Making Matches - Recommending the right personality

Paul Schweiger

TECHNISCHE
UNIVERSITÄT
DARMSTADT

1 Topic Summary

As globalization and connectedness advance, online communities grow in importance and the offline world embraces online opportunities. A surplus of possible contacts emerges, making it challenging to have an optimal online experience by engaging with fitting personalities. This interaction between users is a cornerstone of many different activities, such as multiplayer gaming, social networks, crowdsourcing communities or online dating platforms, making matchmaking efforts important to further financial and social goals. Recommender Systems that account for an adequate operationalization of the relevant personality traits can be used to improve said goals all across different domains of usage.
The ever-growing market of multiplayer gaming experiences have created online spaces where different personalities are forced to work together. While forming an effective team is the apparent goal of competitive online gaming, for the most part of the community *having fun* is even more important. At the same time, the internet's anonymity makes it a great place for trolls and toxic players, resulting in intra-community problems and a âĂIJsaltyâĂİ experience. Not only toxic players, but a combination of different player goals and styles can lead to bad experiences: a single aggressive player in a defensive team will not have the needed support in battle, while players aiming for a story-driven experience wonâĂŹt be able to cooperate with performance-oriented hardcore gamers.
Considering the fact that a large amount of players is available at any given moment, it should be possible to incorporate such preferences in the matchmaking process - which is currently mainly driven by player experience and performance.
A recent study by Wang et al. [WYS15] looked at the enjoyment of multiplayer sessions in League of Legends (LoL) based on player personality. They followed a subset of Sternberg's [Ste99] problem-solving styles in order to categorize the playstyle of different users. While others tried to gain player personality knowledge by simply asking them [RCFP07] or other players [PLBH11], Wang et al. automated this process with gameplay statistics. They assigned specific in-game actions to each of their player categories and determined a player's style based on his action profile. Choosing the data used to calculate a "good" experience is highly domain specific and depends on available objective game statistics.[DCTL+12]
Match enjoyment for LoL was modeled as a function of game length, with shorter games being less enjoyable due to one team dominating the other. The results show a clear tendency of specific globally-active and risk-taking players to positively influence the overall game enjoyment. A neural network tasked with predicting match enjoyment achieved better scores when player style data was included, suggesting that this might benefit matchmaking algorithms. However, it is important to note that the used measures are highly subjective to LoL, as interviews with players of both LoL and other games confirm.
Most Online Dating Services experience comparable problems: Learning who goes good with one another is a major success criterion for dating applications. Possible matchups need to be built according to their preferences rather based upon global performance. Studies predicting user tastes based on the tastes of similar users, a technique called "Collaborative Filtering", showed better results than global matchmaking. [BP07]
Proper personality-based matchmaking in online environments has a positive influence on overall user satisfaction. Successful Recommender Systems need to incorporate personal and detailed data, rather than just performance measurements. On the other hand the limiting effect of Recommender Systems on content diversity needs to be considered, especially in this social use case. [NHH+14]

2 Introduction

The modern world has a plethora of opportunities, topics, people, products, and other things, while people have a limited amount of attention and time to spend. In order to help us guide our attention and resources, basically every digital product tries to recommend meaningful content to users. Recommender Systems play a critical role in modern society and have transcended in many different domains.
Online communities grow in importance and the offline world embraces online opportunities. A surplus of possible social contacts emerges, making it challenging to engage with the right person at the right time. Interaction between users is a cornerstone of many different activities, such as multiplayer gaming, social networks, crowdsourcing communities or online dating platforms. But besides entertainment, social contacts can also further educational goals.
SOZIALES LERNEN IST BESSER als Alleine [Quellen]
E-Learning and Online Courses – Wo passt das sinnvoll hin? :S
Social Recommendation is more complicated, since it needs to account for preferences of more than just a single person. Recommender Systems need account for an adequate operationalization of the relevant personality traits, domain-specific skills and surrounding circumstances.

Focus on E-Learning and Reciprocal Recommendation, because...
Introduction to Recommendation in E-Learning (How good is research? Overlap with other topics -> shorten this section)
Introduction to Recommendation in Reciprocal environments (How good is this researched? How many things do we know?)
This report will discuss possible solutions to improve learning via reciprocal peer recommendation in Learning-Environments. As one of the first (???) and the most recent implementation of such an endeavor, the report will provide a detailed overview of XXXXXXXXX by XXXXXXXX. Important findings leading up towards the work, possible extensions and research in other fields that might prove to be beneficial to this study will be discussed.

3 Research leading towards the paper

3.1 Learning

TODO: Provide overview of modern learning:

- learner types
- E-Learning vs. offline learning
- peer learning
- recommender systems for e-learning so far

3.2 Recommender Systems

TODO: Provide short overview of RS
Where do they come from, what have they been used for?
REMARK: This section might be removed due to overlap with other sections. Maybe RS for E-Learning and reciprocal RS will be enough

3.3 Reciprocal Recommendation

TODO: Provide Overview of Reciprocal Recommendation

- Goal of recirpocal recommendation
- different ways to do this: Just a score for other people, or actual reciprocal recommendations, where bothpartners see each other as a recom.
- Other fields (Gaming, Dating)
- Common methods (Collaborative Filtering)
- Common problems encountered when doing this (operationalize personality / relevant aspects, acquire personal information about people, self-reported vs. implicit, ...)

4 Reciprocal Peer Recommendation for Learning Purposes

4.1 Introduction

Introduction: Motivation for this specific study. What did they want to accomplish?
Explain goals and problems of XXX et al. implemented RiPPLE to prove their theories.

4.2 RiPPLE

RiPPLE ["Recommendation in Personalised Peer Learning Environments"] was designed and developed as a web-based prototype online learning recommendation. RiPPLE is an adaptive, student-facing, open-source platform with the aim to enable students to engage with others in meaningful learning experiences. To enhance the learning experience, RiPPLE functions as a learning platform, helping students to co-create and find meaningful learning-content and to find peers to learn with. This analysis will focus on RiPPLE as a peer recommendation platform.
Based on user input, RiPPLE will calculate potential matchups for its users. Depending on the competency derived by a user's performance on learning content, his or her available timeslots and a user's preference on the topics and partner's competency he would like to provide or seek peer support or find a learning partner in, RiPPLE calculates a score for a matchup and will recommend a predefined amount of persons to each user. As RiPPLE currently recommends learning opportunities for the upcoming week, updates to user preferences or competencies are represented once per week.
An important aspect of the recommendation algorithm is it's compatibility function, calculating a one-directional score for each combination of potential study partners, u1 and u2. In the first step, the algorithm will check whether a potential matchup is viable following two hard constraints:

- a shared timeslot has to be available for both u1 and u2
- the topic-specific joint competency must be greater than a prefedined threshold T. According to BLUMENFELD [QUELLE], peer learning sessions will only become effective once the learners can share a minimum understanding of the topic.

For every pair of users satisfying these constraints, RiPPLE will calculate the user's respective one-directional scores. These represent how fitting u2 is as a study partner for u1 and vice-versa. (Since the users could have defined different preferences for their competency differences, scores don't need to be symmetric.) The scores take into account how good a matchup will be in terms of overall competency level, and how the other user matches the current users preferences. These calculations will be calculated across all topics relevant for u1 and u2.
These two one-directional scores (the score of u2 as a partner for u1 and the score of u1 as a partner of u2,) could now be used to find the the best partner for a specific user. To further recommend a matchup that is beneficial for both u1 and u2, the harmonic mean of both scores is considered as the "reciprocal score" of u1 and u2, a value that is now symmetric. [QUELLE PRABKAHAR 45] The harmonic mean, contrary to the arithmetic mean, pays respect to differences between it's values, making a larger gap between values less desirable. Peer-combinations with approximately similar scores in each direction will receive better final values, pushing matchups that are beneficial to both participants more relevant.
TODO: BILD VON SCORES; SCORES ALS VERTEILUNGEN
In the last step, RiPPLE returns a predefined amount of matchups with the best reciprocal values for each user. Although these reciprocal values are now symmetrical, the recommendations don't have to be: While from u1s standpoint the matchup with u2 and an (arbitrary) reciprocal score of 30 could be the very best opportunity, u2 could still have a matchup with u3 and a value of 50.
For more information on RiPPLE, the algorithm and further clarification of different variables, please reference [QUELLE].
RiPPLE will be evaluated in live conditions in the course of this year. To check whether the implementation could work under real conditions, XXXX conducted an evaluation using randomly generated data. This will be discussed in the upcoming section

4.3 Evaluation

In order to test RiPPLE's applicability for actual use, Potts et al. designed an experimental setup in which RiPPLE would try to propose recommendations for randomly generated test data. Specific quality measures were designed to assess different fields in which RiPPLE would have to show its capabilities. With satisfying results, RiPPLE would be able to be used under live conditionsin the course of 2018.

4.3.1 Data

To conduct the experimental evaluation, random data had to be generated; diverse enough to highlight edge cases but within reasonable bounds. For a set of 1000 users, 10 learning topics and 10 possible timeslots, each user received a random distribution of relevant values. Topic-specific competencies were expressed as a value from 0 to 100 derived from a truncated normal distribution around a random mean with fixed variance. Each user received competencies for every topic. These were then sorted from low to high, and a random number of these topics were chosen to be part of the user's request. The highest competencies of every user were classified as "providing support" roles, the lowest as "seeking support" and the median topic received a "co-study" role. In absence of empirical data, competency difference preferences were modeled as a fixed value per chosen role, as opposed to a explicitly stated preference for each user. Every "user" was made available during random timeslots.

4.3.2 Quality Measures

To fully satisfy as a tool to recommend students to one another, Ripple must be able to form meaningful and successful matches for as many users as possible in reasonable time. On the other hand, minor drawbacks in the defined metrics were considered to be tolerable due to the experimental and randomly generated data and some further adjustments that could be made to compensate bad values.
As Evaluation Metrics for their experimental evaluation Potts et al. decided on four values that can further be used as general Quality Measures for reciprocal recommendation algorithms:

Scalability

With increasing enrollment numbers in higher education, RiPPLE will have to be suitable for large sets of learners. High runtime and costs for evaluating datasets with reasonable amounts of students means slower responses and a worse user experience. An optimal solution could provide immediate recommendations to any user, at any moment.
The runtime of the algorithm increased in a quadratic fashion, as U, the total amount of users, increased: $O(n^2)$. The number of recommendations per user, however, does not significantly impact the runtime. (Although the paper states that it *did* in fact affect runtime, looking at the plotted data suggests that this might be a formatting error.)
Currently, RiPPLE calculates recommendations at the end of each week for the upcoming week, making the algorithm's runtime rather unimportant. However, further improvements are planned.
In a 1000 user experiment, RiPPLE was able to provide recommendations for a single user in 0.045 seconds - which is a pretty good time.

Reciprocality

The best possible recommendations are reciprocal: Users contacting a recommended user would also appear on this user's list of potential study partners. Reciprocality was tested for both, the baseline non-reciprocal and the joint reciprocal average scores. Whenever a user appears in the recommendations of a user on their own recommendation list that was built according to the respective score, the recommendation was considered to be reciprocal.
The precision for every user under the used score is calculated by dividing his reciprocal recommendations through k, the total amount of recommendations that user received. The system's total precision is defined as the average precision across all users.
QUELLE
In all tested cases, the reciprocal score had a higher precision than the baseline score. This is not surprising, since

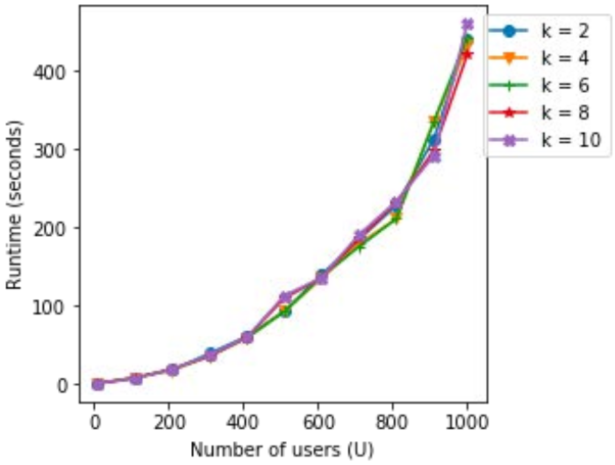

Figure 4.1: Scalability: The algorithm's runtime depending on the number of users U and the amount of recommendations per user k. Note how k has almost no influence on the runtime, while it grows exponentially with increasing U. Source: [PKR+18]

using the harmonic mean of both one-directional scores chooses reciprocal scores with medium values compared to non-reciprocal scores with a single high value. [REFERENZ AUF ERKLÃĎRUNGSABSCHNITT] Increasing k also increases the precision, since more recommendations per user lead to a higher chance of reciprocal recommendations. On the other hand, increasing U with a fixed k reduces reciprocal precision, since there are more possible users to recommend.

Coverage

Recommending potential learning partners to one another should be made for as many users as possible, not abandoning anyone. As such, coverage is a very important metric to consider. Since (almost) every user will receive recommendations, most users will be covered in one way or another. (The exception to this are users with completely incompatible timeslots, role preferences (i.e. being the only person looking for an equally skilled study partner) or users who can't meet the minimum competency when coupled with their available potential partners.) A good fit can only be ensured when the same user is recommended to others, ideally forming a reciprocal recommendation, which is represented in another quality measure. Coverage however is defined as the percentage of users that appear in other's recommendations at least once.
For a low amount of users and lots of recommendations per user, coverage is close to 0.9, meaning most users are recommended to others. As U increases or k decreases, the coverage sinks. However, more than 40% of users appear in other's recommendations under all tested circumstances.

Quality

The quality of a recommendation is not only based on its fit, but also on how good the resulting team could perform. According to BLUMENFELD QUELLE, learners should meet a minimum competency level in order to be an effective group, as specified by the employed minimum matchup threshold T and leniency factor alpha. [IST ES GUT DIE HIER ZU ERWÃĎHNEN? HÃĎNGT DAVON AB WIE ICH DIE BILDER EINBINDE] Quality is thus defined as the user's average joint competencies across their matched topics. The goal is to generate matches, that are as capable as possible in their respective fields of study.

4.4 Discussion

Recap of their discussion
Current results are scientifically weak: Lots of assumptions about initial values (e.g. for T), no aims of bounds set to determine whether final values are actually good.

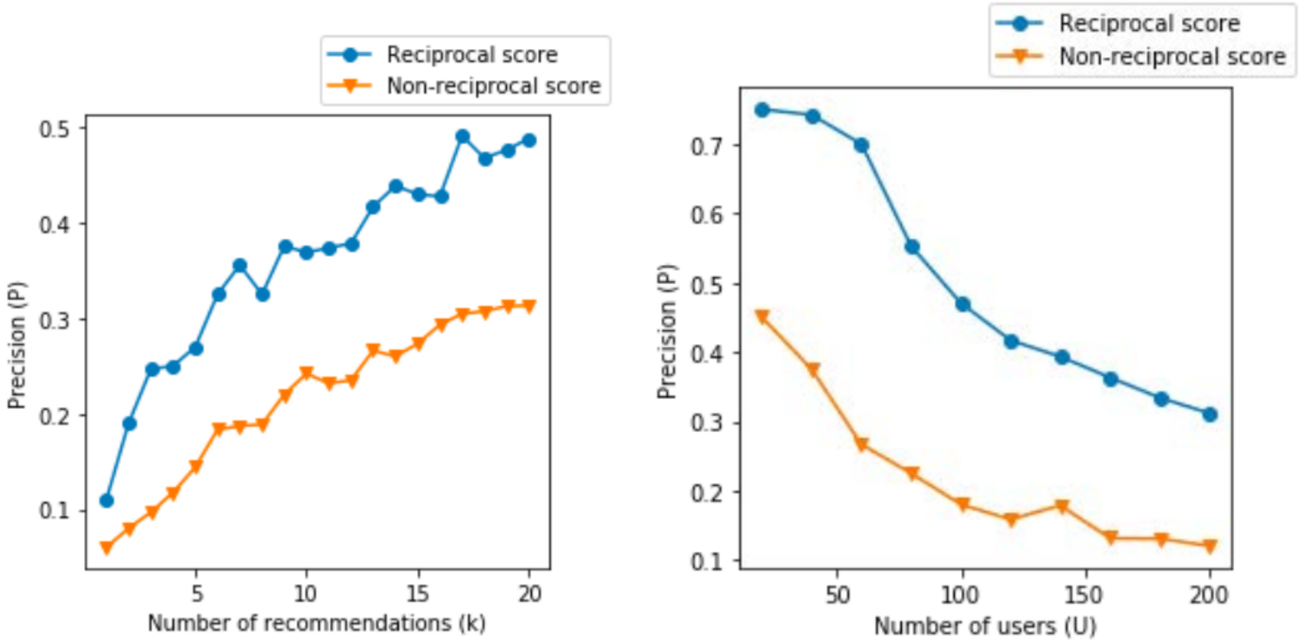

Figure 4.2: Reciprocality: The precision (= the fraction of reciprocal recommendations out of the total recommendations averaged over all users) of the baseline non-reciprocal recommendations (orange) vs. of the reciprocal, averaged scores. Note how the reciprocal scores are always better. Source: [PKR+18]

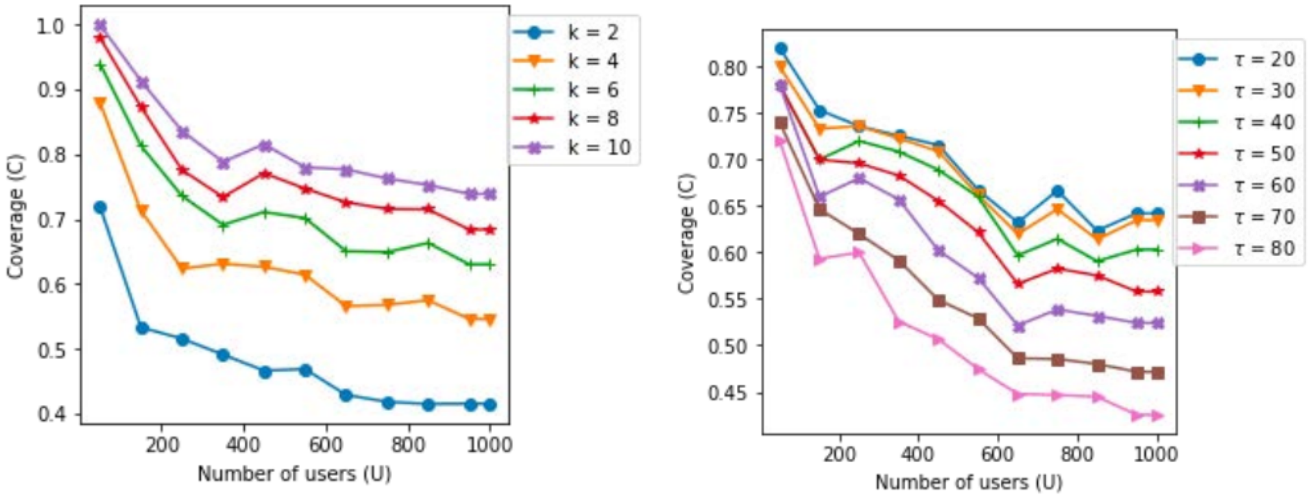

Figure 4.3: Coverage: The percentage of users who appear in other's recommendations. With more users, coverage sinks (likelihood of hard to match users increases). Increasing received recommendations or lowering the minimum joint competency of matches increases coverage. Mind the y-axis cutoff. Source: [PKR+18]

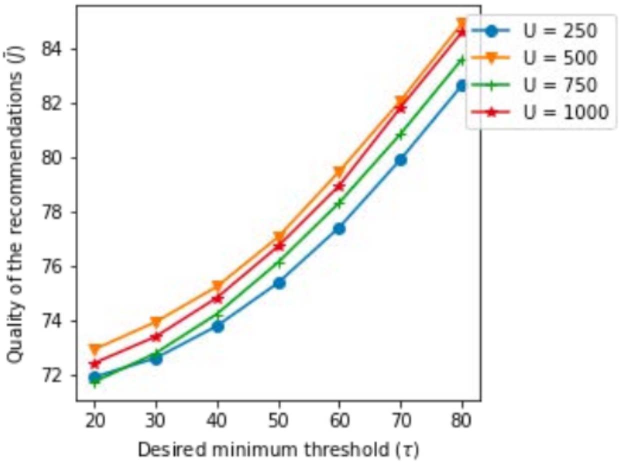

Figure 4.4: Quality: The quality of matchups over different values for U and T. Note how changing U won't affect matchup quality. Choosing a higher minimum joint competency threshold T for successful matchups increases overall quality. Comparing this finding to 4.3 however, suggests that higher quality comes at the cost of less coverage. Source: [PKR$^+$18]

While the values reported from evaluation with artificial data present good metrics to measure the algorithm's performance and suitability for live data, they don't actually evaluate the algorithm, since no targets have been set. The question whether a coverage of little above 0.4 will be enough in practice, remains unanswered. Same goes for all the other values: With little to no theory behind these metrics, their final values are hard to analyze in a larger context.

Perfect Matchup for one topic and "meh" matchup for other topic will get mediocre vote, instead of recommending topic-wise. (Matchups bei denen die ROLLEN gar nicht passen werden aber ausgeklammert, immerhin)

Is the quality a good measurement, actually? We want low level learners to have the opportunity to learn something... Should the score of a match not be always close to a medium value? OR does this metric rely too much on the actual values of their skills? Should me measure quality as "How does this matchup go in terms of preferred skill level?

Another neglected factor is the human factor. Both user buy-in and competence in handling the tool and its demands might influence its use in practice. While this study's goal was explicitly to test the theory and future praxis tests are planned, this topic should be discussed, a major shortcoming of the paper at hand.
A lack in user buy-in is something that always should be considered, especially in a student context. If a student didn't want to engage with foreign people, was not motivated to study with partners or to adjust his or her schedule, all recommendations to and of that student would not accomplish anything. Meeting requests would be ignored, and opportunities for matchups would expire. Even user manipulation needs to be considered as an possibility, but is something that has to be dealt with online.
While missed opportunities are a problem of the students themselves, rather than of the platform providing recommendations, the other human factor needs to be addressed directly by the tool.
As humans are unreliable, self-reported metrics always underly lots of variance and errors. A user's competency in a specific topic, his or her preferences, or the willingness to commit a specific timeslot to learning might change daily, dependent on mood, time of day and lots of other factors. [QUELLE?] Other variables, like a user's preferred skill difference towards a learning partner, are especially hard to specify. How is a user supposed to know what his or her learning preferences are? How would he know which number refers to the desired difference in skill rating? From a psychological standpoint, this operationalization is bound to fail.

Does not consider buy-in by users. This is okay for the tool, it will still recommend people, but the actual use of this whole measure will get undermined.
self-evaluating data about students is prone to errors and was not checked or detailed: misleading competency values might spoil results.
Abandoned users due to edge-cases: Someone with low competency prefering to teach, high competency preferring to get teached. Ripple does account for these cases by only recommending the best fits, so even low fits can still be recom-

mended.
Transparency: Will students know how good their matching value is?
Possible error for highly compatible users who appear in many recommendations: They have few reciprocal recs, but will get lots of meeting requests.
ARE THERE ANY DISCUSSIONS ABOUT THIS PAPER YET?

5 Extensions

TODO: Include possible follow-up research, directions for further research or innovative ideas from other fields to include in future recirpocal peer recommendation for elearning studies.

- NEO-FFI etc.: operationalize personalities
- Instead of student-driven requests: Use recommendations to build study courses
- From Gaming: Instead of 2-people-matches, build learning groups with matching skillsets to further benefit on other topics and to enable social grouping

Bibliography

[BP07] Lukas Brozovsky and Vaclav Petricek. Recommender system for online dating service. *arXiv preprint cs/0703042*, 2007.

[DCTL+12] Olivier Delalleau, Emile Contal, Eric Thibodeau-Laufer, Raul Chandias Ferrari, Yoshua Bengio, and Frank Zhang. Beyond skill rating: Advanced matchmaking in ghost recon online. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(3):167–177, 2012.

[NHH+14] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pages 677–686. ACM, 2014.

[PKR+18] Boyd Potts, Hassan Khosravi, Carl Reidsema, Aneesha Bakharia, Mark Belonogoff, and Melanie Fleming. Reciprocal peer recommendation for learning purposes. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, 2018.

[PLBH11] WO'Kelley II Patrick, Steven D Lamb, Michal Bortnik, and Johan Peter Hansen. System and method for providing feedback on game players and enhancing social matchmaking, November 29 2011. US Patent 8,066,568.

[RCFP07] Jens Riegelsberger, Scott Counts, Shelly D Farnham, and Bruce C Philips. Personality matters: Incorporating detailed user attributes and preferences into the matchmaking process. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 87–87. IEEE, 2007.

[Ste99] Robert J Sternberg. *Thinking styles*. Cambridge University Press, 1999.

[WYS15] Hao Wang, Hao-Tsung Yang, and Chuen-Tsai Sun. Thinking style and team competition game performance and enjoyment. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3):243–254, 2015.