Noname manuscript No.
(will be inserted by the editor)

Supporting Users in Finding Successful Matches in Reciprocal Recommender Systems

Akiva Kleinerman · Ariel Rosenfeld · Francesco Ricci · Sarit Kraus

the date of receipt and acceptance should be inserted later

Abstract Online platforms which assist users in finding a suitable match, such as online dating and job recruiting environments, have become increasingly popular in the last decade. Many of these environments include recommender systems which, for instance in online dating, aim at helping users to discover a suitable partner who will likely be interested in them. Generating successful recommendations in such systems is challenging as the system must balance two objectives: 1) Recommending users with whom the recommendation receiver is likely to initiate an interaction; and 2) Recommending users who are likely to reply positively to the recommendation receiver initiated interaction. Unfortunately, these objectives are partially conflicting since very often the recommendation receiver is likely to contact users who are not likely to respond positively, and vice versa. Furthermore, users in these environments vary in the extent to which they contemplate the other sides preferences before initiating an interaction. Therefore, an effective recommender system must effectively model each user and balance these objectives. In our work, we tackle this challenge through two novel components: 1) An explanation module, which leverages an estimate of why the recommended user is likely to respond positively to the recommendation receiver; and 2) A novel reciprocal recommendation algorithm, which finds an optimal balance, individually tailored to each user, between the partially conflicting objectives mentioned above. In an extensive empirical evaluation, in both simulated and real-world dating web-platforms with 1204 human participants, we find that both components contribute to attaining these objectives, and that the combinations thereof are more effective than each one on its own.

Address(es) of author(s) should be given

1 Introduction

Reciprocal Recommender Systems (RRSs) denote a class of recommender systems which recommend people to people as opposed to traditional recommender systems which recommend items to people [35]. RRSs are very common in web-based platforms such as online-dating and job recruitment, among other domains.

RRSs are distinctively different from traditional recommender systems. In *item-to-people* recommendations, the success of the recommendation is commonly determined by the acceptance of the recommendations (items) by the receiver (also termed *service user*). For example, a recommendation in an online shopping platform would be considered successful if it translates into a purchase from the service user. By contrast, in RRSs, a successful recommendation is one that brings about a *successful interaction* between the two users, meaning that both the service user accepted the recommendation and initiated an interaction with the recommended user, and, most importantly, the recommended user replied positively [35]. In an online dating platform, this means that the service user has expressed interest in the recommended user (e.g., by sending a message) and the recommended user has expressed interest on her end as well (e.g., by replying with a positive message).

Generating successful recommendations for an RRS is challenging. The RRS needs to balance two partially conflicting sub-goals: 1) recommending users which the service user will be likely to find attractive; and 2) recommending users which will be likely find the service user attractive. Unfortunately, these two sub-goals do not necessarily align. For example, Alice may be very interested in Bob (who is a very popular user in an online dating platform), yet Bob might be uninterested as he has many offers from different users who may better suit his preferences. Furthermore, a one-size-fits-all balancing approach, such as giving equal weight to both sub-goals in the recommendation generation, is unlikely to result in favorable outcomes since users widely vary in the extent to which they consider the preferences of the other side before initiating an interaction [22,48]. Specifically, while some users will initiate interactions with users who are more likely to respond positively, other users may completely ignore the projected preferences of the other side and will initiate interactions based on just their own preferences. The differences between users may be attributed to various psychological factors such as the emotional cost of initiating an interaction, the fear of rejection, and similar phenomena which are outside the scope of this work (see [22,21]).

In this article, we tackle the challenge of generating successful recommendations in RRSs through two computational components based on the following approaches: 1) Accommodating recommendations with tailored *explanations* aimed at mitigating the gap between the preferences of the service user and the generated recommendations (which also account for the assumed preferences of the recommended users); and 2) Providing *fully personalized* recommendations through a novel user modeling and recommendation technique aimed at finding the optimal balance between each service user's preferences and the recommended users' preferences.

Providing explanations in a recommender system has been shown to be effective in increasing the acceptance of the generated recommendations (e.g., [46]). Explanations generally provide reasons for why the system has estimated that the recommended item fits the user's preferences. Explanations commonly emphasize specific features of the recommended items or present similar items to those in which the user has shown interest [47]. Existing work in this field has focused, to the best of our knowledge, entirely on the non-reciprocal recommendation case. As a result, state-of-the-art explanation approaches are targeted at the preferences of the service user alone. However, it is our claim that, in reciprocal environments, additional information such as *why* a recommended user is likely to reply positively can be incorporated within an explanation scheme, and can possibly influence the user to accept the recommendation. To utilize this potentially useful information, we introduce and extensively evaluate a novel explanation method based on the preferences of both the service user and the recommended users, denoted *reciprocal explanations.* Through extensive empirical evaluation in both a simulated and a real-world dating platform, with 318 human participants, we examined the effects of both standard explanation techniques as well as our novel reciprocal explanations approach on different types of users and in different environmental settings. The results clearly support our claim that reciprocal explanations can significantly increase the acceptance of an RRS's recommendations. At the same time, we identify an intriguing phenomenon where in settings with minimal to no cost for initiating an interaction (e.g., no monetary or emotional cost), reciprocal explanations are counter-productive. These results combine to provide the first contribution of this article.

Recommendation generation methods for RRSs have been proposed in prior literature, many of which account for the preferences of both sides of the recommendation. However, a common theme among these algorithms is that they assume equal importance to the perceived preferences of both sides, thus providing only a "semi-personalized" recommendation method. As we discuss above, users may vary significantly in how they consider the preferences of the recommended users. Therefore, in order to achieve a fully personalized RRS, we propose a novel user modeling and recommendation method which relies on users' past data combined with machine learning and optimization techniques. In an extensive human study with 398 additional participants we found that our novel user modeling and recommendation method significantly increases the number of successful interactions compared with state-of-the-art RRS techniques. At the same time, as a side effect, as one might expect, our proposed method brings about a decrease in the number of accepted recommendations. Specifically, while the recommendations are better suited to achieve more successful interactions, they deviate from solely representing the preferences of the service user, resulting in some recommendations being deemed as not attractive enough by the service user. The newly developed method and empirical results constitute the second contribution of this article.

Lastly, we examine the integration of reciprocal explanations as a way to mitigate the downsides of our personalized reciprocal recommendation generation method. In an additional experiment with 488 human participants (who, again, did not participate in this study thus far), we show that a RRS, which integrates the two proposed methods for generating recommendations and their explanations, can bring about significant improvements compared to the use of each method on its own. Altogether, all of the phases of our investigation included 1204 unique participants.

The remainder of the article is organized as follows: Section 2 covers related work, mainly focusing on explanation provision and recommendation generation in RRSs. Next, in Section 3, we present our proposed explanation method. Section 4 demonstrates the benefits and limitations of our method through an extensive human study. Section 5 introduces our novel recommendation method suitable for RRSs, followed by an additional, large scale human study in Section 6. In order to overcome both methods' limitations, in Section 7 we present the integration of both methods within a single system which is tested in the real-world. Finally, Section 8 provides a discussion of the results and identifies directions for future work.

2 Related Work

We will now discuss the two prominent sub-fields of recommender systems related to our task: 1) Explanations in recommender systems and 2) Reciprocal Recommender Systems (RRSs).

2.1 Explanation Provision

Explainable Artificial Intelligence (XAI) is an emerging field which aims at making automated systems understandable to humans in order to enhance their effectiveness [15]. This research field was highly prioritized in the recent American National Artificial Intelligence Research and Development Strategic Plan [32, p. 28]. The need for explanations is also acknowledged by regulatory bodies. For example, the European Union passed a General Data Protection Regulation[1] in May 2016 including a "right to explanation", by which a user can ask for an explanation of an algorithmic decision made about him [13]. In recent years, providing an explanation has become a standard in many online platforms such as Google and Amazon.

A wide variety of methods for generating *explanations* for a given recommendation were proposed and evaluated in the literature. Two practices are commonly applied in this realm: First, existing explanation methods focus on the recommendation receiver alone. To the best of our knowledge, none of the existing methods were developed or deployed for RRSs. One exception to the above is Guy et al. [16], who presented a RRS which is transparent (i.e.,

[1] http://ec.europa.eu/justice/data-protection/

provides accurate reasoning as to how the recommendation was generated). Unfortunately, the authors did not compare the effects of their method with other explanation methods, nor did they consider the unique characteristics of RRSs. Secondly, existing explanation methods are often tailored for specific applications and therefore cannot be easily adapted or evaluated in different domains (for example, in [20] the authors have presented an explanation for recommendations of movies which is based on the main actor of the movie). In the following section (Section 3), we relieve these two practices by designing and evaluating novel general-purpose explanation methods for RRSs.

Many studies have demonstrated the potential benefits of providing explanations to automated recommendations. For example, Herlocker et al. [20] found that adding explanations to recommendations can significantly improve the *acceptance rate* of the provided recommendation and the *satisfaction* of the users thereof. Sinha et al. [43] further found that transparent recommendations can also increase the user's *trust* in the system. These results were replicated under various domains and explanation methods (e.g., [9,42,11]). The results of these works and others have combined to suggest two widely acknowledged guidelines for developing explanation methods: (1) Explanations which include *specific features* of the recommended item/user are highly effective, even if these features are not the actual reason that the recommendation was generated [11,20,36]; and (2) It is important to limit the length of the explanation in order to avoid an information overload which can make explanations counter productive [36,11]. We follow these guidelines in our designed explanation methods, presented in the following section.

2.2 Reciprocal Recommender Systems

Applications that require RRSs have unique characteristics, which present opportunities and challenges for providing successful recommendations [35]. Perhaps the most significant difference between RRSs and traditional *item-to-people* recommender systems is that RRS's recommendations must satisfy both parties, the service user and the recommended user. Another important issue that should be addressed in RRSs is limiting recommendations of "popular" users, meaning users who receive a lot of messages, regardless of the recommendations [30].

In the past decade, many research studies have investigated the field of RRSs and specifically the domain of online-dating. In typical online-dating environments, users can create a profile, browse other users' profiles and interact with users by sending messages. Some online-dating environments include an option for explicitly rating profiles or pictures. Brozovsky and Petricek [7] show that in such environments, collaborative filtering algorithms, which leverage similarity between users assessed from their explicit ratings, are significantly more effective in comparison to other algorithms which were commonly used in online-dating sites. However, online-dating sites do not generally include explicit ratings. Therefore, the recommendation methods for online-dating environments commonly elicit the users' preferences from their interaction history. Krzywicki et al. [29] show that a collaborative filtering method, which derives the similarities between users from their interactions, is applicable and effective in the domain of online-dating.

Later, Pizzato et al. [35] show the importance of taking into account the reciprocity of recommendations in online-dating environments. Namely, in order to generate successful recommendations, the recommender system must measure both to what extent the recommended user fits the service user's preferences and to what extent the service user fits recommended user's preferences. In their study, the authors present a content-based algorithm, named RECON, which, for a potential match, calculates the compatibility of both sides' attributes to the other side's presumed preferences. This method is described in detail in the following section (2.2.1). Pizzato et al. also define a new evaluation metric to assess the performance of the recommender system in providing recommendations which lead to successful interactions. We use this evaluation metric in the evaluation process of our novel recommendation method in Section 6. Xia et al. [48] show that a collaborative filtering method, which contemplates the preferences of both sides of the match, outperforms the content-based algorithm described above. We will present this method in detail in the following subsection (2.2.1) and we will use this method in order to evaluate both our explanation method and our recommendation method.

In [30], Krzywicki et al. present a different approach for recommendations in reciprocal environments. In their work, they present a two-staged recommendation algorithm which first generates recommendations using collaborative filtering and later re-ranks the recommendation with a decision tree "critic". They compare the algorithm with a baseline profile matching algorithm, which matches users according to common attributes and shows that their algorithm is superior. However, this method was not compared to the previous algorithms described above.

A novel and important research area that is related to RRS is *multi-stakeholder* recommender systems. Here the analysis focuses on the various stakeholders involved in the system operation, i.e., not only the end users, but also the owner of the system platform or the suppliers of the recommended items [8]. In [49], the authors use online dating as a case-study for *multi-stakeholder* recommender systems, and propose a recommendation algorithm which balances the preferences of the users and those of the system owner. In this work, we focus only on the users themselves and the balance of their preferences in an individualized manner.

Another popular application of RRSs is in job recruitment sites. Similar to online-dating, a successful match requires mutual interests of both the employee and the employer. Hong et al. [23] introduce several algorithms for recommendations of jobs to employees. They conclude that a job recommender system should apply different recommendation approaches for different users according to their characteristics. Later, in [34], the authors introduce a hybrid content-based and collaborative filtering recommendation method for job seekers, in which chances for a reply of the companies to an application is estimated with a support-vector-machine (SVM) prediction model. They show that their method outperforms previous methods. The 2016 ACM Recommender Systems Challenge [2] focused on the problem of job recommendations. The participant teams were given a large dataset from XING[2], a career-oriented social network, that consisted of anonymized user profiles, job postings and interactions between them. The goal of the teams was to predict job postings with which a user will interact. This problem is somewhat similar to the problem of predicting a user's reply to a message on an online-dating site, which we address in the second part of this work (Section 5.2), but the main techniques proposed in that challenge were actually focusing on the sequential dimension of data and adopted solutions dveloped in the area of session-based, or sequence-aware, recommender systems [39].

2.2.1 Recommendation Methods for RRSs

In this study we focus on recommendations for online dating. As such, we use two state-of-the-art *recommendation* methods that have been developed and tested in online-dating: RECON and TWO-SIDED COLLABORATIVE FILTERING. We will use these recommendation methods in our work, for evaluation of our novel recommendation and explanation methods.

RECON

An RRS in online-dating may provide a user x with a list of recommendations for suitable matches where each recommendation consists of a single user y.

RECON [35] is an effective content-based algorithm, which considers the preferences of both sides of the match. The algorithm was empirically shown to be superior to baseline algorithms which only consider the service user's preferences. In the RECON algorithm, each user x in the system is defined by two components:

1. A predefined list of personal attributes which the user fills out in her profile, denoted as follows:
$$A_x = \{a_v\}$$
where a_v is the user's associated value with attribute a.
2. The preference of user x regarding every attribute a of potential counterparts, denoted $p_{x,a}$, which is represented by the user's message history in the environment:
$$p_{x,a} = \{(a_v, n) : n = \#\text{messages sent by } x \text{ to users with } a_v\}$$
That is, $p_{x,a}$ contains a list of pairs, each consisting of a possible (discretized) value for a and the number of messages sent by x to users characterized by a_v.

[2] https://www.xing.com

Example 1 Bob is a male user who has sent messages to 10 different female users. For simplicity, let us assume each user is only characterized by two attributes: smoking habits and body type. Bob sent messages to female users with smoking habits as follows: 1 smokes regularly, 3 smoke occasionally and 6 never smoke. Regarding their body type: 4 were slim, 4 average and 2 athletic. Bob's preferences would be presented as follows:

$$p_{Bob,smoke} = \{(regularly, 1), (occasionally, 3), (never, 6)\}$$

$$p_{Bob,body-type} = \{(slim, 4), (average, 4), (athletic, 2)\}$$

The RECON algorithm derives the predicted preferences of each pair of users x and y using a heuristic function that reflects how much their respective preferences and attributes are aligned.

Collaborative Filtering RRS

Standard collaborative filtering utilizes similarity relations between users or items in order to generate recommendations. As mentioned above, the preferences of a user in online-dating are commonly inferred from her interaction history [29], as we assume an initial message from user x to user y indicates that y fits user x's preferences. In [48], Xia et al. present a similarity measure for users in online-dating. The similarity between two users x_1 and x_2 is defined as follows:

$$Similarity_{x_1,x_2} = \frac{ReFrom_{x_1} \cap ReFrom_{x_2}}{ReFrom_{x_1} \cup ReFrom_{x_2}}$$

where:

$$ReFrom_x = \{y : y \text{ has received a message from } x\}$$

Similarly, the group of users that sent a message to x is defined as follows[3]:

$$SentTo_x = \{y : y \text{ has sent a message to } x\ \}$$

[3] In [48], $ReFrom_x$ is denoted as Se_x and $SentTo_x$ is denoted as Re_x.

Algorithm 1 Reciprocal Collaborative Filtering Recommendation

```
Input: service user $x$
Output: top-k recommendations
$Recs \leftarrow \emptyset$
for all $y \in RecommendationCandidates$ do
    $score_{x,y} \leftarrow 0, score_{y,x} \leftarrow 0$
    for all $u \in SentTo_y$ do                              ▷ calculate $x$'s interest in $y$
        $score_{x,y} \leftarrow score_{x,y} + Similarity_{x,u}$
    for all $v \in SentTo_x$ do                              ▷ calculate $y$'s interest in $x$
        $score_{y,x} \leftarrow score_{y,x} + Similarity_{y,v}$
    $score_{x,y} \leftarrow \frac{score_{x,y}}{|SentTo_y|}$                          ▷ normalize scores
    $score_{y,x} \leftarrow \frac{score_{y,x}}{|SentTo_x|}$
    if $score_{x,y} = 0$ or $score_{y,x} = 0$ then
        $reciprocalScore_{x,y} \leftarrow 0$
    else
        $reciprocalScore_{x,y} \leftarrow \frac{2}{score_{y,x}^{-1} + score_{x,y}^{-1}}$
                                              ▷ save the harmonic mean of both scores
    $Recs \leftarrow Recs + (y, reciprocalScore_{x,y})$
sort $Recs$ and return top-k
```

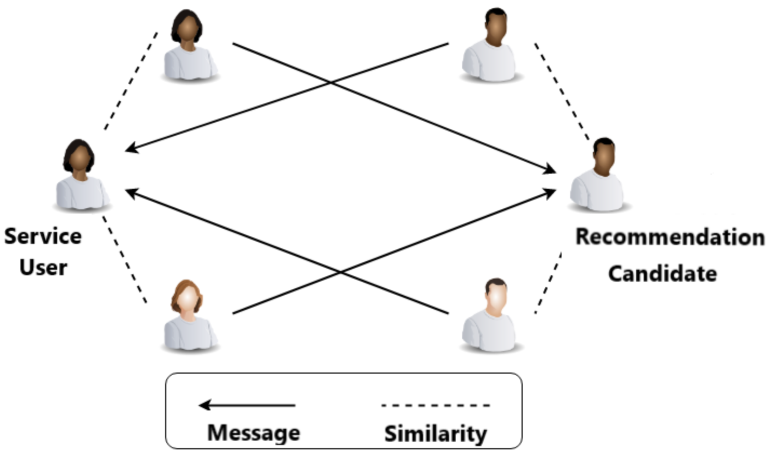


Fig. 1: **Reciprocal collaborative filtering visualization**

Using this similarity measure, Xia et al. [48] introduced a recommendation method for online-dating which utilizes collaborative filtering to measure both the preferences of the service user and the preferences of the recommended user. As mentioned above (Section 2.2), they found that this method significantly outperforms RECON. Algorithm 1 describes this method and Figure 1 illustrates an example for the calculation of the mutual interest between two users[4]. We will refer to this method as Reciprocal Collaborative Filtering (RCF).

3 Generating Reciprocal Explanations

Let us assume a RRS has decided to recommend user y to user x, based on one of the algorithms discussed above. The recommendation may be provided with or without an accompanying explanation. If the explanation only addresses the potential interest of user x in user y (and not vice versa) we refer to it as a *one-sided explanation* and denote it as $e_{x,y}$. Similarly, if the explanation addresses the potential interest of user x in user y *and vice versa*, we refer to it as a *reciprocal explanation* and denote it as $re_{x,y}$.

Naturally, a reciprocal explanation may be decomposed into a pair of one-sided explanations, $e_{x,y}$ and $e_{y,x}$.

The generic framework for providing recommendations with *reciprocal explanations* is provided in Algorithm 2.

Algorithm 2 Reciprocal Explanations

```
Input: User $x$, $GenerateRecommendations$: a Recommendation method, returns a list of
    recommended matches, $Explain$: an explanation method
1: $Output \leftarrow \emptyset$
2: $R \leftarrow GenerateRecommendations(x)$
3: for all $r \in R$ do
4:     $e_{x,r} \leftarrow Explain(x, r)$
5:     $e_{r,x} \leftarrow Explain(r, x)$
6:     $re_{x,r} \leftarrow (e_{x,r}, e_{r,x})$
7:     $Output = Output \cup (r, re_{x,r})$
8: return $Output$
```

Providing a recommendation with a *one-sided explanation* is naturally derived from Algorithm 1 by omitting Row 5 and amending Row 6 accordingly.

In order to implement Algorithm 2, one needs to define both the recommendation method and the *Explain* method. Specifically, one would need to choose the underlying methods to be used in order to provide either a one-sided explanation or reciprocal explanations.

Therefore, as a preliminary step for our main experiment, relying on the general guidelines from previous work, we designed and evaluated explanation methods which were tailored specifically for the domain of online dating. In our investigation, we used an *Explain* method which returns a list of k attributes of the recommended user which can presumably best explain why

[4] This method utilizes user-to-user similarities. Another option for finding the mutual interest is to use item-to-item similarities, meaning the attractiveness similarity of the recommended user to the group of users who received messages from the service user. This option was also examined in [48]. Both of these methods significantly outperformed RECON and there was no significant difference between them. We chose the first method because it performed slightly better than the second.

the recommendation is suitable. This method was shown to be very effective in prior work [45,11]. In order to control information overload, we limited the number of attributes included in the explanation to three, as suggested in [37].

We performed a sequence of experiments which aimed at finding the most effective explanation method in our domain. For example, we investigated the potential of "interaction-based explanations" such as "You may be interested in Bob since you chatted with similar users before." A detailed description of these methods and their evaluation process are presented in Appendix 2. Our results from this preliminary step showed clearly that the "correlation-based" explanation method, which we describe in the following paragraph, is superior to the other methods which were evaluated. Therefore, from this point onward we adopt the correlation-based method as the *Explain* method in our investigation.

The Correlation-based Explanation Method

The correlation-based *Explain* method is inspired by the commonly used Correlation Feature Selection method in Machine Learning [18]. In our context, we would like to measure the correlation between the presence of attribute value a_v in a user's profile and the likelihood that x will choose to send him/her a message. To that end, for each user x, we need to define which users x has viewed in the past and whether she chose to send them a message. Also, we need to identify which of the viewed users is characterized by each attribute value a_v.

Formally, for each user x, we first define $V_x = \{v\}$ as the set of users that x has viewed. We also define:

$$m_x(i) = \begin{cases} 1, & x \text{ sent a message to } v \in V_x \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Meaning m_x is a binary vector of length $|V_x|$ and the value of index i in the vector m_x is 1 if x sent a message to the i^{th} user he viewed and 0 otherwise.

We also define:

$$s_{x,a_v}(i) = \begin{cases} 1, & \text{User } v \in V_x \text{ is characterized by } a_v \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Meaning s_{x,a_v} is also a binary vector of length $|V_x|$ and the value of index i in the the vector s_{x,a_v} is 1 if the i^{th} user viewed by x has attribute a_v and 0 otherwise. Using m_x and s_{x,a_v} we define the correlation-based method described in Algorithm 3.

Algorithm 3 Correlation-based Explanation Method

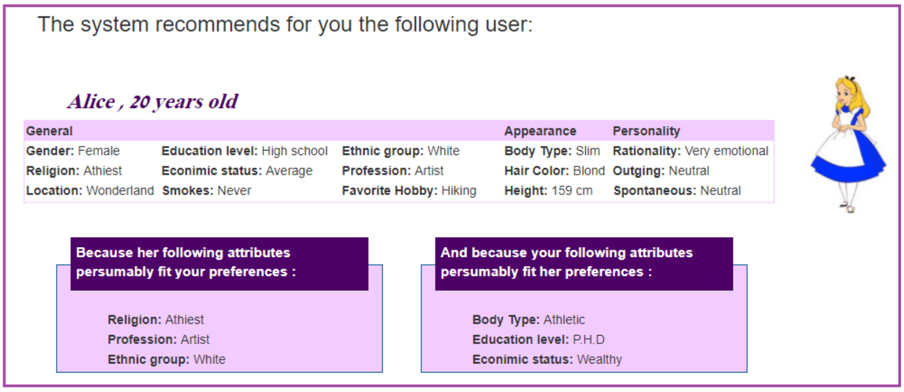
```
Input: two users $x$ and $y$ , number of attributes for explanation $k$.
 1: $temp \leftarrow \emptyset$
 2: obtain $m_x$
 3: for all attributes a $\in A$ do
 4:     obtain the value $a_v$ of attribute a in $A_y$
 5:     obtain $s_{x,a_v}$
 6:     $w_{a_v} = CORRELATION(m_x, s_{x,a_v})$
 7:     $temp = temp \cup (a_v, w_{a_v})$
 8: sort $temp$ by the values $w_{a_v}$
 9: $e_{x,y}$ = top-$k$ attribute values of $temp$
10: return $e_{x,y}$
```

This algorithm attempts to explain why user y is suited for user x. The Algorithm iterates over all attributes of user y (line 3), and for each attribute it calculates the correlation between x's preferences and y's attribute. Later it sorts all of the correlation measures (line 9) and returns the attributes of y which are most correlated with x's preferences (line 10). The CORRELATION function, used in line 6, measures the correlation between x's presumed preferences and y attributes, For this study, we adopted the well-known Pearson correlation [5], yet other measures such as Cosine and Jaccard similarity may be applied.

In order to illustrate the correlation-based method we continue example 1 from above. Assume an RRS has decided to recommend Alice, who never smokes and is slim, to Bob. Recall that Bob sent 6 messages to users who never smoke and 4 to slim users. Now say Bob viewed a total of 25 users, of whom 18 never smoke and 4 were slim. In other words, Bob sent messages to only a third of the users he viewed who never smoke, and to all users he viewed who are slim. Thus, for $k = 1$ the correlation-based method would find a stronger correlation between the presence of "slim body" and Bob's messaging behavior, hence "slim body" would be provided as an explanation.

4 Empirical Investigation of Reciprocal Explanations

In order to evaluate and compare the one-sided and reciprocal explanation methods, we performed three experiments: two in a simulated online-dating environment developed specifically for this study and one in an operational online-dating platform. Each environment has its own benefits: Results from the operational online-dating platform naturally reflect the real-world impact of both explanation methods, whereas in the simulated environment one receives detailed and explicit feedback from the users, which otherwise would be impractical to gather in an active online-dating platform. In addition, a simulated environment is not constrained by privacy issues which are present in online environments. We discuss these experiments below.



The system recommends for you the following user:

Alice , 20 years old

General			Appearance	Personality
Gender: Female	**Education level:** High school	**Ethnic group:** White	**Body Type:** Slim	**Rationality:** Very emotional
Religion: Athiest	**Econimic status:** Average	**Profession:** Artist	**Hair Color:** Blond	**Outging:** Neutral
Location: Wonderland	**Smokes:** Never	**Favorite Hobby:** Hiking	**Height:** 159 cm	**Spontaneous:** Neutral

Because her following attributes persumably fit your preferences :

Religion: Athiest
Profession: Artist
Ethnic group: White

And because your following attributes persumably fit her preferences :

Body Type: Athletic
Education level: P.H.D
Econimic status: Wealthy

Fig. 2: A recommendation with a reciprocal explanation in MM.

4.1 The MATCHMAKER Simulated Environment

We created a simulated online-dating platform, which we call MATCHMAKER (MM for short). Using MM, users can view profiles of other users, interact with each other by sending messages and receive recommendations from the system for suitable matches, as common in online dating environments. With the collaboration of experts in online-dating who did not co-author this paper, we designed MM's features to reflect those of popular online-dating platforms. Figure 2 presents a snapshot of a recommendation in the MM platform. The explanations for the recommendation in MM were generated from: 1) a general word template; 2) A personalised list of features for each user, which were chosen according a method described in Algorithm 3.

MM is a web-based platform and can be accessed at *www.biu-ai.com/Dating*.

In order to develop a RRS for MM, it is necessary to obtain the attributes and preferences of both of the participants in the experiment and the potential recommended users. In order to create profiles in MM which would be as realistic as possible, we used the public attributes of profiles from real online-dating sites, such as *www.date4dos.co.il*.

However, note that the data set does not contain the users' message history or preferences, hence, by using only that information, the designed RRS would be very limited. To overcome this challenge we performed the following data collection: We recruited 121 participants, 63 males and 58 females ranging in age between 18 and 35 (average 23.3), all of whom are self-reportedly single and heterosexual. All participants were university students recruited by posting ads in relevant classes. First, the participants entered MM and filled out a personal attributes questionnaire that is common in online-dating platforms (e.g., age, occupation). Later, the participants viewed the profiles obtained from the real online-dating site as discussed above and sent fictitious messages to the profiles that they perceived as suitable matches[5]. Participants were instructed to view at least thirty profiles and to send messages to at least ten relevant profiles in order to generate sufficient data for deriving their preferences. An average of 50.72 profiles (s.d.= 30.99) were viewed and 11.92 messages (s.d.=3.96) were sent by each participant. The data of three of the participants was removed because they did not comply with our instructions.

Following the above data collection procedure, we obtained 118 participant profiles and preferences. We anonymized the participants' profiles and preferences and used them as the initial profiles in MM for later investigation.

4.2 Evaluation in a Simulated Online-dating Environment

One of the main challenges in designing a realistic online-dating environment is the challenge of incorporating and modeling the costs and potential gains associated with accepting recommendations in the platform. Specifically, previous research has shown that different costs, especially the emotional one, such as that produced by fear of rejection, play prominent factors in determining the behavior of users in online dating platforms [21,48]. Since the costs and potential gains involved with the acceptance of a recommendation (i.e., sending a message to the recommended user) may vary significantly between users, we consider two models: First, a model in which no explicit cost is introduced. Specifically, users are asked to rate the relevance of the recommended profiles without encountering any explicit cost or gain, as in the preliminary investigation described above. We then considered a model in which explicit costs and potential gains are associated with accepting recommendations and users are incentivized to maximize their performance. The first model will assist us in understanding the effects of the explanation method when the cost is negligible, and the second when the cost is significant.

4.2.1 Negligible Cost

In order to compare the two *Explain* methods, we used the MM simulated system discussed above. We asked 59 of the 118 participants who took part in the data collection phase to reenter the MM platform to receive system recommendations. 54 percent of the participants were female and the average age of the participants was 23.1 (s.d.= 2.58). Each participant was randomly assigned to one of two conditions: 1) one-sided explanations (30 participants); and 2) reciprocal explanations (29 participants). The participants received five recommendations, generated by the RECON algorithm, with an explanation corresponding to their condition. Participants were asked to rate the *relevance* of each recommendation separately, on a five point Likert scale from 1

[5] Participants were aware that the profiles were simulated although based upon real data and that the messages were not actually sent to recipients. They were guided to send simulated messages to profiles they viewed as relevant matches for them.

(extremely irrelevant) to 5 (extremely relevant), followed by the user experience questionnaire. The questionnaire included questions which are commonly used for measuring prominent factors in user experience. The questionnaire and the results from the questionnaire are presented in detail in Appendix 1. In this setting, our working hypothesis was that the reciprocal explanation would have a significantly different effect on the participants in comparison with the one-sided explanations. However, since the recommendations in this setting did not involve cost, our hypothesis was non-directional, meaning we did not expect that the influence of reciprocal explanations would necessarily be positive or negative.

RECON was chosen as it is a well-known content-based recommendation method for online dating. For this evaluation, we needed a content-based recommendation method since the recommended profiles were specifically created for generating these recommendations and therefore recommendations could not be generated by collaborative filtering methods which require interaction data.

Results: All data was found to be distributed normally according to the Anderson-Darling normality test [40]. In contrast to what one may expect, the one-sided explanation outperformed the reciprocal explanation. Specifically, using a *two-tailed unpaired t-test*, we found that the reported relevance of the one-sided explanation condition was significantly higher than the (reported relevance of) reciprocal explanation condition (one-sided: mean= 3.76, s.d.= 0.62 vs. reciprocal: mean=3.34, s.d.= 0.85, $p \leq 0.04$). In addition, the results from the questionnaire (described in detail in Appendix 9) showed that the participants in the one-sided explanation condition were more satisfied with the recommendations and trusted the system more, compared to the reciprocal explanation condition.

Due the relatively small sample sizes it is extremely difficult to assess the differences between subgroups of the conditions and participants. For example, it is difficult to derive insights as to the possible difference in how females benefit from reciprocal explanation compared to the one-sided explanation condition. The experiment in the active online-dating application, described in Section 4.3, includes a significantly larger sample size and thus enables us to statistically analyze such subgroups.

4.2.2 Explicit Cost

For this experiment, we recruited 67 new participants who had not yet participated in this study (35 male and 32 female) ranging in age from 18 to 35 (average= 24.8 s.d.=4.74). Participants were then randomly assigned to one of the two conditions: one-sided explanations or reciprocal explanations. As was the case in the negligible-cost setting, participants created profiles, browsed profiles and sent messages to users they viewed as potential matches. However, in the recommendation phase, the participants were given an incentive to maximize an artificial score which was effected by costs and gains as follows: Upon receiving a recommendation, each participant had two options – either send a message to the recommended user or not. If the participant did not send a message, she did not gain or lose any points. If the participant did send a message, the recommended user returned a positive or negative reply according to a probability derived from the recommended user's preferences. Specifically, we used the interest of the recommended user in the participant, as estimated by the RECON algorithm. Participants were informed that the probability is based on the preferences of the recommended user. If the recommended user replied positively, the participant gained points proportional to how RECON estimated that the recommended user fit the user's preferences (between three and four points). If the recommended user replied negatively, the participant lost three points. This scoring scheme was chosen in order to encourage users to send messages to other users in whom they are interested while considering the probability of being rejected. Participants were paid with respect to their score. Participants who received positive replies to their messages were paid between 3 and 5 American dollars. Complete technical details about this scoring and payment methodology are available on the MM website. Each participant then received 5 recommendations accompanied by an explanation according to their assigned condition. In this setup, we define the *acceptance rate* as the number of recommended users to which the participant chose to send messages. Later the participants filled out the user experience questionnaire as done in the previous setups. In this setting, our working hypothesis was that the reciprocal explanation condition would outperform the one-sided explanation condition, since we believed that reciprocal explanations would decrease the participants' uncertainty and reduce concerns regarding the cost.

Results: As opposed to the results of the previous experiment, the results here show a significant benefit to the reciprocal explanations method compared to one-sided explanations. Specifically, the acceptance of the reciprocal explanation condition was reported to be significantly higher than the one-sided condition (one-sided: mean=2.83 s.d.=0.87 vs. reciprocal: mean=3.49 s.d.=1.02, $p \leq 0.01$). Also, participants' trust in the system was found to be higher under the reciprocal explanation condition (detailed results from the questionnaire are presented in Appendix 1). The results are public and can be accessed at *http://www.biu-ai.com/Dating/Home/Results*.

4.3 Evaluation in an Active Online-dating Application

After completing both experiments in the MM environment, we contacted *Doovdevan*, an Israeli online-dating application, and received permission to conduct a similar experiment within their application, using active users as participants.

Doovdevan is a web and mobile application customized for android and iOS operating systems. Similar to other online-dating applications, users of this platform can create profiles, search for possible matches and interact with other users via messages. Doovdevan currently consists of about $40,000$ users and is growing rapidly. We chose to perform our experiment on Doovdevan since it is relatively new and none of the users had received recommendations from the system prior to the experiment. This was important since previous recommendations can affect the trust of the users in the system and subsequently effect their attitude towards new recommendations [28,9]. The recommendation algorithm that was implemented in the Doovdevan application was the RECIPROCAL COLLABORATIVE FILTERING method described above in Section 2.1.

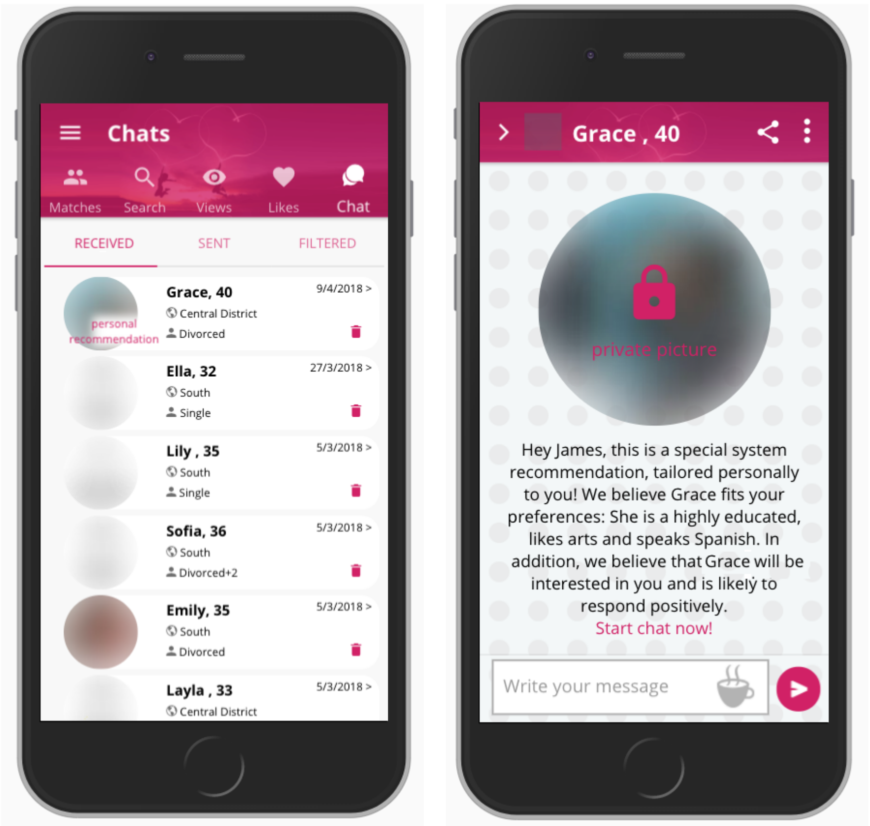
We randomly selected a group of 133 active users on the site (i.e., users who logged on to the platform at least once in the week prior to the experiment), 73 males and 60 females, ranging in age from 18 to 69 (mean= 36.27, s.d.= 13.01), and randomly assigned them to one of the two examined conditions: one-sided explanations or reciprocal explanations. Due to privacy concerns, we were not permitted to reveal the recommended user's preferences to the recommendation receiver. Therefore, the reciprocal explanation included two (asymmetrical) parts: First, an explanation of the presumed interest of the recommendation receiver in the recommended user, including specific attributes of the recommended user, as done in the simulated MM environment. Second, a statement that the system believes that the recommendation receiver fits the recommended user's preferences, thus he or she is likely to reply positively.

4.3.1 The Service User's Interface

The service user's interface supports three interaction stages:

1. The system generates a recommendation and sends it to the service user's inbox. In addition, the service user receives a notification on her smartphone. The recommendation has a unique label that distinguishes it from other incoming messages in the inbox (left snapshot in Figure 3). The recommendation includes a brief description of the recommended user: low-resolution picture, name, age, location, marital status. The service user may decide to click on the recommendation.
2. If the user clicks on the recommendation, she moves to a new screen showing a higher quality picture of the recommended user and an explanation accommodating the recommendation (right snapshot in Figure 3). The service user can then decide whether or not to send a message to the recommended user.
3. If the service user opts to send a message to the recommended user, the recommended user can reply with a message or ignore the chat request.

As in the previous experiment, each participant received five recommendations. However, unlike previous experiments, with Doovdevan only one recommendation was sent per day, based on the advice from the site owner who suggested that users would find it odd to receive multiple recommendations in a single day after not receiving a single recommendation thus far. Unlike the MM environment, in Doovdevan we could not explicitly ask participants for their experience. Therefore, we measure the *acceptance rate* of the provided



Hey James, this is a special system recommendation, tailored personally to you! We believe Grace fits your preferences: She is a highly educated, likes arts and speaks Spanish. In addition, we believe that Grace will be interested in you and is likely to respond positively.

Fig. 3: **Screen shots of the recommendation's user interface.** The left image is a screen shot of the inbox of a user who received a recommendation. In this case, the recommendation appears at the top. The right image is what the user sees after clicking on the recommendations, accommodated with a reciprocal explanation. The pictures are blurred for reasons of privacy.

recommendations as the number of recommendations that resulted in the recommendation receiver sending a message to the recommended user divided by the number of recommendations the recommendation receiver had viewed. Although the recommendations in this real-world setting did not involve any monetary cost, we expect that the *emotional cost*, which is an established prominent factor in decision-making in online-dating environments [22], will have an effect similar to the explicit cost in the simulated environment (Section 4.2.2). Therefore, we hypothesized that the reciprocal explanation condition will have a higher acceptance rate, similar to the results of the simulated environment.

All data was found to be distributed normally according to the Anderson-Darling normality test. We compared both conditions using a t-test. The

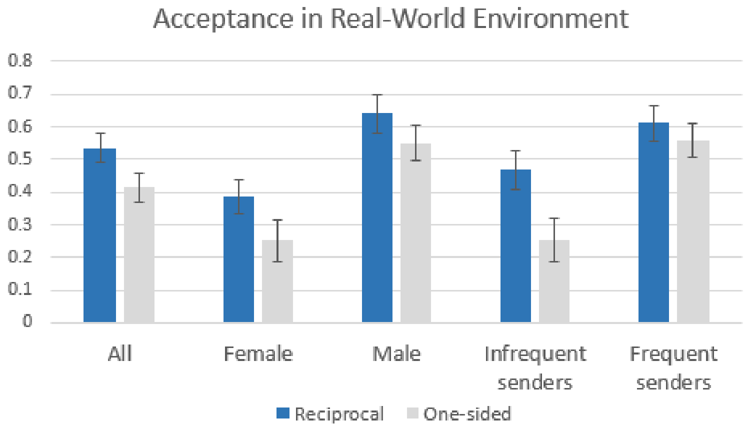


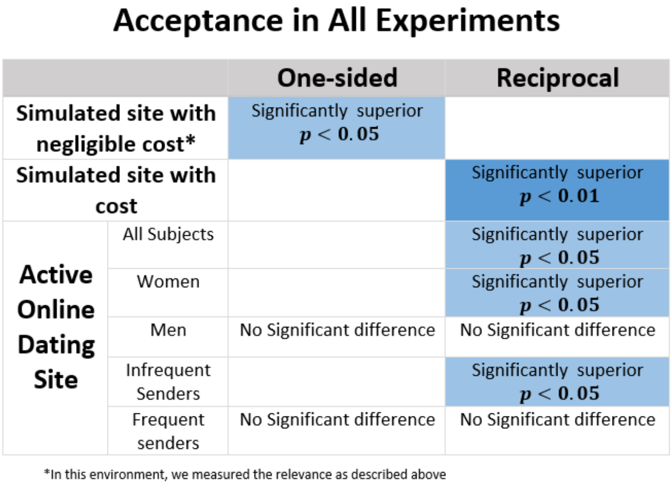
Fig. 4: Reciprocal vs. one-sided explanations in a real online-dating environment. Error bars represent the standard error.

results show that users who received reciprocal explanations presented significantly higher acceptance rates compared to users who received one-sided explanations ($p < 0.05$). Specifically, on average, users who received reciprocal explanations sent messages to 54% of the recommended users they viewed while the same was true for only 42% of the recommended users under the one-sided explanations condition.

Interestingly, we find that reciprocal explanations outperform one-sided explanations for *women* while they do not show a statistically significant difference for men. Specifically, for women we find an average acceptance rate of 39% under the reciprocal explanation condition while only 25% under the one-sided explanations condition. For men, we find that the reciprocal explanation method achieves an average acceptance rate of 64% compared to 55% under the one-sided explanation method, but the difference is not statistically significant.

We further analyze the explanations' effect on users who sent more or fewer messages than the median number of messages sent by users in the system. We found that for the group who sent fewer messages than the median, the reciprocal explanation significantly outperformed the one-sided explanation, averaging a 47% acceptance rate compared to 25% under the one-sided explanations condition. For the complementary group, the reciprocal explanation averaged approximately 61% compared to 56% in the one-sided explanation, without a significant difference between the two. The results are presented in Figure 4.

We also examined the number of log-ins of the participants in the week following the recommendation, which can be considered as an additional potential indicator of the impact of the explanation method. The results show that the participants under the reciprocal explanations condition logged-in significantly more often than those under the one-sided explanations, with an



Acceptance in All Experiments

		One-sided	Reciprocal
Simulated site with negligible cost*		Significantly superior $p < 0.05$	
Simulated site with cost			Significantly superior $p < 0.01$
Active Online Dating Site	All Subjects		Significantly superior $p < 0.05$
	Women		Significantly superior $p < 0.05$
	Men	No Significant difference	No Significant difference
	Infrequent Senders		Significantly superior $p < 0.05$
	Frequent senders	No Significant difference	No Significant difference

*In this environment, we measured the relevance as described above

Fig. 5: Summary of results from all experiments evaluating reciprocal explanations

average of 56 log-ins compared to 23 log-ins under the one-sided explanations condition ($p <= 0.05$). It is worth saying that, on one hand, these results can indicate that the users who received reciprocal explanations were more satisfied with the system. On the other hand, an alternative explanation can suggest that the users who received reciprocal explanations were expecting more responses from the users they contacted, since the system suggested that the latter are likely to be interested in them. Given the available data, we cannot conclusively say which interpretation is the correct one.

4.4 Explanation Provision: Conclusions

In summary, the main results from all experiments are presented in Figure 5. The results from both the synthetic and real-world investigations suggest that the choice of explanation method depends on the users' cost for following the recommendations. Specifically, in environments where the cost of accepting a recommendation is high, the reciprocal explanations favorably compare to one-sided explanations. We argue that this is because the additional information in the reciprocal explanation makes the user feel more confident in the outcome of accepting the recommendation, and subsequently this increases her willingness to "take the risk".

The results are consistent with previous research which found that many users in online-dating platforms have an emotional cost for sending a message, mainly due to the fear of rejection [21,22,3]. Specifically, when the fear of rejection was removed, as in our first simulation, the one-sided explanation method was found to be superior. In addition, our findings align with recent research which found that the cost associated with the advice has a significant effect on the acceptance of the recommendation [44].

We further find that not all users respond to explanations in the same way, possibly suggesting that a "one-size-fits-all" explanation method is not likely to be found. Specifically, the cost associated with accepting a recommendation may vary between users. Previous work in the online dating domain has revealed that men tend to focus more on their own preferences compared to women who take into account their own attractiveness to the other side of the match [48,25]. We find support for these insights in our study as well. We further find that users who are more "choosy" in their messaging behavior tend to benefit more from reciprocal explanations compared to other users. These differences between males and females or frequent and infrequent senders possibly indicate an underlying factor of emotional cost for sending messages, which is more likely to be prominent in infrequent message senders and females [22].

5 Reciprocal Recommendation Generation

The results from the previous experiments indicate that users in reciprocal environments, and specifically in online-dating, are very different in the extent to which they consider the other side's preferences. This aligns with research focused on online-dating, which has demonstrated that users differ in the extent to which they consider the likelihood that the contacted partner will reply positively before they send a message [22,48]. Existing recommendation generation methods do not account for this apparent difference and hence cannot be considered fully personalized. Namely, existing algorithms assign equal importance to the perceived preferences of both sides when generating recommendation while, in fact, users vary in how they act upon their preferences.

In order to illustrate the need for personalization in RRSs, let us consider the following example:

Example 2 Alice and Bob are users in an online-dating environment. Bob sends messages to a wide range of users without considering whether or not his messages will be accepted. In addition, Bob is unpopular (i.e., rarely receives messages) and the rate of positively replied messages to Bob is low. Therefore, in order to maximize successful interactions, an optimal RRS should generate recommendations for Bob that mainly focus on the chances for a reply. Alice, on the other hand, is very cautious about sending messages and only sends to a narrow range of users. In addition, she has a high rate of positive message replies. Therefore, in contrast to Bob, for Alice a RRS should generate recommendations that give more importance to the chances she will find a potential partner appealing.

In order to overcome this limitation in existing methods, we designed a new recommendation method called Reciprocal Weighted Score (RWS for short). RWS separately calculates the service user's presumed interest in the recommended user and the likelihood that the recommended user will reply positively. Then, for each service user, RWS balances these two scores in order to maximize the likelihood of initiating a successful interaction through the provided recommendation. This optimal balance is tailored individually for each service user according to her interaction history, and thus RWS is fully personalized. RWS will be described in detail in the following subsection.

In addition, we found that the current methods lacked real-world evaluation. Namely, to the best of our knowledge, prior RRS methods were only evaluated in an offline fashion, based on historical data, and were not integrated into an operational system in order to provide recommendations and investigate their real world effect. Recently, there has been a growing recognition in the recommender systems community that conclusions from laboratory-based evaluations may not be confirmed in live-user studies [41,31]. Therefore, we found it necessary to conduct online experiments in an operational dating site in order to evaluate the real world effect of our recommendation method alongside the state-of-the-art methods.

5.1 Optimal Weighting Approach

In this section, we introduce RWS, a novel algorithm for RRSs. In order to measure the compatibility of a recommendation of user y to user x we initially calculate two measurements separately: we use user-to-user similarity in order to estimate x's interest in y and an AdaBoost machine learning model for predicting whether y will respond positively to x.

In contrast to previous methods, which assign equal importance to the preferences of both sides for all users, in RWS the relative importance of the two is tailored for each user individually. Namely, based on each user's previous interactions, we optimize the relative importance of the two scores for this specific user and tune the weights accordingly. Figure 6 is a diagrammatic representation of our recommendation method.

As mentioned above, RWS measures the preferences of the service user using collaborative filtering. This score is calculated identically to the way in which the service user's preferences are calculated in the RCF method described above (lines 4-5 in Algorithm 4). We denote the score of user x's interest in user y as $CF_{x,y}$. The second score is calculated by an AdaBoost machine learning prediction model (described in Section 5.2), which predicts the chances of a positive reply from user y to user x following an initial message from user x to user y. We denote this measure as $PR_{y,x}$.

In contrast to the RCF method, RWS uses two different prediction models since the prediction tasks are inherently different. The prediction of the service user's preferences is similar to non-reciprocal recommendation tasks and therefore we use collaborative filtering methodology. On the other hand, the prediction of the reply of a recommended user can be better framed as a 2-class classification problem.

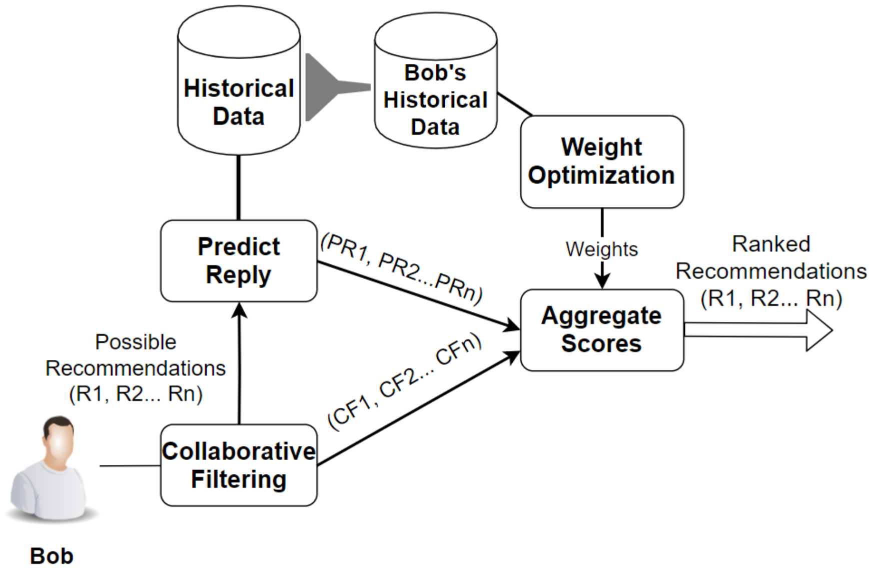


Fig. 6: A diagrammatic representation of RWS. The "Predict Reply" component is described in Section 5.2 and the "Weight Optimization" component is described in Section 5.4

5.2 Predicting Replies of Recommended Users

Our reply prediction model was trained on 35,000 samples of messages contained in the dataset provided by Doovdevan (the active online dating environment, described above in Section 4.3), each including a list of features. We used the AdaBoost classifier, which we found to outperform other machine learning algorithms we tried on our dataset. Since we are interested in the probability estimation of the algorithm and not in the classifications themselves, we use the standard AUC measure and present the performance results below. The samples in the dataset are classified into two classes: 1) positive reply and 2) no reply or negative reply[6]. As our goal in this work is to increase positive interactions, we do not distinguish between a negative reply or no reply at all.

The features of a message sent to a recommended user can be divided into two main groups: 1) features describing the sender and 2) features describing the recipient. Each of these groups can be divided into two subgroups: 1) attributes of the user from her public profile, for example: age, gender, height, profession; and 2) features describing the activity and popularity of the user, such as the number of received messages, sent messages, views and logins. We

[6] We manually classified all of the samples, which included a response into two classes: 1) positive response; and 2) negative response.

first consulted a domain expert, who does not co-author this paper, in order to find potentially influential features, and later we reduced the number of the features to 54 using the backward elimination feature-selection method [27]. In Table 1 we present the most prominent features, ordered by their information gain [17].

	Feature
Features of recipient	1) Percent of positively replied messages before current message. 2) Log-ins to the environment in the week before the message. 3) Number of profiles he/she viewed.
Features of sender	4) Number of users who viewed him/her. 5) Number of messages he/she received.

Table 1: Prominent features used in the reply prediction model, ordered by their information gain.

We denote the vector of feature values for a specific service user x as:

$$\mathbf{x} = (x_1, x_2, x_3 ... x_s)$$

where s is the number of the sender's features.

Similarly, we denote the vector of feature values for recommendation candidate y as:

$$\mathbf{y} = (y_1, y_2, y_3 ... y_r)$$

where r is the number of the recipient's features.

For any given service user x and potentially recommended user y, we denote the probability for a positive response of y to a message from x as:

$$PR_{y,x} = h\left(\mathbf{x}, \mathbf{y}\right)$$

where h is the function learned by the AdaBoost model, which returns the probability (value between 0 and 1) for a positive reply.

Our dataset is highly imbalanced, with only 7% of the initial messages classified as positively replied. Therefore, in order to balance our dataset, we used the/a standard oversampling class-balancing technique [4]. The area under the curve (AUC) of the model is 0.833[7].

In the next section we describe how RWS leverages this prediction model in order to generate recommendations.

[7] For comparison, following are the best AUC scores received by other prediction models which we tested: 1) random forest classifier: 0.798; 2) logistic regression: 0.795; 3) multi-layer-perceptron classifier: 0.791; 4) Gaussian naïve Bayes classifier: 0.672.

5.3 Optimally Balancing Receiver and Recommended Users' Importance

In Algorithm 4 we give the general scheme for our recommendation algorithm, where *ServiceUserFeatures* (row 10) is a function which obtains the service user x's feature vector $\mathbf{x}$, as denoted above, and *RecommendedUserFeatures* (row 11) obtains the recommended user's feature vector $\mathbf{y}$.

Algorithm 4 Reciprocal Weighted Score Recommendations Scheme

```
Input: service user $x$
Output: top-k recommendations
 1: $Recs \leftarrow \emptyset$
 2: for all $y \in RecommendationCandidates$ do
 3:     $CF_{x,y} \leftarrow 0$
 4:     for all $n \in SentTo_y$ do                          ▷ calculate $x$'s interest in $y$
 5:         $CF_{x,y} \leftarrow CF_{x,y} + Similarity_{x,n}$
 6:     $CF_{x,y} \leftarrow \frac{CF_{x,y}}{|SentTo_y|}$                              ▷ normalize score
 7:     if $CF_{x,y} = 0$ then
 8:         $reciprocalScore_{x,y} \leftarrow 0$
 9:     else
10:         $\mathbf{x} \leftarrow ServiceUserFeatures(x)$
11:         $\mathbf{y} \leftarrow RecommendedUserFeatures(y)$
12:         $PR_{y,x} \leftarrow PredictReply(\mathbf{x}, \mathbf{y})$
                                                          ▷ predict $y$'s response to $x$
13:         $\alpha \leftarrow OptimizedWeight(x)$
14:         $reciprocalScore_{x,y} \leftarrow (\alpha \cdot CF_{x,y} + (1 - \alpha) \cdot PR_{y,x})$
                                                                ▷ aggregate scores
15:     $Recs \leftarrow Recs + (y, reciprocalScore_{x,y})$
16: sort $Recs$ and return top-k
```

The *PredictReply* (row 12) function returns the probability of a positive reply according to our predictive model function h. The *OptimizedWeight* function (row 13) retrieves a weight, optimized specifically for the service user, as described in Section 5.4. Later (row 14), our method utilizes these weights to aggregate the CF and PR scores into a single score that resembles the reciprocal interests of the match.

Notice that for a given user x, the algorithm only predicts the probability of a reply for potentially recommended users y where $CF_{x,y}$ is not null (rows 7-9). In this way we reduce the size of possible candidates to a smaller subset of users. The importance of this reduction will be discussed in detail in Section 6.3.

In the following section we will describe the method we use to optimize these weights.

5.4 Weight Optimization

We aim at finding, for each service user x, a weight α_x which balances $CF_{x,y}$ and $PR_{y,x}$ (for each recommendation candidate y) so that it will optimize

x's successful interactions. We denote the weighted score of user x for the recommended user y as $RWS_{x,y}$, and it is calculated as follows:

$$RWS_{x,y}(\alpha_x) = \alpha_x \cdot CF_{x,y} + (1 - \alpha_x) \cdot PR_{y,x}$$

Note that both the CF and PR scores are normalized and we use the standard score [10] rather than the original score.

In order to find a specific weight optimized for x, we observe from the user's *interaction history* the influence of each score (CF and PR) on her successful interactions. We first define the following sets:

$$SuccInter_x = \{y : x \text{ has sent an initial message to } y \\ \text{and } y \text{ replied positively}\}$$

$$V_x = \{y : x \text{ has viewed } y\}$$

In addition, we denote with $RWS_{x,*}(\alpha_x)$ the scoring function of user x when a particular α_x is employed. Moreover, we denote with $Rank_y\ (RWS_{x,*}(\ \alpha_x))$ the rank position of y in the list of the viewed users $v \in V_x$, sorted by decreasing value of the scoring function of user x.

We now define our target optimization problem for a specific user x, which we denote as $IndividualOptimization$:

$$\begin{aligned} \underset{\alpha_x}{\text{minimize}} \quad & \sum_{v \in V_x} \mathbb{1}_{v \in SuccInter_x} Rank_v\ (RWS_{x,*}(\ \alpha_x)) \\ \text{subject to} \quad & 0 \geq \alpha_x \geq 1 \end{aligned}$$

By solving this optimization problem, we find the weight α_x which will rank the users with whom x had successful interactions higher than all other users that x has viewed. For the implementation of the optimization, we used Brent's (numerical analysis) method [6], which finds a local minimum in a given interval.

In Figure 7 we show the distribution of the alpha weight for 765 male and 566 female users randomly chosen from the Doovdevan environment. These figures demonstrate the importance of individual weighting: it is clear that in order to maximize the successful interactions, the balance of the CF and PR scores must vary substantially among different users. We can also observe that for most of the women the CF score is a stronger indicator for a successful interaction than PR, while for most of the men PR is a better indicator. This difference can possibly be attributed to the fact that women are relatively more attentive to the preferences of the other side [48, 22] and therefore their CF measure already captures (to a certain degree) the chances for a positive reply.

Notice that the optimization problem described above is only effective for users who have had at least one successful interaction. In order to also recommend to users who have not had previous successful interactions, we

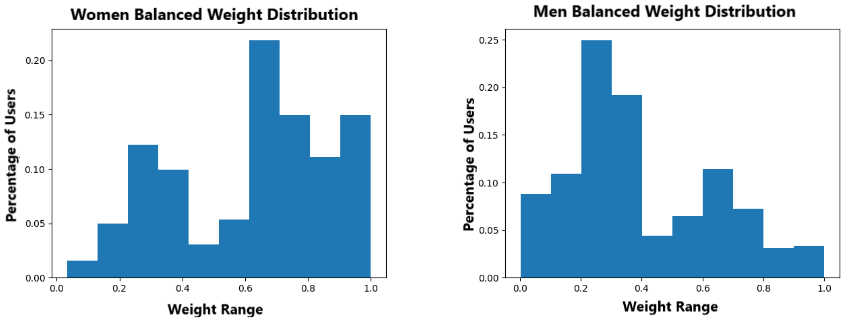


Fig. 7: **Distribution of the alpha weight for women and men in the Doovdevan environment**

define a similar optimization problem *over all users*, which we denote as *GlobalOptimization*:

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} & & \sum_{u \in U} \sum_{v \in V_u} \mathbb{1}_{v \in SuccInter_u} Rank_v \left(RWS_{u,*}(\alpha) \right) \\ & \text{subject to} & & 0 \geq \alpha \geq 1 \end{aligned}$$

where U is the set of all users and $Rank_v\left(RWS_{u,*}(\alpha)\right)$ is the rank position of v in the ranked list of all the users in U sorted by decreasing value of the $RWS_{u,*}(\alpha)$ score. In our environment, the calculated value of α for this global optimization problem is 0.3978. This result shows that, as expected, the PR score is a better predictor of a successful interaction than the CF score for the average user.

Additional techniques for mitigating the cold start problem could include semi-personalized weighting schemes relying on gender, age and other socio-demographic information.

6 Evaluation of the Reciprocal Recommender

In order to evaluate RWS we performed 2 experiments: The first experiment was done offline, using historical data from Doovdevan. The second experiment was online, where active users from Doovdevan received recommendations from the system. As mentioned above, we prefer the online evaluation, since the results in the offline evaluation do not necessary reflect real world impact. Nevertheless, the offline evaluation was necessary in order verify that the RWS method is efficient before the online experiment (which is more “expensive”). In addition, the offline evaluation examines different variations of RWS, as explained in the following subsection.

6.1 Offline evaluation

As an initial step, we evaluated our recommendation method, RWS, in an offline fashion using historical data of 7668 active users in the Doovdevan environment. As a baseline approach, we used the RCF method (described above in Section 2.2.1), which has been shown to be superior to all methods proposed previously (see Section 2). Recall that there are two differences between our proposed method (RWS) and the RCF method. The first difference is the way the preferences of both sides are balanced: RWS tailors an optimal balance for each user and RCF gives equal importance. The second difference is the prediction model used to predict the chances of a reply: RWS uses an AdaBoost prediction model which predicts the chances for a positive reply, while RCF uses collaborative filtering. In order to evaluate the influence of each of these factors alone, we added two hybrid recommendation methods to the offline evaluation, which combine factors from both RWS and RC:

1. *Weighted-RCF.* In this method, the prediction model for the preferences of both sides utilizes collaborative filtering, as in the RCF method. However, unlike the RCF method, here the prediction scores are balanced by a weight, which is optimized for each service user individually, as in the RWS method.
2. *Equal-PR.* In this method, we use CF to predict the service user's interest and an Adaboost prediction model to predict the reply as in the RWS method. However, unlike the RWS method, both prediction scores are given equal weights, as in the RCF method.

For a given service user, we ranked the possible recommended users by comparing their aggregated scores.

We evaluated the performance of each method by comparing the top-K users in the recommendation list with the users actually contacted by the service user according to Doovdevan's data. The evaluation was calculated per day. In other words, for each day where the user was active, we used all the data collected before that day for training the model and generating recommendations. Later, we compared the recommendations with the users contacted by the service user during that specific day. For each user we evaluated the prediction accuracy for ten days and averaged all of the results into a single result.

Evaluation Metrics for offline evaluation

For a given user x, we define the following sets of users:

1. R_x is the set of users who were recommended to x.
2. RM_x is the set of users who were recommended to x and received a message from x during the evaluation period.
3. M_x is the set of users who received a message from x in the evaluation period.

4. RI_x is the set of users recommended to x and their recommendations were followed by a successful interaction, i.e., x sent a message to the recommended user and the recommended user replied positively.
5. I_x is the set of users with whom x initiated a successful interaction during the evaluation period.

In order to evaluate the recommendation methods, we use four measures for evaluation of RRSs defined in [35, 48]. The first two measure the accuracy of the recommendation methods in recommending users who will receive messages from the service user x:

$$MPrecision_x = \frac{|RM_x|}{|R_x|} \quad MRecall_x = \frac{|RM_x|}{|M_x|}$$

The second two measures measure the accuracy of the recommendation methods in predicting successful interactions, and are calculated as follows:

$$RPrecision_x = \frac{|RI_x|}{|RM_x|} \quad RRecall_x = \frac{|RI_x|}{|I_x|}$$

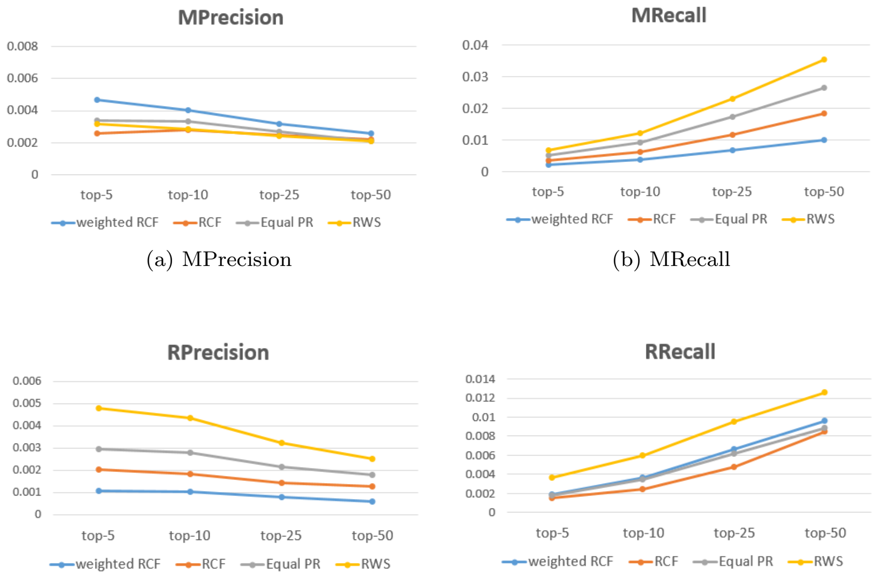
6.2 Offline Evaluation Results

We evaluated the performance of the recommendation methods by comparing the mean of the measures defined above using a repeated measures analysis of variance (ANOVA) test [12] followed by the Tukey's HSD post-hoc test [1]. All of the measures for both conditions were found to be distributed normally according to the Anderson-Darling normality test [40], and the ANOVA test determined that there were significant differences in all measures.

We first compared the MPrecision and MRecall, which measure the success of the recommendation methods in recommending users who are likely to be contacted by the service user. We evaluated top-k recommendation, where k was set to either 5, 10, 25 or 50. We found that the Weighted-RCF method outperformed all other methods significantly in both MPrecision and MRecall[8] ($p < 0.01$). In addition, the RWS method outperformed RCF and Equal-PR in MRecall. We did not find a significant difference between any of the other groups.

We also compared the RRecall and RPrecision measures, which evaluate the success of the recommendation method in recommending users who are likely to be contacted and reply. We found that our proposed method, RWS, significantly outperformed all other recommendation methods ($p < 0.01$) for all values k of top-k. In addition, we found that both hybrid recommendation methods, weighted-RCF and Equal-PR, outperformed RCF in the top-10 and top-25 recommendations in RRecall and RPrecision ($p < 0.05$). No significant difference was found between any of the other groups. Figure 8 presents all of the results of the offline evaluation. We note that it is a common feature

[8] In all top-k recommendations except the top-50



(a) MPrecision

(b) MRecall

(c) RPrecision

(d) RRecall

Fig. 8: **Offline Evaluation.** The error bars are too small to visualize in the graphs.

in online dating sites that messaging and reply rates are relatively low (e.g. [48]), hence the rather low values of precision and recall measured in our study should not come as a surprise.

These results led us to believe that the individual weighting methodology can be effective in improving the prediction accuracy of both the message sending by the service user and the successful interactions. In addition, we concluded that the reply prediction model we use is more effective than collaborative filtering. Specifically, regarding the number of successful interactions, the RWS method, which combines both our reply prediction model and the individual weighting methodology, is signficantly superior to all other methods. Therefore in the online evaluation, described in the following section, we focus on evaluating the RWS method.

6.3 Online Experimental Setup

After finishing the offline evaluation, we contacted Doovdevan again and requested to perform an online experiment. We originally intended to compare RWS to many recommendation generation methods proposed in previous work. However, due to the constraints imposed on us by Doovdevan, we were limited to only two conditions. Hence, we compared RWS to the RCF method. We recall that in [48], RCF demonstrated its superiority over several recommendation methods, including “one-sided” recommendation methods that consider the preferences of only one of the parties.

Our online experiment involved a group of 398 active users randomly chosen from Doovdevan, ranging in age from 18 to 70 (mean = 34.9, s.d.= 12.9), of which 24% (n=97) were female. The male-female ratio was chosen to approximately match that of the full Doovdevan dating platform. We randomly divided the participants into two conditions. Both condition groups received recommendations. The dependent variable was the recommendation method: the first group received recommendations based on the RCF method, and the second received recommendations based on the proposed method, RWS.

All participants received the top three recommendations generated by the recommendation method of their respective condition. The recommendations were proposed one per day for three days.

6.3.1 The Optimized Weight of the RWS Condition

Before generating the recommendations, we calculated the optimized weight for all users in the RWS condition, as described in (Section 5.4). About 89% of the participants received individual optimized weights (the remaining participants had no successful interactions). The average optimized weight was 0.411 and the median weight was 0.349. This indicates that to most of the users in our proposed method condition, the method gave higher importance to the score that measures the probability that a recommended user will reply to a service user, while the RCF method gives equal importance to the preferences of the service user and the recommended user.

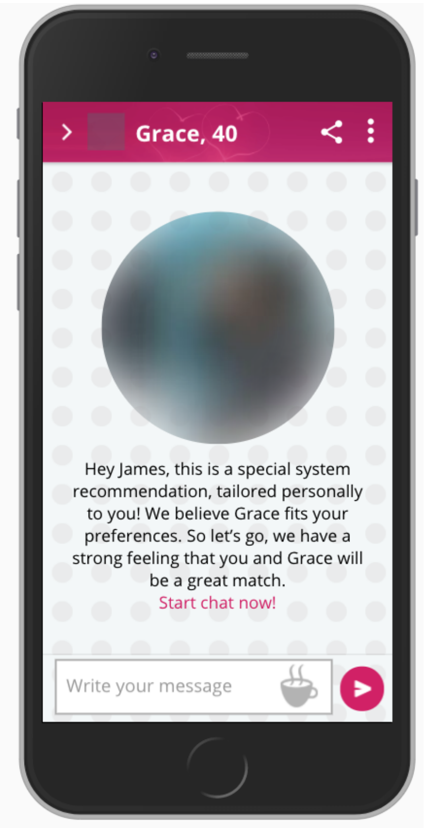
6.3.2 Online Experiment Evaluation Metrics

Due to the special nature of online evaluations and the users’ recommendation interface, the evaluation metrics we used are slightly different than the metrics we used for the offline evaluation (Section 6.1). We measured three important indicators of the success of a reciprocal recommender system:

1. The number of recommendations that were clicked on by the service user;
2. The number of the recommendations that the service user accepted where she initiated a chat with the recommended user; and
3. The number of messages sent by the service user to which the recommended user replied.

As our objective at this stage of our work was to increase the amount of successful interactions, our main focus is on the third indicator.

The recommendation interface in Doovdevan includes three stages: First, the system generates the recommendation and sends a message to the service user’s inbox, and she receives a notification on her smartphone. Then, if the user clicks on the recommendation, she moves to a new window with a higher resolution photo and a text justifying the system recommendation. Then, the user can decide to send a message or ignore the recommendation.



Grace, 40

Hey James, this is a special system recommendation, tailored personally to you! We believe Grace fits your preferences. So let's go, we have a strong feeling that you and Grace will be a great match.

Start chat now!

Write your message

Fig. 9: **Screen shot of a recommendation in the active online-dating platform.**

At this stage of the study, we focused only on the effect of the underlying *recommendation* method, and not on the *explanation* effect. Therefore we did not modify the user's interface in any way. The justification text which accommodated the recommendation was predefined by Doovdevan. This justification was *not* personalized, and did not refer to any specific preferences of the user. It simply stated that the recommendation was made based on the service user's personal characteristics and attempted to encourage the user to initiate a chat (see Figure 9).

For a given service user x, in addition to the sets of users defined above for the offline evaluation, we define the following four sets of users:

1. RO_x is the set of recommended users who were recommended to x and viewed by x in her inbox[9].

[9] Some users did not view all of the recommendations, either because they did not log-in during the week following the recommendations or because they did not view their inbox

Measure \ Condition	RCF	RWS
RO	320	356
RV	**174**	147
RM	**171**	138
M	889	1536
RI	1	**8**
I	99	184

Table 2: The summation of the results for all users in both conditions, evaluated a week after provision of the recommendations

2. RV_x is the set of recommended users whom x viewed in her inbox and then clicked on in order to browse for more detailed information.

We first measure the number of messages that were clicked on by the user after viewing the message in the inbox:

$$VPrecision_x = \frac{|RV_x|}{|RO_x|}$$

This measure evaluates the performance of the methods in recommending users who seem interesting enough to the service user so that she clicks on them in order to receive more information. The measure is only applicable for users who have viewed at least one recommendation in the inbox (some participants have logged in but did not view their inbox). In addition, we measured the accuracy of predicting the service user's messages similar to the measures of the offline evaluation:

$$MPrecision_x = \frac{|RM_x|}{|RV_x|} \quad MRecall_x = \frac{|RM_x|}{|M_x|}$$

$$RPrecision_x = \frac{|RI_x|}{|RM_x|} \quad RRecall_x = \frac{|RI_x|}{|I_x|}$$

6.4 Online Study Results

We examined the results a week after the provision of the recommendations. In Table 2 we show the summation of all of the results for all users in each condition.

We evaluated the performance of the recommendation methods by comparing the mean of the metrics defined above using a standard t-test. All of the results for both conditions were found to be distributed normally according to the Anderson-Darling normality test [40].

We first evaluated the average $VPrecision$. Our results show that the RCF method obtained significantly higher results (RCF: mean=0.57, s.d.= 0.42 vs. RWS: mean= 0.43, s.d.= 0.42 , $p < 0.05$), meaning that the recommendations

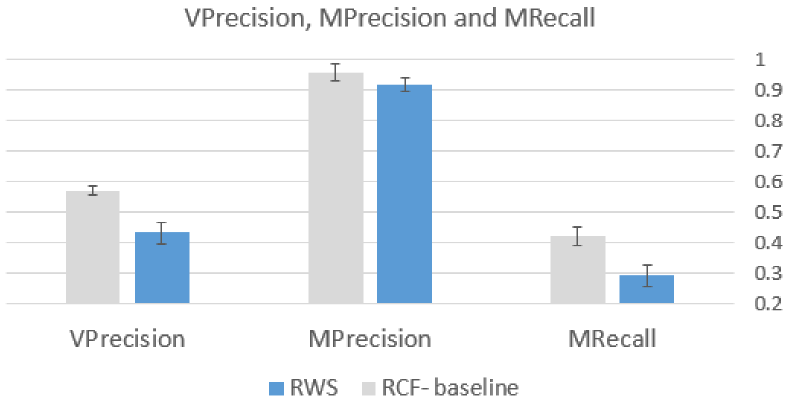


Fig. 10: VPrecision, MPrecision and MRecall Metrics. Error bars represent the standard error

that were provided by the RCF method looked more interesting to the users when they were scanned in their inbox.

Regarding the *MRecall* metric, we found that the RCF method significantly outperformed RWS (RCF: mean=0.42, s.d.= 0.38 v.s RWS: mean=0.29, s.d. =0.35, $p < 0.05$). The *MPrecision* metric of both conditions is similar, with no statistically significant difference (mean=0.96, s.d.= 0.28 vs. mean= 0.92 , s.d.= 0.23, $p < 0.05$). The mean and the standard error of *VRecall*, *MRecall* and *MPrecision* are presented in Figure 10.

These results indicate that the users recommended by RCF were evaluated as more appealing compared with those recommended by RWS. This finding was not surprising, as our method aims at optimizing successful interactions. In addition, as described above in Section 6.3.1, for most of the participants our proposed method gives relatively less importance to the service user's interest.

Considering the *RRecall* and *RPrecision*, which measure the effectiveness of the algorithms in providing recommendations that lead to a successful interaction, we found that RWS significantly outperforms RCF with respect to *RPrecision* (RCF: mean=0.01, s.d.= 0.05 vs. RWS: mean= 0.06, s.d.= 0.21, $p < 0.05$). Also, with respect to *RRecall*, RWS gave better results, but the difference is not significant (RCF: mean=0.02, s.d.= 0.14 vs. RWS: mean= 0.06, s.d.= 0.21, $p < 0.1$). Note that *RRecall* is less important for our evaluation, as our optimization is based on precision rather than recall. The mean and the standard error of *RRecall* and *RPrecision* are presented in Figure 11.

In addition, we found that the average weight (α) assigned to a participant who had a successful interaction following a recommendation was 0.194, while the average weight of all participants was 0.411, as mentioned above. This means that for these users, our method gave a substantially higher importance to the reply prediction model in comparison to the remaining participants.

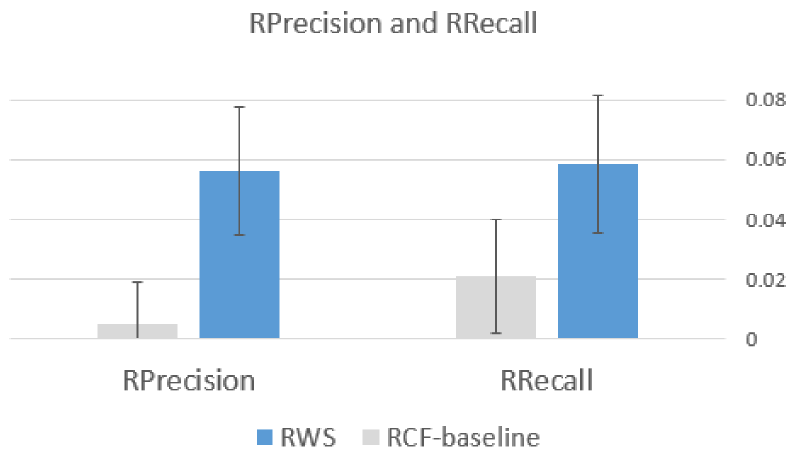


Fig. 11: RRecall and RPrecision Metrics. Error bars represent the standard error

6.4.1 Popularity of the recommended users

We have also analyzed the popularity of the users who were recommended by both methods. The popularity of the users is commonly estimated by the number of messages received during a specific time period [30]. We measured the total number of messages received in the thirty days before the recommendation provision. We found that the popularity of the active recommended users[10] in the RCF condition was significantly higher than the RWS condition (RCF: mean= 59.49, s.d.= 45.14 vs. mean= 32.72, s.d.= 35.06, $p < 0.01$). This result indicates that our method recommends less popular users. In fact, it is very important to be selective when recommending popular users, especially in online dating applications, where popular users are typically overwhelmed by incoming messages [33].

6.4.2 Runtime

We measured the average runtime for generating recommendations for a single user by both methods. The average runtime of the RCF method was 1.47 seconds, while in our proposed method the average runtime was 6.97 seconds, including an average of 2.61 seconds for the optimization calculation. However, in practice, the runtime has no impact on the user's experience since the recommendations in our application are delivered as messages and are not requested by the users.

[10] We only focus on the active recommended users, since non-active users receive fewer messages regardless of their popularity and attractiveness.

Conclusion of the RWS method evaluation

The main outcome of the previous experiment is that the RWS method significantly improves the number of successful interactions in comparison to the RCF method. However, as we expected, it also reduces the amount of accepted recommendations. An additional benefit of focusing the recommendation on the prediction of the chances for a reply is that it reduces the recommendation of popular users.

Due to limitations imposed by the site owner, we were only able to compare the RWS to a single baseline method. We plan to compare RWS to other methods in future work.

7 Combining Reciprocal Explanation and the Reciprocal Recommendation Algorithm

The decrease in acceptance of recommendations generated by the RWS algorithm was expected since, as discussed before, the RWS algorithm generally gives increased importance to the chance of a reply. However, even though our main objective was to increase the successful interactions, we were still unsatisfied by this decrease, since the acceptance rate (of the recommendation by the service user alone) is an important factor, also in RRSs [35].

In order to remedy this negative feature of the recommendation method we propose the use of reciprocal explanations, introduced in the first part of our work (Sections 3 and 4), to improve the effect of the RWS recommendation method. We hypothesized that accompanying the recommendation with a reciprocal explanation can overcome this issue. This hypothesis was based on the results from the experiments which focused on the explanation style alone (Section 4), which showed that users are more likely to send messages to users when they believe that the chances for a positive reply are high (even if the recommendation does not strictly fit their preferences).

Therefore, after completing the previous experiments we decided to investigate the integration of our reciprocal recommendation algorithm (RWS) with the reciprocal explanation method.

7.1 Experimental Setup

In order to evaluate the combination of RWS with reciprocal explanations, we collaborated once again with Doovdevan and conducted the following experiment. Recall that in the previous experiments (Section 6), we compared RWS with a baseline algorithm, and in all conditions the recommendations received by the participants were accompanied with a general text format, predefined by the system (as described above in Section 6.3.2). Hence, in order to accurately capture the effect of a reciprocal explanation with the RWS method, we decided to add two additional conditions: a condition of the RWS algorithm with one-sided explanations (presented in Section 3) and a condition of the RWS algorithm with reciprocal explanations.

The new experiment included a group of 488 active users who entered the system in the week prior to the experiment and did not participate in any of the previous experiments. The participants were divided randomly into two equal groups. The users ranged in age between 18 and 70 (mean= 44.15 , s.d.= 14.01) of which 27% were female (n=132). Similar to the previous experiment, all of the participants received three recommendations over the course of three days. The recommendations were generated by the RWS algorithm. The dependent variable was the explanation method: One condition received recommendations with reciprocal explanations and the other received recommendations with one-sided explanations. We will call the first condition REWS (Reciprocal Explanation Weighted Score) and the second condition OEWS (One-sided

Condition	Recommendation method	Explanation method	Description
Baseline	RCF	General	Section 2.2.1
RWS	RWS	General	Section 5
OEWS	RWS	One-sided	Section 7
REWS	RWS	Reciprocal	Section 7

Table 3: Summary of all conditions in the two last experiments.

Explanation Weighted Score). Table 3 summarizes the differences between all conditions of the experiments in the Doovdevan environment.

7.2 REWS vs.OEWS

We evaluated the results a week after the last recommendations were sent, as we did in our previous experiments. The summary of all of the results, of the new conditions and the previous conditions, are presented in Table 4.

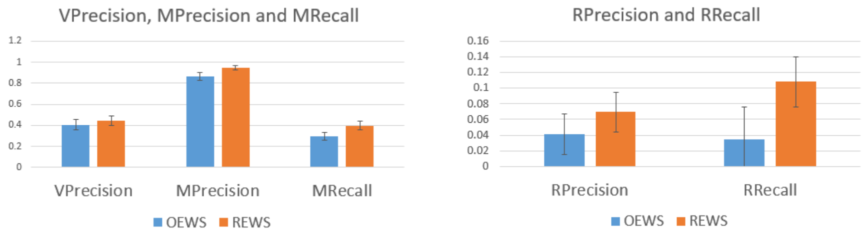
We first compared the results of the two new conditions. The dependent variable was the explanation method. We used the same evaluation metrics as defined above in Section 6 to compare the conditions. As in the previous experiments, we evaluated the performance of the recommendations by comparing the mean of the metrics (defined above in Section 6) using a t-test.

Regarding the $VPrecision$ metric, we did not expect a significant difference between the two new conditions, since the measure reflects the portion of recommendations which were clicked on by the user as viewed in the inbox. As mentioned above (Section 4.3.1), the recommendation in the inbox did not include an explanation (the user only receives an explanation if she clicks on the recommendation) and therefore the explanation style has no effect on this metric.

7.2.1 Results

As expected, no significant difference was found between the two new conditions in the $VPrecision$ metric (REWS: $mean = 0.44$, $s.d = 0.51$, OEWS: $mean = 0.41$, $s.d = 0.53$, $p = 0.26$).

Regarding the $MRecall$ and $MPrecision$ (which measure the amount of messages sent to recommended users who were clicked on by the user), we found that the REWS condition significantly outperformed the OEWS condition. This result aligns with the results from the evaluation of the reciprocal explanation described above in Section 4.3. In both experiments we find that when the recommendation method is independent (both for the RCF and RWS methods), the reciprocal explanation is significantly superior to one-sided explanation in increasing the number of accepted recommendations by the service user.



(a) VPrecision, MPrecision, MRecall

(b) RPrecison and RRecall

Fig. 12: **New Experiment Result**

	$VPrecision$	$MPrecision$	$MRecall$	$RPrecision$	$RRecall$
Baseline	0.57	0.96	0.42	0.01	0.02
RWS	0.43	0.92	0.29	0.06	0.06
OEWS	0.41	0.86	0.29	0.04	0.03
REWS	0.44	0.95	0.32	0.07	0.11

Table 4: Summary of the mean value of all measures in all conditions of the two last experiments.

Considering $RRecall$ and $RPrecision$, which evaluate the effectiveness of the algorithms in providing recommendations that lead to successful interactions, we found no significant differences. This result was expected since the recommendation methods in both conditions are the same. However, in the $RRecall$ measure the REWS condition ($mean = 0.11$, $s.d = 0.3$) outperformed the OEWS condition, with a marginally significant difference ($mean = 0.03$, $s.d = 0.11$, $p \leq 0.1$). Also regarding the $RPrecision$, the REWS condition performed better than the OEWS. However, the difference was not significant ($p = 0.22$). The results are presented in Figure 12.

Surprisingly, we observed that regarding the $MRecall$ and $MPrecision$ metrics, the OEWS condition was inferior to the RWS condition. Even though the difference was not significant ($p = 0.13$), it possibly indicates that the one-sided explanation deteriorates the performance of the general explanations used in the RWS condition. We discussed these results with the CEO of Doovdevan, an online-dating domain expert, and he suggested that the users in this environment did not like the one-sided explanations for two reasons: 1) The explanations include specific textual features while users often decide to initiate an interaction based on the picture alone; and 2) Some users do not like being told that the system believes a specific attribute, such as economic level, is their most important decision factor.

7.3 REWS vs. Baseline

We concluded from the the results that a *reciprocal* explanation yields better results than *one-sided* explanations for recommendations generated by the RWS algorithm. Yet, at this point, we wanted to assure that REWS, which combines our novel recommendation method and our novel explanation method, is superior to the baseline condition, which generated recommendations with the RCF algorithm with a general predefined explanation. Therefore, we further compared the REWS condition with the baseline condition.

Contrary to the comparison of Baseline and the RWS algorithm in the previous experiment (Section 6), we found that in both $MRecall$ and $MPrecision$, no significant difference was found between the conditions. Regarding the $RRecall$ and $RPrecision$ metrics, the REWS (similar to RWS) was significantly superior to the baseline condition (in $VRecall$: $p \leq 0.01$, in $VPrecision$: $p \leq 0.05$). Regarding the $VPrecision$ metric, the baseline condition significantly outperformed the REWS condition ($p \leq 0.4$). However, as mentioned above, this metric is not expected to be effected by the explanation style and therefore we were not surprised by this result.

In summary, we can conclude from the last experiment that REWS, which integrates reciprocal explanations and the RWS recommendation method, is significantly superior in increasing the number of successful interactions without decreasing the number of recommendations accepted by the service user with respect to the baseline condition.

8 Discussion and Future Work

The main outcomes of this work are threefold: First, we find that reciprocal explanations can increase the acceptance of recommendation. However, such explanations should be used cautiously, since under circumstances where the acceptance does not involve a significant cost, they could actually have a counteractive effect. Second, we find that a recommendation method which individually balances the importance of the interests for each user is successful in increasing the number of successful interactions, but on the other hand reduces the number of accepted recommendations by the service user alone. Lastly, we find that the combination of the novel recommendation method with reciprocal explanation yields better results than either of the methods does when used alone.

Our main contribution in the first part of this work is the introduction of reciprocal explanations and the evaluation of their effectiveness. We acknowledge that the explanation methods used in this study are relatively simple and we are currently working on more sophisticated methods. Specifically, in this work we used a generalized explanation method which did not differentiate between users' presumed cost of rejection. We intend to extend this research and build a fully-personalized user model, which will model the user's considerations in a RRS based on her historical interactions (calculated in a similar manner to the individual weight optimization we used in the RWS method) and provide reciprocal or one-sided explanations accordingly. We also plan to thoroughly investigate explanation methods which refer to the user's previous interactions, in contrast to the content-based explanations which we considered in this work.

It is important to note that since we focused on online-dating, the above results, regarding both explanation and recommendation methods, cannot be not directly generalized to other application domains which can benefit from RRSs, such as job recruitment or roommate matching. Specifically, RRSs can be used in domains were there is wide variety of the inherent emotional cost of reject a proposal which will presumably influence the effectiveness of reciprocal explanations. In addition, the optimal balance between the service user's and recommended user's preferences, which is calculated in the RWS method, can be heavily dependent on the application domain (meaning that the same user can be assigned to different weights according to his behavior in different domains). Furthermore, the users' preferences may change over time, requring the adaptation of the personalized balance in a continuous fashion. To that end, we intend to explore additional RRSs of varying characteristics in future work.

It is also important to note that our experiments were performed in a heterosexual online dating environment, and therefore the conclusions cannot be generalized automatically to non-heterosexual environments. We would like to specifically investigate non-heterosexual environments in the future.

An additional important limitation of reciprocal explanations is the issue of the recommended users' privacy. Specifically, one should balance between the need to provide meaningful explanations that also account for the recommended user's preferences, yet at the same time, avoid breaching their privacy.

We intend to investigate *coalitional* reciprocal environments, where a user seeks to form or join a group of partners with whom to form a coalition. For example, a system which recommends potential research collaborators for scholars. In these environments, users often have preferences for a group of partners and therefore the explanations should be adapted accordingly.

9 Appendix 1: User Experience Questionnaire for Evaluation of Reciprocal Explanations

Our questionnaire, which evaluated the effect of explanation on the user experience (Section 4), included 5 Likert-scale questions, with a scale ranging from 1 ("strongly disagree") to 5 ("strongly agree"). These questions measured five prominent factors of user experience in recommender systems: user *satisfaction* from the recommendations, perceived *competence* of the system, perceived *transparency* of the system, and *trust* in the system [9,38,26]. In addition, the users were asked specifically about the *explanation usefulness*, namely the extent to which the users considered the explanations to be helpful. The questions are presented in Table 5. The second question, which is 'negatively worded', was reverse-scored [19]. In order to make sure that the different questions actually evaluate different measures, we calculated the cross-scale Pearson correlation coefficients [14] which show that the answers to the questions are not strongly correlated. The full correlation table is presented in Table 6.

Measure	Statement
Satisfaction	1) I like the profiles the system recommended to me.
System competence	2) The system is useless for me.
Trust	3) I trust the system to recommend all profiles that are of interest to me.
Transparency	4) I understand why the system recommended the profiles it did.
Explanation Usefulness	5) The explanations that were provided along with the recommendations were good.

Table 5: User Experience Questionnaire

	Question 2	Question 3	Question 4	Question 5
Question 1	0.427	0.448	0.426	0.291
Question 2		0.44	0.271	0.283
Question 3			0.449	0.292
Question 4				0.44

Table 6: **Cross-scale Pearson Correlation Coefficients**

9.1 Questionnaire Results in the Simulated Environment

In this section we present the results of the questionnaire, which evaluated the user-experience in the simulated environment, for both negligible cost and explicit cost settings.

9.1.1 Negligible Cost

In the negligible cost setting, the one-sided condition outperformed the reciprocal condition also in the user experience, as in the *relevance* measure (Section 4.2.1). Specifically, the satisfaction (mean= 4 s.d.= 0.85 vs. mean= 3.57, s.d.=0.86 , $p \leq 0.05$) and perceived competence (mean= 4.13 s.d.= 0.83 vs. mean=3.27, s.d.=0.9 , $p \leq 0.01$) were found to be significantly superior for the one-sided explanation condition. No statistically significant difference was found between the conditions for the remaining measures. The results are presented in Figure 13.

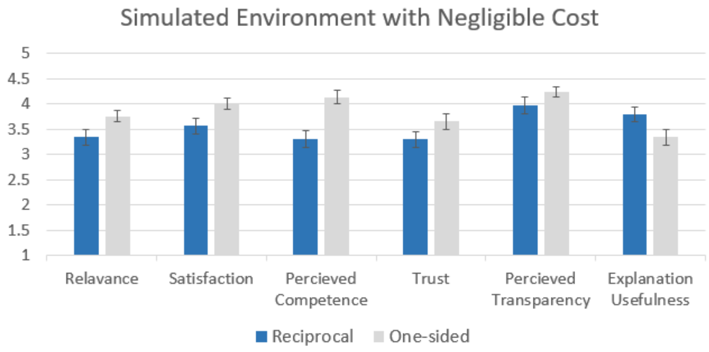


Fig. 13: Reciprocal vs. one-sided explanations in MM with negligible cost. Error bars represent the standard error.

9.1.2 Explicit Cost

In the explicit cost setting, in addition to the acceptance (Section 4.2.2), the participants' trust in the system was found to be higher under the reciprocal explanation condition (one-sided: mean=2.93 s.d.=1.14 vs. reciprocal: mean=3.38 s.d.=1.01 , $p \leq 0.05$). No statistically significant difference was found between the conditions for the remaining measures.

The results are presented in Figure 14.

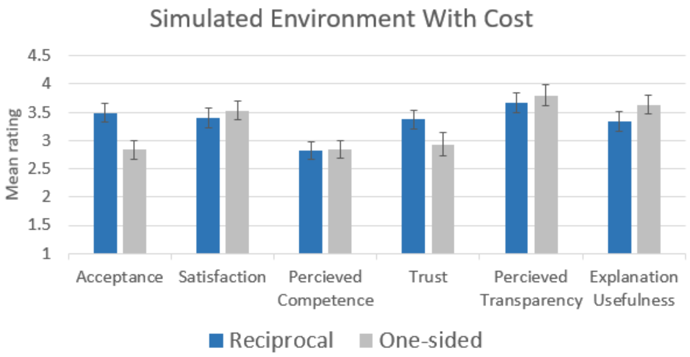


Fig. 14: Reciprocal vs. one-sided explanations in MM with explicit cost. Error bars represent the standard error.

10 Appendix 2: Choosing the Explanation Method

Before the evaluation of one-sided and reciprocal explanations in RRSs, we performed a preliminary investigation in order to find the best suited explanation method for online-dating, the domain on which we focus throughout this paper.

10.1 Comparison of Correaltion-based and Transparent Explanation Methods

In addition to the correlation-based explanation method, which is described in Section 3, we designed a similar explanation method based on the same guidelines (described above in Section 2.1). We called this explanation method the "transparent" explanation method.

The transparent explanation method, which aims to reflect the actual reasoning for the recommendations provided by the RECON algorithm, works as follows: to explain to user x a recommendation of user y, the method returns the top-k attributes of y which are the most prominent among users who received a message from user x.

Algorithm 5 Transparent Explanation Method

```
Input: two users $x$ and $y$, number of attributes for explanation $k$.
temp ← $\emptyset$
obtain $P_x$ from user $x$
obtain $A_y$ from user $y$
for all attributes a $\in A$ do
    obtain the value $a_v$ of attribute a in $A_y$
    obtain $P_{x,a}$ from $P_x$.
    find $(z, n) \in P_{x,a}$ s.t. $z = a_v$
    $temp = temp \cup (a_v, n)$
sort temp by the values $n$
$e_{x,y}$ = top-$k$ attribute values of temp
return $e_{x,y}$
```

To illustrate the difference between the transparent and the correlation-based explanation methods, we revisit Example 1. Assume an RRS has decided to recommend Alice, who never smokes and is slim, to Bob. Recall that Bob sent 6 messages to users who never smoke and 4 to slim users. For $k = 1$, the transparent explanation method would provide “never smoke” as an explanation because Bob sent more messages to users who never smoke than to users who are slim. Now say Bob viewed a total of 25 users, of whom 18 never smoke and 4 were slim. In other words, Bob sent messages to only a third of the users he viewed who never smoke, and to all users he viewed who are slim. Thus, the correlation-based method would find a stronger correlation between the presence of “slim body” and Bob’s messaging behavior, hence “slim body” would be provided as an explanation.

In order to compare the transparent and the correlation-based explanation methods, we used the MM simulated system discussed in Section 4.1. We asked 59 of the 118 participants who took part in the data collection phase and did not take part in the negligible-cost experiment (Section 4) to enter the MM platform, where each participant then received a list of five personal recommendations generated by the RECON algorithm along with either transparent explanations (30 participants) or correlation-based explanations (29 participants). Participants were randomly assigned to one of the two conditions. As in the negligible-cost experiment, participants were asked to rate the *relevance* of each recommendation separately, on a five-point Likert scale from 1 (extremely irrelevant) to 5 (extremely relevant). Next, participants answered the questionnaire (available in Appendix 1), debriefing them on their user experience.

Results

All collected data was found to be approximately normally distributed according to the Anderson-Darling normality test [40]. All reported results were compared using an unpaired t-test. The results from the questionnaire showed that participants in the correlation-based condition were more satisfied than those in the transparent explanation condition and perceived the system as more useful ($p \leq 0.05$). We did not find a significant difference in the way participants rated the relevance of the provided recommendations nor did we find a significant difference in the reported trust of the system.

10.2 Additional Explanation Methods Evaluated in the MM Environment

10.2.1 Comparison to Baseline

Prior to our main experiment, we first compared the correlation-based explanations with a baseline condition: recommendations without any explanation. We recruited an additional group of 30 participants who were asked to enter the MM environment. We used the same experimental methodology described in Section 4. We measured all evaluation measures with the exception of the *explanation usefulness* (which was not relevant to the baseline condition). We found that the correlation-based condition significantly outperformed the baseline condition in the *relevance* measure ($p \leq 0.05$).

10.2.2 Comparison to Collaborative Filtering Explanation Style

We further examined another explanation method, similar to a method which was presented in previous work [20]. This explanation method justifies the recommendation by simply stating that “similar users” to the service user have shown interest in the recommended match. We call this explanation style "collaborative filtering", because the explanation indicates that the recommendation was generated using collaborative filtering methodology, where recommendations are based on similarity measures. Unlike the previous methods, these explanations do not include any information about the attributes of the recommended users. Of course, this explanation does not reflect the actual reasoning for the recommendation, since the underlying algorithm is content-based. Nevertheless, previous work has shown that explanations which are not related to the underlying algorithm can also be highly effective [20].

For the evaluation of this explanation method, we recruited an additional group of 25 subjects. All of the experimental setup was identical to the setup in the previous experiment, with the only difference being the explanation method.

Our results show that the correlation-based recommendation method was significantly superior to the collaborative filtering explanation style. Specifically, the relevance rate in the correlation-based condition was significantly higher than the collaborative filtering (correlation: mean=3.34 vs. collaborative filtering: mean=2.36, $p \leq 0.01$). In addition, the subjects in the experiment with the correlation-based method were significantly more satisfied than the users in the experiment with the collaborative filtering method (correlation: mean=4 vs. collaborative filtering: mean=3.28, $p \leq 0.01$).

11 Appendix 3 : Features in the Reply Prediction Model

Public Profile Features of Sender and Receiver:

These features are part of the users' profile and are public to all the users in the environment.

1. Age
2. Gender
3. Marital status
4. Number of children
5. Height
6. Smoking habits
7. Number of pictures in profile
8. Are pictures public? (The users on the site had an option of keeping their pictures private)
9. Religious observance level
10. Dating goal
11. Living area
12. Self-description length (Number of characters in the user's self description)
13. Preferences description length (Number of characters in the user's description of his/her preferences)
14. Economic status
15. Ethnic background

Each feature corresponds to two features in the model - one for the sender and one for the receiver.

Interaction and Activity features of Sender:

1. Number of profiles he/she viewed.
2. Number of users who viewed him/her.
3. Number of users he/she liked.
4. Number of users who liked him/her.
5. Number of messages he/she sent.
6. Number of messages he/she received.
7. Number of his/her messages which were positively replied to before current message.
8. Percent of positively replied messages before current message.
9. Number of received messages which he/she did not view.
10. Number of users who viewed him/her which he/she did not view.
11. Number of users who liked him/her which he/she did not view.

Interaction and Activity features of Recipient:

1. Number of users who viewed him/her.
2. Number of users he/she viewed.
3. Number of users who liked him/her.
4. Number of users he/she liked.
5. Percent of messages he/she replied to positively from all received messages.
6. Number of messages he/she received.
7. Did he/she send a message to the sender before?
8. Did he/she like the sender before?
9. Has he/she replied positively to any message before?
10. Was he logged-in while the message was received?
11. Log-ins to the environment in the week before the message.
12. Average duration of logins in previous week (before the message).
13. Number of sent messages in previous week (before the message).

Acknowledgements

This article extends the preceding papers: [25] and [24], with the following additions:

1. In Section 6.1 we present an extensive offline evaluation, based on data from 7668 users, of our novel recommendation method and additional variations of RWS. The evaluation results demonstrate the efficiency of RWS and justify our decision to use RWS in the online evaluation.
2. In Section 7 we describe an additional large-scale live experiment, including 488 participants, in which we investigate the integration of both our novel recommendation generation method and explanation method. This experiment strengthens our conclusions from previous experiments and provides credibility for the use of both methods together.
3. In Appendix 2 (Section 10) we present the full process which led us to use the correlation-based explanation method for the evaluation of the reciprocal explanation. This process includes a sequence of the experiments, involving 114 participants, in which we compared a few explanation methods and found that the correlation-based method was superior to all other methods investigated.

References

1. Hervé Abdi and Lynne J Williams. Tukeys honestly significant difference (hsd) test. *Encyclopedia of Research Design. Thousand Oaks, CA: Sage*, pages 1–5, 2010.
2. Fabian Abel, András Benczúr, Daniel Kohlsdorf, Martha Larson, and Róbert Pálovics. Recsys challenge 2016: Job recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 425–426. ACM, 2016.
3. Joshua Akehurst, Irena Koprinska, Kalina Yacef, Luiz Pizzato, Judy Kay, and Tomasz Rej. Ccra content-collaborative reciprocal recommender for online dating. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
4. Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.
5. Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.
6. Richard P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971.
7. Lukas Brozovsky and Vaclav Petricek. Recommender system for online dating service. *arXiv preprint cs/0703042*, 2007.
8. Robin Burke, Himan Abdollahpouri, Edward C. Malthouse, K. P. Thai, and Yongfeng Zhang. Recommendation in multistakeholder environments. In Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, pages 566–567. ACM, 2019.
9. Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18(5):455–496, 2008.
10. Wilerid J Dixon and J Massey Frank. *Introduction To Statistical Analsis.* McGraw-Hill Book Company, Inc; New York, 1950.
11. Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. How should I explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.
12. Ellen R Girden. *ANOVA: Repeated measures.* Number 84. Sage, 1992.
13. Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *Workshop on Human Interpretability in Machine Learning at the International Conference on Machine Learning*, 2016.
14. Robert Goodman. Psychometric properties of the strengths and difficulties questionnaire. *Journal of the American Academy of Child & Adolescent Psychiatry*, 40(11):1337–1345, 2001.
15. David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2017.
16. Ido Guy, Inbal Ronen, and Eric Wilcox. Do you know?: recommending people to invite into your social network. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 77–86. ACM, 2009.
17. Mark A Hall. Feature selection for discrete and numeric class machine learning. 1999.
18. Mark Andrew Hall. *Correlation-based feature selection for machine learning.* PhD thesis, University of Waikato Hamilton, 1999.
19. James Hartley. Some thoughts on likert-type scales. *International Journal of Clinical and Health Psychology*, 14(1):83–86, 2014.
20. Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.
21. Gunter J Hitsch, Ali Hortaçsu, and Dan Ariely. Matching and sorting in online dating. *American Economic Review*, 100(1):130–63, 2010.
22. Günter J Hitsch, Ali Hortaçsu, and Dan Ariely. What makes you click?-mate preferences in online dating. *Quantitative marketing and Economics*, 8(4):393–427, 2010.

23. Wenxing Hong, Siting Zheng, Huan Wang, and Jianchao Shi. A job recommender system based on user clustering. *JCP*, 8(8):1960–1967, 2013.
24. Akiva Kleinerman, Ariel Rosenfeld, and Sarit Kraus. Providing explanations for recommendations in reciprocal environments. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018.
25. Akiva Kleinerman, Ariel Rosenfeld, Francesco Ricci, and Sarit Kraus. Optimally balancing receiver and recommended users' importance in reciprocal recommender systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 2018.
26. Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504, 2012.
27. Daphne Koller and Mehran Sahami. Toward optimal feature selection. Technical report, Stanford InfoLab, 1996.
28. Sherrie YX Komiak and Izak Benbasat. The effects of personalization and familiarity on trust and adoption of recommendation agents. *MIS quarterly*, pages 941–960, 2006.
29. Alfred Krzywicki, Wayne Wobcke, Xiongcai Cai, Ashesh Mahidadia, Michael Bain, Paul Compton, and Yang Sok Kim. Interaction-based collaborative filtering methods for recommendation in online dating. In *International Conference on Web Information Systems Engineering*, pages 342–356. Springer, 2010.
30. Alfred Krzywicki, Wayne Wobcke, Yang Sok Kim, Xiongcai Cai, Michael Bain, Ashesh Mahidadia, and Paul Compton. Collaborative filtering for people-to-people recommendation in online dating: Data analysis and user trial. *International Journal of Human-Computer Studies*, 76:50–66, 2015.
31. Sean M McNee, John Riedl, and Joseph A Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM, 2006.
32. National Science and Technology Council. The National Artificial Intelligence Research And Development Strategic Plan. 2016.
33. OkCupid. Okcupid blog: a women advantage. *https://theblog.okcupid.com/a-womans-advantage-82d5074dde2d*, 2015. Accessed: 2018-04-25.
34. Gözde Özcan and Sule Gündüz Ögüdücü. Applying different classification techniques in reciprocal job recommender system for considering job candidate preferences. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 235–240. IEEE, 2016.
35. Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. RECON: a reciprocal recommender for online dating. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 207–214. ACM, 2010.
36. Pearl Pu and Li Chen. Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100. ACM, 2006.
37. Pearl Pu and Li Chen. Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556, 2007.
38. Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164. ACM, 2011.
39. Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Comput. Surv.*, 51(4):66:1–66:36, 2018.
40. Nornadiah Mohd Razali, Yap Bee Wah, et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.
41. Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.
42. Amit Sharma and Dan Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1133–1144. ACM, 2013.
43. Rashmi Sinha and Kirsten Swearingen. The role of transparency in recommender systems. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 830–831. ACM, 2002.