommendation lists and scores. Then, we normalized the scores. We used the grid search technique to try different weight combinations to compute the ranking scores. The sum of the weights is 1. For example, if we give NovelNet a weight of 0.75, then the weight of STAN would be 0.25. Finally, we computed MRR@k and Recall@k where k is 1,5,10, and 20. The parameters of NovelNet were set the same as the parameters of NovelNet baseline. For STAN and AR, we refer to the session-based recommendation framework, session-rec [$https://github.com/rn5l/session-rec$] to get the recommendation lists and scores.

	Recommendations	Label
0	[69, 68, 214, 284, 422, 662, 1386, 504, 1106, 2477, 358, 522, 1357, 206, 189, 118, 35, 352, 1233, 609]	[69]
1	[72, 71, 70, 248, 567, 66, 253, 383, 225, 73, 2175, 348, 76, 268, 564, 265, 244, 364, 370, 267]	[73]
2	[164, 163, 167, 162, 169, 165, 168, 166, 171, 170, 118, 214, 602, 522, 83, 662, 412, 141, 712, 95]	[169]
3	[169, 163, 164, 162, 165, 167, 171, 166, 170, 168, 3582, 3436, 12110, 8563, 2969, 118, 2360, 13867, 602, 522]	[167]
4	[200, 214, 456, 201, 243, 274, 260, 2001, 269, 1808, 184, 722, 1830, 952, 953, 219, 650, 1829, 1580, 358]	[200]
5	[223, 214, 222, 456, 2076, 200, 260, 201, 1386, 243, 2905, 358, 5120, 2299, 274, 722, 1808, 202, 2001, 1580]	[223]
6	[297, 294, 298, 295, 296, 214, 176, 456, 206, 504, 1808, 358, 201, 2001, 1829, 25, 260, 274, 662, 1580]	[297]
7	[127, 351, 214, 352, 353, 456, 354, 201, 260, 243, 358, 274, 1808, 722, 200, 2001, 953, 1386, 952, 1830]	[127]
8	[200, 880, 214, 651, 456, 274, 243, 201, 260, 71, 184, 722, 2001, 265, 952, 219, 567, 1830, 383, 1808]	[200]
9	[71, 1001, 218, 214, 248, 456, 567, 243, 383, 2175, 66, 268, 76, 265, 260, 274, 182, 201, 722, 184]	[71]
10	[583, 253, 248, 71, 66, 567, 383, 225, 2175, 76, 268, 73, 265, 255, 564, 348, 254, 267, 364, 414]	[253]
11	[348, 248, 71, 567, 383, 253, 66, 76, 2175, 225, 73, 564, 268, 364, 267, 254, 371, 735, 414, 900]	[348]

Fig. 4: Novel lists are recommended by the hybrid model (0.75*NN+0.25*STAN)

Table 2: A performance comparison between the baseline model and the hybrid models.

	MRR@k				Recall@k		
	@1	@5	@10	@20	@5	@10	@20
1.0 * NovelNet	**46.35**	**50.21**	**50.69**	**51.09**	56.15	59.85	65.53
1.0 * STAN	43.01	48.65	49.32	49.67	57.50	62.55	67.56
1.0 * AR	40.45	43.24	43.7	44.07	48.1	51.66	56.95
0.9 * NovelNet + 0.1 * STAN	46.18	50.21	50.68	51.07	56.36	60.06	65.57
0.75 * NovelNet + 0.25 * STAN	45.91	50.13	50.63	51.01	56.52	60.35	65.58
0.5 * NovelNet + 0.5 * STAN	44.74	49.62	50.17	50.53	56.85	61.09	66.07
0.25 * NovelNet + 0.75 * STAN	43.62	49.21	49.81	50.14	57.46	62.0	66.77
0.1 * NovelNet + 0.9 * STAN	43.2	49.1	49.71	50.1	**58.4**	**62.6**	**68.23**
0.9*NovelNet+0.1*AR	46.17	50.14	50.65	51.04	56.23	60.2	65.76
0.75*NovelNet+0.25*AR	45.91	50.0	50.55	50.94	56.23	60.49	66.02
0.5*NovelNet+0.5*AR	45.31	49.56	50.13	50.51	55.97	60.37	65.87
0.25*NovelNet+0.75*AR	44.09	48.31	48.89	49.24	54.94	59.28	64.25
0.1*NovelNet+0.9*AR	42.11	45.89	46.46	46.85	52.42	56.69	62.27

VI. Results and Discussion

1) NovelNet and STAN individual performance: Each row of figure 2 shows the ranked novel recommendations for one user session. The number sequence on the leftmost column is recommendations emitted by NovelNet, and the number sequence on the rightmost column is recommendations produced by STAN. Both sequences are ordered from high ranked to lower ranked recommendations. The middle column is the label indicating the novel chosen by the user.

The figure illustrates the complementary nature of NovelNet and STAN. Although both models produce similar predictions, there are instances where one model misses or ranks a label low while the other ranks it highly. For example, in Session 0, both models make similar predictions that are close to the label. However, NovelNet places novel 69, which the user eventually chose, in second place, while STAN ranks it as the top recommendation. In Session 1, both models mispredict the user's choice. NovelNet ranks the label, novel 73, at 12th place, while STAN ranks it at 5th place. The interesting cases occur in Session 3 and Session 11, where one model does not recommend the user's choice, but the other model does. In Session 3, NovelNet places the user's choice, novel 167, in fifth place, while the STAN model does not have it in its recommendation. In Session 11, while NovelNet does not recommend novel 348, it is ranked as the top recommendation by STAN. These examples demonstrate that there are instances where NovelNet or STAN misses recommending the target novel.

The result shows that different approaches in producing recommendations might produce different results, and users' behaviors might fit one model better than another. This provides evidence that combining the results of all recommendation models for each user session allows us to capture all user's behavior models and that can lead us to a better recommendation.

2) NovelNet and AR individual performance: Similar to NovelNet and STAN, the same relation is shown for NovelNet and AR as shown in Figure 5 in the Appendix section. There are cases where NovelNet is able to place the target label at the high ranks while AR fails to do so. There are also cases where NovelNet fails to place the user's choice in its recommendations while AR places the choice on top of its recommendations. This clearly shows that having two complementary models might help to improve the final prediction.

3) The hybrid model of NovelNet and STAN: Figure 4 illustrates the effect of capturing the results from NovelNet and STAN in each user session using the weighted sum. NovelNet scores have a weight of 0.75 while the STAN scores have a weight of 0.25. In other words, the result shows the effect when we weigh NovelNet's recommendations more than STAN's recommendations. In Figure 2, NovelNet failed to recommend the last two novels. As we can see now, the target novels are all on the recommendation list. Out of 12 recommendations, there are 8 recommendations where the first novel matches the label. For the same set of recommendations outputted by NovelNet only, only 4 out of 12 recommendations have the top item that matches the label. For row 2 and row 3 in our weighted model result, the labels are at position 5 and 6 respectively while in STAN, the labels do not appear on the respective recommendation.

4) The hybrid model of NovelNet and AR: Figure 6 in the Appendix illustrates the similar effect of the hybrid model of NovelNet and AR. The hybrid model is able to improve from the individual performance of NovelNet and AR. Specifically, the hybrid model is able to place the target label within the first six places in every session, while individually, NovelNet and AR might miss the target label entirely in some user sessions.

5) Discussion: The results from Table 2 highlight that the hybrid models of STAN and NovelNet are able to achieve higher Recall@k (for k = 5, 10, 20) than the baseline model. The result implies that the hybrid models are good at including relevant novels in recommendation lists. Specifically, the hybrid system (0.1*NovelNet + 0.9*STAN) is able to achieve a Recall@20 of 68.23%, which is a 3% increase over the baseline of 1.0*NovelNet. This result shows the advantage of a hybrid model where it reduces the chance of missing out from the user choices by increasing the chance of having top recommendations of each model in the final recommendation. We believe that this is the key reason for the performance improvement.

Individually, NovelNet and STAN capture users' behaviors in different dimensions. NovelNet takes both the novel ID and interaction features as input, which sets it apart from STAN and AR which only take the novel ID as input. By using bidirectional GRU to capture the user interaction sequence, NovelNet is able to take into account the complexity of user interaction when reading a novel that they have already started. Thus, NovelNet is better at ranking relevant novels at higher positions than STAN or AR. As shown in previous subsections, one model can entirely miss the target label in its recommendation. By capturing and combining different dimensions of users' behaviors, we reduce the chance of missing out on the target label when a user behaves in a way that could not be captured by one model. As a result, the best combination of NovelNet and STAN is able to outperform NovelNet in terms of Recall@k. However, not all hybrid models perform equally well in terms of Recall@k. Simply combining any two models would not guarantee to improve upon the baseline. Specifically, the best combination of NovelNet and AR, the 0.75*NovelNet+0.25*AR combination, does perform better than NovelNet in terms of Recall@k (for k = 5, 10, 20), but the margin of improvement is much less than 1%. Even though NovelNet and AR are different approaches in capturing users' behaviors, we believe that the difference is not as meaningful as in the case of NovelNet and STAN, and NovelNet and AR are not complementary enough to each other to make a significant improvement upon the baseline NovelNet model. Additionally, STAN outperforms AR and NovelNet in terms of Recall@k, as it is designed to take into account the behavior of other users with similar interests. We think adding more models to NovelNet and AR could improve the hybrid model further. However, this is out-of-scope of this paper, and we will leave this for future work.

Results in Table 2 also show that there is no dominant model. In other words, simply putting more weights on the state-of-the-art model does not guarantee to improve the Recall@k. To be specific, in the combination of NovelNet and STAN, giving STAN overwhelmingly more weights produces the best result. In contrast, in the combination of NovelNet and AR, putting more weights on NovelNet produces the best result, and when the weight for NovelNet is decreased from 0.9 to 0.1, the Recall@k of the hybrid model also decreases. This result implies that putting more weights on a specific model would not guarantee to improve in the Recall@k metric.

The MRR@k result in Table 2 shows the inherent disadvantage of using the weighted sum approach for the hybrid model. In the previous subsection, we show that there are a lot of cases where one model ranks a novel highly while the other model ranks the novel lowly or does not even include the novel in the recommendation. This case would drastically decrease the score of the novel in the weighted hybrid model compared to its score in the model that ranked the novel highly. The result in Figure 4 and Figure 6 show that the hybrid model in such cases would not place the novel at a high ranking. Therefore, while the hybrid model is able to recommend the novel, the fact that the novel is not highly ranked degrades the performance of the hybrid model in terms of MRR@k. Data in Figure 2 and Figure 5 shows that this phenomenon occurs rather frequently, and this is subsequently reflected in Table 2, where all hybrid models are not able to improve the MRR@k on top of the baseline NovelNet model. One should not expect the hybrid model to improve the ranking of the users' choices on top of both models, but they should find the users' choices among the recommendations. Table 2 also illustrates the trend or MRR@k results when using hybrid models. When increasing the weight for NovelNet from 0.1 to 0.9 in the combination of NovelNet and STAN. there is a significant increase in the MRR@k of hybrid models. This is because NovelNet itself has the highest MRR@k, which proves its effectiveness in recommending novels to users. Therefore, in order to prioritize relevant novels with higher rankings, we would give more weight to NovelNet, such as 0.9*NovelNet + 0.1*STAN. This would allow us to leverage the strengths of both models and provide users with more accurate and personalized recommendations.

VII. Conclusion

In this paper, we proposed a weighted hybrid model for an online novel recommendation by combing NovelNet and STAN or NovelNet and AR. While the hybrid models (NovelNet and STAN) are better at recommending all relevant items, it needs improvement in terms of ranking them higher on the list. NovelNet itself is better at ranking relevant novels to the top, so we would give more weight to NovelNet in the hybrid model.

There are several future works that could be done. First, if we have more computation power, we could run the NovelNet with more epochs. This might improve the ability of the hybrid model being able to rank the target label higher in the list. Secondly, we could try other hybrid approaches such as switching or cascade or we could try to combine NovelNet with another recommendation system algorithms. Last but not least, one of the four major modules in the NovelNet model is Sequence Encoder. Li et al. used bidirectional GRU which is a type of recurrent neural network (RNN) for sequence encoder [1]. Another type of RNN used in recommendation systems is the Long Short-Term Memory (LSTM) network. LSTM has more parameters than GRUs and can be slower to train. However, they are also can capture more complex relationships in the data. For example, LSTM has a forget gate parameter that is able to decide which information can be removed or forgotten when training. GRUs, on the other hand, have fewer parameters and can be faster to train, but may not perform as well on tasks that require modeling long-term dependencies.

References

[1] Y. Li, C. Yin, Y. He, G. Xu, J. Cai, L. Luo, and S.-h. Zhong, "Modeling user repeat consumption behavior for online novel recommendation," in *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022, pp. 14–24.

[2] D. Garg, P. Gupta, P. Malhotra, L. Vig, and G. Shroff, "Sequence and time aware neighborhood for session-based recommendations: Stan," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 1069–1072.

[3] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, and D. Kluver, "Exploring author gender in book rating and recommendation," in *Proceedings of the 12th ACM conference on recommender systems*, 2018, pp. 242–250.

[4] A. Putri, Z. A. Baizal, and D. Rischasdy, "Book recommender system using convolutional neural network," in *2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*. IEEE, 2022, pp. 1–6.

[5] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. De Rijke, "Repeatnet: A repeat aware neural recommendation machine for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4806–4813.

[6] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–38, 2021.

[7] H.-R. Zhang, F. Min, X. He, and Y.-Y. Xu, “A hybrid recommender system based on user-recommender interaction,” *Mathematical Problems in Engineering*, vol. 2015, 2015.

[8] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, “Empirical analysis of session-based recommendation algorithms: A comparison of neural and non-neural approaches,” *User Modeling and User-Adapted Interaction*, vol. 31, pp. 149–181, 2021.

[9] S. Gandhi and M. Gandhi, “Hybrid recommendation system with collaborative filtering and association rule mining using big data,” in *2018 3rd International Conference for Convergence in Technology (I2CT)*. IEEE, 2018, pp. 1–5.

[10] M. Ludewig and D. Jannach, “Evaluation of session-based recommendation algorithms,” *User Modeling and User-Adapted Interaction*, vol. 28, pp. 331–390, 2018.

APPENDIX

	Recommendations_NN	label	Recommendations_Ar
0	[128, 129, 265, 66, 71, 567, 248, 383, 25, 808, 504, 255, 268, 468, 358, 371, 206, 348, 225, 267]	[128]	[128, 129, 22376, 22401, 22400, 22393, 15357, 22381, 13718, 10060, 22355, 18314, 20250, 20901, 22362, 22361, 22356, 22402, 1, 21923]
1	[197, 214, 198, 456, 201, 243, 274, 260, 269, 358, 184, 200, 176, 1829, 722, 1808, 178, 952, 2001, 219]	[197]	[197, 176, 555, 33059, 127, 47745, 338, 662, 214, 4055, 260, 788, 537, 25, 380, 47080, 5399, 254, 23964, 2847]
2	[223, 214, 456, 200, 260, 201, 222, 1386, 243, 358, 274, 722, 1808, 202, 2001, 1580, 953, 338, 1830, 219]	[223]	[223, 5120, 2076, 178, 222, 40776, 2276, 1630, 2671, 129, 10001, 35973, 579, 200, 4266, 28769, 28768, 791, 380, 3191]
3	[856, 201, 1136, 1137, 1138, 214, 456, 455, 260, 243, 358, 200, 1808, 274, 269, 176, 206, 722, 2001, 953]	[1138]	[1138, 184, 200, 201, 456, 2129, 5916, 1829, 269, 260, 1864, 856, 915, 2001, 2113, 2306, 1157, 1140, 214, 1827]
4	[164, 169, 162, 163, 167, 166, 165, 168, 171, 170, 118, 214, 602, 522, 83, 412, 712, 141, 95, 662]	[164]	[162, 3856, 609, 129, 376, 3128, 19339, 7087, 6866, 1449, 1484, 23293, 166, 2701, 3947, 380, 660, 296, 164, 2595]
5	[199, 214, 456, 358, 1386, 201, 662, 422, 260, 243, 206, 504, 352, 338, 274, 176, 1233, 202, 1808, 1580]	[199]	[7254, 199, 206, 1005, 2863, 21265, 914, 25, 1754, 865, 8126, 3016, 1086, 1972, 827, 2286, 34220, 2839, 866, 358]
6	[2662, 72, 244, 25, 214, 71, 567, 946, 383, 243, 456, 2175, 265, 66, 248, 268, 76, 264, 182, 260]	[244]	[244, 370, 71, 567, 76, 2175, 253, 66, 3589, 248, 268, 383, 129, 25, 364, 900, 358, 371, 73, 255]
7	[370, 73, 248, 71, 66, 567, 383, 2175, 76, 348, 225, 253, 268, 460, 900, 265, 255, 564, 583, 264]	[73]	[73, 71, 253, 567, 66, 255, 248, 268, 414, 383, 184, 348, 2175, 182, 264, 371, 370, 254, 76, 908]
8	[269, 504, 456, 214, 243, 260, 201, 71, 184, 274, 265, 200, 567, 722, 248, 383, 1808, 1830, 66, 952]	[504]	[504, 9, 247, 888, 908, 218, 4132, 1822, 1580, 3178, 5381, 798, 4047, 278, 363, 1208, 933, 2349, 642, 475]
9	[9203, 9206, 9204, 1386, 793, 933, 1868, 2227, 78, 214, 1100, 4831, 2079, 9205, 1574, 206, 118, 504, 662, 176]	[6402]	[6402, 1100, 114, 1868, 3178, 5056, 22070, 6378, 66, 7230, 3882, 805, 2079, 1577, 803, 7805, 10760, 9415, 3953, 347]
10	[4172, 5371, 16012, 12415, 1631, 650, 13246, 14275, 3391, 7848, 118, 2193, 206, 239, 602, 16010, 468, 712, 83, 1172]	[5373]	[5373, 5459, 4172, 5461, 17264, 590, 9810, 18220, 5300, 18365, 463, 1095, 660, 14867, 1106, 5452, 6293, 204, 674, 2525]
11	[10992, 1344, 391, 648, 908, 5161, 1842, 387, 8400, 187, 1643, 12217, 10993, 1333, 383, 231, 214, 6281, 1386, 206]	[4027]	[4027, 2902, 10992, 21978, 3256, 1353, 7261, 5795, 2349, 11017, 13143, 1319, 6709, 34079, 10873, 236, 1807, 2944, 23480, 8152]
12	[167, 169, 164, 162, 163, 165, 166, 171, 168, 170, 118, 602, 522, 83, 712, 214, 412, 141, 95, 136]	[163]	[167, 162, 164, 169, 163, 165, 170, 168, 166, 22393, 15357, 22381, 13718, 22376, 10060, 1, 22401, 18314, 20250, 20901]
13	[248, 253, 291, 383, 225, 71, 564, 567, 66, 186, 72, 73, 268, 76, 267, 181, 735, 187, 348, 900]	[225]	[291, 175, 895, 423, 1883, 196, 213, 248, 71, 567, 8080, 2317, 66, 268, 76, 371, 2175, 383, 573, 946]
14	[670, 296, 660, 653, 654, 14, 408, 269, 669, 664, 667, 659, 663, 668, 661, 666, 118, 655, 665, 602]	[653]	[670, 25, 351, 352, 669, 3464, 12183, 5592, 2210, 3721, 6458, 14, 468, 1028, 13619, 2788, 1410, 1962, 14244, 4576]

Fig. 5: Novel lists are recommended by NovelNet and AR

	Recommendations	label
0	[128, 129, 265, 66, 71, 567, 248, 383, 25, 808, 504, 255, 268, 468, 358, 371, 206, 348, 225, 267]	[128]
1	[197, 214, 198, 456, 176, 260, 201, 243, 274, 269, 358, 184, 200, 1829, 722, 1808, 178, 952, 2001, 219]	[197]
2	[223, 5120, 2076, 214, 222, 178, 40776, 2276, 1630, 200, 2671, 129, 10001, 35973, 579, 456, 4266, 28769, 28768, 260]	[223]
3	[856, 201, 1138, 1136, 184, 200, 456, 1137, 214, 260, 269, 455, 2001, 2129, 243, 5916, 358, 1808, 274, 1829]	[1138]
4	[162, 164, 169, 163, 167, 166, 165, 168, 171, 3856, 609, 129, 170, 376, 3128, 19339, 7087, 6866, 1449, 1484]	[164]
5	[199, 7254, 206, 214, 1005, 2863, 21265, 914, 25, 1754, 865, 8126, 3016, 1086, 1972, 827, 358, 456, 2286, 34220]	[199]
6	[244, 2662, 72, 25, 71, 567, 76, 2175, 370, 66, 383, 248, 268, 214, 253, 946, 3589, 243, 456, 265]	[244]
7	[73, 370, 71, 248, 66, 567, 383, 253, 2175, 348, 255, 76, 268, 225, 264, 460, 900, 265, 564, 583]	[73]
8	[269, 504, 456, 214, 243, 260, 201, 71, 184, 274, 265, 200, 567, 722, 248, 383, 1808, 1830, 66, 952]	[504]
9	[9203, 9206, 9204, 6402, 1100, 114, 1868, 3178, 5056, 22070, 6378, 2079, 1386, 66, 7230, 3882, 805, 1577, 803, 7805]	[6402]
10	[4172, 5371, 16012, 12415, 5373, 5459, 1631, 650, 13246, 14275, 5461, 3391, 7848, 118, 2193, 206, 239, 602, 16010, 468]	[5373]
11	[10992, 1344, 391, 648, 908, 4027, 5161, 2902, 1842, 387, 8400, 187, 1643, 21978, 3256, 12217, 10993, 1333, 1353, 7261]	[4027]
12	[167, 169, 162, 164, 163, 165, 166, 171, 168, 170, 118, 602, 522, 83, 712, 214, 412, 141, 95, 136]	[163]
13	[291, 248, 253, 383, 71, 225, 567, 66, 564, 186, 268, 72, 73, 175, 76, 267, 181, 895, 735, 187]	[225]
14	[670, 296, 14, 660, 653, 654, 25, 408, 669, 269, 351, 352, 664, 667, 3464, 659, 12183, 5592, 663, 2210]	[653]

Fig. 6: Novel lists are recommended by the hybrid model (NovelNet and AR)