Why should I *not* follow you? Reasons For and Reasons Against in Responsible Recommender Systems

Gustavo P. Polleti
Universidade de São Paulo
gustavo.polleti@usp.br

Douglas L. de Souza
Universidade de São Paulo
douglas.luan.souza@usp.br

Fabio G. Cozman
Universidade de São Paulo
fgcozman@usp.br

ABSTRACT

A few Recommender Systems (RS) resort to explanations so as to enhance trust in recommendations. However, current techniques for explanation generation tend to strongly uphold the recommended products instead of presenting both reasons for and reasons against them. We argue that an RS can better enhance overall trust and transparency by frankly displaying both kinds of reasons to users. We have developed such an RS by exploiting knowledge graphs and by applying Snedegar's theory of practical reasoning. We show that our implemented RS has excellent performance and we report on an experiment with human subjects that shows the value of presenting both reasons for and against, with significant improvements in trust, engagement, and persuasion.

1 INTRODUCTION

Human subjects find it hard to make a decision when a very large number of options is available; a Recommender System (RS) provides valuable help by selecting a small set of options that are then evaluated by the user [20]. However, even if an RS produces sensible recommendations, users may reject them if their rationale is not understood [22]. It is thus clearly desirable to have RSs that offer sensible, transparent and trustworthy recommendations; one strategy that seems particularly promising is for the RS to generate explanations that clarify the recommendations [25].

Explanations presumably enhance transparency and trust. However, explanation generation techniques now in use in RSs focus solely on advocacy for the recommended options. By describing only the benefits of those options, they may fail to offer a balanced perspective to the user, ultimately squandering overall trust. A user may be at first happy to get some positive clarification about recommended products, but if she never sees information about possible downsides, she will ultimately lose interest in the recommendations.

We argue that an RS should provide *responsible* explanations in the sense that both reasons *for* and reasons *against* explicitly escort recommendations. We take Snedegar's theory of reasons for/against [23], a philosophical theory of practical reasoning, and realize it in the context of RSs. To do so, we start with existing

procedures that generate reasons *for* by analyzing paths in knowledge graphs [1, 15, 18]. We then modify such procedures so as to detect paths (or their absence) that count as reasons *against*. Snedegar's theory relies on five schemes of reasons against; we examine their computational implementation, identifying the most promising strategies. We also describe an RS we have implemented and its practical operation with reasons for/against. Additionally, we have carried out experiments with human subjects that show our approach to responsible recommendations to yield higher overall trust in generated explanations.

The paper is organized as follows. Section 2 presents some basic notions on recommender systems, explainability, transparency, and trust. In Section 3 we propose strategies to generate reasons for/against. We then present our empirical results, and offer concluding remarks in the last section.

2 A BIT OF BACKGROUND: RECOMMENDATIONS, TRUST, INTERPRETABILITY, EXPLANATIONS

An RS has a set of users and a set of items, usually producing a score $r(u, i)$ that captures the affinity between user u and item i [20]. An RS often relies on the score to rank a number N of items to be presented to the user. The definition of affinity varies wildly, depending on the application domain [8, 21]. The current state of the art is to learn the affinity between users and items from past experience using latent variable models, often dependent on matrix factorization and embedding techniques [5–7]. These techniques map items to a (numeric) latent space where similar items appear near to each other, usually by optimizing distances between related objects as they are mapped [16].

Opaque models, such as the ones produced by embeddings, create obstacles to the interpretability of recommendations [3]. Here we take interpretability as the degree to which a human can understand the cause of a decision [13]. A device may be transparent in that the user can access all elements of its operation, yet its output may have low interpretability. When interpretability is low, one possible strategy is to generate explanations for the decisions. There are several techniques for explanation generation [14]; for instance, some of them investigate the sensitivity of outputs to inputs or to elements of a model — the explanation is an indication of which parts of input/model affect the output. Other techniques aim at more elaborate explanations. Some of them are dependent on a particular model; for instance, some techniques focus on neural networks, producing explanations that involve particular neurons and layers. Other techniques for explanation generation are model agnostic; that is, they only look at inputs and outputs of the model to be explained. We focus on model-agnostic explanations in this work.

It is commonly stated that performance and interpretability are opposing goals [19]; for instance, an accurate classifier is a complex and hard to interpret one. However, matters are more delicate in the context of RSs, as performance itself depends on trust [17], and high interpretability is bound to increase trust (when interpretation fails, existing RSs may fail in surprising ways [4]). Previous efforts have explored various ways to obtain high performance and high interpretability [10, 11, 28], in some cases generating explanations that support recommendations [1, 15, 24].

3 EXPLANATIONS WITH REASONS FOR AND REASONS AGAINST

Recent RSs that rely on explanations do offer useful information to the user; however, we argue that they run into a difficult balancing act [12]. This is not unlike the salesperson who always proposes products with complimentary words, as opposed to the salesperson who frankly discusses the advantages and disadvantages of products. A perceptive customer will gradually favor a salesperson who chooses sincerity over persuasion — exactly the behavior we propose for responsible RSs.

The solution, then, is to build RSs that state reasons *for* recommended items together with reasons *against* the same items. This is the main idea in this paper; to make it concrete, we first discuss techniques that generate reasons *for* (Section 3.1) and then we propose novel ideas on the generation of reasons *against* (Section 3.2).

3.1 Reasons For: What They Are, and How to Generate Them

Reasons for a given recommendation can be produced using an auxiliary knowledge graph (KG), a strategy that has been explored in previous efforts [1, 15, 18].

The idea is to use a KG containing all entities handled by the RS so as to find connections between users and items. A knowledge graph (KG) consists of a set of *entities* $\mathcal{E} = \{e_1, \ldots, e_{N_e}\}$ and a set of binary *relations* $\mathcal{R} = \{r_1, \ldots, r_{N_r}\}$. Using RDF notation [26], an edge in the graph can be interpreted as a triple $\langle h, r, t \rangle$ where h, r and t are, respectively, the *subject (head)*, *predicate (relation)* and *object (tail)*. The existence of a triple $x_{h,r,t} = \langle h, r, t \rangle$ is indicated by a random variable $y_{h,r,t}$ with values in $\{0, 1\}$. A *path type* π is a sequence of relations $r_1 - r_2 - \ldots - r_l$, some of which may be the inverses of relations in $\mathcal{R}$ (the inverse of relation r is denoted by r^-). A given path π *holds* for entities h and t if there exists a set of entities $e_1, e_2, \ldots$ so that all the variables $\{y_{h,r_1,e_1}, y_{e_1,r_2,e_2}, \ldots y_{e_{l-1},r_l,t}\}$ have value 1. We assume a set Π of permissible path types is specified (by the RS designers) so that those path types capture sensible connections between entities [18].

Suppose an RS suggests item e_i to user e_u (note that items and users are represented by entities in the assumed auxiliary KG). A reason *for* this recommendation is simply taken to be a path $\pi \in \Pi$ that takes e_i to e_u in the KG. Thus we have an function f that starts with the KG and the path π, takes inputs e_i and e_u, and returns a set of reasons for the recommendation of e_i to e_u. While this function can be implemented in several ways, in our implementation (described later) we employed depth-first search in the KG [18].
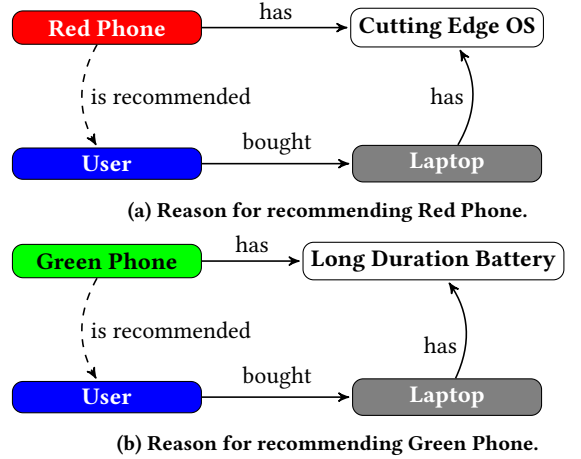

(a) Reason for recommending Red Phone.

(b) Reason for recommending Green Phone.

Figure 1: Examples of reasons for and reasons against in item-based recommendation.

To illustrate, Figure 1a shows through graphs an example where the recommendation of the Red Phone to a user is explained by the path $\pi_3 = (\text{bought}, \text{has}, \text{has}^-)$, which goes through entities User, Laptop, Cutting Edge OS and Red Phone.

3.2 Reasons Against: What They Are, and How to Generate Them

We now focus on the main technical challenge in this work: how to generate reasons *against* a particular recommendation. To do so, we resort to the literature on practical reasoning in Philosophy, where we find Snedegar's rather comprehensive theory of reasoning [23]. Snedegar presents five schemes by which reasons *against* can be generated by an agent contemplating competitive options:

Scheme 1 (S1) : a reason against an item A is a reason for a competing option;

Scheme 2 (S2) : a reason against an item A is only a reason for NOT A (not for any particular other option);

Scheme 3 (S3) : a reason against an item A is just a reason for the disjunction of the other options (say $B \lor C \lor D$);

Scheme 4 (S4) : a reason against an item A is a reason for each, i.e. all, of the alternatives to it.

Scheme 5 (S5) : a reason against an item A explains (or is part of the explanation as to) why A promotes or respects some objective less well than some other option.[1]

These schemes have been defined by Snedegar at a highly abstract level; we must taken them to a concrete level. We present our implementations in the remainder of this section.

Our implementation of S1 generates a reason against a given item by generating reasons for other options. For instance, take the case where the RS has recommended two phones — Red and Green — as in Figure 1. A reason against the Red Phone then would be that the Green Phone has a "Long Duration Battery".

Scheme S2 is more delicate: how to define the negation of an item in the context of recommendations? The vague nature of this question led us to skip this scheme.

---

[1]This scheme requires one to specify a quantitative objective.

Our implementation of S3 goes through all competing options, collecting reasons for them that are not reasons for the option of interest; we then trim the list of reasons against to an arbitrary small number of reasons (e.g. 3). In our running example we can imagine there is a Blue Phone and as reasons against the Red Phone we have that the Green Phone, the Blue Phone or both of them have long duration batteries. In practice S1 and S3 produce identical reasons against.

The implementation of S4 is similar to that of S3 to the extent that S4 takes reasons for all competing options into account (reasons against according to S4 are also reasons against according to S3). An example of reason against the Red Phone using S4 would be that both the Blue Phone and the Green Phone from the example above have adequate battery duration. The stringent nature of this scheme, where the intersection of reasons is required, makes it hard to generate reasons against in practical circumstances.

Scheme S5 depends on a quantitative objective that can be the basis of explanations; this objective is used to determine whether a reason is for or against an option. Consider in our phone example that the user has the objective of long battery life for her phone; with that piece of information, the RS can present the user with the reason against buying the Red Phone because it has a short duration battery. We have implemented S5 by assuming that an objective function is known; however, this is not a realistic assumption and future work should address the elicitation of objectives at running time.

To illustrate the implemented algorithm, suppose an RS recommended N items in an ordered set $\mathcal{I} : \{i_1, i_2, ...i_N\}$ to user u. In Schema S1 (and S3) we define as reason against an item i_r the union of reasons for each of its alternatives $\mathcal{I} \setminus \{i_r\}$ that are not reasons for i_r itself. Hence we must iterate over the alternatives, extracting reasons for each one of them $\Phi \leftarrow \Phi \cup \Phi_{u,i} \forall i \in \mathcal{I} \setminus \{i_r\}$. Note that at this point we assume that function f, as described in Section 3.1, is available. We then remove from Φ the reasons for our recommendation of interest, if any. The remaining reasons $\Omega = \Phi/\Phi_{u,i_r}$ are the reasons against i_r – as presented in the Algorithm 1.

Regarding the implementation of Schema 4 (S4), we follow a very similar procedure, except that instead of considering the union of reasons for its alternatives, we take the intersection. That is, we just replace the line 15 of the Algorithm 1 so as to take the intersection of sets $\Phi \leftarrow \Phi \cap \Phi_{u,i} \forall i \in \mathcal{I} \setminus \{i_r\}$.

To close this section, consider an extended example using Scheme S1. We focus on Scheme 1 due to the fact that it captures most of the content of Scheme S3 as well; as noted already, Scheme 2 does not seem conducive to a concrete implementation, and Scheme 5 requires elicitation of user objectives — finally, as discussed later in connection with our experiments, Scheme 4 does not seem very promising in practice.

Example 3.1. We have built an RS to suggest University classes called Ganimedes. A student asks for courses by presenting a few topics to Ganimedes; the RS then uses information from syllabuses and an associated knowledge graph to produce recommendations. The knowledge graph, called USPedia, collects information about topics and their relationships; it was automatically harvested from Wikipedia pages [18]. We have defined a number of permissible paths for explanations (Section 3.1). For instance, one of them is

---

Algorithm 1 Explanation Generation using Scheme S1

1: **procedure** REASONS-FOR($i, u, \Pi, \mathcal{G}$)
2:     $\Phi_{u,i} = \{\}$ ▷ Set of reasons for i
3:     **for all** $\pi \in \Pi$ **do**
4:         $\phi \leftarrow f(u, i, \pi | \mathcal{G})$ ▷ Function described in Section 3.1
5:         $\Phi_{u,i} \leftarrow \Phi_{u,i} \cup \phi$
6:     **end for**
7:     **return** $\Phi_{u,i}$
8: **end procedure**
9: **procedure** REASONS-AGAINST-S1($i_r, u, \mathcal{I}, \Pi, \mathcal{G}$)
10:     $\Omega_{u,i_r} \leftarrow \{\}$ ▷ Set of reasons against i_r
11:     $\Phi = \{\}$
12:     $\Phi_{u,i_r} \leftarrow$ REASONS-FOR($i_r, u, \Pi, \mathcal{G}$)   ▷ Set of reasons for i_r
13:     **for** $i \in \mathcal{I} \setminus \{i_r\}$ **do**   ▷ Iterate over i_r alternatives
14:         $\Phi_{u,i} \leftarrow$ REASONS-FOR($i, u, \Pi, \mathcal{G}$)
15:         $\Phi \leftarrow \Phi \cup \Phi_{u,i}$
16:     **end for**
17:     $\Omega_{u,i_r} \leftarrow \Phi \setminus \Phi_{u,i_r}$
18:     **return** Ω_{u,i_r}
19: **end procedure**

(subject, broader$^-$, broader); as this permissible path indicates that a subject is of the same broader category as another topic of interest. That is, subject(X, Y) broader(Z, Y) broader(Z, W) means that Y is a topic of X, Z has the same broader categories of Y and W and, finally, that W is of the same broader category of a topic of X.

We assume that a course is likely to be about a given subject when it deals with topics that are related to that subject. For instance, a student who is a machine learning (ML) enthusiast would be satisfied with a course that is about statistical models even if the course is not focused on ML itself.

Figure 2 conveys a number of explanations generated by our RS. In this case, the student asked our RS for courses about *Stochastic Resonance*, and was suggested classes with codes PME3430 and PME3479. The RS found two reasons for PME3430 (Fig. 2a and Fig. 2b) and one for PME3479 (Fig. 2c). Note that both recommendations share the *reason for* depicted in Fig. 2b and 2c, thus it cannot be a reason against for none of them. On the other hand, the one in Fig. 2a is a reason for only PME3430; therefore, it is a reason against PME3479. □

4 EXPERIMENTS

In this section we describe experiments with simulated and real users; we first examine the feasibility of our techniques in Section 4.1 and then we discuss the reaction of human users to our approach in Section 4.2.

4.1 Evaluation of feasibility: simulated interactions

We have first evaluated our proposal from two perspectives: (1) the fraction of recommendations for which we can find at least one explanation (we refer to it as *coverage*) and (2) the average number of reasons we can find to support/attack a given recommendation (we refer to it as *support*) [18, 27]. These metrics offer a glimpse at the workings of our proposal in a real-world scenario from a
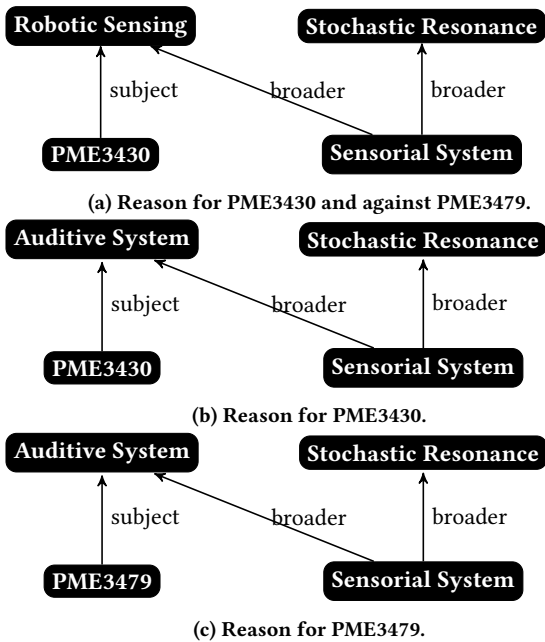
(a) Reason for PME3430 and against PME3479.

(b) Reason for PME3430.

(c) Reason for PME3479.

Figure 2: Examples of S1 with two reasons for and against the recommendation PME3430.

objective perspective. To carry out our experiments, we trained an RS based on TransE [2] embedding from the USPedia knowledge graph employed in Example 3.1, using the same set-up as in Ref. [18]. We built our simulated interactions by asking for the Top-4 recommendations of randomly sampled 100 cases. Next, for each interaction, we used our proposed method to retrieve both reasons for and against.

Regarding *reasons for*, Table 1 shows that we obtained 79.33% coverage and a support mean of 2.0, similar results to those reported in previous works [18, 27]. As for *reasons against*, we ran our experiments considering Schemas S1 and S4. Both the coverage (85.1%) and support (2.3) obtained for S1 are higher than those from *reasons for*. This result was expected since S1 implementation considers more aggregated reasons for alternatives than it removes from the recommendation being explained.

On the other hand, Scheme S4 could *not* generate a single reason against at all (coverage 0%!). As Scheme S4 requires that a reason against an option must be a reason for all of its alternatives, it imposes a restriction so rigorous that it is in fact unfeasible in practice.

Explanation Type	Coverage	Support
Reason For	79.3%	2.0 ± 1.0
Reason Against (S1)	85.1%	2.3 ± 1.4
Reason Against (S4)	0%	-

Table 1: Coverage and Support for reasons for and reasons against using Schemas S1 and S4.

Metric	Question
transparency	The explanation on the right helped me understand why the items were recommended better than the explanation on the left
persuasion	Based on the explanation on the right, I was more prone to follow the recommendation than based on the explanation on the left
engagement	The explanation on the right helped me learn more about the recommended items than the explanation on the left
trust	The explanation on the right contributed more to increase my confidence in the recommendations than the explanation on the left
effectiveness	The explanation on the right made me more confidence about making the best choice than the explanation on the left

Table 2: The five explanation metrics that subjects had to take into account in the experiment.

4.2 Evaluation with Human Subjects

One could expect the fact that an RS can be built with reasons for/against does not mean that human subjects would be satisfied with it; to determine whether indeed our approach is a valuable one, we carried out an experiment to address the following questions:
1) Do reasons for/against have value for users?
2) Do reasons against reduce an RS persuasion?
3) Do users perceive a conflict of interest in their interaction with an RS?
4) Do reasons for/against influence user choices?

Our experiment took 31 subjects, all of which are undergraduate students, and asked them to evaluate two RS implementations, one displaying only reasons for recommendations, and the other displaying reasons for and against them. Subjects were presented with an e-commerce mock-up where they received recommendations concerning smartphones. Each subject first received a recommendation and one reason for, and was asked to select an item; then the subject received a recommendation with one reason for and one reason against, and was again asked to select an item. Note that we avoided presenting too many reasons at once. Figure 3 depicts the information presented.

Each subject then evaluated the two RSs individually using five *explanation metrics* [25] that are presented in Table 2. Each subject ranked each RS with respect to each explanation metric using a survey-based Likert psychometric scale [9] from 1 to 5 (standing for "Strongly disagree", 2 "Disagree", 3 "Neither agree nor disagree", 4 "Agree", and 5 "Strongly agree"). This scale was used to reduce central tendency and social desirability biases where subjects do not want to be identified with extreme positions. Finally, each subject could write a short free text with thoughts about the RSs.

Figure 4a shows the percentage of responses given by subjects. Responses, notably for *engagement*, *trust* and *effectiveness*, are concentrated around scores 4 and 5. This result indicates that users mostly agree that showing reasons against a recommendation adds
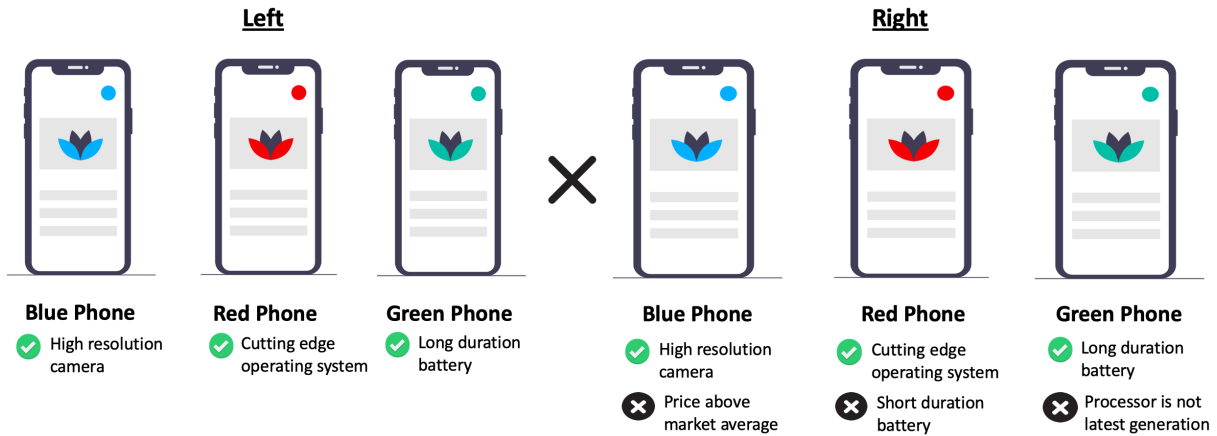
Why should I *not* follow you? Reasons For and Reasons Against in Responsible Recommender Systems

FAccTRec2020, September 26, 2020, Online

Left

Right

Blue Phone	**Red Phone**	**Green Phone**
✅ High resolution camera	✅ Cutting edge operating system	✅ Long duration battery

Blue Phone	**Red Phone**	**Green Phone**
✅ High resolution camera	✅ Cutting edge operating system	✅ Long duration battery
❌ Price above market average	❌ Short duration battery	❌ Processor is not latest generation

Figure 3: Experiment: just one reason for (left); one reason for and one reason against (right).

value with respect to trust, engagement and effectiveness of RS. Figures 4a and 4b show that there was a divergence amongst users about whether the proposed explanation paradigm increases transparency. As our method is model-agnostic (it makes no assumptions about the RS internal behavior), the explanations were unable to shed light on how items were actually recommended. As the transparency score peaked around 3, this does not mean reasons for/against were adverse to transparency; it means that they were as good as just reasons for.

We expected a possible drawback of our proposal would be a reduction in persuasion (as reasons against might make the users less likely to follow recommendations). Figure 4b shows that the down whisker is longer for persuasion than it is for trust, engagement and effectiveness. However, note that the boxplot for persuasion is skewed up; thus most users felt more convinced when reasons against were present. By doing a further analysis of textual comments, we found out that persuasion increases are produced by higher trust in the RS. Consider two comments:

> 1) *I always think that recommendations that bring positive and negative aspects are fairer, and could influence me more into buying the product, once I feel I am not being misled.*
> 2) *As the first example [the first RS] shows only strong points for each product, it leads the user to have a certain mistrust about the suggestions.*

Comments also indicate that many users expect the RSs to try to lead them into a decision, sensing a conflict of interest in the process. Consider the following comment:

> 3) *Differently from marketing which always idealize the product, this one seems to show the reality about it, thus I feel I understand the recommended product in its real form.*

These comments corroborate our hypothesis that, indeed, reasons against have a significant positive impact on the user decision-making process. As a matter of fact, a full 45% of our test subjects changed their choices after we presented reasons against.
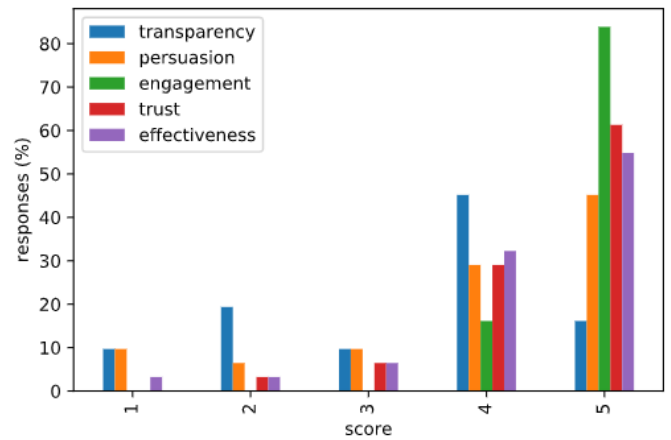
(a) Visual representation for explanation metrics average scores.
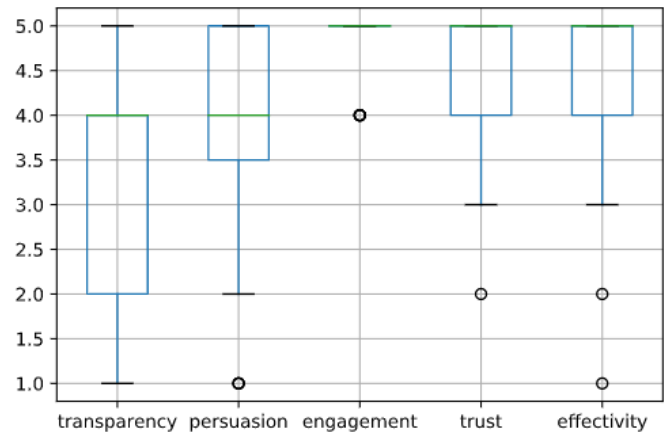
(b) Boxplots for the explanation metrics.

Figure 4: Results from the experiment with human subjects.

5 CONCLUSION

In this paper we have proposed a novel feature for RSs, whose goal is to enhance trust by acting responsibly; namely, we investigated the generation of reasons *for* and *against* recommendations. By displaying such reasons, an RS not only helps the user to reach the most rewarding decision, but the RS acts on its own interest in building trust.

We have developed ways to generate reasons for/against using an auxiliary KG by adapting Snedegar's theory of practical reasoning. Our implementation demonstrates that additional calculations needed to generate such reasons do not affect overall performance. By implementing Snedegar's theory we have found difficulties with some of his schemes for reasons against; we suggest that his Scheme 1 is the most appropriate in practice at the moment. Moreover, our experiment with human subjects demonstrated that reasons against can significantly increase trust, engagement, and even persuasion. Overall we demonstrated that adding reasons against items does improve RSs.

Future work should investigate how much information should be given to users when presenting reasons for/against. It would also be useful to explore mental models of the user so as to extract quantitative objectives to use in Snedegar's Scheme S5. Moreover, it would be important to evaluate our proposals at scale.

6 ACKNOWLEDGMENTS

REFERENCES

[1] M. Alshammari, O. Nasraoui, and S. Sanders. 2019. Mining Semantic Knowledge Graphs to Add Explainability to Black Box Recommender Systems. *IEEE Access* 7 (2019), 110563–110579. https://doi.org/10.1109/ACCESS.2019.2934633

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[3] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. http://arxiv.org/abs/1702.08608 cite arxiv:1702.08608.

[4] Evan Estola. 2016. When Recommendation Systems Go Bad. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 367. https://doi.org/10.1145/2959100.2959117

[5] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 161–169. https://doi.org/10.1145/3109859.3109882

[6] Veronika Henk, Sahar Vahdati, Mojataba Nayyeri, Mehdi Ali, Hamed Shariat Yazdi, and Jens Lehmann. 2018. Metaresearch Recommendations using Knowledge Graph Embeddings.

[7] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge Graph Embedding Based Question Answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 105–113. https://doi.org/10.1145/3289600.3290956

[8] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[9] Rensis Likert. 1932. A Technique for the Measurement of Attitudes. *Archives of Psychology* 140 (1932), 1–55.

[10] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender Systems with Social Regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining* (Hong Kong, China) *(WSDM '11)*. Association for Computing Machinery, New York, NY, USA, 287–296. https://doi.org/10.1145/1935826.1935877

[11] Paolo Massa and Paolo Avesani. 2007. Trust-Aware Recommender Systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems* (Minneapolis, MN, USA) *(RecSys '07)*. Association for Computing Machinery, New York, NY, USA, 17–24. https://doi.org/10.1145/1297231.1297235

[12] Silvia Milano, Mariarosaria Taddeo, and Luciano Floridi. 2020. Recommender systems and their ethical challenges. *AI & SOCIETY* 35 (2020). https://doi.org/10.1007/s00146-020-00950-y

[13] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1 – 38. https://doi.org/10.1016/j.artint.2018.07.007

[14] Christoph Molnar. 2019. *Interpretable Machine Learning*. [Online]. https://christophm.github.io/interpretable-ml-book/.

[15] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2019. Linked open data-based explanations for transparent recommender systems. *International Journal of Human-Computer Studies* 121 (2019), 93 – 107. https://doi.org/10.1016/j.ijhcs.2018.03.003 Advances in Computer-Human Interaction for Recommender Systems.

[16] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A Review of Relational Machine Learning for Knowledge Graphs. https://doi.org/10.1109/JPROC.2015.2483592 cite arxiv:1503.00759Comment: To appear in Proceedings of the IEEE.

[17] John O'Donovan and Barry Smyth. 2005. Trust in Recommender Systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (San Diego, California, USA) *(IUI '05)*. Association for Computing Machinery, New York, NY, USA, 167–174. https://doi.org/10.1145/1040830.1040870

[18] G. P. Polleti, H. N. Munhoz, and F. G. Cozman. 2020. Explanations within Conversational Recommendation Systems: Improving Coverage through Knowledge Graph Embedding. In *2020 AAAI Workshop on Interactive and Conversational Recommendation System*. AAAI Press, New York City, New York, USA.

[19] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-Agnostic Interpretability of Machine Learning. arXiv:1606.05386.pdf http://arxiv.org/abs/1606.05386.pdf

[20] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2010. *Recommender Systems Handbook* (1st ed.). Springer-Verlag, Berlin, Heidelberg.

[21] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems* (Vancouver, British Columbia, Canada) *(NIPS'07)*. Curran Associates Inc., Red Hook, NY, USA, 1257–1264.

[22] Rashmi Sinha and Kirsten Swearingen. 2002. The Role of Transparency in Recommender Systems. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI EA '02)*. ACM, New York, NY, USA, 830–831. https://doi.org/10.1145/506443.506619

[23] Justin Snedegar. 2018. Reasons for and reasons against. *Philosophical Studies* 175, 3 (2018), 725–743.

[24] Maartje ter Hoeve, Anne Schuth, Daan Odijk, and Maarten de Rijke. 2018. Faithfully Explaining Rankings in a News Recommender System. *CoRR* abs/1805.05447 (2018). arXiv:1805.05447 http://arxiv.org/abs/1805.05447

[25] Nava Tintarev and Judith Masthoff. 2007. A Survey of Explanations in Recommender Systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop (ICDEW '07)*. IEEE Computer Society, Washington, DC, USA, 801–810. https://doi.org/10.1109/ICDEW.2007.4401070

[26] W3. 2019. RDF 1.1 Concepts and Abstract Syntax. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[27] Wen Zhang, Bibek Paudel, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Interaction Embeddings for Prediction and Explanation in Knowledge Graphs. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Jan 2019). https://doi.org/10.1145/3289600.3291014

[28] Yongfeng Zhang and Xu Chen. 2018. Explainable Recommendation: A Survey and New Perspectives. *CoRR* abs/1804.11192 (2018), 1–101. arXiv:1804.11192 http://arxiv.org/abs/1804.11192