

Interpretable Contextual Team-aware Item Recommendation: Application in Multiplayer Online Battle Arena Games

ANDRÉS VILLA, VLADIMIR ARAUJO*, FRANCISCA CATTAN, and DENIS PARRA*, Pontificia Universidad Católica de Chile

The video game industry has adopted recommendation systems to boost users interest with a focus on game sales. Other exciting applications within video games are those that help the player make decisions that would maximize their playing experience, which is a desirable feature in real-time strategy video games such as Multiplayer Online Battle Arena (MOBA) like as DotA and LoL. Among these tasks, the recommendation of items is challenging, given both the contextual nature of the game and how it exposes the dependence on the formation of each team. Existing works on this topic do not take advantage of all the available contextual match data and dismiss potentially valuable information. To address this problem we develop TTIR, a contextual recommender model derived from the Transformer neural architecture that suggests a set of items to every team member, based on the contexts of teams and roles that describe the match. TTIR outperforms several approaches and provides interpretable recommendations through visualization of attention weights. Our evaluation indicates that both the Transformer architecture and the contextual information are essential to get the best results for this item recommendation task. Furthermore, a preliminary user survey indicates the usefulness of attention weights for explaining recommendations as well as ideas for future work. The code and dataset are available at: <https://github.com/ojedaf/IC-TIR-Lol>.

CCS Concepts: • **Information systems** → **Recommender systems**; • **Computing methodologies** → *Machine learning*.

Additional Key Words and Phrases: Item Recommendation, Deep Learning, MOBA Games

ACM Reference Format:

Andrés Villa, Vladimir Araujo, Francisca Cattan, and Denis Parra. 2020. Interpretable Contextual Team-aware Item Recommendation: Application in Multiplayer Online Battle Arena Games. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20)*, September 21–26, 2020, Virtual Event, Brazil. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3383313.3412211>

1 INTRODUCTION

The annual report of Newzoo shows that global e-sports revenues and its audience will grow to \$1.1 billion and 495 million people in 2020, respectively [16]. MOBA is one of the most significant social gaming genres contributing to that growth. An example of this phenomenon is the League of Legends World Championship, which was the biggest tournament of 2019 with more than 105 million hours live on Twitch and YouTube. This type of games can rise up to 64 million active players per month worldwide, with over a billion monthly gaming hours [23]. Most of its popularity is due to social dynamics that motivate new players to engage in long-term commitment to the game [24].

In this context, several studies have leveraged artificial intelligence to recommend videogames [8], as well as to improve the personal experience of players, in applications like difficulty adjustment [2, 22], intelligent agents [17], and in-game recommender systems [3, 7]. Regarding recommender systems, one challenge is to suggest the users the most

* Also with Millennium Institute Foundational Research on Data, IMFD.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

Manuscript submitted to ACM

suitable set of items for their characters considering the context of a specific match. Existing approaches attempt to solve the problem simply by using character descriptors, thus ignoring relevant contextual information from matches. Also, they focus on recommending a single character. However, such recommendations do have a common goal and a group (team) recommendation may be appropriate.

In this paper, we focus on exploiting contextual information present in each match in order to generate richer representations of the characters, thus improving item recommendations for each participant in a team. Such information corresponds to the specific champion used, the role, and the team that each player belongs to. Inspired by the Transformer neural architecture [25], we propose to use its encoder layer to model the relationships between the descriptor vectors for each of the aforementioned features. Additionally, its multi-headed attention mechanism helps to acquire information that makes it possible to interpret what the model is focusing on. We extensively evaluate our system by conducting comparisons with state-of-the-art methods on a real and challenging dataset. We also conduct a preliminary user survey to gain insights about the recommendation performance and the usefulness of attention-based explanations.

The contributions of our work are: (i) Introducing the method TTIR (Team-aware Transformer-based Item Recommendation), which significantly outperforms existing works on several ranking metrics and provides support to the importance of the team and role contexts, (ii) Designing a visual explanation mechanism in order to help users understand and follow team-aware item recommendations, and (iii) Providing ideas for future work by conducting a preliminary user survey to gain insights from the quality of the recommendation and the explanations provided.

2 MOBA GAMES: OVERVIEW AND RECOMMENDATION PROBLEM

The MOBA genre corresponds to strategy video games in which each player controls a single character as part of a team competing against another team of players on a battle arena. Among the different video games cataloged under the MOBA genre, League of Legends (LoL) has dominated the market since 2012 and is considered one of the most popular electronic games worldwide. Each match consists of ten players divided into two teams of five people (*Blue* and *Red*). The main goal of the game is to battle head-to-head across a fixed battlefield to destroy the base of the enemy team. Each player selects their unique champion from a total of more than 146 available, according to the player's preferences and the composition of the allied team.

The pace of the game is encouraged by an in-game currency reward system, which is used to buy items that increase the statistics and performance of the champion. This is one of the main ways for the players to increase their attack and defense power, thereby increasing their contribution to winning the game. Players can choose up to six items from approximately 233 available. However, several of the items can be combined to obtain a total of stronger 89 finished items, which we use in this work. Both the choice of champions and the items pose the challenge of the number of possible combinations, which users face making decisions based on experience. This is particularly complex for new players and presents interesting opportunities for in-game recommender systems.

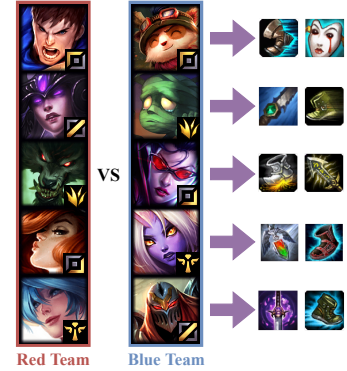


Fig. 1. Example of two teams matchup and item recommendations for the Blue team. Symbols on the bottom right corner of each champion represents their role.

3 RELATED WORK

In-game recommendation for MOBA games. In recent years, methods for in-game recommendations have received interest, where most works focused on character suggestion [7, 11, 20]. However, there has been little work on item

recommendation, recently showing two approaches based on data mining methods. One for the recommendation of the future item given an initial set of items [14] and another for the recommendation of a fixed item set [3]. We closely follow the methodology from [3]; however, unlike their approach that uses only a few attributes of the data, we leverage meaningful contextual information about the game such as the allies, enemies, and the role of champions.

Group recommendation. The increase in social networking increased the importance of group recommendations in various domains [1, 15]. The most common systems are applied to recommendation of movies [21], music [10], and travel [6]. All those systems attempt to recommend products or services to a group that has a common aim at a particular moment while increasing the individual satisfaction of each user. Another interesting domain is videogames, which is still an open issue. Recently, a multi-profile team-based recommender system for PvP games was proposed [12] to help teams improve by suggesting play styles and weapons to use. That approach is not directly related to our proposal because it was applied for a MMOG game using user profile data. Instead, we focus on the MOBA genre with an approach that does not use information from the user but from the characters in the game for item recommendations.

Recommendation systems with Transformer. Recently, the Transformer [25] has been a foundation for many competitive methods. This architecture has been shown to efficiently encode various types of information useful for recommendation systems [26]. A personalized re-ranking model is proposed in [19], which captures in its encoding layer the interactions between users and items to produce an interpretable re-ranked recommendation list. Other works use this model, including the user’s behavior sequence to learn more in-depth representations for each item in the sequence [4, 5]. Unlike previous works we apply it to in-game interpretable item recommendation with newer contexts.

4 MODEL ARCHITECTURE

Figure 2 shows the **Transformer for Team-aware Item Recommendation** architecture (TTIR). This model is made up of three major parts: the input representation layer, the encoder layer, and the output layer for recommendation. It takes as input the information of a match, which consists of the champions, their assigned role, and the team they belong to. Then it recommends a list of six items to each of them. The details of our architecture are in the following paragraphs.

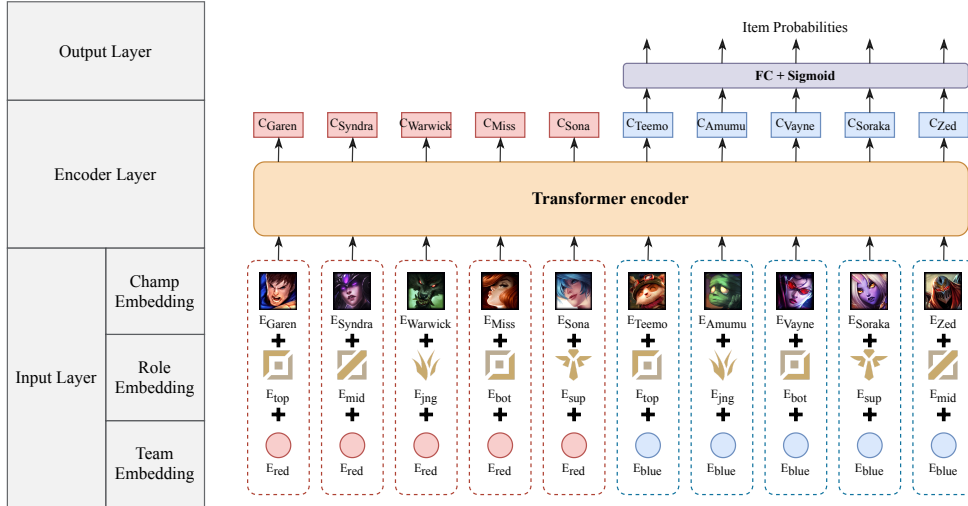


Fig. 2. Network architecture of TTIR.

Input Layer. The goal of the input representation layer is to prepare an unambiguous representation for each participant within the match, considering its champion, role and the team they belong to. To represent these features we take inspiration

from the BERT language model [9]. It represents sequences of words as sentences, adding to the representation of each word the information about its position and the sentence it belongs to. In the same spirit, we use three learned embeddings: the champion embedding E_{champ} , the role embedding E_{role} and the team embedding E_{team} . At last we add all of these embeddings to obtain the champion embedding E_{input} , which has the model dimension d_{model} as shown in the equation 1. The lower part of the Figure 2 shows a visual example of the input representation.

$$E_{input} = E_{champ} + E_{team} + E_{role}, \text{ where } E \in \mathbb{R}^{d_{model}} \quad (1)$$

Encoder Layer. This part of the model is a Transformer encoder based on the original implementation [25]. We omit an exhaustive description of the Transformer architecture because its use has become ubiquitous in recent years. The goal of this encoder is to compute interactions between allied and enemy champions of the match with the self-attention mechanism. The output of this encoder are contextualized embeddings C of the input embeddings E_{input} that captures complex relations between the champions. This architecture has two principal hyper-parameters, the number of layers l and number of heads h . The influence of these parameters is studied later in the ablation analysis.

$$C = \text{TransformerEncoder}(E_{input}), \text{ where } C \in \mathbb{R}^{d_{model}} \quad (2)$$

Output Layer (Item Recommendation). The purpose of the item recommendation layer is to generate a list of items for each champion of a team in a match. In order to do this, we feed each contextualized embedding C^i of a team, where i denotes the champion, through a linear layer, followed by a sigmoid function. As it is shown in the equation 3, the final output are the probabilities that the champion selects each of the items in N_{items} . Given that the champions can only use up to six items, the model recommends the six most probable items for each of the five players in a team (equation 4).

$$P_{items}^i = \text{Sigmoid}(W_{rec}C^i), \text{ } W_{rec} \in \mathbb{R}^{N_{items} \times d_{model}}, C^i \in \mathbb{R}^{d_{model}}, i \in [1, 5] \quad (3)$$

$$\text{Recommendation}^i = \text{top}(P_{items}^i, 6) \quad (4)$$

Like [3], the model was supervised only with the items selected by each champion of the winning team. This way the likelihood of recommending the best items is maximised in order to win.

5 EXPERIMENTS






In this section, we describe the dataset used and the offline evaluation conducted on the final TTIR model. Along with the results, an ablation study is presented to better understand the behavior of the network.

Dataset. In order to train and evaluate our model, we used a publicly available dataset provided by Kaggle¹. This consists of 184,070 game sessions in the ranked category, a competitive alternative to the normal match. Although the dataset does not provide a specific structure for recommendation tasks, we adequate it for this purpose. The raw dataset includes several files with much information about each match, so we choose the most relevant for this work. Specifically, our final dataset contains a match per instance, consisting of the identifier of each of the 10 participants with their role, team, and items used. We filter the basic, advanced, and consumable items, as well as all the matches that did not belong to the LoL 7th season. The complete data was divided into two subsets for training and testing, taking into consideration that the least common champion must be present in both of them. The overview of the final dataset is shown in Table 1.

Training and Evaluation Settings. The best configuration of TTIR consists on 2 heads, 1 layer, an embedding size of 512 and dropout of 0,5. We trained the model using Adam optimizer with a learning rate of 3e-4 until convergence. For

¹www.kaggle.com/paololol/league-of-legends-ranked-matches

Table 1. Overview of the dataset

LoL Ranked Matches 7th Season	
# Items	89
# Champions	136
# Matches	157,584
Roles	Top  , Mid  , Jungle  , Support  , Bot 
Train / Test	1,261,280 / 314,560

evaluation, we compare our model with decision tree (D-Tree), logistic regression (Logit) and shallow artificial neural network (ANN) baselines of [3]. We also implemented an additional baseline based on Convolutional neural networks (CNN) for a stronger comparison. We used different evaluation metrics to measure the relevance and ranking of our recommender system: Precision@k, Recall@k, F1@k, and MAP@k, with k=1,6,10. While k=6 might seem a strange cut, it is the maximum number of items a player can use during a match. Likewise, k=10 shows how the six articles are distributed in the firsts ten positions of the list.

5.1 Results

Results are shown in Table 2a. Our model achieves consistent and statistically significant improvements compared to the best baseline CNN. The trend indicates that the biggest performance difference is observed in top rank positions (k=1,6) and this difference decreases as the cut-off increases (k=10). This result is positive for TTIR, since it is difficult for a model to know when it has to recommend at least six items, as it normally depends on external factors like the game duration and the player expertise. The difference in performance between TTIR and other models is due to the ability of the Transformer to include relevant contextual information of the match into the representation of each champion.

Table 2. Experiment Results

(a) Results for top @k recommendation. TTIR is significantly better than the second best method, CNN.

	Method					T-test($df = 314557$)	
	D-Tree	Logit	ANN	CNN	TTIR	p-Value	(t-Stat)
Precision@1	0.516	0.672	0.771	0.790	0.803	5.30e-20	(9.158)
Recall@1	0.135	0.178	0.205	0.209	0.214	9.54e-18	(8.580)
F1@1	0.210	0.277	0.318	0.331	0.338	7.23e-20	(9.124)
MAP@1	0.516	0.672	0.771	0.790	0.803	5.30e-20	(9.158)
Precision@6	0.319	0.393	0.476	0.484	0.492	2.41e-22	(9.723)
Recall@6	0.491	0.607	0.732	0.744	0.756	1.93e-27	(10.854)
F1@6	0.379	0.468	0.566	0.586	0.596	2.57e-27	(10.828)
MAP@6	0.648	0.714	0.785	0.795	0.805	3.77e-30	(11.410)
Precision@10	0.204	0.285	0.341	0.348	0.351	2.49e-11	(6.674)
Recall@10	0.520	0.726	0.864	0.882	0.889	1.43e-24	(10.232)
F1@10	0.289	0.403	0.481	0.499	0.503	3.34e-15	(7.878)
MAP@10	0.636	0.672	0.743	0.754	0.764	1.32e-34	(12.270)

(b) Ablation study of TTIR

	P@6	R@6	MAP@6
Default ($h=2, l=1$)	0.492	0.756	0.805
Multiheads ($h=1$)	0.462	0.726	0.778
Multiheads ($h=4$)	0.492	0.756	0.806
Layers ($l=2$)	0.493	0.757	0.806
Layers ($l=3$)	0.493	0.758	0.807
w/o $E_{enemies}$	0.487	0.749	0.798
w/o E_{role}	0.484	0.742	0.794
w/o $E_{enemies}, E_{role}$	0.479	0.736	0.787
CNN w/ $E_{enemies}, E_{role}$	0.484	0.744	0.795

Ablation analysis. To understand the influence of contextual dimensions as well as several hyperparameters of the Transformer model, we conducted an ablation analysis which results are presented in Table 2b. The performance of TTIR does not increase with bigger numbers of attention heads (by default $h = 2$), but it declines when the heads decrease to $h = 1$. This confirms the importance of focusing on the different features of each champion. Increasing the number of layers (by default $l = 1$) to $l = 2$ or $l = 3$ has an almost negligible improvement, but with a bigger cost on the number of parameters. This suggests preserving the default number of layers. In terms of contextual dimensions, we notice that removing the E_{role} context has a slightly higher impact than the role of the $E_{enemies}$ context, but removing both



Fig. 3. Visualization of the attention weights for each member of the Blue team on each member of both teams (bottom row).

contexts makes the model perform even worse than CNN with these contexts. These results indicate that it is not only the Transformer architecture of TTIR which makes a difference in performance, but the combined effect of both architecture and contextual information what makes TTIR work.

6 PRELIMINARY USER SURVEY

In order to get insights from LoL players about the relevance of our recommendations and the usefulness of the attention weights from TTIR to explain the suggestions, we designed a visualization to explain the recommendations. Based on this, we conducted a preliminary survey.

The visual explanation of team-aware item recommendations. Figure 3 shows an example of attention weights visualization to explain the model recommendations. It consists of: (i) two teams, of five players each, at the bottom, (ii) a heat map in the center, and (iii) one of the teams with its six recommended items by each player on the right side. The heat map uses different color intensities to show the relevance of each champion upon each recommendation list. Darker colors represent more relevance. In Figure 3, the model pays more attention to the enemies since the items are used to beat them, and the model recommends by maximizing the chance of winning.

The survey procedure. The survey consisted of showing subjects four cases similar to the one displayed in Figure 3. Subjects were told that they belonged to the *Blue team* and that they were facing the *Red team*. Then they had to judge the quality of the recommendations for the *Blue team* and the usefulness of the explanations provided by the heatmap. The survey was advertised in public Facebook group pages of college students who were LoL players. They replied their interest with an e-mail and we sent them back an online form where users had to agree with an informed consent about the study, and then answer the questions in Table 3 in a scale of 1 to 10 (1: completely disagree, 10: completely agree) for each of four cases similar to the one in Figure 3. We also asked open questions in order to get a less structured feedback from the participants: (Q4) *If you **do not find** this visualization useful to explain the recommendation, tell us why*, and (Q5) *If you **find** this visualization useful to explain the recommendation, tell us why*.

Results of the user survey. 16 people answered our survey. 25% identified as female, 68% as male and one subject did not disclose the gender. 13 out of 16 participants were between 20 and 30 years of age, one was between 18-20, the rest were older or did not disclose their age. Six subjects indicated that they started playing in the last 2-4 years, while the other 10 indicated having been playing for 6 years or more. Although we acknowledge having a small number of subjects (N=16), each one responded 4 cases of recommendations and their open comments provided evidence of paying deep

attention to the user study. The results to questions Q1-Q3 are presented in Table 3. With respect to **Q1**, we observe that people have a fairly positive perception of recommendation relevance ($M=7.98\pm1.22$), but this perception seems to be more positive to less experienced users who started playing not before 2-5 years ago ($M=8.46\pm1.3$). In terms of **Q2**, we observe also a positive ($M=7.4\pm1.42$) and rather uniform impression among newer and experienced players, with respect to the perception of subjects towards influence of enemies and allies champions towards recommendation for the *Blue team*. Finally, in **Q3** we notice that the perception of recommendation interpretability from the visualization is not as good as the perception of relevance ($M=6.9\pm2.15$), but again we observe a more positive impression from newer subjects ($M=7.33\pm2.87$) compared to the most experienced users. These results are consistent with previous studies indicating user expertise as a factor influencing perception about recommendations [13, 18]. To dig deeper into these results we analyzed the open user comments.

Table 3. Results of the preliminary user survey (N=16), ratings in range [1-10]

Question	Global M \pm SD (N=16)	Subjects by year of first play		
		2009-11 (N=5)	2012-14 (N=5)	2015-2017 (N=6)
Q1. How good were the recommendations for the <i>Blue team</i> ?	7.98 \pm 1.22	7.7 \pm 1.24	7.7 \pm 1.16	8.46 \pm 1.3
Q2. Is it understandable the influence of every team member upon each champion being recommended ?	7.44 \pm 1.72	7.4 \pm 1.55	7.1 \pm 0.8	7.75 \pm 2.49
Q3. Is it useful the information provided by the visualization in order to understand the item recommendations made ?	6.9 \pm 2.15	6.7 \pm 1.98	6.6 \pm 1.65	7.33 \pm 2.87

Synthesis of user comments. On the positive side, we received comments of the usefulness of the explanations since they made sense to users based on their game experience: *“useful build to prevent enemy ganking...”*, *“you can see exactly the focus of each champion with respect to the main enemies on the facing team...”*, *“...with this build, Vlad hinders the enemy, making Lucian suffer. Then Ezreal with that build can damage both Lux and Fizz”*. We also received critical comments which provide important ideas for future work, several of them require information not currently available in our dataset: *“this explanation missed armor penetration and grievous wounds...”*, *“it doesn’t show with which item I need to start and the sequence to progress...”*, *“recommendations do not show magic resistance...”*, *“...in some cases, such as Soraka and Teemo, it would make more sense to show attention on the relationship with themselves”*.

7 CONCLUSIONS

In this work we introduced TTIR, a contextual recommendation model which provides team-aware item recommendations in MOBA games such as LoL. TTIR successfully models the complex contextual relationships present in the matches, and the attention weights allow us to provide explanations of suggested items. Furthermore, with a preliminary user study we had an initial idea of the perception of relevance of actual LoL players as well as important feedback towards the visual recommendations based on TTIR attention weights. Our initial analysis indicates that expert users require more details for understanding and following the recommendations, while less experienced users find them coherent and useful. Among ideas for future work we consider providing further details in our recommendation explanations such as item statistics, as well as sequential item recommendation.

ACKNOWLEDGMENTS

This work has been supported by the Millennium Institute for Foundational Research on Data (IMFD) and by the Chilean research agency ANID, FONDECYT grant 1191791.

REFERENCES

- [1] Xavier Amatriain and Josep M. Pujol. 2015. Data Mining Methods for Recommender Systems. In *Recommender Systems Handbook*. Springer US, 227–262. https://doi.org/10.1007/978-1-4899-7637-6_7
- [2] Vladimir Araujo, Alejandra Gonzalez, and Diego Mendez. 2018. Dynamic Difficulty Adjustment for a Memory Game. In *Communications in Computer and Information Science*. Springer International Publishing, 605–616. https://doi.org/10.1007/978-3-030-05532-5_46
- [3] Vladimir Araujo, Felipe Rios, and Denis Parra. 2019. Data Mining for Item Recommendation in MOBA Games. In *Proc. of the 13th ACM Conference on Recommender Systems (Copenhagen, Denmark) (RecSys '19)*. ACM, New York, NY, USA, 393–397. <https://doi.org/10.1145/3298689.3346986>
- [4] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-Commerce Recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (Anchorage, Alaska) (DLP-KDD '19)*. Association for Computing Machinery, New York, NY, USA, Article 12, 4 pages. <https://doi.org/10.1145/3326937.3341261>
- [5] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: Personalized Outfit Generation for Fashion Recommendation at Alibaba iFashion. In *Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Anchorage, AK, USA) (KDD '19)*. ACM, New York, NY, USA, 2662–2670. <https://doi.org/10.1145/3292500.3330652>
- [6] Y. Chen, A. Cheng, and W. H. Hsu. 2013. Travel Recommendation by Mining People Attributes and Travel Group Types From Community-Contributed Photos. *IEEE Transactions on Multimedia* 15, 6 (2013), 1283–1295.
- [7] Zhengxing Chen, Truong-Huy D Nguyen, Yuyu Xu, Christopher Amato, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 2018. The art of drafting: A Team-Oriented Hero Recommendation System for Multiplayer Online Battle Arena Games. In *Proc. of the 12th ACM Conference on Recommender Systems - RecSys '18*. ACM Press. <https://doi.org/10.1145/3240323.3240345>
- [8] Germán Cheuque, José Guzmán, and Denis Parra. 2019. Recommender systems for Online video game platforms: The case of STEAM. , 763–771 pages.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [10] Sarik Ghazarian and Mohammad Ali Nematbakhsh. 2015. Enhancing Memory-Based Collaborative Filtering for Group Recommender Systems. *Expert Syst. Appl.* 42, 7 (May 2015), 3801â\$3812. <https://doi.org/10.1016/j.eswa.2014.11.042>
- [11] D. Gourdeau and L. Archambault. 2020. Discriminative neural network for hero selection in professional Heroes of the Storm and DOTA 2. *IEEE Transactions on Games* (2020), 1–1.
- [12] Rishabh Joshi, Varun Gupta, Xinyue Li, Yue Cui, Ziwen Wang, Yaser Norouzzadeh Ravari, Diego Klabjan, Rafet Sifa, Azita Parsaeian, Anders Drachen, and Simon Demediuk. 2019. A Team Based Player Versus Player Recommender Systems Framework For Player Improvement. In *Proceedings of the Australasian Computer Science Week Multiconference (Sydney, NSW, Australia) (ACSW 2019)*. Association for Computing Machinery, New York, NY, USA, Article 45, 7 pages. <https://doi.org/10.1145/3290688.3290750>
- [13] Bart P Knijnenburg, Niels JM Reijmer, and Martijn C Willemsen. 2011. Each to his own: how different users call for different interaction methods in recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 141–148.
- [14] W. Looi, M. Dhaliwal, R. Alhajj, and J. Rokne. 2018. Recommender System for Items in Dota 2. *IEEE Transactions on Games* (2018), 1–1. <https://doi.org/10.1109/TG.2018.2844121>
- [15] Judith Masthoff. 2011. Group recommender systems: Combining individual models. In *Recommender systems handbook*. Springer, 677–702.
- [16] Newzoo. 2020. Global Esports Market Report. <https://strivesponsorship.com/wp-content/uploads/2020/03/Global-Esports-Market-Report-2020.pdf>.
- [17] OpenAI, :, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dăbziak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Jęszefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondã de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. arXiv:1912.06680 [cs.LG]
- [18] Denis Parra and Peter Brusilovsky. 2015. User-controllable personalization: A case study with SetFusion. *International Journal of Human-Computer Studies* 78 (2015), 43–67.
- [19] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, and Dan Pei. 2019. Personalized Re-ranking for Recommendation. In *Proc. of the 13th ACM Conference on Recommender Systems (Copenhagen, Denmark) (RecSys '19)*. ACM, New York, NY, USA, 3–11. <https://doi.org/10.1145/3298689.3347000>
- [20] I. Porokhnenko, P. Polezhaev, and A. Shukhman. 2019. Machine Learning Approaches to Choose Heroes in Dota 2. In *2019 24th Conference of Open Innovations Association (FRUCT)*. 345–350.
- [21] J. Shi, B. Wu, and X. Lin. 2015. A Latent Group Model for Group Recommendation. In *2015 IEEE International Conference on Mobile Services*. 233–238.
- [22] Mirna Paula Silva, Victor do Nascimento Silva, and Luiz Chaimowicz. 2017. Dynamic difficulty adjustment on MOBA games. *Entertainment Computing* 18 (Jan. 2017), 103–123. <https://doi.org/10.1016/j.entcom.2016.10.002>

- [23] Paul Tassi. 2014. Riotâ€™s League of Legendsâ€™ Reveals Astonishing 27 Million Daily Players, 67 Million Monthly. <http://www.forbes.com/sites/insertcoin/2014/01/27/riots-league-of-legends-reveals-astonishing-27-million-daily-players-67-million-monthly>.
- [24] April Tyack, Peta Wyeth, and Daniel Johnson. 2016. The Appeal of MOBA Games: What Makes People Start, Stay, and Stop. In *Proc. of the 2016 Annual Symposium on Computer-Human Interaction in Play (Austin, Texas, USA) (CHI PLAY '16)*. ACM, New York, NY, USA, 313–325. <https://doi.org/10.1145/2967934.2968098>
- [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *31st Conference on Neural Information Processing Systems (Long Beach, CA, USA) (NIPS '17)*.
- [26] Yichao Zhou, Shaunak Mishra, Manisha Verma, Narayan Bhamidipati, and Wei Wang. 2020. Recommending Themes for Ad Creative Design via Visual-Linguistic Representations. *Proceedings of The Web Conference 2020* (Apr 2020). <https://doi.org/10.1145/3366423.3380001>