Explanations within Conversational Recommendation Systems: Improving Coverage through Knowledge Graph Embeddings

Gustavo Padilha Polleti[1]
Hugo Neri Munhoz[1]
Fabio Gagliardi Cozman[1]
[1]Escola Politécnica da Universidade de São Paulo

{gustavo.polleti, hugo.munhoz, fgcozman}@usp.br

Abstract

We propose techniques for explanation generation in conversational recommendation systems. The idea is that explanations can be generated by exploring a knowledge graph, but to improve coverage one must complete the knowledge graph using embeddings. We present a method that searches for explanations using a knowledge graph and associated embeddings; the search is designed to be fast enough to be useful in an interactive context. We describe experiments that validate our proposal, showing that it is superior to alternatives.

Introduction

Conversational recommendation systems resort to dialogue to grasp the user needs and to build accurate recommendations. Such systems can clearly benefit from the ability to state reasons for any particular recommendation. Indeed, explanations can increase user's trust and confidence (Tintarev 2007; Tintarev and Masthoff 2007).

A conversational recommendation system cannot impose long delays on its users; if an explanation is to be provided, then it must be produced quickly. No user will wait a minute to grasp the reasons why a particular product has been suggested. Alas, the literature on explainable AI has paid little attention to the time required to generate explanations; existing methods may take dozens of seconds to explain a single decision (Ribeiro, Singh, and Guestrin 2016; 2018). This may be tolerable during exploratory data analysis, but it is certainly too long for an interactive dialogue.

Recent techniques generate explanations by establishing connections between chosen and recommended items with respect to large scale knowledge graphs (Musto et al. 2019; Alshammari, Nasraoui, and Sanders 2019). However, because the knowledge graphs themselves are often incomplete (Murphy, Talukdar, and Mitchell 2012), many recommendations may remain unexplained due to missing links.

In this paper, we aim at increasing the fraction of recommendations that can be explained by a conversational recommendation system that extracts information from a knowledge graph. We employ knowledge base completion techniques based on embeddings (Nickel et al. 2015) to help generate explanations for recommendations. We focus on embeddings such as TransE (Bordes et al. 2013) as they have state-of-art performance in knowledge base completion.

In short, we investigate the quick generation of explanations through knowledge embeddings in the context of conversational recommendation systems; the embeddings are used to improve coverage, thus guaranteeing the explanation of more recommendations. We have evaluated our proposal concerning time efficiency to ensure that it is suitable for interactive scenarios. Furthermore, we have tested our proposal with human subjects to evaluate the quality of generated explanations.

The paper is organized as follows. The next section presents basic material on conversational recommendation systems, social explanations and knowledge graph embeddings. We then propose our explanation method. Finally, we present our empirical validation set-up and discuss empirical results in two sections on Empirical Evaluation, and offer concluding remarks in the Conclusion.

Background

Any rational process, from daily life interactions to scientific discourse, must rely on explanations both as a source of information and as a cognitive process open for inspection. Despite their importance, explanations are not usually found in recommendation systems that employ latent representations such as matrix factorization (Koren and Bell 2011) or embeddings (Le and Mikolov 2014). The user only gets suggestions, often produced through complex algorithms, that she can either adopt or reject without a provided rationale.

We now discuss a few relevant aspects of explanations and knowledge graph embeddings.

Explanations, Motivations, and the Like

Explanations are essentially communicative processes; within a communicative process, reasons can explain the action *or* point out a motivation for action (Alvarez 2007; 2009; 2010). These two factors are not the same. For example, suppose that John suggests a course Exoplanets101 to Mary because he finds out that Mary has the same interest in astronomy as himself. The fact that John knows that

Mary has the same interest as himself is a reason that explains John's action of suggesting. However, that fact about John's mental state of knowledge is not the reason as to why John suggests a course to Mary as the reason is a fact about Mary, namely that she is interested in Astronomy. That is the *motivating reason*. So, in this example, we have two different (though related) reasons that play different roles: a) that Mary is interested in Astronomy and b) that John knows that Mary is interested in Astronomy as himself. One reason motivates John to suggest Mary Exoplanets101 (the shared interest); the other explains why he does it (the knowledge of such interest). In this example, we learned only at a superficial level the motivation for suggesting a course. If inspected, there is an undefined variety of ground reasons for the motivation, for instance, that John wants to have someone else sharing the same amount of knowledge about exoplanets as he does, or he had a good experience with the instructor, and so on. Clearly motivations can be rather opaque; in our daily lives we do not want to inquire about others' innermost motivation for actions, except in some circumstances such as trials.

If we transpose this reasoning to an algorithm that makes recommendations, the motivation for a recommendation is its computational process, and it could be as opaque as any other human cognitive process regarding the motivation of an action. Even if we try our best to make a conversational recommendation system generate motivating reasons for its suggestions, they would never be enough. However, a rationalized reason would be enough to explain the recommendation. $Student_a$ suggested $Student_b$ Exoplanets because $Student_b$ is interested in Astronomy. Because exoplanets are a topic of Astronomy, the suggestion is justified. This line of thought is at the epistemological level of discussion.

Of course, there are other pragmatic aspects of an explanation: Is it accepted? Is it necessary/sufficient? Such aspects can only be assessed through experiments with human subjects (as we discuss later).

Explanation Generation Techniques

Explanation generation techniques are intended to shed light on the reasons why a complex model, such as a Deep Neural Network (DNN), emits a particular decision. One possibility is to decompose the mechanism that led to the decision. Such an approach is usually said to be model-specific, because it is limited to a specific class of models as it relies on internal information about it.

In contrast to model-specific approaches, model-agnostic approaches make no assumption about the model internal structure, instead taking the model as a black-box. There the goal is to develop an "interpreter" or surrogate that can produce explanations for whatever black-box device makes decisions. Surrogate methods consists of training an interpretable model, perhaps a linear or logistic regression, that can mimic the black-box at least locally around a decision. That is, the local surrogate may be trained only for the instance we aim to explain: First, we add noise to the input instance so that a "neighborhood" of artificial data points is generated; next, we collect the labels produced by the black-box for each point in the neighborhood; finally, we fit an interpretable model. These operations of course take time that must be spent after the particular decision.

For example, Listwise Explainer (LISTEN) (ter Hoeve et al. 2018) explains rankings faithfully by training an interpretable local-surrogate model — similarly to LIME (Ribeiro, Singh, and Guestrin 2016). Despite promising results, LISTEN is still not suitable for explanation generation at scale in real-time environments due to the high computational cost at online training a local-surrogate model for each recommendation. Alternatively, ter Hoeve et al. (2018) proposes Q-LISTEN, where a Neural Network learns the underlying explaining function: while the time to produce an explanation decreases considerably, the surrogate itself becomes a black-box.

In short, local-surrogate based methods are typically time expensive because they demand that a new interpretable model is trained from scratch to explain an single decision. To speed up matters, one might consider training a single global-surrogate model that aims at capturing the black-box behavior as a whole and draw explanations from it for all instances. While this approach could dramatically reduce the explanation time, it is hard to expect that a simple and interpretable model can faithfully capture the complex black-box behavior. Indeed, global-surrogate based methods, like XKE, display relatively low fidelity (Gusmão et al. 2018).

As noted in the Introduction, concern about time efficiency is really important in interactive systems such as the ones we contemplate. We thus look at faster techniques based on knowledge graphs.

Detour: Knowlege Graphs and their Embeddings

A Knowledge Graph (KG) G is here taken as a set of entities $\mathcal{E}$, relations $\mathcal{R}$ and facts $\mathcal{T}$. A fact is an atomic representation of a relationship between entities. We model a fact as a triple $\langle h, r, t\rangle$, where the head entity h is the subject, relation r is the predicate and tail entity t is the object. For instance, the information that "exoplanets is a topic of astronomy" can be described as the triple $\langle exoplanets, topic\_of, astronomy\rangle$, in which both exoplanets and astronomy are entities, and $topic\_of$ the relation connecting them.

Large-scale KGs, such as Freebase (Bollacker et al. 2008) and DBpedia (Auer et al. 2007), are often incomplete (Nickel et al. 2015); this clearly limits their application to real-world tasks. Knowledge Graph Embeddings (KE) now achieve state-of-the-art performance in KG completion tasks such as triple classification and link prediction (Wang et al. 2019). Embedding techniques learn representations for entities, so that relationships between them can be predicted by operations in the latent space. Embeddings are learned as a result of an optimization process that maximizes the total plausibility of all known facts in the original KG. Thus, a KE model must define a plausibility scoring function $f_r(h, t \mid \Theta)$, where Θ represents all model parameters and h, r and t are head, relation and tail respectively.

Despite being originally proposed for knowledge base completion, we should note that embeddings are also used to produce recommendations (He, Kang, and McAuley 2017; Henk et al. 2018) and to answer questions (Huang et al. 2019). Proposals in the literature typically employ the KE

plausibility scoring function to rank entities and to return those with the highest plausibility as recommendations. Even though these approaches are accurate, they are not interpretable as they operate in the latent space of embeddings, and they do not attempt to generate explanations for/with embeddings (as we do).

Explanations via Knowledge Graphs and Embeddings

There are recommendation systems that rely on large-scale knowledge graphs for explanation generation; for instance, ExpLOD (Musto et al. 2019) and ASEMF_UIB (Alshammari, Nasraoui, and Sanders 2019). Both employ semantic information about items to find similarities between user profiles (e.g., previously liked items). For example, consider that a hypothetical recommendation system suggests "Titanic" to someone who has watched "Avatar." If the knowledge base contains the fact that "James Cameron" directed both movies, then ExplLOD might utter: "I recommend you Titanic because you are fond of movies directed by James Cameron like Avatar." The explanation might be even more transparent:"I recommend you Titanic because you have been watching James Cameron's movies lately." Despite producing human-friendly explanations, this sort of approach relies on the completeness of the KG to work properly, a strong assumption considering the incompleteness of large-scale KGs (Murphy, Talukdar, and Mitchell 2012). If the KG does not contain the fact that "James Cameron" is the director of "Titanic", the recommendation system may fail to explain the recommendation.

One alternative is to use knowledge embeddings to complete the original KG. CrossE (Zhang et al. 2019) introduces an embedding-based explanation search method for a specific type of interaction between entities and relations called *crossover interactions*. For instance, the fact "user A" is friend of "user B", who likes "Titanic", can be considered as an explanation for the fact "user A likes the movie Titanic" if and only if it can be found at least one crossover interaction in the KG to support it, where users who like the same movies are mutual friends. Although CrossE explores the latent embedding space, it still relies on the KG, which limits the number of instances it can find explanations. Also, CrossE restricts its search to crossover interactions only, while other proposals in the literature suggest that more expressive types of graph features can produce better explanations (Gusmão et al. 2018) (Gardner and Mitchell 2015).

Table 1: Qualitative comparison among proposals in the literature.

Approach	Real-Time	High Coverage
LISTEN	No	Yes
ExpLOD	Yes	No
ASEMF_UIB	Yes	No
XKE	Yes	No
CrossE	Partially	No

Table 1 highlights a research gap on explanation methods that are suitable for interactive, real-time settings and that display high coverage. This is exactly where the contributions in this paper fit in.

Embedding-based Explanations based on Depth-First Search

We wish to use KGs to generate explanations in conversational recommendation systems (CRSs). Given that KG incompleteness seems to be the cause of missing explanations in existing methods, we expect KEs to be useful in increasing the number of explained recommendations due to the ability of KEs to infer new facts with good accuracy (Bordes et al. 2013; Wang et al. 2014). Presumably, we can find good explanations by searching for facts in the (necessarily "complete") KE latent space.

As a digression, we note that embeddings themselves are black boxes, so one might argue that the overall scheme is not really interpretable. Indeed we cannot provide an explanation from "first principles" using embeddings, but we can certainly provide an explanation that is based on solid semantic information in the available KG using their embeddings. For instance, consider a student interested in space shuttles; for this student, the class Exoplanets101 is recommended. Perhaps the available KG does not contain a connection between topics in Exoplanets101 and space shuttles, but the learning KE may indicate that, based on connections in the KG, there is a strong relationship between topics in Exoplanets101 and space shuttles — a relationship that matters when explaining the recommendation.

We now describe our proposal for explanations in CRSs.

An Abstract Recommendation Scheme

While interacting with the CRS, the user must inform her preferences. Suppose that such preferences can be mapped to an entity e_h in an available large-scale KG. Assume that our base recommendation system runs link prediction using an embedding (built from the same KG) and returns the Top-N ranked entities as recommendations. This is a conceptual scheme that corresponds to the vast majority of recommendation procedures.

To illustrate the assumed recommendation mechanism, suppose Exoplanets101 is recommended to a student whose preference lies in astronomy. Then:

$$T = [f_{subject}(e_{astro}, e_i \mid \Theta), e_i \in \mathcal{E}];$$

$$sort\_desc(T) = \begin{bmatrix} Exoplanets101 \\ Aeronautics102 \\ \vdots \\ e_m \end{bmatrix}.$$

In this toy example, T is the list containing plausibility values for all entities in $\mathcal{E}$. We sort T in descending order, and identify that Exoplanets101 is more related to astronomy than Aeronautics102 and so on.

That is, the recommendation procedure basically recommends the N entities that best fit as a tail entity in the triple

$\langle h, r, ? \rangle$, where r is a relation modelling how tail entities meets user preferences h. For instance, in our example on astronomy, the user desires classes about a theme of interest, so the relation r in this case could be "topic of class".

The Possible Explanations

We take an explanation to be, in our context, a path of length L composed of relations $r_i \in \mathcal{R}$ connecting e_h and e_t. For instance, the explanation for our toy example (the explanation that Exoplanets101 is about exoplanets, and exoplanets is a topic of astronomy) could be modeled as the path of length 2:

$$astronomy \xrightarrow{subject} exoplanets \xrightarrow{subject} Exoplanets101$$

We must specify the set of possible paths $\pi \in \Pi_L$ that, if found, are considered to be explanations. We assume that such a set is specified by declaring the kinds of sequences of relations that are permissible. It is important to adequately specify Π_L because there might exist paths that do not provide any sense of causality, even though they connect e_h and e_t; such meaningless paths should not be included in Π_L. Also, the more paths we have in Π_L, the higher the computation time required. In small domains, i.e., KG with a small number of relations, an expert may define Π_L manually, however in bigger ones, we expect that automated approaches will be useful, such as graph feature selection methods (Gardner and Mitchell 2015). It is worth mentioning that when we filter the paths included in Π_L, we may end up missing explanations. So, we must consider a trade-off between coverage and time efficiency while conducting an explanation search. For this paper, we assume that Π_L is available to the conversational recommendation system.

Searching for Explanations

We go through every path $\pi \in \Pi_L$ starting from e_h, using a depth-first search (DFS), and if at the end of the path we find e_t, the sequence of nodes visited from e_h to e_t is recognized as an explanation. Here the search-tree height is known beforehand and equal to the path length L; for this reason we use DFS instead of say breadth-first search.

It is important to recall that, due to KG incompleteness, we run this search in the space of all completions of the KG as produced by the given embedding. However, the KE is a real-valued continuous latent space and not a graph; how can we perform DFS on it?

Clearly, a graph $\hat{G}$ can be build using the KE. Basically, the KE can perform two main tasks, link prediction (LP) and triple classification (TC). TC means to classify a given triple as true or false, i.e., tells if an edge in the KG holds. Thus, with TC alone, we can build $\hat{G}$ by merely classifying all possible relationships between entities $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$, but we go further. With LP, we can also assign plausibility scores to each edge in $\hat{G}$ so that we can discriminate which links are stronger. We assume that the more plausible an edge is, the more expected or obvious is the relationship it describes. As we want our explanations to be easy to understand, we prioritize edges with a high plausibility in the DSP.

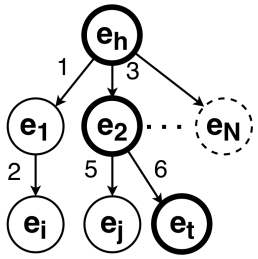

Figure 1: Depth-First Search toy example.

Formally, a path is a sequence of relations $\pi = \{r_1, r_2, ...r_L\}$. A path exists only if we can find a sequence of entities $\Omega = \{e_1, e_2, ..., e_L\}$, where for all entities $e_i \in \Omega$, the triples $\langle e_{i-1}, r_i, e_{i+1} \rangle, \forall i \in 1, 2, ..., L$ also holds. We consider that a triple exists if the plausibility score for it is greater than the threshold δ_r, the same as in TC. That is,

$$f_{r_i}(e_{i-1}, e_i) > \delta_r, \ \forall e_i \in \Omega;$$

$$e_0 = e_h, \ e_{L+1} = e_t.$$

We start by assigning the head entity e_h as root node. Then we expand its outgoing edge with the highest plausibility score considering the first relation r_1 in the path. We then repeat the procedure for the expanded node and the second relation in the path, and so on.

To illustrate this procedure, consider the example of a depth-first search in Figure 1. The nodes are sorted from the highest plausibility score in the left to lowest in the right. The numbers in the arrows represent the order each node is visited. In this particular example, an explanation is the path $e_h \rightarrow e_2 \rightarrow e_t$.

Empirical Evaluation: Set-Up

We carried out an evaluation intended to answer a number of research questions (discussed in the next section). In this section we describe the conversational recommendation system (and associated tools) we have built.

We implemented two CRSs in an educational domain; both recommend courses offered by the Universidade de São Paulo. Our CRSs aim at helping students to find classes of interest given the large collection of courses offered by Universidade de São Paulo. We developed the chatbots using the Dialogflow platform.[1] We adopted a timeout constraint of 5 seconds on the response time to ensure responsiveness. Both conversational recommendation systems receive as input a single preference of theme from the user, then answer with the best found recommendation and its explanation.

We also built a knowledge graph and an associated embedding; those tools were used to produce recommendations and explanations. One CRS uses the original knowledge graph as a source for explanations, and another CRS uses the knowledge embedding.

We briefly describe the construction process of the KG consisting of information about courses and faculty of

[1] https://dialogflow.cloud.google.com/

Universidade de São Paulo. The KG structure consists of five types of entities: learning-object, professor, concept, and category. The learning-object includes both graduation courses offered by the university and articles authored by faculty members. Concepts and categories represent an ontology for the learning-objects content. Thus, we say that a faculty member is involved in multiple learning-objects in which he or she can either teach a course or author an article and that each learning-object is about multiple concepts.

To build the KG, we opted for using an automated semi-structured approach (Nickel et al. 2015) as it has also been employed by many popular large scale KGs, such as DBpedia (Auer et al. 2007) and YAGO2 (Hoffart et al. 2013). The selected approach aims at automatically extracting information from semi-structured data, like infoboxes, via rules or regular expressions. Firstly, we collected the description and 543 teachers' names for 1740 engineering courses from the university graduation support website.[2] We also retrieved from the academic repository Scopus[3] 7648 articles authored by the faculty members. Finally, we performed entity linking to DBpedia [4] using the articles' keyword and course description content, so the hierarchy of concepts and categories were incorporated from DBpedia. The whole process resulted in a Knowledge Graph with 34182 entities, 3 relations, and 152468 triples. Even though the algorithm produces accurate knowledge graphs, typically, they are incomplete (Nickel et al. 2015).

Once we constructed our Knowledge Graph, we trained a TransE (Bordes et al. 2013) knowledge embedding model with 500 dimensions for 1000 epochs. We opted for using a batch size of 500, 0.001 as alpha, 1.0 as margin and the optimizer ADAGRAD to perform the training. We selected TransE because it is commonly used as benchmark in the literature (Gusmão et al. 2018; Wang et al. 2019).

Using the trained KGE, we implemented a neighborhood-based recommendation system. Our recommendation system performs the link prediction task $\langle head, relation, ? \rangle$, in which the $head$ is a conceptual entity representing a preference of theme provided by the user, $relation$ is the "subject" relationship modeling learning-objects content. Therefore, we consider the plausibility score provided by the KGE to rank entities and, then, realize a Top-N recommendation, as described in our proposal.

To illustrate the kinds of explanations generated by the CRSs, Figure 2 depicts an explanation example. Entities and relations found in the KG appear in the figure, while the textual explanation derived from them appears in the caption.

Empirical Evaluation: Experiments

In this section we report on experiments that were designed to address the following research questions:

1. Can we find at least one explanation for a greater fraction of recommendations when we search the knowledge embedding than the original graph given timeout constraints?

[2] At https://uspdigital.usp.br/jupiterweb/.

[3] https://www.scopus.com/home.uri

[4] http://dbpedia.org/

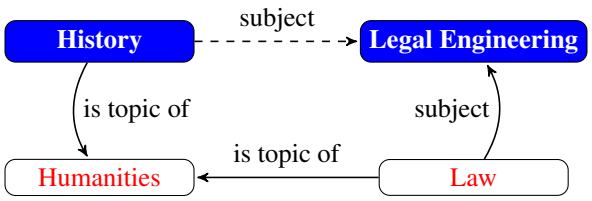

Figure 2: "Legal engineering is recommended as it is about Law and both Law and History are topics of Humanities"

2. How long does it take to find explanations using the knowledge embedding?Is time-to-response acceptable?
3. The quality of the explanations found using the knowledge embedding deteriorates when compared to those using the original graph?

We have run two sets of experiments, one with simulated data (aiming at the first two questions above), and the other with real data collected from human subjects (aiming at the third question above).

Coverage and Execution Time

We designed user-simulated experiments to evaluate the fraction of recommendations that our proposed method can find at least one explanation for — we call it *Recall* or *Coverage*. Also, we evaluated the time our proposal takes to find multiple explanations for recommendations.

Figure 3 presents the behaviour of the recall for our proposed method (*embedding recall*) compared to the baseline (*graph recall*), also the average number of explanations found (*avg. explanation* n^o) and average execution time (*avg. exec. time*) for our proposed method, when varying time constraints (timeout).

While the baseline method, which uses only the original graph to search for explanations, is by far faster than our proposed one, we can observe that the graph recall achieves a certain degree of "saturation" at 42%, which is a significantly lower level than the embedding one at 99%. Here we consider a "saturation level" the point where one does not have timeout constraints, i.e., virtually infinite time to search for explanations. Thus, we verify that the original KG cannot find explanations for less than half of recommendations in our experiment; also, it is not sensitive to time constraints. On the other hand, the embedding recall, despite having a slow start (close to 0 for timeouts shorter than 2, 3 seconds), grows greater than the graph recall for timeouts longer than 3 seconds. Indeed, for a timeout of 5 seconds, a timeout that can be considered acceptable for an interactive application, we observe that our proposed method can explain almost two times more recommendations than if using the original graph; this answers our first research question **Q1**. Note that the average number of explanations and the average execution time behave linearly, considering the timeout value. This points out that it may be expensive, in terms of computation cost, to find multiple explanations for the same recommendation.

Figure 4 shows the boxplots, with suppressed outliers for better visualization, of the execution time of our proposed

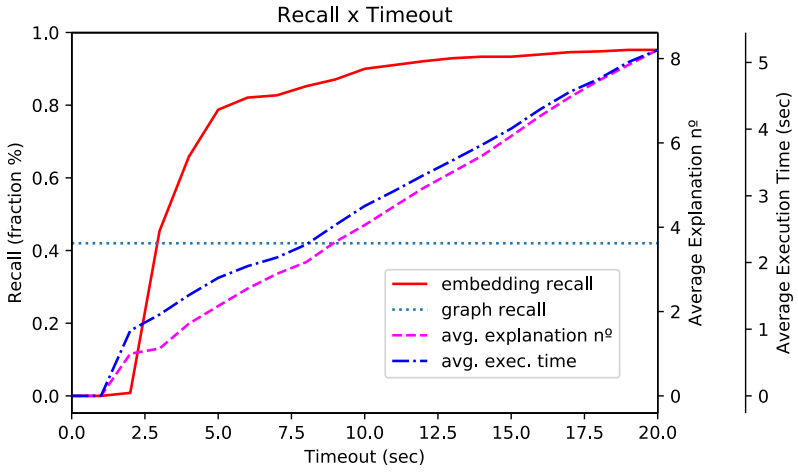

Figure 3: Recall comparison between our proposal (embedding recall) and the baseline (graph recall). Also present average explanation number found (avg. explanation n^o) and average execution time (avg. exec. time) for our proposal

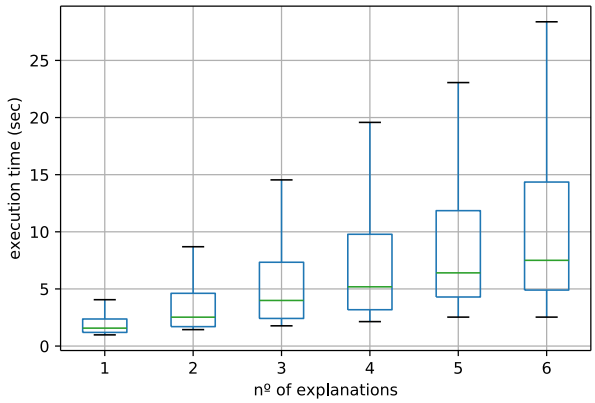

Figure 4: Execution time of our proposal considering explanation number constraints.

method for different numbers of explanations. In this experiment, we aim to evaluating how long it takes to find a given number of explanations for a recommendation. We can observe that all boxplots are skewed down, and the top whiskers are longer than the bottom ones; also, the variability of execution time increases as more explanations are demanded. Considering an acceptable response time (for instance, 5 seconds), for a small number of explanations (one to three), the median value is acceptable, and for a single explanation, even the maximum value is acceptable. Therefore, our approach can produce multiple explanations but in a small quantity, answering the second question **Q2**.

Quality of Explanations

As described previously, we produced two CRSs, one with the automatically generated KG as a source of explanations, and the other with our proposed (embeddings-based search) method as a source of explanations. Our goal was to compare both techniques.

We conducted a user study involving 26 students, in which each user evaluated two CRSs, one employing our proposed method and the other one using the original KG as a source for explanations. As in (Tintarev and Masthoff 2007), the users were asked to evaluate the following explanation metrics: *transparency*, *persuasiveness*, *engagement*, *trust* and *effectiveness* (Musto et al. 2019). The metrics are summarized in Table 2.

We designed a survey-based Likert psychometric scale (Likert 1932). Users could assign grades ranging from 1 to 5 in which 1 stands for "Strongly disagree", 2 "Disagree", 3 "Neither agree nor disagree", 4 "Agree", and 5 "Strongly agree". This scale helps to avoid central tendency bias that may happen in this situation whenever users do not want to present themselves with extreme positions, hence acting by the social desirability bias.

We asked the 26 students to interact with both chatbots and, at the end of dialogue, to provide scores from 1 to 5 for each one of the previously mentioned explanation aims (Tintarev and Masthoff 2007); we used the questionnaire described as Table 2. One interaction consists of the user asking for a recommendation for 5 different themes, so we collected a total of 130 interactions.

Considering the exploratory nature of the survey, we describe below the performance indicators from the users' interaction with the CRS. Table 3 presents the average scores provided by the students in our user study for each one of the explanation aims. Figure 5 depicts table 3 on a continuum representing visually the scale. Comparing both algorithms' overall mean, the KE approach (PRED) was better from the user's perspective $\mu = 2.7$ corresponding to the "neutral" evaluation at the Likert scale; what answers the third question **Q3**. On the other hand, for the graph approach (TRUE) $\mu = 2.26$ closer to the "disagree" at the Likert scale. Taking the variable in isolation, Effectiveness got the highest average value for both $\mu_{pred} = 2.9$ and $\mu_{pred} = 2.64$. It signalizes that users perceived the explanations as coherent. The

Table 2: Questionnaire details.

Aim	Question
transparency	Did the explanation help you to understand the recommendation?
persuasion	On the basis of the explanation, would you follow the recommendation?
engagement	Did the explanation have a pedagogical effect?
trust	Did the explanation contribute to increase your confidence in the recommendation system?
effectiveness	Did the explanation sound coherent ?

Table 3: Average scores for explanation aims from our user study.

Algorithm	Transparency	Persuasion	Engagement	Trust	Effectiveness
PRED	2.92	2.28	2.84	2.52	2.92
TRUE	2.21	2.36	2.17	1.92	2.64

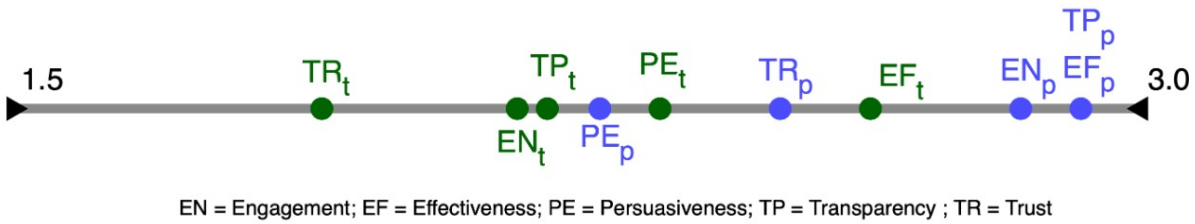

Figure 5: Visual representation for explanation aims average scores.

TRUE approach had a bad evaluation when the trust was at stake ($\mu_{trust} = 1.92$). Since TRUE suffers from KG incompleteness, it cannot posit explanations for every suggestion. When compared to a better performance of the KE approach ($\mu_{trust} = 2.52$), we might conjecture that users prefer any explanation instead of no explanations at all.

Users tended to evaluate all the indicators with low variations for both approaches ($\sigma_{pred} = 1.10$, $\sigma_{true} = 1.09$). For a further interpretation of users' behavior, we can assume that $\sigma >= 1.0$ as a token of higher discrimination of the indicators. 18.5% of users evaluate more carefully the KE approach against merely 7.4% for the TRUE. One can say that users evaluated the overall experience in general, and more strongly in the TRUE approach. Among those who discriminated the indicators properly in KE approach the average was significantly higher $\mu_{\sigma>=1} = 3.28$, which transparency ($\mu = 4.0$) and engagement ($\mu = 3.8$) were better evaluated.

We shall restate here that these are just a first exploratory analysis of users' experience. Some insights from the feedback (such as it is better to have any explanation rather than no explanation) must be further explored in future work.

Conclusion

In this paper we proposed and evaluated techniques that produce fast and effective explanations in the context of conversational recommendation systems. Our experiments support the claim that KEs, if properly employed, indeed increase explanation coverage, while also satisfying reasonable time constraints. In addition, the experiment with human subjects presented evidence that explanations drawn from embeddings not only remain coherent and meaningful from the user perspective, but also increase trust in the CRS, transparency perception and overall satisfaction.

The present work represents a step towards efficient explanation generation methods that are suitable for interactive and conversational recommendation systems. Future work should include the exploration of novel approaches for explanation selection that can handle time constraints. Also, we intend to investigate how argumentative explanations can improve the interaction between users and CRSs.

Acknowledgments

This work was carried out with the support of Itaú Unibanco S.A.; the first author has been supported by the Itaú Scholarship Program (PBI), linked to the Data Science Center (C2D) of the Escola Politécnica da Universidade de São Paulo. The second author has been supported by the São Paulo Foundation (FAPESP), grant 2018/09681-4. The third author has been partially supported by the Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq), grant 312180/2018-7. The work was also supported by the FAPESP, grant 2019/07665-4, and also by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) - finance code 001. We are also grateful to the Center for Inovation at Universidade de São Paulo (InovaUSP) for hosting our lab.

References

Alshammari, M.; Nasraoui, O.; and Sanders, S. 2019. Mining semantic knowledge graphs to add explainability to black box recommender systems. *IEEE Access* 7:110563–110579.

Alvarez, M. 2007. The causalism / anti-causalism debate in the theory of action: What it is and why it matters. In Leist, A., ed., *Action in Context*. De Gruyter.

Alvarez, M. 2009. How many kinds of reasons? *Philosophical Explorations* 12:181–93.

Alvarez, M. 2010. *Kinds of Reasons: An Essay on the Philosophy of Action*. Oxford University Press.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In Aberer, K.; Choi, K.-S.; Noy, N.; Allemang, D.; Lee, K.-I.; Nixon, L.; Golbeck, J.; Mika, P.; Maynard, D.; Mizoguchi, R.; Schreiber, G.; and Cudré-Mauroux, P., eds., *The Semantic Web*, 722–735. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, 1247–1250. New York, NY, USA: ACM.

Bordes, A.; Usunier, N.; García-Durán, A.; and Weston, J. 2013. Translating Embeddings for Modeling Multi-Relational Data. *Advances in Neural Information Processing Systems* 2787–2795.

Gardner, M., and Mitchell, T. 2015. Efficient and expressive knowledge base completion using subgraph feature extraction. 1488–1498.

Gusmão, A. C.; Correia, A. C.; De Bona, G.; and Cozman, F. G. 2018. Interpreting Embedding Models of Knowledge Bases : A Pedagogical Approach. In *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, number Whi, 79–86.

He, R.; Kang, W.-C.; and McAuley, J. 2017. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, 161–169. New York, NY, USA: ACM.

Henk, V.; Vahdati, S.; Nayyeri, M.; Ali, M.; Yazdi, H. S.; and Lehmann, J. 2018. Metaresearch recommendations using knowledge graph embeddings.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; and Weikum, G. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence* 194:28 – 61.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *ACM International Conference on Web Search and Data Mining*, WSDM '19, 105–113. New York, NY, USA: ACM.

Koren, Y., and Bell, R. M. 2011. Advances in collaborative filtering. In Ricci, F.; Rokach, L.; Shapira, B.; and Kantor, P. B., eds., *Recommender Systems Handbook*. Springer. 145–186.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, II–1188–II–1196. JMLR.org.

Likert, R. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 140:1–55.

Murphy, B.; Talukdar, P.; and Mitchell, T. 2012. *Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding*. Mumbai, India: The COLING 2012 Organizing Committee. 1933–1950.

Musto, C.; Narducci, F.; Lops, P.; de Gemmis, M.; and Semeraro, G. 2019. Linked open data-based explanations for transparent recommender systems. *International Journal of Human-Computer Studies* 121:93 – 107. Advances in Computer-Human Interaction for Recommender Systems.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2015. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *CoRR* abs/1503.00759.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1135–1144. New York, NY, USA: ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)*.

ter Hoeve, M.; Schuth, A.; Odijk, D.; and de Rijke, M. 2018. Faithfully explaining rankings in a news recommender system. *CoRR* abs/1805.05447.

Tintarev, N., and Masthoff, J. 2007. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW '07, 801–810. Washington, DC, USA: IEEE Computer Society.

Tintarev, N. 2007. Explanations of recommendations. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, 203–206. New York, NY, USA: ACM.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

Wang, Y.; Ruffinelli, D.; Gemulla, R.; Broscheit, S.; and Meilicke, C. 2019. On evaluating embedding models for knowledge base completion. In *Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 104–112. Florence, Italy: Association for Computational Linguistics.

Zhang, W.; Paudel, B.; Zhang, W.; Bernstein, A.; and Chen, H. 2019. Interaction embeddings for prediction and explanation in knowledge graphs. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 96–104. New York, NY, USA: ACM.