GUSTAVO PADILHA POLLETI

EXPLANATION GENERATION FOR CONVERSATIONAL RECOMMENDATION SYSTEMS BASED ON KNOWLEDGE EMBEDDINGS

São Paulo
2022

GUSTAVO PADILHA POLLETI

EXPLANATION GENERATION FOR CONVERSATIONAL RECOMMENDATION SYSTEMS BASED ON KNOWLEDGE EMBEDDINGS

Dissertation presented to the Escola Politecnica da Universidade de São Paulo to obtain the Master of Science degree.

Concentration Area:

Computer Engineering

Advisor:

Prof. Dr. Fábio Gagliardi Cozman

Revised Version.

São Paulo
2022

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 21 de Julho de 2023

Assinatura do autor: Gustavo Padilha P. Oll.

Assinatura do orientador: ____________________

Catalogação-na-publicação

Polleti, Gustavo Padilha
Explanation Generation for Conversational Recommendation Systems based on Knowledge Embeddings / G. P. Polleti -- versão corr. -- São Paulo, 2022.
82 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Explainable AI 2.Interpretability 3.Relational Machine Learning 4.Embedding Models 5.Knowledge Bases I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

ACKNOWLEDGMENTS

First of all, I would like to thank my parents who have always supported and trusted me in all moments throughout my journey.

A special thanks to my advisor Dr. Fabio Gagliardi Cozman for the advice and for guiding me during my evolution in this research area. I am grateful to the brilliant group of colleagues who are part of Data Science Center (C2D) and of Center for Artificial Intelligence and Machine Learning (CIAAM), specially to Douglas Luan de Souza, Andrey Ruschel and Hugo Neri who collaborated in this research. In addition, I want to thank Escola Politécnica for providing adequate infrastructure and environment for research.

Finally, I would like to thank everyone who somehow contributed to my evolution and to this research.

"Epigraphy"

Lotus Sutra-We have gained an
immeasurably rare jewel, something
unsought that came of itself.

ABSTRACT

Conversational agents or chatbots are increasingly employed in commercial applications to answer questions and to recommend items. Despite their success, they usually behave as black-boxes from the user perspective, typically failing to produce high quality human-computer interactions. Thus interpretability is a major concern for the next generation of recommendation systems. This work addresses challenges related to the development of a recommendation system that can explain its own suggestions. Furthermore, this work evaluates the impact of different explanation generation techniques both in simulated interactions and in tests with human subjects. This work present novel model-agnostic methods that address challenges of explanation generation in the context of knowledge embedding based conversational recommendation systems, such as: explanation fidelity, graph incompleteness, time to response constraints and reasons against generation. Finally, this research evaluates the technical feasibility of such methods with simulated experiments and shows preliminary on user perception of the generated explanations.

Keywords – Conversational Recommendation System, Explanation, Recommendation System, Interpretability, Knowledge Graph, Knowledge Embedding.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

USP	Universidade de São Paulo
RS	Recommendation System
CF	Collaborative Filtering
CB	Content-Based
CRS	Conversational Recommendation System
KB	Knowledge Base
KG	Knowledge Graph
KE	Knowledge Embedding
KBC	Knowledge Base Completion
GS	Global-Surrogate
LS	Local-Surrogate
NLU	Natural Language Understanding
DM	Dialogue Management
NLG	Natural Language Generation

CONTENTS

1 INTRODUCTION

"Causal explanation takes the form of conversation and is thus subject to the rules of conversation."

-- Hilton

We all need to make a number of decisions everyday. What recipe should I cook? Should I buy X instead of Y? What movie should I watch? While most of these decisions are harmless, e.g. a poor movie choice to watch on a Saturday night, some of them may have serious implications. For example, the physician's decision whether to operate on a patient or not, or the HR recruiter's choice among all applicants of whom will be hired or not. These are examples of high stake decisions that must not be taken lightly.

Unfortunately, high stake decisions are not restricted to positions of power, such as those of a physician or a recruiter, but they are also a part of common daily routine. Suppose, for instance, that you are an undergraduate student and you need to choose which subjects apply for the next semester. If you register into a class that does not fit your interests, you will likely have a sub-optimal learning experience and your next application window will only be open next semester. A bad application decision can be very expensive both in terms of time spent and learning experience.

One could argue that the solution for high stake scenarios should simply be to evaluate all the available options carefully. In fact, this could work well when the set of options is small; however, in most real-world cases the set of options is too large to be completely inspected within acceptable time.

A Recommendation System (RS) can be used to play the role of an expert that filters the available options to only contain the most promising ones, so that users can focus their attention and make assertive decisions more efficiently. RSs typically employ algorithms that behave as black-boxes from the user perspective. It is commonly stated that performance and interpretability are opposing goals in machine learning [1]; however, matters are more delicate in the context of RSs, as performance itself depends on trust [2] and high interpretability is bound to increase trust — when interpretation fails, existing RSs may fail in surprising ways [3]. It is hard to expect any user to blindly follow the recommendations of a black-box when an important decision is at stake. In such scenarios, the user wishes to understand the rationale for recommendations in order to trust them.

To achieve high performance while maintaining interpretability, previous proposals have explored explanation generation methods [4,5]. Such methods are typically independent from the RS algorithm, so that the RS can remain complex while its recommendations are escorted by proper explanations.

In this work, we investigate explanation generation in one type of RS, the Conversational Recommendation System (CRS). Such systems resort to dialogue to grasp the user needs and to build accurate recommendations. Differently from traditional RSs that typically operate in an offline manner, a CRS is subject to the strong time constraints of interactive applications; if an explanation is to be provided, it must be produced quickly. No user will wait a minute to grasp the reasons why a particular product has been suggested. The research on explainable AI often pays little attention to the time required to generate explanations and its viability in interactive applications; existing methods may take dozens of seconds to explain a single decision [1,6]. In this work, we consider these circumstances.

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 generating some mistrust. A user may grow suspicious of an explanation that is not transparent about possible downsides of the recommendations. We here 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 [7], a philosophical theory of practical reasoning, and realize it in the context of CRSs. For this, we start with existing procedures that generate reasons *for* by analyzing paths in knowledge graphs and their embeddings [4,8]. 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 in an interactive dialog application and identify the most promising strategies. We also describe a CRS we have implemented and its practical operation with reasons for/against. We additionally, carried out experiments with human subjects that show our approach to responsible recommendations to yield higher overall trust in the explanations generated.

1.1 Goal

The main goal of this research is to improve explanation generation methods for conversational recommendation systems. We cover two major research topics:

- **Online Explanation Generation**: we focus on explanation generation under time constraints; we address the challenges of interpretability methods in online applications such as interactive dialog systems.
- **Responsible Explanations**: we offer explanations for recommendations that lead to a more trustworthy and transparent human-computer interaction.

To address the first topic, we argue that methods based on knowledge embeddings (instead of large-scale knowledge graphs) can lead to efficient explanation generation while maintaining high fidelity, model-agnosticism and coverage. To approach the second topic, we rely on a conversational model [9, 10] coupled with reasons for and against, so as to produce improvements in the user's trust and transparency perception.

The expected outcome of this research includes the development of model-agnostic explanation generation methods for knowledge-embeddings in the context of recommendation systems that provide faithful and high coverage explanations while dealing with time to response constraints. We hope this work will represent a step forward trustworthy conversational recommendation systems that effectively assist its user's decision-making process. The trust dimension, which we expect to improve with this research, may increase the social acceptance of conversational commerce and contribute to the development of this emergent industry.

1.2 Organization of this Manuscript

This document is organized as follows: Chapter 2 presents the main conceptual aspects and relevant background to our research, such as conversational recommendation systems, knowledge graphs and interpretability. In Chapter 3 we discuss our proposals, with their strengths and weaknesses. Then Chapter 4 describes the machinery used in this work implementation, alongside with our evaluation methods. Finally, Chapter 5 shows our preliminary results and Chapter 6 summarizes our contributions and suggests future work.

2 BACKGROUND

"If I have seen further than others, it is by standing upon the shoulders of giants."

-- Sir Isaac Newton

This section presents notation and terminology, and some theoretical foundations about the main concepts needed in this work. Firstly, we briefly describe the commonest taxonomy for recommendation systems and highlight the class of knowledge-aware content-based recommendation systems; there lies the focus of this research. Next, we explore the underlying mechanisms of knowledge-aware recommendation systems, discussing in detail their core components: knowledge graphs and embeddings.

Once we have covered the building blocks of recommendation systems and knowledge graphs, we dive into the limitations of these embedding-based models from an interpretability perspective. We focus on explanation generation strategy as source of interpretability for opaque models such as embeddings. Next, we discuss the strengths and weaknesses of the commonest approaches involved in the explanation process.

2.1 Recommendation Systems

Recommendation systems provide suggestions for items so as to support user decision-making [11]. User interests are usually expressed as a historic profile of actions or ratings. Despite their success [12] (Amazon's CEO reported in 2006 that almost 35% of their sales originated from recommendations), recommendation systems often face difficulties with cold starts and changes in user interests. Recent efforts have explored adaptive behavior, reinforcement learning and dialogue systems [13, 14].

Recommendation systems can be classified into six groups [11]: Content-based (CB), Collaborative filtering (CF), Demographic, Knowledge-based, Community-based and Hybrid. Content-based systems are built under the assumption that a given user would prefer similar items rather than different ones. While intuitive, the CB assumption may not hold if user preferences are too diverse or drifts over time. In contrast to CB approaches, collaborative filtering assumes a given user would prefer items previous selected by other users with similar tastes to hers.

All RSs, from the six categories, aim at identifying similarities, but their targets are different. While content-based systems are based on item description, collaborative filtering, on the other hand, considers similarities between users interests, so that it recommends items regardless of whether they are similar or not to the user profile. Knowledge-based recommendation relies on expert beliefs about how items meet users needs and preferences. Community-based and demographic systems are based on features of the user herself. Hybrid RSs simply combine strategies.

Even though collaborative filtering techniques are the most popular among traditional applications, content-based systems are notably powerful techniques in domains where data about users preferences is scarce. Consider a scenario where you do not have a historical record of user choices, e.g. a new user starts using your recommendation system, it would be impossible to use a collaborative filtering system due to the lack of data; this challenge is known as cold-start [15]. As content-based systems do not depend exclusively on user profiles, they can avoid difficulties like cold-starts; besides, they can be associated with collaborative filtering techniques [16, 17].

An emergent class of state-of-the-art content-based techniques, knowledge-aware ones, typically benefit from heterogeneous information networks or general knowledge graphs to model item contents [18, 19]. These graph-like structures provide enough flexibility to incorporate item descriptions from multiple sources and formats (e.g. tabular, image, text, etc) so that the recommendation content can be modeled more expressively than traditional user-item interaction matrices.

One way to build content-based systems is to rely on latent feature models that map semantically rich features into numerical vectors. Such embeddings are expected to map similar items to nearby vectors; thus one can select items that are similar to any given item.

In this work we analyze graph-like approaches from the unified perspective of knowledge graphs and their related latent feature models, called knowledge embeddings. Both concepts are discussed in detail in Section 2.2.

2.2 Knowledge Graphs

In this work, we loosely follow the *RDF* notation [20], focusing on datasets where a triple representing a fact is written as $\langle h, r, t \rangle$ where h, r and t are, respectively, the *subject* *(head)*, *predicate (relation)* and *object (tail)*. A knowledge graph $\mathcal{KG}$ consists of the set of all entities $\mathcal{E} = \{e_1, \ldots, e_{N_e}\}$, and the set of relations $\mathcal{R} = \{r_1, \ldots, r_{N_r}\}$, where N_e and N_r represent the number of entities and relations in the knowledge graph, respectively. The existence of a triple $x_{h,r,t} = \langle h, r, t\rangle$ is indicated by a random variable $y_{h,r,t} \in \{0, 1\}$.

To illustrate, the statement:

> Jane Doe's parents, John and Mary Doe, are married to each other.

can be expressed via the set of facts represented by the triples in Table 1. We can combine all these triples into a knowledge graph (see Figure 1), where nodes represent entities and directed edges represent relationships between them. The direction of an edge indicates whether an entity stands as subject or tail in the given relationship.

triple
$\langle jane\_doe, parent, mary\_doe\rangle$
$\langle jane\_doe, parent, john\_doe\rangle$
$\langle john\_doe, spouse, mary\_doe\rangle$
$\langle mary\_doe, spouse, john\_doe\rangle$

Table 1: Sample knowledge base. Represent triples of type $\langle subject, relation, tail\rangle$. Analogous to the Figure 1 knowledge graph.

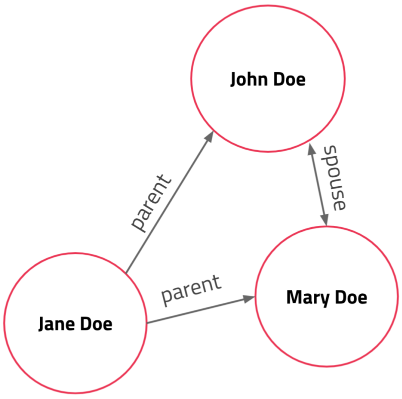


Figure 1: Sample knowledge graph.

Large-scale knowledge graphs (KG), such as Freebase [21], are often built by automatic knowledge base construction [22]. Automatic knowledge base construction aims at extracting factual information from unstructured or semi-structured textual data typically using natural language processing (NLP) techniques. A KG built automatically, although less dependent on human experts than curated KGs, has usually limited applicability due to missing or incorrect facts. The collaborative construction of KGs may also lead to missing or incorrect data that may be unknown by the volunteers filling the KG.

While existing triples encode known true or positive relationships between entities, there is no explicit representation for false relationships in a knowledge graph. For instance, the statement: "Jane Doe has a sibling called Bob" can be encoded by the triple $\langle jane\_doe, sibling, bob\_doe\rangle$. However the exact opposite, i.e. "Jane Doe **doesn't** have a sibling called Bob", typically, can only be efficiently encoded by the absence of the triple. Therefore, there is no clear distinction between false and unknown facts in a KG.

Indeed, there are two paradigms for the interpretation of non-existing triples: the *open world assumption* (OWA) or *closed world assumption* (CWA) [22]. The OWA considers a missing triple as simply unknown, whilst CWA considers missing facts as negative.

Knowledge graphs are often incomplete, i.e. many triples representing true relationships are missing. For instance, the place of birth attribute is missing for 71% of all people in Freebase in 2012 [23], a KG that was constructed collaboratively. Be it due to the limitations of KG building techniques in extracting all knowledge from unstructured sources or due to the inherent incompleteness of the source itself, it is hard to guarantee that all true facts are mapped and that the absent triples are composed by only false relationships.

It is really hard to assume closed-world for a large-scale KG, such as NELL [24], because it would imply an NLP technique able to capture all the knowledge textually described in the web and that the web has no missing information — two assumptions that are unlikely to hold. In this work we adopt the OWA for KGs.

Several approaches have been developed to address the task of *Knowledge Base Completion* (KBC), i.e. to distinguish between true and false facts among the unknown triples in a KG. The main assumption behind those methods is that it is possible to predict new facts from a statistical model based on existing facts (triples) [22]. There are two major approaches for KBCs: the first one focuses on observable graph features, while the second one works by converting semantically rich factual information into low-dimensional vector spaces. We now examine both approaches.

2.2.1 Graph Feature Models

These models aim at predicting new facts by extracting features from the observed graph [22]. To illustrate, consider again our example about Doe's family tree (see Figure 1). Assume we have a knowledge graph representing several family trees and we want to predict parenthood relationships. Imagine that the triple $\langle jane\_doe, parent, john\_doe\rangle$ is one of the missing facts in our KG and our task is to predict if John is parent of Jane.

Graph feature models leverage patterns from existing relationships in the KG to predict missing facts. In our toy example, it means to go through all other family trees in the graph and see what relationships typically connect people with their parents. For example, it reasonable to expect that married parents are a common pattern. Thus, the fact that John is the spouse of Jane's parent, Mary, suggests a parenthood relationship between them.

A popular graph feature model is the *Path Ranking Algorithm (PRA)* [25]. PRA searches for paths in the graph connecting entities whose relationship we are trying to predict. We say that path π_l is PRA-styled if it consists in a number l of arbitrary relations (directed edges of the graph) in the form $r_1 \rightarrow r_2 \rightarrow ... \rightarrow r_l$. In our example, the married parents pattern, *parent* → *spouse*, is a PRA-styled path of length 2 because it connects father and daughter in the graph.

A number of popular techniques based on graph feature models are inspired by the Path Ranking Algorithm (PRA) [25]. In particular the *Subgraph Feature Extraction (SFE)* [26] method has displayed promising performance and computational efficiency. SFE operates by performing random walks in the knowledge graph to extract PRA-styled features, to latter construct a feature matrix where the existence of each path is the feature value. Then, the feature matrix is used to train a classifier. The SFE classifier attributes a weight to each feature, corresponding to how relevant it is to the prediction task. To illustrate, it is expected that suggestive paths like the married parents pattern, *parent* → *spouse*, will receive a highly positive weight while spurious paths will be zeroed. The features extracted by SFE are taken to be easily interpretable [22].

Alternatively to the PRA-styled features, a more expressive approach, known as one-sided path features, consists in relaxing the condition where the path must connect both head and tail entities present in the predicted relationship. Consider the task of predicting if the triple $\langle e_h, r, e_t \rangle$ holds. While the path π_l needs to connect both e_h and e_t to be considered a PRA-styled feature, we might only require it to connect a single one of them to be a one-sided feature. To illustrate one-sided features expressiveness, imagine that our task now is to predict Jane's religion based on her family and other demographic attributes (e.g. gender, profession, etc). Now, lets focus on the profession attribute. We could model the correlation between religion and profession using either the PRA-styled feature $profession \leftarrow profession \rightarrow religion$ or the one-sided path $profession \rightarrow e_i$, where e_i represents any entity in the graph. While the first tells whether people with the same profession usually have the same religion, the later measure the correlation between people who have a specific profession e_i and their religion. While the profession information is irrelevant overall, it can be highly correlated with religion for a few particular cases. For instance, if we know in advance that Jane is an engineer, it does not tell much about her religion, however, if she is a nun, it is reasonable to predict her religion as catholic. Unlike the PRA-styled features that cannot differentiate professions with high correlation (e.g. nun, rabbi, pope, etc) from the overall, one-sided paths are expressive enough to capture these patterns.

Despite being valuable techniques in the field of knowledge base completion, graph feature models heavily rely on feature engineering and on intensive computing to operate. Besides PRA-styled and one-sided features, the literature is filled with other feature designs, each one more appropriate than the other depending on the specific domain. It is thus hard to generalize. Furthermore, the applicability of graph feature models is limited by the incompleteness of knowledge graphs. As we discussed, these models are intended to address incompleteness by predicting new facts; however, the models themselves rely on observable features from the original graph to make predictions. When the graph is severely incomplete, it is not unlike to find that most triples without a single feature value [27].

The performance of graph feature models has recently being surpassed by a new class of techniques based on knowledge embeddings. In the next section we will discuss how embeddings operate and their key limitations.

2.2.2 Knowledge Embeddings

Approaches based on knowledge embeddings (KE) now display state-of-the-art performance in knowledge base completion [28]. The main intuition behind embeddings is that interactions betweem latent features capture actual relationships [22]. Embeddings operate by learning such latent features from observed data and using them to infer missing facts in the original knowledge graph. To illustrate, consider again our knowledge graph representing family trees. We could encode the entities in the triple $\langle jane\_doe, parent, john\_doe \rangle$ using a bi-dimensional real-valued embedding $\mathbf{e}_i \in \mathbb{R}^2$ where the two latent features are "*kin*" and "*age*", as follows:

$$\mathbf{e}_{jane} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}, \mathbf{r}_{parent} = \begin{bmatrix} 0.0 \\ 0.2 \end{bmatrix}, \mathbf{e}_{john} = \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix}. \tag{2.1}$$

Here, we say the first dimension of the embedding corresponds to "kin" while the second is directly related to "age". In our toy embedding model, we can tell that both Jane and John are of the same kin since their embedding first dimension has the same value 0.1. In addition, by looking at the other dimension, we can also say that John is older than Jane ($0.5 > 0.2$). We say that kin and age are latent because they cannot be directly observed from data (there is no explicit relation in the KG representing them).

Note that unlike our toy example, embedding dimensions typically do not carry explicit meaning and are hard to interpret.

Because modeling techniques can be both diverse and complex, many embeddings have been proposed, each one with distinct characteristics [29, 30].

Typically, a model based on an embedding defines a particular scoring function $f_r(h, t \mid \Theta)$ to measure the plausibility of fact $\langle h, r, t \rangle$, where Θ is a set of parameters or dimensions. The greater the plausibility score for a given fact, the more likely it is to hold. For example, in our toy example, we might infer parenthood relationships by looking at people with the same kin but older. This simple rule can be modeled as a linear translation of the age dimension. For instance, to tell whether John is Jane's parent, we can increase the Jane's age dimension while maintaining her kin ($\mathbf{r}_{parent} = [0.0, 0.2]$) and see if John's embedding is anywhere near, i.e. $\mathbf{e}_{jane} + \mathbf{r}_{parent} \approx \mathbf{e}_{john}$. If we apply euclidean norm as distance measure, this translational inference model can be described by the plausibility function $- \parallel \mathbf{h} + \mathbf{r} - \mathbf{t} \parallel_{\frac{1}{2}}$, which corresponds to the TransE [29] model. There we have:

$$\begin{aligned} f_{parent}(jane, john \mid \Theta) &= - \parallel \mathbf{e}_{jane} + \mathbf{r}_{parent} - \mathbf{e}_{john} \parallel_{\frac{1}{2}} \\ &= - \left\| \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.2 \end{bmatrix} - \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix} \right\|_{\frac{1}{2}} \\ &= -\sqrt{(0.1 + 0.0 - 0.1)^2 + (0.2 + 0.2 - 0.5)^2} \\ &= -0.1. \end{aligned}$$

Embedding models build the latent features through an optimization process that maximizes the total plausibility of all known facts. It thus has to find, for each entity, the optimal embedding representation that leads to a positive inference for all (or at least most) triples in the KG. The plausibility score should be a measure of how likely a given triple is to hold; however, by looking at the value alone we miss a comparison scale to tell whether a fact has high or low plausibility. For example, we can tell that the plausibility of John being Jane's parent is -0.1, but that does not answer whether John is indeed Jane's father. One approach to effectively answer that question is to fit a plausibility threshold δ_r, specific for each relation r, that represents the minimum plausibility score for a triple to hold or to be told true; i.e. $\mathit{holds}(h, r, t)$ when $f_r(h, t_N \mid \Theta) > \delta_r$. For example, if we observe on a validation set that parenthood relationships that presents plausibility greater than -0.15 are likely to hold, we can infer that John is Jane's father. This inference procedure is known as triple classification (TC).

Despite being originally proposed for knowledge base completion, we should note that embeddings are also used to produce recommendations [19, 31] and to answer questions [32]. While triple classification is enough for tasks related to knowledge base completion, these other applications rely on plausibility ordered entity rankings. For example, consider a system that employs a knowledge embedding to recommend films. Even if the embedding is accurate at triple classification, i.e. telling whether a given user would like a movie or not, the system is more concerned if the embedding is able to rank order among the ones the user will enjoy the most. This task of rank ordering entities accordingly to the plausibility score is known as link prediction (LP) or entity ranking.

So far we have discussed inference mechanisms over knowledge graphs and their embeddings. As we mentioned, embeddings operate on a latent-feature space to make inferences. Because the latent features or embedding dimensions are specified as a result of an optimization process and carry no explicit meaning, the plausibility function or the embedding itself is a rather opaque model.

2.3 Interpretability

Opaque models, such as the ones produced by embeddings, create obstacles to the interpretability of recommendations [33]. Here we take interpretability as the degree to which a human can understand the cause of a decision [9]. 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.

An explanation can be perceived as the answer to a "why" question [9]. From this definition, the act of explaining takes the form of a social interaction between at least two agents: the explainer and the explainee. The explanation model proposed by Miller (Figure 2) argues that the explanation is a compound process with two elements, the cognitive and the social. While the first describes the process of identifying the causes why a given decision was made, the second is the process of conveying or communicating such reasons to the explainee. We now examine both processes.

2.3.1 The Cognitive Process

Several techniques have been developed to shed light on opaque models by generating explanations. These techniques can be divided in two groups: model-specific and model-

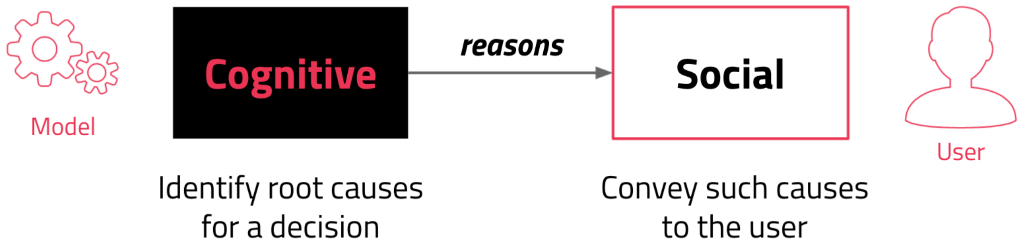

Model
Cognitive
reasons
Social
User
Identify root causes for a decision
Convey such causes to the user

Figure 2: Visual representation of the explanation processes.

agnostic. The first one is bound to specific classes of models; these techniques usually have access to internal or structural information about the explained device. On the other hand, the model-agnostic techniques can be applied to any machine learning model, as they consider the explained device as a black-box, i.e. do not make any assumptions about its internal behavior.

A popular model-agnostic approach is the construction of interpretable surrogate models. These surrogate techniques vary in scope: some aim to explain the model of interest as a whole (a *global* or *holistic* approach) while others focus on a single or a set of predictions.

2.3.1.1 Global Surrogate

The global surrogate technique consists of training an intrinsically interpretable model using the black-box predictions as ground truth, so that the global surrogate mimics the black-box model. If the global surrogate is intrinsically interpretable, explanations about the black-box model can be drawn from it. However, because the black-box model is presumably complex, it is not to be expected that another model (the global surrogate) can mimic its behavior in a faithful manner while remaining simple and interpretable; Figure 3 depicts the global surrogate method in high-level. Also, if a faithful interpretable model is achieved, one could ask why keep the black-box model itself.

2.3.1.2 Local Surrogate

One popular model-agnostic strategy consists of reducing the scope of the surrogate model [1]. The interpretable model is then expected to mimic the black-box behavior only partially.

The main intuition behind local surrogates is that an interpretable and simple model should be faithful to a complex model at least locally. Here, intuitively, the simple model extracts the part of black box behavior that explain the particular prediction in hand. So,

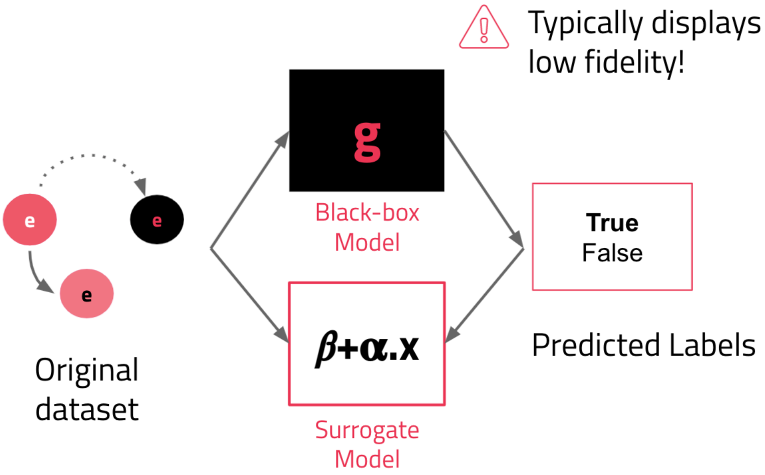


Figure 3: Global surrogate high-level schema.

even though each local-surrogate is only able to explain a single or a few cases, their value consist in being able to faithfully untangle the complex black box and summarize only its behavior of interest. For instance, suppose that one intends to use a local surrogate to explain a single prediction of a black-box. First, the input data of interest is perturbed, generating a set of variations of the original data (a *"neighborhood"*). This set of data points around the original input are then fed to the black-box model that provides labels. Finally, an interpretable surrogate model is trained with the data points around the original input and their respective labels given by the black-box. Figure 4 depicts the local surrogate method in high-level.

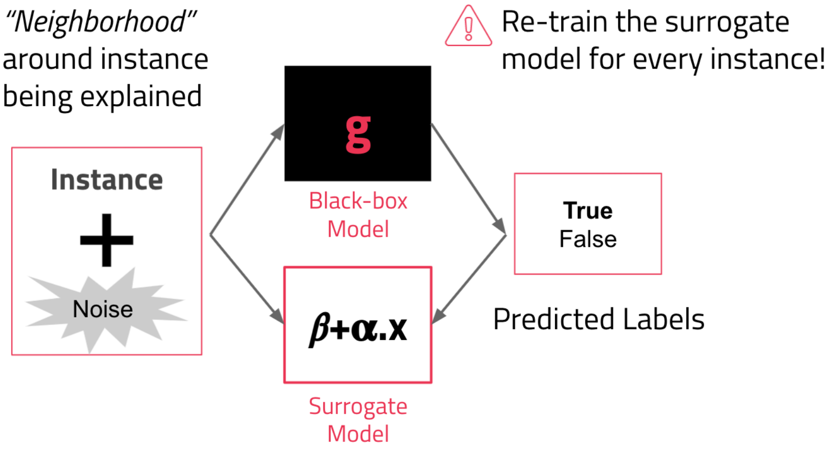

"Neighborhood" around instance being explained
Instance
+
Noise
Re-train the surrogate model for every instance!
g
Black-box Model
β+α.x
Surrogate Model
True
False
Predicted Labels

Figure 4: Local surrogate high-level schema.

Often, local surrogate models with complexity constraints are defined as follows:

$$explanation(x) = \arg\min_{g \in G} L(f, g, d_x) + \Omega(g). \tag{2.2}$$

The explanation given for the instance x is the interpretable model $g* \in G$, where G

is the set of all possible models that minimizes the loss function L and the complexity constraint Ω. The loss function measures the unfaithfulness of the surrogate model g to the black-box function f considering the neighborhood around instance x limited by the distance parameter d_x. The complexity function Ω balances the trade-off between interpretability and fidelity; it may be for instance a measure of model sparsity.

It is worth noting that if Expression (2.2) covers the entire training set instead of the neighborhood d_x, the resulting model $g*$ corresponds to a global surrogate.

One of the most popular local surrogate techniques is LIME [1]. Roughly speaking, LIME runs a sensitivity test around the instance of interest, then it presents as explanations the most significant features for each label. Even though it is effective in producing explanations virtually for all kinds of data, e.g. tabular, textual or visual data, LIME is limited to non-relational classifiers. LIME explains based on the assumption that features themselves are interpretable, a weak assumption in the context of embeddings. For instance, LIME, when applied to a binary text classifier, produces explanations such as "Word XYZ is significant for the prediction"; similarly, when applied to embeddings, LIME explanations would be such as "Dimension 123 is significant for the prediction" — a sentence that is hard to interpret because the dimensions are latent features and, thus, convey no meaning for users.

The dimensions of an embedding are part of the underlying structure of the model, so explanations should rely only on features from the semantic field instead. Multi-relational classifiers require techniques for mapping real-valued latent features to the semantic field in order to produce human-friendly explanations.

Furthermore, local-surrogate based methods are often time expensive because they demand a new interpretable model to be trained from scratch to explain a single decision. The operations for generating the *neighborhood* around the instance to be explained and, consecutively, for training the surrogate model represent an overhead that may be unfeasible for domains that require low response time, notably online and interactive applications. For instance, Listwise Explainer (LISTEN) [6], which is based on LIME, explains rankings faithfully by training an local-surrogate model – similarly to LIME [1]. Despite promising results, LISTEN is not suitable for explanation generation at scale in real-time environments due to the high computational cost at online training a surrogate model for each recommendation. Alternatively, 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.

2.3.2 Social Processes in Explanation

The techniques we discussed earlier in the Section 2.3.1, alongside with most of the research work available in the literature, focus on the cognitive step of the explanation generation but do not completely address the underlying social process involved. For instance, LIME claims to be able to explain the predictions of any classifier to lay users, however it does not fully consider to whom its explanations are directed or in what contexts. LIME depicts its explanation as an array of weighted features in a visual form. While this format may be appropriate for a machine learning practitioner debugging a model offline, it may not provide all the details needed for instance by a regulator.

The social process includes the act of explaining as the interaction between agents unfolds; thus, while designing an explanation, it is important to take into account what is the target audience, the communication channel where the interaction takes place and which social roles these agents (the explainer and the explainee) play. To address these questions, we examine below the explanation conversational model and briefly discuss some ethical concerns related to the social process.

2.3.2.1 Conversational Model

The conversational model of an explanation [10], which the Miller framework is built upon, argues that an explanation is inherently a form of dialogue, thus it is tied to the rules of a conversation as stated in Grice's maxims [34]. In short, to say an explanation is conversational means that it is faithful to the explainer beliefs about what caused the given decision, it is relevant, simple and presented in a meaningful manner to the explainee. It is worth noting that even though the term "conversational" may suggest natural language, this model is not restricted to it.

Further extensions of the explanation conversational model incorporate argumentation in the explanatory dialogue [35–38]. These works claim that explanations are not only intended to communicate the causes behind a given decision but also to justify why it is the correct one. Notably, when the explainer is the same agent who made the decision or is bounded to it, the explanation is likely to be designed so that it presents reasons supporting its own claims.

This argumentative model highlights that the explainer and the explainee might disagree or have conflicting interests about the decision being explained, so that the explanation serves to defend one or another claims. Moreover, one could argue that the persuasive property of explanations might be exploited by the explainer to convince the explainee even when the decision should in fact be refuted; indeed, this is a concern we discuss in the next section.

2.3.2.2 Responsible Explanations

In Section 2.1, we discussed that interpretability, and consequently explanations, are desirable when it is necessary to establish a trust relationship between an AI system and its end user. However, it is hard to expect any user would trust a system whose explanations are designed to persuade her into agreeing with its decisions. The system would be like the salesperson who always proposes products with complimentary words, as opposed to the salesperson who frankly discusses the advantages and disadvantages of products. We conjecture that a perceptive customer will gradually favor a salesperson who chooses sincerity over persuasion. Indeed, in the marketing and retail literature, empirical studies [39–42] suggest that negative reviews tend to be more effective than uniformly positive ones.

The conversational model for explanations does offer useful information to the user; however, we argue that it runs into a difficult balancing act [43]. For instance, lets assume that while the company's interest lies in increasing its own sales, the client wants to make the best purchase. The company and the client goals may oppose each other if no product available fits the client interests because the best purchase would be not to buy anything at all. In such a scenario, if the explanation is designed to persuade the client to buy a product that does not fit her interests, it potentially leads to mistrust and, thus, jeopardize the explanation's own purpose. For example, in retail industry, there is evidence that customers do trust more online reviews from other buyers than seller's statements [44].

We refer to *responsible* explanation generation techniques as those techniques that avoid mistrust due to conflicts of interest. In Section 3.3 we address this issue with a novel proposal.

2.4 Related Work

In this section we investigate the literature on interpretability of knowledge graph embeddings and on recommendation systems. We first discuss the insights and limitations of explanation generation techniques focused on knowledge embeddings and, then, we address the work on explainability of recommendation systems related to knowledge graphs. Finally, we sum up the limitations of the most relevant references in a qualitative comparative table and identify a research gap.

2.4.1 Interpretability of Knowledge Embeddings

Many proposals related to the interpretability of knowledge embeddings follow the model-specific approach, e.g. ITransF [45], SimplE [46] and CrossE [47].

SimplE asssumes that each dimension of an entity embedding can be considered a feature and the corresponding element of a relation representation is a measure of how important that feature is to the relation. Even though this characteristic provides a certain degree of transparency, it does not seem to be really interpretable. The model allows one to include background knowledge into the embeddings, however since its interpretability focuses on embedding dimension, it is not possible to drawn meaningful explanations for its predictions.

Similarly to SimplE, ITransF also deals with interpretability on the latent feature level. ITransF proposes a sparse attention mechanism to represent shared concepts among relations. For instance, both relations “nominated for” and “honored for” represent a concept of high quality work even as they are distinct. The attention mechanism of ITransF allows the identification of latent features, or concepts, however as these features are given in the embedding level, they are not really interpretable. Back to our example, even though we identify a strong link between the relations “nominated for” and “honored for”, it is hard to infer what is the actual concept shared. Despite SimplE and ITransF techniques provide a certain degree of interpretability, as their insights are mostly related to the embedding internal structure and are given in terms of real-valued vector or heat maps, their explanations are only understandable by data scientists or by experts.

On the other hand, CrossE exploits a particular type of interaction between relations and entities called *crossover interactions* (CI) to explain embedding predictions in the semantic field instead of the real-valued one. For instance, suppose one has to explain the triple $\langle person\, X, isFatherOf, person\, Z\rangle$. An explanation that supports this triple could be the path $\xrightarrow{\text{hasWife}}\xrightarrow{\text{hasChild}}$ connecting *head* and *tail* entities. Despite being highly interpretable [26] and faithful, once these support paths are reconstructed considering not only the embedding, but also the knowledge graph, which is often incomplete, CrossE method cannot explain all predicted triples; it is also not affected by negative instances, i.e. cannot explain why a given triple is *not* true. Furthermore, 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 [27] [26].

In contrast to the model-specific approaches described previously in this section, the XKE method [27] is model-agnostic. XKE consists of training a global surrogate logistic regression on SFE graph features while using the embedding labels, so that explanations can be drawn from the interpretable classifier. Even though XKE is easy to interpret, it displays relatively low fidelity [27].

2.4.2 Interpretability of Recommendation Systems

Apart from more traditional surrogate techniques like LISTEN (discussed in Section 2.3.1.2), there are recommendation systems that rely on large-scale knowledge graphs for explanation generation; for instance, ExpLOD [4] and ASEMF_UIB [8]. Both employ semantic information about items to find similarities between user profiles (e.g., previously liked items). For example, consider that an 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 completeness to work properly, a strong assumption considering the incompleteness of large-scale knowledge graphs [22, 23]. If the graph does not contain the fact that "James Cameron" is the director of "Titanic", the recommendation system may fail to produce an explanation.

Both ExpLOD and ASEMF_UIB are based on an external source of knowledge, i.e. knowledge graphs, to produce explanations and, thus, do not make any assumptions about the RS internal mechanisms. These approaches are model-agnostic but may not be faithful if the knowledge graph used to explain is not also being considered by the recommendation system.

2.4.3 Research Gap

Table 2 highlights a research gap: explanation methods that are suitable for interactive, real-time settings, that display high coverage, that are both model-agnostic and faithful. This is exactly where the contributions in this work fit in. Also, note that none of the related work provides *responsible explanations* as they do not discuss the social aspects that impact trust in RSs.

Table 2: Comparative summary of related works

Method	Lay User	Model-Agnostic	Faithful	Real-Time	High Coverage
SimplE [46]	No	No	-	No	Yes
ITransF [45]	No	No	-	No	Yes
CrossE [47]	Yes	No	Yes	Yes	No
XKE [27]	Yes	Yes	No	Yes	No
LISTEN [6]	Yes	Yes	Yes	No	Yes
ExpLOD [4]	Yes	Yes	No	Yes	No
ASEMF_UIB [8]	Yes	Yes	No	Yes	No

3 PROPOSAL

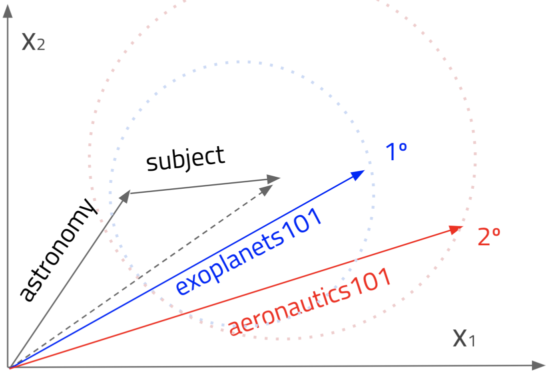
"To achieve great things, two things are needed:
a plan, and not quite enough time."

-- Leonard Bernstein

Before we examine our proposals, consider the following abstract recommendation system based on knowledge graph embeddings. First, while interacting with the recommendation system, the user must inform her preferences. Suppose that these preferences are related to an entity e_h in an available large-scale knowledge graph and that the KG is built so that the edges of e_h constitutes known interactions between items with the given preference. For instance, we should find in the KG triples $\langle e_h, r, e_t \rangle$ that links items of interest to user preferences. Assume that our base recommendation system runs link prediction using an embedding (built from the same graph) and returns the Top-N ranked entities as recommendations. So, in our set up, the recommended items are the entities in the graph that yields the highest plausibility score $f_r(e_h, e_t \mid \Theta)$. This is a conceptual scheme that corresponds to the vast majority of recommendation procedures.

We propose the following use case, consider a recommendation system that suggests a ranked set of University disciplines to a student based on her informed subject of interest. In our use case, a University proposes this recommendation system to support the enrollment process, the goal is to help students find disciplines that teach a given subject. To illustrate the assumed recommendation mechanism, suppose class Exoplanets101 is recommended to a student whose preference lies in astronomy. In this toy example, $\mathbb{T}$ is the list containing plausibility values for all entities in $\mathcal{E}$. We sort $\mathbb{T}$ in descending order, and identify that Exoplanets101 is more related to astronomy than Aeronautics101 and so on. Figure 5a presents this toy example in a bi-dimensional TransE embedding, where the recommended items are ranked by their proximity to the translated vector. Note that the recommendation mechanism in Equation 5b is agnostic to any embedding model since it only assumes a generic plausibility function.

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 "subject".



$$\mathbb{T} = \begin{bmatrix} f_{subject}(e_a, e_1 \mid \Theta) \\ \dots \\ f_{subject}(e_a, e_m \mid \Theta) \end{bmatrix}$$

(a) Bi-dimensional TransE embedding.

(b) Ranking mechanism via plausibility score.

Figure 5: Abstract recommendation mechanism toy example.

We now present three interpretability methods. These proposals start from the abstract recommendation mechanism described previously.

3.1 Faithful Explanations through Local Surrogates

Our first proposal is a novel model-agnostic explanation method for knowledge embeddings inspired by the local-surrogate approach adopted by LIME [1]. We address a series of challenges due to the complex nature of embedding techniques and provide a method to effectively produce faithful explanations for link predictions.

Explanations drawn from local surrogates usually are given in the form of weighted features, which implies that the feature itself must be meaningful to the user. Even though this is true for most traditional classifiers, in some cases, e.g. knowledge embeddings, the features considered by the model to realize their predictions are too complex to be understandable or bear no explicit meaning for the target audience. In short, latent features or embedding dimensions are inappropriate for human-friendly explanations. Features in the explanations need to be different from the features (real-valued vectors) used by the knowledge embedding.

To address this issue, we argue that the knowledge embedding itself should be used to extract interpretable representations for entity embeddings, so that we can generate meaningful explanations while remaining faithful to the model. This feature extraction procedure is described below, but first consider some important definitions.

A knowledge embedding, for the top-1 tail prediction task, can be defined as a set of black-box classifiers $g_r \in \mathcal{G}$, one for each relation $r \in \mathcal{R}$, where $g_r(h|\Theta)$ returns the tail entity $e_t \in \mathcal{E}$ that gives the greatest plausibility score f_r for the triple $\langle e_h, r, e_t \rangle$. That is,

$$g_r(h|\Theta) = \arg\max_{e_i \in \mathcal{E}} f_r(e_h, e_i \mid \Theta). \tag{3.1}$$

As there exists a real-valued vector representation for each entity $e_i \in \mathcal{E}$ in the knowledge embedding parameters Θ, we can define the classifier function g_r so that it takes as input the head entity embedding. That is,

$$g_r(\mathbf{e}_h) = \arg\max_{\mathbf{e}_i \in \Theta} f_r(\mathbf{e}_h, \mathbf{e}_i). \tag{3.2}$$

We have defined $g_r(\mathbf{e}_h)$ as a classifier that takes the head entity embedding $\mathbf{e}_h$ (or tabular data) and outputs the most plausible tail entity (or label), thus g_r and a tabular data traditional classifier are alike.

Example 1. *To illustrate our definitions, consider the toy example where we are interested in the tail prediction for the triple* $\langle astronomy, topic\_of\_class, ? \rangle$*. Let us define:*

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

$$sort\_desc(\mathbb{T}) = \begin{bmatrix} f(e_{astro}, e_{exoplanets101}|\Theta) \\ f(e_{astro}, e_{aeronautics102}|\Theta) \\ \vdots \\ f(e_{astro}, e_m|\Theta) \end{bmatrix}, \quad g_{topic}(astro|\Theta) = exoplanets101. \tag{3.3}$$

The list $\mathbb{T}$ *represents the plausibility score calculated for all entities. Thus, to discover the most plausible candidates for courses about astronomy, we sort* $\mathbb{T}$ *in descending order and identify that the class most presumably related is exoplanets101, then aeronautics102 and so on. Our function* g_r *returns only the top 1 ranked entity, in this case exoplanets101.*

At this point, we should be able to train a local surrogate to g_r. However, as its input is given in terms of latent features, i.e. embedding dimensions, we still cannot extract meaningful explanations. Thus, to proceed we need to answer the following questions:

1. **Q1**: What should be considered an interpretable representation for embeddings?

2. **Q2**: How to extract these interpretable representations from the embedding space?

To answer the first question we take graph feature models as alternatives [26]. Although other feature types could be used, e.g. PRA-style features, we opted for one-sided path features due to their simplicity. As adopted in Section 2.2.1, the interpretable representation provided by one-sided path features for an certain entity e_h is simply $\phi_h = [e_\pi : \pi \in \Pi_L]$. Note the parameter L, which represents the maximum number of relations in the path, enforces a complexity constraint, because it limits both the path's maximum length and the number of features.

Example 2. *For instance, consider the comparison between the interpretable representation of astronomy ϕ_{astro}, and its real-valued vector $\mathbf{e}_{astro}$:*

$$\Pi_2 = \begin{bmatrix} topic\_of\_class \\ difficulty \\ \vdots \\ from\_category \end{bmatrix} : \phi_{astro} = \begin{bmatrix} exoplanets101 \\ hard \\ \vdots \\ astrophysics \end{bmatrix}, \mathbf{e}_{astro} = \begin{bmatrix} 0.9 \\ 1.2 \\ \vdots \\ 0.1 \end{bmatrix}. \qquad (3.4)$$

In order to answer the second question, we propose to use the knowledge embedding itself to extract the graph features. Formally, a path π_l is a sequence of relations $\{r_1, r_2, ...r_L\}$. Since each path π_l consists of a sequence of relations $\xrightarrow{r_1}\xrightarrow{r_2} ... \xrightarrow{r_l}$, where $r_i \in \mathcal{R}$, we can use our classifiers g_r (the knowledge embedding itself) to extract the interpretable features for a given entity embedding.

Example 3. *Back to our example, the embedding feature extraction for the compound feature category – topic of the entity astronomy. First we discover the category of astronomy using the function $g_{category}$, then we inquire for her classes using g_{topic}. That is,*

$$g_{category}(astro|\Theta) = astrophysics \rightarrow g_{topic}(astrophys|\Theta) = mechanics101. \qquad (3.5)$$

Once we know how to map the embeddings to their interpretable representations, we are ready to proceed. Suppose we wish to explain why e_t is a plausible tail entity for the triple $\langle e_h, r, ?\rangle$. First, we sample K data points around $\mathbf{e}_h$, similarly to LIME applied to tabular data [1], thus generating a dataset $\mathcal{Z}$ of perturbed samples $\hat{\mathbf{z}}_k$. It is worth to mention that unlike LIME, we sample around the input original representation, instead of its interpretable one.

Next, for each perturbed sample $\hat{\mathbf{z}}_k \in \mathcal{Z}$ we realize the feature extraction procedure previously described. That is,

$$\phi_k = [e_\pi : g_\pi(\hat{\mathbf{z}}_k), \pi \in \Pi_L]. \qquad (3.6)$$

As a result of the previous step, we have the interpretable representation $\phi_k \in \Phi$ for each perturbed sample $z \in \mathcal{Z}$. Finally, we train a intrinsically interpretable classifier, such as a sparse logistic regression (SLR), $g'_r \leftarrow SLR(\Phi)$ and draw explanations from it in terms of feature importance, e.g. the top n high-valued coefficients (Algorithm 3.1).

Algorithm 3.1 Local-Surrogate Explanation Generation

```
procedure EXTRACT-FEATURES($\hat{\mathbf{z}}_k$: perturbed sample, $\Pi_L$: explanatory path set)
    $\phi_k = \{\}$
    for all $\pi \in \Pi_L$ do                              ▷ Equivalent to Equation (3.6)
        $e_\pi \leftarrow \hat{\mathbf{z}}_k$
        for each edge $r_j \in \pi$ do
            $e_\pi \leftarrow g_{r_j}(e_\pi)$
        $\phi_k \leftarrow \phi_k \cup e_\pi$
    return $\phi_k$
procedure EXPLAIN-INSTANCE($x_{h,r,t}$: query, $L$: path length, $\Theta$: embedding)
    $\langle h, r, t \rangle \leftarrow x_{h,r,t}$
    $\mathbf{e}_h \leftarrow h|\Theta$                                 ▷ Retrieve embedding for head entity
    $\Phi \leftarrow \{\}$
    $\Pi_L \leftarrow$ GRAPH-FEATURES($L$)                     ▷ Generate set of path features
    for $k \in 1, 2, 3, ..., K$ do
        $\hat{\mathbf{z}}_k \leftarrow$ SAMPLE-AROUND($\mathbf{e}_h$)                ▷ Generate perturbed sample
        $\phi_k \leftarrow$ EXTRACT-FEATURES($\hat{\mathbf{z}}_k, \Pi_L, \mathcal{G}$)
        $\Phi \leftarrow \Phi \cup \langle \phi_k, g_r(\hat{\mathbf{z}}_k), d(\mathbf{e}_h, \hat{\mathbf{z}}_k) \rangle$
    $g'_r \leftarrow SLR(\Phi)$   ▷ Train interpretable classifier with $\phi_k$ as features and $t$ as target
    Draw explanations from $g'_r$ in terms of feature importance
```

3.2 High Coverage Explanations through Embeddings

Even though the method described in the previous section is able to provide faithful explanations for a recommendation system based on knowledge embedding, it is expensive in terms of computational cost. As we discussed in Section 2.3.1.2, local-surrogate models based on LIME require a new model to be trained for every explanation, typically requiring a prohibitive overhead in online applications.

Now we examine a simplification of the previous approach that produces faithful explanations in a time scale feasible for interactive applications. In addition, this second method enhances the interpretability techniques discussed in Section 2.4.2 by addressing the incompleteness of large-scale knowledge graphs.

Now, we take an explanation to be a PRA-styled 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 explanatory paths $\pi \in \Pi_L$ that, if found, are taken to be explanations. We assume that such a set is specified by declaring the sequences of relations that are permissible. It is important to adequately specify Π_L because there might exist paths that do not provide any sensible explanation, 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 [26].

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. In any case, we assume that Π_L is available.

To generate an explanation, 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 taken 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 knowledge embedding. With triple classification (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 link prediction, 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 DFS. This procedure is similar to the one described in Section 3.1, with the difference that previously we used one-sided paths as features and now we consider PRA-styled ones.

Again, we formally define a path as 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 $e_i \in \Omega$, each triple $\langle e_{i-1}, r_i, e_{i+1} \rangle$ holds. We consider that a triple holds if the plausibility score for it is greater than a given threshold δ_r, the same as in Triple Classification task. 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 the 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 6. 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 $e_h \rightarrow e_2 \rightarrow e_t$.

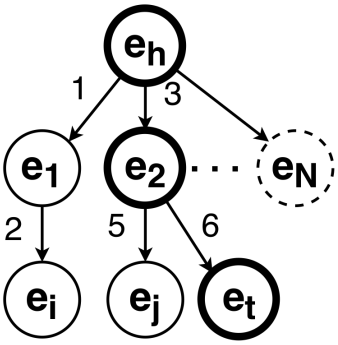


Figure 6: Depth-First Search toy example.

In short, in our first proposed method we had to train a surrogate model and compute feature importance to know which features could explain the knowledge embedding prediction. Now we assume that the important set of features Π_L is available beforehand and, thus, avoid the step of re-training the surrogate model for every instance. To illustrate the final result, consider the explanation example depicted in Figure 7; it tells us that Exoplanets101 is recommended as it is about Exoplanets, which is a topic of Astronomy.

3.3 Responsible Explanations through Reasons-Theory

At this point, we have a model-agnostic method that generates faithful explanations (i.e. grounded on embedding inference mechanism) in acceptable response times while keeping high coverage (i.e. able to explain most cases). However, these explanations only offer justifications of the system choices. Because they do not shed light on the possible downsides of recommendations, a perceptive user may feel that the explanations were designed to solely persuade her into agreeing with the recommendation system choices, as we discussed in Section 2.3.2.2. We claim that both reasons *for* and reasons *against* should at least be explicitly presented to the user if we expect her to trust the suggestions.

So far we have been discussing proposals more tied to the cognitive process of an explanation, i.e. focused on extracting reasons why the system predicted its recommendations. Now, we enter in the social process of the explanation whose input are the reasons collected in the cognitive step.

Suppose a system suggests item e_t for the user preference e_h. We define as γ the function that starts with the knowledge embedding parameters Θ and the path π, takes inputs e_h and e_t, and returns a set of reasons for the recommendation of e_t to e_h. The function γ represents the cognitive process of an explanation and, in the context of this work, it is the DFS algorithm described in the previous Section 3.2.

We thus 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 [7], a philosophical study of reasons-theory.

Snedegar presents five schemes by which reasons *against* can be generated by an agent contemplating competitive options:

1. (S1): a reason against an item A is a reason for a competing option;
2. (S2): a reason against an item A is only a reason for NOT A (not for any particular other option);
3. (S3): a reason against an item A is just a reason for the disjunction of the other options (say $B \vee C \vee D$);
4. (S4): a reason against an item A is a reason for all of its alternatives.

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 translate them to a concrete level. We present our implementations in the remainder of this section.

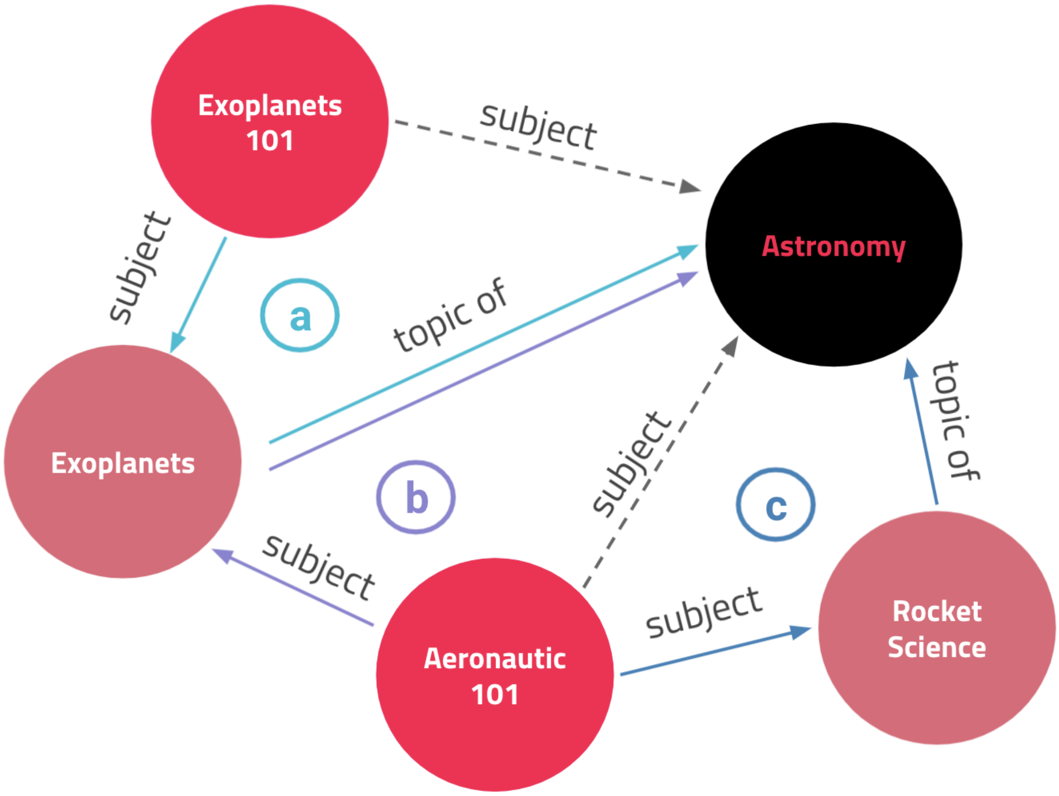


Figure 7: Sub-graph highlighting reasons for Exoplanets101 (**a** turquoise) and Aeronautic101 (**b** purple and **c** dark blue). For example, the turquoise path $Exoplanets101 \xrightarrow{subject} Exoplanets \xrightarrow{topic\_of} Astronomy$ is a reason for Exoplanets101.

Our implementation of S1 generates a reason against a given item by generating reasons for other options. For instance, take the case where the system has recommended two courses — Exoplanets101 and Aeronautics102 — as in Figure 7. A reason against Exoplanets101 then would be that Aeronautics102 is about “Rocket Science”. The intuition behind S1 is similar to the concept of opportunity cost. If you choose Exoplanets101 instead of Aeronautics102, you will miss the opportunity to learn about rocket science.

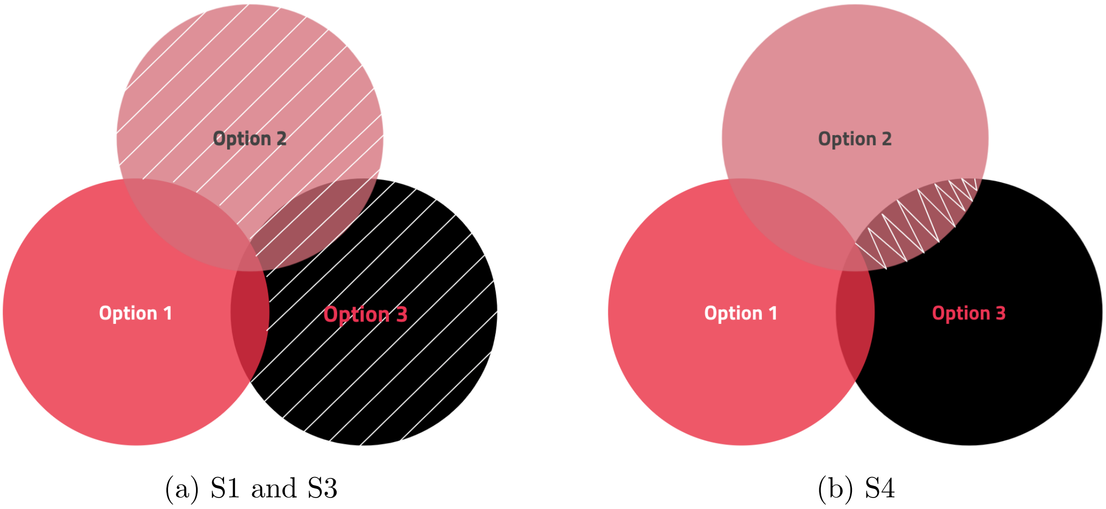
Scheme S2 is more delicate: how to define the negation of an item in the context of recommendations? The vague nature of this question led us to skip this scheme.

[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 third recommended course Astrobiology101 and as reasons against Exoplanets101 we have that Aeronautics102, Astrobiology101 or both of them are about rocket science. In practice, in our approach both 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 Exoplanets101 using S4 would be that both Astrobiology101 and Aeronautics102 from the example above are about rocket science. The stringent nature of this scheme, where the intersection of reasons is required, makes it hard to generate reasons against in practical circumstances.

To better illustrate the differences between S1, S3 and S4, consider a toy example where a system recommended three options (1, 2 and 3) and produced multiple reasons for each one of them, as showed in Figure 8. Note the sets of reasons for each option are not necessarily disjoint, so intersections are possible, i.e. the same reason for may be applied to multiple options. For S1, the reasons against option 1 are the union of the reasons for its alternatives (option 2 and 3) that are not reasons for itself, which is represented by the dashed area in Figure 8a. On the other hand, for S4, it is the intersection, as presented in the Figure 8b.



(a) S1 and S3

(b) S4

Figure 8: Venn diagram representing the set of reasons for each option. Reasons against option 1, according to schemes S1, S3 and S4, are dashed.

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 toy example that the user has the objective of learning about exoplanets; with that piece of information, the RS can present the user with the reason against choosing Aeronautics102 because it is less related to exoplanets than Exoplanets101, even though Aeronautics102 addresses the subject marginally. The implementation of S5 is quite different from the other schemes; we examine it in detail later in this section.

To illustrate the implemented algorithms for S1, S3 and S4, suppose an RS recommended N items in an ordered set $\mathcal{I} : \{i_1, i_2, ...i_N\}$ to user u. In scheme S1 (and S3) we define as reason against an item i_r the union of reasons for each of its alternatives $\mathcal{I} \backslash \{i_r\}$ that are not reasons for i_r itself. Hence we must iterate over the alternatives, extracting reasons for each one of them $\Phi \leftarrow \Phi \cup \Phi_{u,i}\, \forall i \in \mathcal{I} \backslash \{i_r\}$. Note that at this point we assume that the function representing the cognitive process of explanation γ, as described earlier in this section, 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 3.2.

Algorithm 3.2 Explanation Generation using Scheme S1

```
 1: procedure REASONS-FOR($i$: rec. item, $u$: user, $\Pi$: paths, $\Theta$: parameters)
 2:     $\Phi_{u,i} = \{\}$                                  ▷ Set of reasons for $i$
 3:     for all $\pi \in \Pi$ do
 4:         $\phi \leftarrow \gamma(u, i, \pi | \Theta)$         ▷ Function describing the cognitive process
 5:         $\Phi_{u,i} \leftarrow \Phi_{u,i} \cup \phi$
 6:     return $\Phi_{u,i}$
 7: procedure REASONS-AGAINST-S1($i_r$: rec. item, $u$: user, $\mathcal{I}$: rec. set, $\Pi$: paths)
 8:     $\Omega_{u,i_r} \leftarrow \{\}$                         ▷ Set of reasons against $i_r$
 9:     $\Phi = \{\}$
10:     $\Phi_{u,i_r} \leftarrow$ REASONS-FOR($i_r$, $u$, $\Pi$, $\Theta$)     ▷ Set of reasons for $i_r$
11:     for $i \in \mathcal{I} \backslash \{i_r\}$ do                   ▷ Iterate over $i_r$ alternatives
12:         $\Phi_{u,i} \leftarrow$ REASONS-FOR($i$, $u$, $\Pi$, $\Theta$)
13:         $\Phi \leftarrow \Phi \cup \Phi_{u,i}$
14:     $\Omega_{u,i_r} \leftarrow \Phi \backslash \Phi_{u,i_r}$
15:     return $\Omega_{u,i_r}$
```

Regarding the implementation of Scheme 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 13 of Algorithm 3.2 so as to take the intersection of sets $\Phi \leftarrow \Phi \cap \Phi_{u,i}\, \forall i \in \mathcal{I} \backslash \{i_r\}$.

In the remainder of this section we discuss Scheme S5.

While all the other schemes are implemented by modeling reasons as an unweighted directed graph, in scheme S5 we must consider weights. To illustrate, consider again our toy example in Figure 7. If we model these reasons disregarding weights, we are limited to categorical comparisons, i.e. one can only tell “Aeronautics102 teaches about rocket science, but Exoplanets101 does not”. This limitation imposes difficulties when we need more granular information to compare two recommended items. For example, we can tell that both Exoplanets101 and Aeronautics102 are about exoplanets, but we cannot compare them since we do not have information about how strong is the link relating each course to its subjects. On the other hand, if we model relations with weights, we can capture that “Exoplanets101 is more related to exoplanets than Aeronautics102”. Therefore, using a more sophisticated weighted directed graph model to represent our reasons, we can leverage the expressiveness of our explanations while improving coverage and support.

We propose to use the plausibility scores from the knowledge embedding itself, the function f, to rank the the recommended entities according to the quantitative objective, available beforehand. While user’s objectives are subjective and, thus, hard to generalize, we assume that in the context of our abstract recommendation mechanism, users pursue strong links connecting their recommended items to their preferences. Thus, in our explanatory framework, the “available objective” for scheme S5 takes the form of paths in the graph while the embedding plausibility score represents a measure of how well each path meets the user goal. For instance, back to our example where a student, who wants to learn about astronomy, finds herself in the position of choosing between two disciplines: “Exoplanets101” and “Aeronautics101”. We know both disciplines are about exoplanets, which is a topic of astronomy, and we know that the fact a discipline is about a topic of astronomy is a compelling reason for choosing that particular discipline. In this example, our proposal for scheme S5 considers that the “available objective” is to learn about topics of astronomy and that the predicted strength of the links connecting a discipline and topics of astronomy (the embedding plausibility score) is the measure of how well each discipline meets this objective.

Formally, first, we collect all the reasons for the recommended item being explained i_r and we rank i_r for each reason according to the embedding plausibility score; this is where we calculated the weights. Secondly, we iterate over all the alternatives to i_r repeating the same procedure we described. If the alternative is better ranked than i_r, it means the alternative is more related to that reason than i_r, so it is a reason against i_r. This procedure is described in Algorithm 3.3.

Algorithm 3.3 Explanation Generation using Scheme S5

```
procedure RANK($\Phi$: reasons for, $u$: user, $i$: item, $\Theta$: parameters)
    $r = \phi_i \in \Phi$                                   ▷ The quantitative objective relation
    $\Psi \leftarrow f_r(u, i|\Theta)$          ▷ Assess the quantitative objective as the plausibility score
    return $\Psi$                                          ▷ Measured plausibility score value
procedure SCHEME-S5($i_r$: rec. item, $u$: user, $\mathcal{I}$: rec. set, $\Pi$: paths, $\Theta$: parameters)
    $\Omega_{u,i_r} \leftarrow \{\}$                                        ▷ Set of reasons against $i_r$
    $\Phi_{u,i_r} \leftarrow$ REASONS-FOR($i_r$, $u$, $\Pi$, $\Theta$)                      ▷ Set of reasons for $i_r$
    $\Psi_{i_r} \leftarrow$ RANK($\Phi_{u,i_r}, u, \Theta$)
    for $i \in \mathcal{I} \backslash \{i_r\}$ do                                ▷ Iterate over alternatives to $i_r$
        $\Phi_{u,i} \leftarrow$ REASONS-FOR($i$, $u$, $\Pi$, $\Theta$)
        $\Psi_i \leftarrow$ RANK($\Phi_{u,i}, u, \Theta$)           ▷ Measure of how well $i_r$ meets user's objective
        if ($\Psi_{i_r} < \Psi_i$) then            ▷ Compare if $i$ better meets the objective than $i_r$
            $\Omega_{u,i_r} \leftarrow \Omega_{u,i_r} \cup (\Phi_{u,i} \backslash \Phi_{u,i_r})$
    return $\Omega_{u,i_r}$
```

4 IMPLEMENTATION AND EVALUATION METHOD

"Instrumental or mechanical science is the noblest and, above all others, the most useful."

-- Leonardo da Vinci

4.1 Implementation

We developed a system that recommends courses offered by USP to undergraduates so as to test our proposals. Our recommendation system is built upon a real-world large-scale knowledge graph of our own making, called USPedia. Firstly, we describe the process of data gathering we carried out. Next, we present USPedia development in detail. Finally, we list the technologies, platforms and toolkits used in the development.

All the experiments and processes for this research were conducted using the computational infrastructure from the *Centro de Ciência de Dados* at the USP Innovation Center (C^2D-Inova). In addition, all the code and data used in this project is open sourced at github (https://github.com/gustapoll/calisto-backend).

4.1.1 Data Sources

All the specific data from USP, such as undergraduate courses and teachers, were gathered from the undergraduate support platform Jupiterweb.[1] Even though these data are public, by the time this work was done (from 2019 to 2020), no open access API was available. Given manual gathering was unfeasible due to the scale of data, we implemented a web scraper using Python 3.7 language and the Selenium library [2] to access Jupiterweb and extract information about all the 1740 undergraduate courses available at the *Escola Politécnica da Universidade de São Paulo* (EPUSP) as of May 20 of 2019.

The information available in Jupiterweb is semi-structured, i.e. while some of of it can be mapped to primitive types, others are presented in Brazilian Portuguese. In this work we concatenated all textual information (course description, program and syllabus)

[1] https://uspdigital.usp.br/jupiterweb/

[2] https://www.seleniumhq.org/

into what we will refer from now on as *description*. To illustrate, Figure 9 shows the wordcloud for the course descriptions.

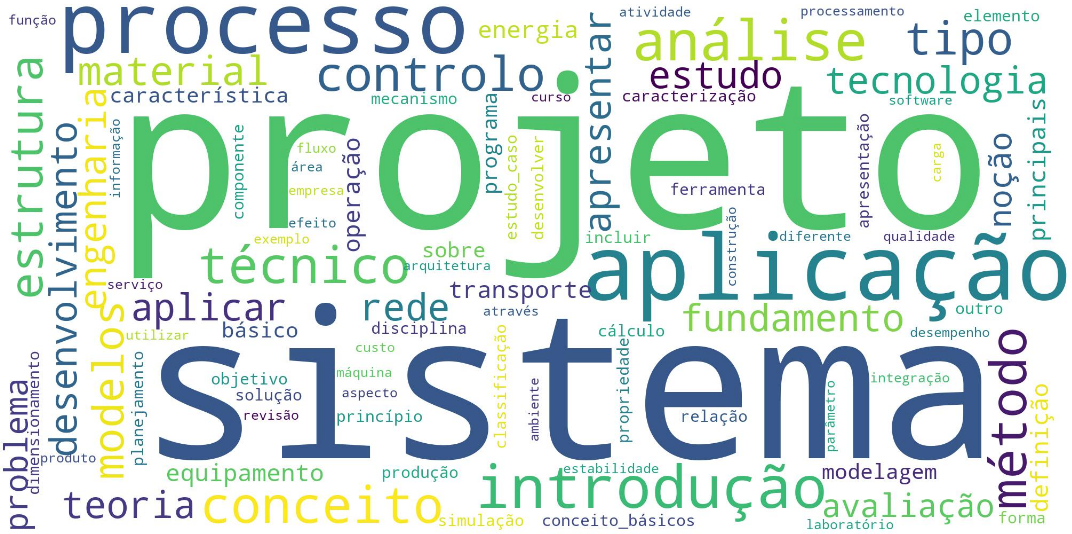


Figure 9: Wordcloud for the course descriptions of all collected EPUSP undergraduate courses.

Besides the description, we also collected 543 teachers names for the undergraduate courses. We consulted the Elsevier academic data respository Scopus [3] to collect information about research articles published by EPUSP teachers. In spite of the fact that Scopus does not have public access, it can be consulted under USP institutional affiliation.

The data collection was executed by a Python script of our own authorship using pybibliometrics [4], which is a popular toolkit to access Scopus API. First, we retrieve the Scopus author id using the teacher's name from our collected database using the query:

```
authfirst(<NAME>) and authlast(<SURNAME>) and af-id (60008088)
```

The fields *authfirst* and *authlast* are filled with the teacher's name, the *af-id* is the affiliation code from USP. When more than one author id for the same teacher is retrieved, we used the least Jaro distance [48] between teacher's name (from our base) and author's full name (from Scopus) as disambiguation criteria. Next, for each author id, we retrieved all their associated publications in Scopus. We collected the abstracts in English for 7648 research articles as of Jun 15 of 2019. It is worth mentioning that we could not find any publication for 90 of the consulted teachers.

It is important to highlight that a clear limitation of this method lies in abbreviations and common names. For example, if a given teacher name is very common (e.g. "João

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

[4] https://pybliometrics.readthedocs.io/en/stable/

Silva") or is abbreviated (e.g. "João N G da Silva"), we expect to retrieve multiple author ids that will correspond to other researchers with very similar names. In such cases, since our disambiguation is simply lexical, it is reasonable to expect that false matches will arise. In addition, our procedure links only one author id to each teacher, so we disregard cases when the same researcher has multiple accounts.

The undergraduate course's descriptions went through a data pipeline (of our own authorship) implemented in Python 3.7 using the packages nltk [5] and spacy[6]. Consider the course descriptions pipeline. First, we employ regex to remove all punctuation, tags, special characters and digits. Next, we use nltk to tokenize and remove stopwords (we employed the default set of predefined stopwords from nltk). Then, we realize POS tagging using spacy and filter only tokens classified as nouns. We applied a spelling [7] to reduce the impact of typos and misspelling. Also, since some descriptions pre-dated the Portuguese spelling reform, we noticed the use of some deprecated language resources, e.g. dieresis. While some of these orthographic inconsistencies were addressed with the spelling (e.g. *idéia*), words with dual spelling (e.g. *idielétrico*, *idieléctrico*) or dieresis (e.g. *freqüencia*) were manually solved. Finally, we applied spacy's lemmatizer on the remaining tokens.

Our pipeline did not contemplate compound words, like artificial intelligence, machine learning, etc. In order to capture these concepts, we applied a bigram identification procedure. We employed nltk to calculate the likelihood ratio [49] for all word pairs in tokenized documents. We then took the top 200 pairs with highest likelihood as collocations. Finally, we applied a keyword identification procedure to select only the most relevant concepts. We considered as keyword the top 15 tokens in each document with the highest TF-IDF [50]. Regarding the article's abstract, we did not employ any text mining technique; instead we used directly the keywords from the article metadata.

The cleaned and structured data was stored in Firestore[8], a document cloud database. The code used in this processing pipeline is available at github[9].

4.1.2 USPedia

Besides the data collected from Jupiterweb and Scopus, we also used the open domain large-scale knowledge graph DBpedia [51]. DBpedia is an encyclopedic database that

[5] https://www.nltk.org/

[6] https://spacy.io/

[7] https://pyspellchecker.readthedocs.io/

[8] https://firebase.google.com/docs/firestore

[9] https://github.com/gustapp/uspedia

stores the knowledge available at Wikipedia[10] in a structured and accessible manner. As undergraduate courses teach concepts that are tied to common domains (e.g. engineering, physics, etc), such concepts should be available in DBpedia and there is no need for us to build an ontology from scratch. In order to select only the subset of DBpedia that contains the ontology needed for this research, we employed a script written in Python 3.7 of our own authorship that queries the DBpedia SparQL API[11].

The USPedia structure consists of five types of entities: learning-object, lecturer, 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. In addition, we built our knowledge graph with 3 relations: *involved*, *subject* and *is_topic_of*. 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 has multiple concepts as subjects. USPedia incorporates an hierarchy of concepts where one or more concepts are topics of a category, and categories are topics of other categories.

To build the KG, we opted for using an automated semi-structured approach [22] as it has also been employed by many popular large scale knowledge graphs, such as DBpedia. The selected approach aims at automatically extracting information from semi-structured data, like infoboxes, via rules or regular expressions. We performed entity linking to DBpedia [12] using the articles keywords and course description content, so the hierarchy of concepts and categories were incorporated from DBpedia. The entity linking procedure consisted of consulting DPedia (see query in Figure 22 on Appendix) for each document keyword and check for entities with matching names. A match or hit is assumed only if the entity label's name, or any of its synonyms, is exactly the same as the keyword. For every keyword, we tested its plural form and, in the case of compound words, we also tested inserting middle prepositions or hyphen. To illustrate, consider the keyword *rede_computador* and its corresponding DBpedia entity *rede_de_computadores*. While the keyword is composed by a bigram whose words are in lemma form, the entity has one word in plural form and a preposition. In order to match them, our entity linking method will iterate over all combinations of plural form and prepositions.

Because it is costly, both in terms of time and computation, to perform this entity linking procedure for all keywords, we opted for limiting the number of evaluated key-

[10] https://pt.wikipedia.org/

[11] https://dbpedia.org/sparql/

[12] http://dbpedia.org/

words. We evaluated the top 10 keywords, according to their TF-IDF score, by document. With this configuration, we were able to link at least one keyword for 99% of all courses.

After this entity linking process, we extracted the relevant ontology from DBpedia using the following relations: `dct:subject`[13] and `skos:broader`[14]. The first represents an attribute of the entity and the second relates to categories and sub topics. To extract our ontology, firstly we expanded the "subject" relation to retrieve the categories for each linked entity. Then, we expanded the nodes from the previous step 5 times using the relation "broader". To illustrate this procedure, consider Figure 10.

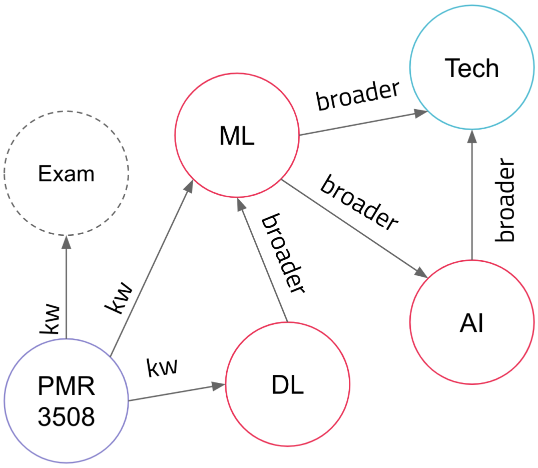


Figure 10: Ontology building sample. Pink nodes (ML, DL and AI) are reached after the first expansion, Turquoise node (Tech) is reached in the second expansion. "kw" stands for keyword. The dashed node (Exam) represents an unlinked keyword.

New concepts, not previously recovered in the entity linking, are retrieved from DBPedia as a result of the node expansion. For instance, the node "Tech" from the example in Figure 10 is present in the graph even though it is not linked to any course. Thus, relevant concepts that are unlikely to be found during keyword extraction, e.g. trigrams, are included in our ontology. In order to properly link them to their respective courses, we executed a lexical search for the course descriptions.

We executed an analogous entity linking process for the articles and their keywords.

We thus obtained two distinct graphs, one for course descriptions and another for articles. As each graph has entities in different languages (Brazilian Portuguese for courses and English for articles), we opted for using the relation `owl:sameAs`[15], which tells if two entities are the same even if described in different languages, to unify both graphs. When

[13] https://dublincore.org/

[14] https://www.w3.org/2009/08/skos-reference/skos.html

[15] https://www.w3.org/2002/07/owl

there is no equivalent match in Brazilian Portuguese, we opted to keep the entity described in English.

The whole knowledge process here described resulted in a Knowledge Graph with 34182 entities, 3 relations, and 152468 triples. Table 3 presents USPedia compared to other popular large-scale knowledge graph benchmarks from the literature. Figure 11 shows a sample sub-graph of USPedia.

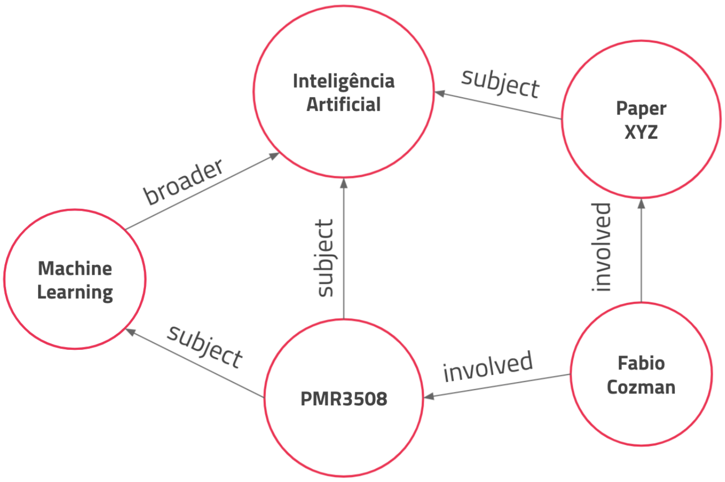


Figure 11: Sample sub-graph of USPedia.

Dataset	$\parallel \mathcal{E} \parallel$	$\parallel \mathcal{R} \parallel$	$\parallel \mathcal{T} \parallel$
USPedia	$34,182$	3	$152,468$
FB13	$75,043$	13	$345,873$
NELL186	$14,463$	186	$41,134$
WN18RR	$40,943$	11	$93,273$

Table 3: Statistics of other knowledge graphs as compared to USPedia; $\parallel \mathcal{E} \parallel$, $\parallel \mathcal{R} \parallel$ and $\parallel \mathcal{T} \parallel$ are the total number of entities, relations and triples respectively.

Despite being a large-scale knowledge graph, USPedia still have some limitations and room for improvement. One clear restriction is its inability to handle words with dual meaning. Our entity linking procedure was only lexical, so if multiple entities match the same word, the disambiguation is done randomly, which could potentially lead to false facts. For example, a course about telecommunication could be mistakenly linked to the chemical element "Radio". In addition, USPedia is only partially described in Portuguese and this clearly limits its application.

Finally, we used the OpenKE [16] toolkit to train the knowledge embeddings used in this research. This Python package contains the implementation of state-of-the-art embedding

[16] https://github.com/thunlp/OpenKE

models and is built upon tensorflow [17].

4.1.3 Recommendation System Set-Up

Once we constructed our knowledge graph, we trained a TransE [29] 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 [27, 28].

Using the trained knowledge embedding, 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 knowledge embedding to rank entities and, then, realize a Top-N recommendation following the abstract mechanism described earlier in our proposal.

4.1.4 Dialogflow

We opted for using the Dialogflow[18] chatbot development platform. Dialogflow has support for Brazilian Portuguese, is free and offers acceptable performance in natural language processing tasks [52].

4.2 Evaluation Method

Our work focus on improving the interpretability of recommendation systems through novel approaches for explanation generation. To evaluate interpretability, we consider three levels (Figure 12): functional, human and application [33]. These levels grow in complexity and implementation cost. The functional level is the cheapest because it evaluates the explanation in objective terms based on a proxy task and, thus, does not require human subjects. However, the functional level alone is inappropriate for testing the subjectivity of explanations, which rely inherently on the perception of users. While the human level provides insights about user perception, the application level tests the system as whole and, thus, considers practical aspects, e.g. response delays, in the evaluation. Furthermore, we highlight in the following sections some of the limitations related to each

[17] https://www.tensorflow.org/

[18] https://dialogflow.com/

evaluation level in the context of this work and discuss approaches from the literature to mitigate the observed caveats [53, 54].

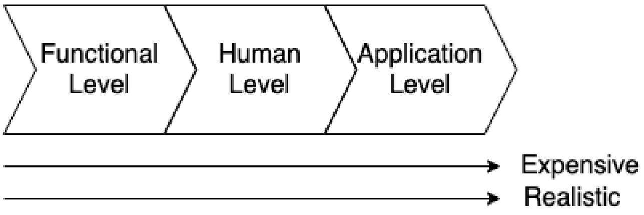


Figure 12: Interpretability evaluation levels.

In this work we explore the three interpretability evaluation levels. At first, we employ functional level tests to evaluate the practical feasibility of our proposals. The methods that pass the functional tests move forward for the human level evaluation. We run a survey where users were asked to compare examples from different explanation methods. Such examples were simplified simulations of interactions with the real application. Additionally, we have carried out a preliminary application level evaluation. We integrated the most promising explanation methods into actual conversational recommendation systems, which were then compared in a survey with real users from our application domain.

Now we examine in detail each one of these evaluation steps and the metrics used.

4.2.1 Functional Level

An online system, such as conversational recommendation systems, cannot impose long delays on its users, as it is an interactive application. In addition, a good interpretability method should be able to explain most if not all the recommendations, i.e. ideally no decision should remain unexplained. None of these characteristics rely on human subjectivity and, thus, they can be evaluated in a functional level.

We define the following three metrics to be evaluated in this level:

1. *Coverage or Recall*: The fraction or percentage of recommendations for which the interpretability method can find at least one explanation.

2. *Support*: The arithmetic mean number of explanations the interpretability method can find for each recommendation.

3. *Response or Execution Time*: The arithmetic mean time the interpretability method takes to find explanations for each recommendation.

The main goal of our functional level is to evaluate the feasibility of an interpretability method. Therefore, we want to verify if the method achieves high coverage while coping with the strong execution time constraints of interactive applications. Support is an auxiliary metric that tells us how the method scales in scenarios where multiple explanations are required.

To realize our functional level evaluation, firstly, we built a dataset of simulated user-interactions from 10% randomly selected entities of USPedia. Next, we run each interpretability method in an offline manner and recorded the metrics of interest.

In the remainder of this section we discuss the limitations and caveats related to our functional level evaluation:

- **Domain Generalization**: We carried out all experiments in the single application domain of USP discipline recommendation, so we cannot ascertain the generalization of our methods towards other domains.
- **Simulation Generalization**: We built simulations randomly selecting entities from USPedia, so we didn't account for any popularity bias that may appear in real world scenarios. It is reasonable to expect that students will ask for recommendations of some subjects (popular) more often than others. Thus, we could possibly observe worse results in real applications if our methods perform poorly for the most popular subjects.

4.2.2 Human Level

The human level evaluation is intended to offer preliminary insights or early validation of an interpretability method from the user perspective. In order to translate the subjective user perception into quantifiable measures, we adopt the following five *explanation aims* [5]: transparency, trust, persuasion, engagement e effectiveness. Each one of them represents a particular goal of an explanation and are commonly used as benchmark in the RS domain [4, 5].

The evaluation was carried out with undergraduate engineering students at USP. Each subject ranked each interpretability method with respect to each explanation metric using a survey-based Likert psychometric scale [55] 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 method.

In the remainder of this section we discuss the limitations and caveats related to our human level evaluation other than the ones already discussed in the previous section; the following list was inspired by previous human studies in the literature [53]:

- **Sample Generalization**: We used a few handpicked recommendation examples to build the surveys used in our human level evaluation, so we cannot ascertain how our proposed methods would generalize for different ones.
- **Demographic Generalization**: All surveys were targeted towards engineering students from USP, so our results are still very limited to a young, male community from southeast Brazil.
- **History and Maturation**: To mitigate the effects of time passage, we ensured that all participants had to complete the survey in one go.
- **Instrumentation**: To reduce the impact of learning effects on our results, i.e. participants improving over repetition, we randomized the order of questions they were asked to answer. Thus, we expect an overestimation due to learning effects but equally distributed, allowing relative comparisons.
- **Experimenter Bias**: To mitigate any unconscious bias being conveyed to participants, experimenters were only allowed to answer technical questions.
- **Misunderstanding**: To reduce the risk of participants misunderstanding the survey instructions due to lack of clarity, we asked some volunteers, without any contact to the actual participants, to fill the survey. In addition, we required every participant to answer correctly a series of practice questions before moving forward with the experiment.
- **Technical Variance**: To reduce variance due to participants using different hardware and software, we asked in advance for preparing an environment with stable internet connection and implemented the survey in a google forms to avoid compatibility issues.
- **Multiple Submissions**: Participants were allowed to answer the survey only once.
- **Selection**: Participants were asked to join in the experiment as an optional rewarded task in an USP engineering course. So, even though the participants had

to volunteer, there were incentives in place. Since our results were drawn from this self-selected population, they might not generalize. On the other hand, all participants were potential real users from our application domain, what strengthen our results.

- **Ecological Validity**: The fact that our surveys were carried out remotely, instead of a physical environment like a laboratory, increases their ecological validity. Since participants were in their usual surrounding, the effects of being in an unfamiliar setting were mitigated.
- **Drop out**: Even though participants were free to drop out the experiment, this effect was mitigated due to the incentive of receiving an additional grade in USP engineering course.

4.2.3 Application Level

Since our target audience, USP undergraduate students, was directly accessible, the application level evaluation was similar to the human level one. However, in the application level subjects were asked to interact with a real recommendation system instead of a mocked scenario. Thus, functional factors such as response time and coverage impacted the user perception evaluation.

About the limitations and caveats related to our application level evaluation, while the concerns regarding sample and simulation generalization are mitigated (i.e. users can freely interact with the system as in a real world scenario), the following issues become much more evident:

- **Instrumentation**: Since users can freely interact with the system, we have little or no control of learning effects as they can ask and evaluate multiple instances.
- **Misunderstanding**: While freely interacting with a conversational system, users are more likely to misunderstand the purpose of the system and, thus, struggle with its limitations. For example, users may get frustrated after asking for recommendations based on discipline difficulty, what is not in the scope of our system.
- **Technical Variance**: Users are much more likely to struggle with internet connection, response delays and compatibility issues while interacting with a real system.

5 RESULTS AND DISCUSSION

"To achieve great things, two things are needed:
a plan, and not quite enough time."

-- Leonard Bernstein

In this section we describe experiments with simulated and real users. First, we present some anecdotal examples that highlight the strengths and weakness of our explanation method in the current set up. Next, we examine the feasibility of our techniques in Section 5.2 and then we discuss the reaction of human users to our approach in Section 5.3.

5.1 Anecdotal Examples

To illustrate the explanations generated by our proposals, Figure 13 depicts a real explanation example generated by our method. Entities and relations found in the graph appear in the figure, while the textual explanation derived from them appears in the caption. In this particular case, the system recommended the discipline entitled "Legal Engineering" to attend the requested preference about "History" subject. Here, the system explains by arguing that the recommended discipline is about "Law", which is a "Humanities" topics just like "History". This example highlights the ability of our methods to leverage ontological connections to offer a rationale for recommendations. In addition, in this example, the relationship between the discipline "Legal Engineering" and "Law" was inferred by the embedding and was not present in the original knowledge graph due to its inherent incompleteness. The characteristic of being able to benefit from the embedding capacity of inferring missing relationships in the graph and using them to formulate explanations, besides increasing coverage, can potentially shed light on what relationships were considered important by the embedding for producing its recommendations.

On the other hand, the ability of leveraging inferred relationships can have drawbacks. Since the reasons produced by our explanation methods can employ relationships that are not necessarily grounded on known facts, i.e. absent in the original graph but predicted by the embedding, explanations may incur into false facts, which can be often easily spotted

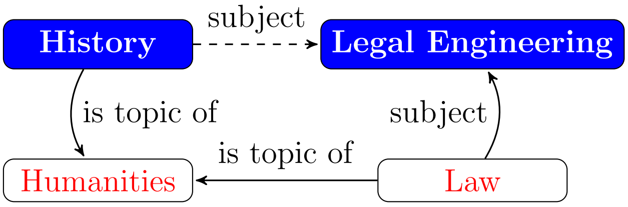


Figure 13: “Legal engineering is recommended as it is about Law and both Law and History are topics of Humanities”

by the system’s end user. Notably when the system is forced to explain a bad recommendation, i.e. a suggestion that poorly fits the preference subject, weak or clearly false relationships tend to arise in the system explanations. For instance, Figure 14 explains the connection between the recommendation of a discipline entitled “Heavy Construction” and the subject of interest “Medicine” by saying that both “Medicine” and “Work Accident” are correlated themes. Despite hurting the chance of a user following the system’s choice, we argue that even non-sense explanations serve the purpose of empowering the user to critique and disregard bad recommendations.

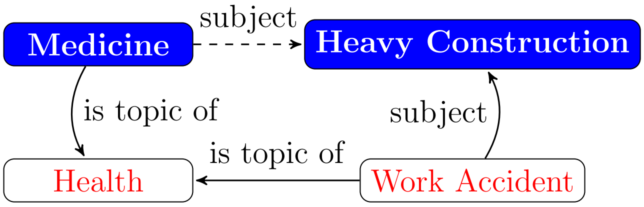


Figure 14: “Heavy Construction is recommended as it is about work accident and both accident and medicine are topics of health”

5.2 Simulated Experiments

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

1. Are our explanation schemes feasible from an implementation perspective?
2. 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?
3. How long does it take to find explanations using the knowledge embedding? Is time-to-response acceptable?

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 — we call it *Support*.

Regarding *reasons for*, Table 4 shows the overall coverage and support for our proposed DFS interpretability method using the knowledge embedding as source for explanations (referred as PRED) compared to the baseline using the original incomplete knowledge graph (referred as TRUE). The results show that we obtained 79.33% coverage and a support mean of 2.0 for the embedding-based approach, compared to 42.3% coverage and 1.8 support in the graph-based. We can observe that indeed our proposal of replacing the original KG by the KE improves coverage significantly.

As for *reasons against*, we ran our experiments considering schemes S1, S4 and S5; all schemes were implemented considering the embedding-based approach. Both the coverage (85.1%) and support (2.3) obtained for S1 (the same for S5) 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.

Type	**Scheme**	**Coverage**	**Support**
Reason For	TRUE	42.3%	1.8 ± 1.0
	PRED	79.3%	2.0 ± 1.0
Reason Against	S1	85.1%	2.3 ± 1.4
	S4	0%	-
	S5	83%	1.0 ± 0.5

Table 4: Coverage and Support for *reasons for* using the embedding-based (PRED) and the graph-based approach (TRUE), and reasons against using schemes S1, S4 and S5. Note Support for each Scheme is presented with its respective standard deviations.

Figure 15 presents the behavior of the recall for our *reasons for* proposed method PRED (*embedding recall*) compared to the baseline TRUE (*graph recall*). It also shows the average number of explanations found (*avg. explanation number*) and average execution time (*avg. exec. time*) for our proposed method, when varying time constraints (timeout). We did not include the local-surrogate proposed method, described in Section 3.1, in this analysis because we observed that its average response time is around 2-3 minutes, which is clearly unfeasible for online applications.

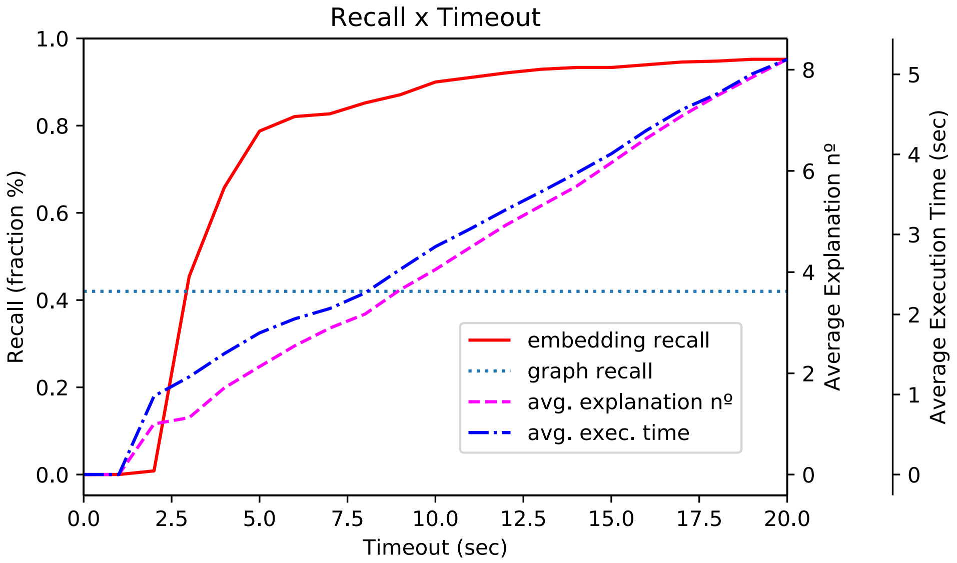


Figure 15: Recall comparison between our proposal (embedding recall) and the baseline (graph recall). Also present average explanation number found (avg. explanation no.) and average execution time (avg. exec. time) for our proposal

While the baseline method, which uses only the original graph to search for explanations, is by far faster than our proposed method, we 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 "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 knowledge graph cannot find explanations for less than half of recommendations in our experiment; also, it is not sensitive to time constraints, i.e. saturates into a flat line within milliseconds. 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 second research question. 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 16 shows the boxplots, with suppressed outliers for better visualization, of the execution time of our proposed method PRED for different numbers of explanations. In this experiment, we aim to evaluate 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 does produce multiple explanations but not too many of them, answering the third question.

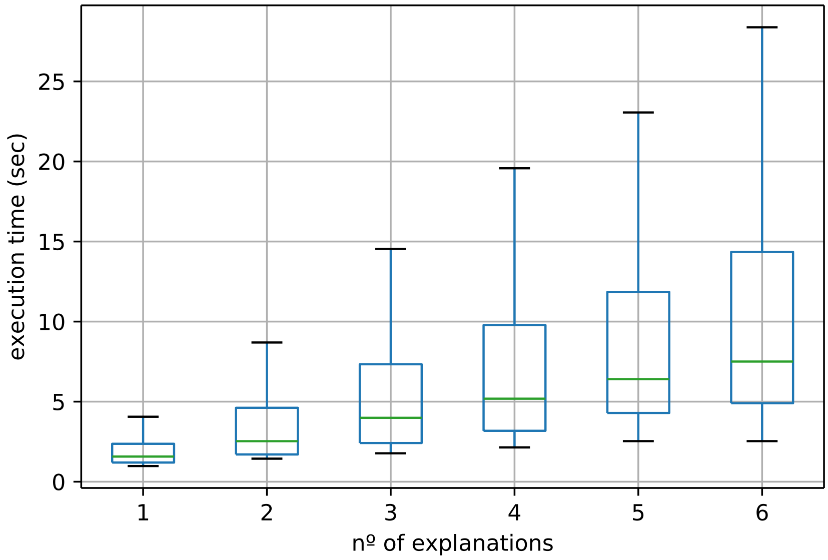


Figure 16: Execution time of PRED method considering explanation number constraints.

5.3 Tests with Human Subjects

During this work we prepared four user tests. On November 2019 we executed an application level experiment focused on the *reasons for* generation and intended to compare the embedding-based approach (PRED) against the graph-based baseline (TRUE). On July 2020 we realized a human level evaluation to validate our hypothesis about the introduction of reasons against in explanations. On November 2020 we run the final application level test where we compared multiple reasons against schemes against the reasons for only baseline. Finally, on April 2021 we realized a final human level test to assert the user perception on multiple reasons against schemes.

We start presenting the results for the human level evaluation in Section 5.3.1 and, then, we discuss the two application level tests together in Section 5.3.2.

5.3.1 Human Level Evaluation

In this section we report on human-level evaluation experiments that were designed to address the following questions:

1. Do users perceive value in the explanations produced by our schemes?

2. If they do perceive value, which scheme performs best?

We conducted a user study involving 54 subjects from the engineering post-graduate program in EPUSP (88% = male, 78% = born in Brazil's southeast, 72% = *age* <= 30, and 98% = high tech affinity). All demographic data was reported by the subjects.

To run our experiment, we asked the subjects to evaluate explanations generated using our proposed schemes S1 and S5 for 6 recommendation cases (check the full list in the Appendices). The set up was designed to run a block-randomized experiment within subjects. Each user evaluated explanations from both S1 and S5 schemes, according to a five point Likert-scale, for three explanation aims: persuasion, trust and transparency, respectively. The order the schemes were presented at each step varied randomly.

Figure 17 depicts the entire experimental set up. Each subject involved in the experiment carried out the following steps:

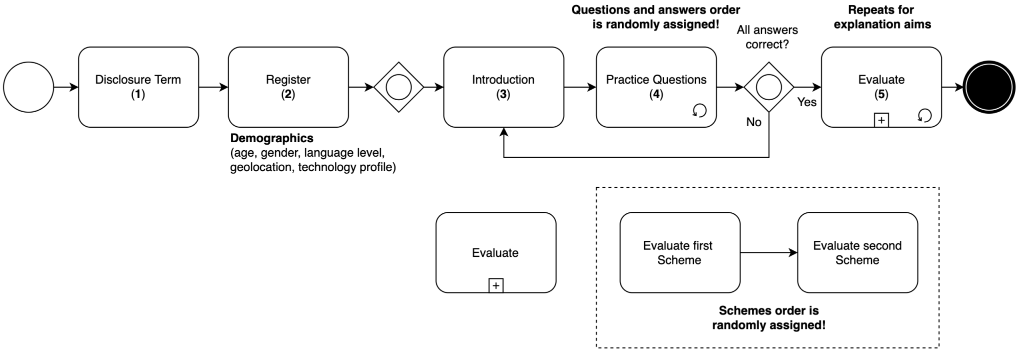

Questions and answers order is randomly assigned!
All answers correct?
Repeats for explanation aims
Disclosure Term (1)
Register (2)
Introduction (3)
Practice Questions (4)
Yes
No
Evaluate (5)
Demographics (age, gender, language level, geolocation, technology profile)
Evaluate
Evaluate first Scheme
Evaluate second Scheme
Schemes order is randomly assigned!

Figure 17: Diagram representing the human-level experimental set up adopted.

(1) Disclosure Agreement: First, we ask the user to accept a disclosure term granting access of their data for academic purposes.

(2) Collection of demographic data: Subjects were asked to provide common demographic information regarding age, gender self-identification, Brazilian macro region as place of birth and tech affinity; check the Appendices for the full questionnaire. Our goal was to better understand our sample and scope our experiment within a population strata.

(3) Introduction and metrics: We presented all the explanations aims [56] and asked each subject to read them carefully.

(4) Practice questions: Subjects were asked to answer multiple answer questions about all the explanation aims. If the user answers all questions correctly, they would proceed, otherwise they were asked to go back to step 3 and repeat the questionnaire. This phase was intended to guarantee that all subjects understood the experiment purpose and metrics.

(5) Explanation scheme evaluation through questionnaire: Subjects were asked to evaluate the S1 and S5 schemes according to the explanation aims: persuasion, trust and transparency, respectively, in a five point Likert-scale. Table 5 contains the details of the questionnaire presented.

Aim	**Question**
persuasion	I feel the explanation persuaded me to follow the recommendation
trust	I feel more confidence in the system after receiving the explanation
transparency	I feel I better understood the recommendation after receiving the explanation

Table 5: Questionnaire details.

Figure 18 presents the arithmetic mean scores obtained from the survey for each explanation aim considering the schemes S1 and S5. The confidence intervals were calculated with bootstrapping at 95% confidence. We can observe that mean scores for all explanation aims were closer to “neutral” or “agree” in the Likert-scale, which indicates that, at least in our specific set up, users perceived value in the explanations.

In addition, note that scheme S5 ($\mu_{persuasion} = 3.3$ and $\mu_{trust} = 3.83$) received greater mean scores than S1 ($\mu_{persuasion} = 2.48$ and $\mu_{trust} = 3.07$) for persuasion and trust. Note the confidence interval whiskers do not overlap; indeed this difference is statistically significant considering a t-test (persuasion = p $0.000838 < 0.05$ and trust = p $0.000971 < 0.05$). On the other hand, for transparency, S1 obtained a greater mean value than S5, however, for this metric we didn't achieved statistical significance (transparency = p $0.2 > 0.05$). All average values are summarized in Table 6. These results indicate that S5 appears to perform better than S1 in persuasion and trust, while having similar results in transparency.

It is important to reinstate that these results are only exploratory and require large-scale user studies to properly evaluate and compare our explanation schemes. First, our sample is biased towards Brazilian southeastern young male post-graduate engineers

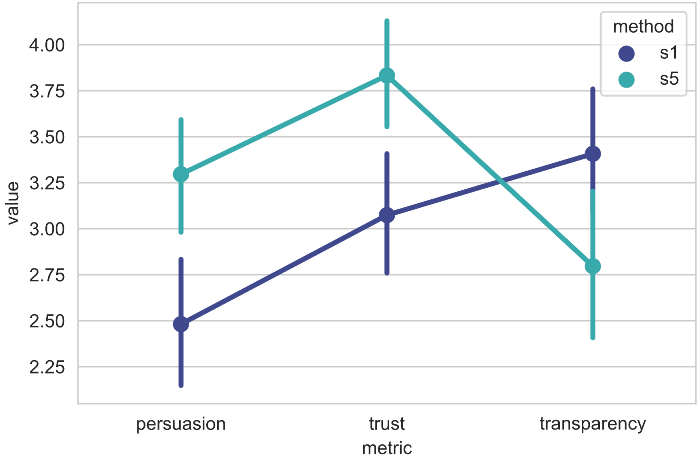


Figure 18: Visual representation for explanation metrics arithmetic mean scores.

Scheme	**Persuasion**	**Trust**	**Transparency**
S1	2.48	3.07	3.41
S5	**3.3**	**3.83**	2.8

Table 6: Arithmetic mean scores for explanations from our user study. The highest scores with statistical significance for each metric are highlighted in bold. Statistical significance was assessed by t-tests, with $p < 0.05$.

with high degree of technological affinity. Further research should evaluate whether the methods proposed in this work would generalize to a broader audience. In addition, the conclusions of this experiment were drawn from explanations that, despite being actual outputs of our schemes, are limited to relatively few examples; therefore, it is uncertain whether they will generalize to real-world scenarios. Finally, our experimental set up consists of a human-level evaluation and, thus, fails to take into account the impact of practical circumstances, such as system response time and explanation coverage, in the user perception.

5.3.2 Application Level Evaluation

In this section we report on experiments that were designed to provide a first glance on the following research questions:

1. Does the quality of the explanations found using the knowledge embedding deteriorate when compared to those using the original graph?

2. Do reasons for/against have value for users in a real application scenario? If yes, which scheme performs best?

As described previously, in our first application level experiment (focused on *reasons for* only), we produced two conversational recommendation systems, one with the automatically generated knowledge graph 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 undergraduate engineering students from EPUSP, in which each user evaluated two systems, one employing our proposed method and the other one using the original knowledge graph as a source for explanations. The users were asked to evaluate the five explanation aims (summarized in Table 7) using a Likert psychometric scale from 1 to 5 [55]. One interaction consisted of the user asking for a recommendation for 5 different themes, so we collected a total of 130 interactions.

Aim	**Question**
transparency	Did the explanation help you 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 RS?
effectiveness	Did the explanation sound coherent?

Table 7: Questionnaire details.

Figure 19 depicts the whole experimental set up. Each subject executed the following steps:

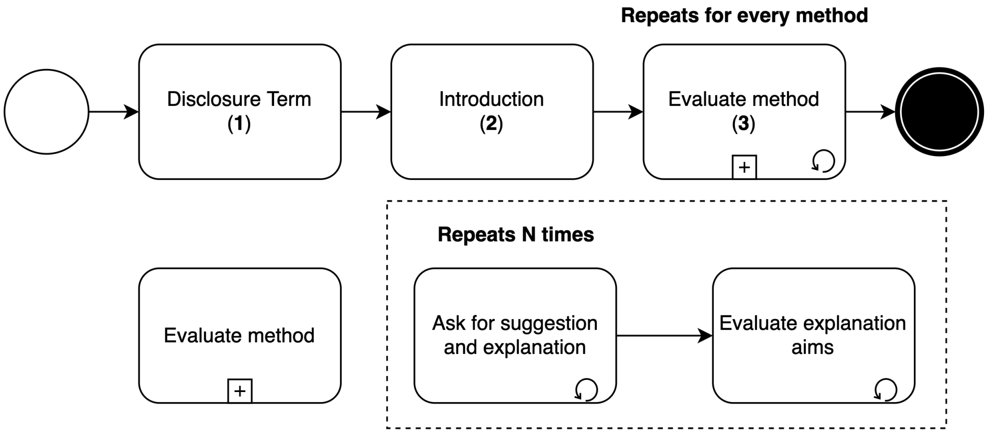


Figure 19: Diagram representing the application-level experimental set up adopted

(1) Disclosure Agreement: First, we ask the user to sign a term accepting to disclose her data for academic purposes.

(2) Introduction: Next, we present a detailed description of each explanation aim to be used as evaluation metric. Also, we allow the user to explore a conversational recommendation system without explanation facilities as warm up and to get familiar with the interface.

(3) Evaluation: Finally, subjects start the evaluation by asking a predefined number N of recommendations and their respective explanations, at the end of the interaction, the user evaluates all explanation aims. Note the experiment is within subjects and each scheme is evaluated in order. Thus, this evaluation does not account for learning effects and can, potentially, lead to positive bias for the later schemes. This step is repeated for every scheme.

It is important to comment that our experiment set up did not collect demographic data about its subjects, hence we cannot report on which population strata we are dealing with. In addition, our sample size was restricted to a few undergraduate students, so all conclusions here presented are only preliminary and further research is required.

Considering the exploratory nature of the survey, we describe below the performance indicators from the users' interaction with the conversational recommendation systems. Table 8 presents the arithmetic mean scores provided by the students in our user study for each one of the explanation aims. The upper half of the Table 8 contains the scores acquired in our first application level experiment, where we compared the embedding-based explanations to the graph-based ones. Figure 20 depicts these scores on a continuum representing visually the scale. Comparing both algorithms' overall mean, the knowledge embedding approach (PRED) was better from the user's perspective, $\mu = 2.7$ corresponding to the "neutral" evaluation at the Likert scale.

Algorithm	**Transparency**	**Persuasion**	**Engagement**	**Trust**	**Effectiveness**
TRUE	2.21	2.36	2.17	1.92	2.64
PRED	2.92	2.28	2.84	2.52	2.92
PRED*	2.87	2.41	2.87	2.58	2.96
S1	2.68	2.50	2.18	2.59	2.40
S5	2.94	2.65	2.68	2.68	2.74

Table 8: Average scores for explanation aims from our user study. The $*$ marker differentiates scores obtained for the same algorithm but in different experiments.

On the other hand, for the graph approach (TRUE) $\mu = 2.21$ is closer to the "disagree" at the Likert scale. Taking the variable in isolation, effectiveness got the highest average value for both $\mu_{pred} = 2.92$ and $\mu_{true} = 2.64$. This signals that users perceived the explanations as coherent. The TRUE approach had a bad evaluation when the trust was at stake ($\mu_{trust} = 1.92$). As TRUE suffers from knowledge graph incompleteness, it cannot

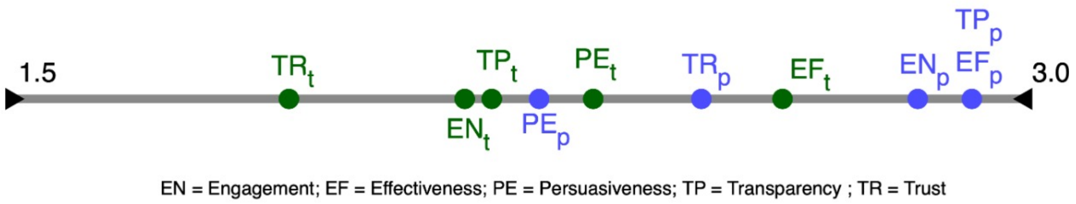


Figure 20: Visual representation for explanation aims average scores in our first experiment. TRUE and PRED results are in green and blue, respectively.

posit explanations for every suggestion. When compared to a better performance of the knowledge embedding approach ($\mu_{trust} = 2.52$), we might conjecture that users prefer any explanation instead of no explanations at all.

The second half of Table 8 presents the scores from our second application level test (focused on both *reasons for/ against*). In this second experiment, we asked 35 undergraduate engineering students from EPUSP to evaluate three conversational recommendational systems, one producing only reasons for as explanations (PRED) and the other two with both reasons for and reasons against (one using S1 scheme and the other S5 scheme). The second experiment was carried out using the same questionnaire (see Table 7) and evaluation procedures as the first one.

The second experiment was focused on comparing multiple techniques of reasons against generation (S1 and S5), and evaluating the presence of reasons against in an explanation against the reasons for only baseline. Note that the baseline PRED was the same embedding-based method as in the first experiment; indeed, we can observe that the mean arithmetic scores for PRED method in both experiments are similar. Figure 21 depicts these scores on a continuum representing visually the scale. We observe that the systems employing scheme S5 approach has the lead from the user's perspective in terms of transparency ($\mu_{transparency} = 2.94$), trust ($\mu_{trust} = 2.68$) and, notably, persuasion ($\mu_{persuasion} = 2.65$). Furthermore, the scheme S5 appears to be better than S1 in all the evaluated metrics.

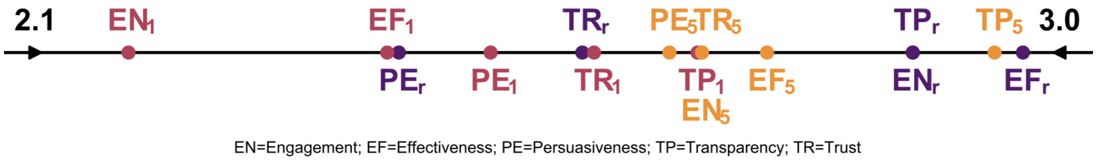


Figure 21: Visual representation for explanation aims average scores in our second experiment. PRED, S1 and S5 results are in purple, red and yellow, respectively.

As a side effect of introducing reasons against in explanations, we observe a drop in engagement and effectiveness, i.e. the baseline (PRED) achieved the highest scores in both metrics ($\mu_{effectiveness} = 2.96$ and $\mu_{engagement} = 2.87$).

If we compare these results with the ones we obtained in the human level evaluation, from Section 5.3.1, improvements in trust and persuasion were observed, however, the drop in effectiveness and engagement is unexpected. We imagine the bad performance in effectiveness and engagement is due to the complexity overhead added in explanations, i.e. explanations with both reasons for and against are harder to grasp than with reasons for only.

For both application-level experiments we were unable to obtain statistical significance with the t-test $p < 0.05$ criteria. The results here presented represent an initial and exploratory analysis to drive further research focused on properly evaluating these explanation schemes in real-world applications.

6 CONCLUSION

"All I know is that I know nothing."

-- Socrates

We here proposed and evaluated several techniques that aim at producing fast, effective and responsible explanations in the context of recommendation systems. Our experiments provided preliminary evidence that knowledge embeddings, if properly employed, can increase explanation coverage, while also satisfying reasonable time constraints. In addition, experiments with human subjects suggested that explanations drawn from embeddings may remain coherent and meaningful from the user perspective, while also increasing trust in the system and the perception of transparency. Furthermore, we explored the generation of reasons *for* and *against* in recommendations as a strategy for responsible explanations. We investigated the hypothesis that a recommendation system, by displaying such reasons, not only helps the user to reach the most rewarding decision, but also acts on its own interest in building trust.

We developed ways to generate reasons for/against adapting Snedegar's theory of practical reasoning. By implementing Snedegar's theory, we addressed practical difficulties with some of his schemes for reasons against and have proposed a novel design based on knowledge graphs and their embeddings. We evaluated the most promising schemes both offline and with human users in the contexts of recommendation systems. Our early results suggest that Scheme 5 is the most appropriate in practice at the moment. Moreover, our initial experiments with human subjects indicate that reasons against can potentially increase trust and persuasion. Overall, we advanced the notion that adding reasons against items does improve recommendation systems.

The present work represents a step towards efficient and responsible explanation generation methods that are suitable for interactive and conversational recommendation systems. In the process of pursuing this goal, we also built a new large-scale knowledge graph, USPedia, which can be a benchmark in the domain of course recommendation.

6.1 Future Work

We emphasize that this work, while a significant step towards explainable recommendation systems, is a preliminary and exploratory research that should be extended through future efforts.

We must first recognize that our current experimental set up is very limited. Due to the limited scale, limited set of participants, and the preliminary nature of this work, we were not able to properly evaluate our proposals on an application level and our promising results from both human and functional levels are only early signs instead of actual conclusions. Future work should provide a comprehensive evaluation so that we can ascertain the impact of explanations produced by our proposals on user perception. To achieve this goal, we suggest a web-based application level experiment whereby students from all over USP can freely ask for discipline recommendations and explanations, similarly to previous works from the literature [53].

As we have solely explored a single and very specific domain (discipline recommendation within the context of USP), there is still uncertainty as to whether our results generalize to other domains. Besides traditional entertainment domains, such as movie and song recommendation, biomedical domains represent a promising opportunity for future work. Recent advances in biomedical hypothesis generation methods, notably for drug discovery, have led to the adoption of complex knowledge embedding models [57,58], thus opening a wide interpretability gap.

Another concern related to the generalization of our proposals lies in the fact that all our results have been based on a single embedding model, TransE. Even though our proposals are model-agnostic by design, their effectiveness (e.g. coverage, response time and user perception) may differ depending on the underlying embedding model. Future work should evaluate our explanation generation methods on a broader range of embedding models. While translational embedding models are similar to TransE, semantic models differ widely among themselves [59]. Thus, we suggest that future research should evaluate our proposals on several semantic embedding models.

Finally, we suggest future work should adapt our proposals towards interactive recommendation systems. Although our proposals suit online applications, they disregard their sequential aspect, typical of interactive or conversational recommendation systems, whereby explanations and how users react to them should guide future recommendations. Future work should adapt our schemes to benefit from this conversational aspect and explore argumentative frameworks for sequential recommendation-explanation.

6.2 Research Disclosure

The main peer reviewed papers reporting results of this work are:

1. "*Explaining Completions Produced by Embeddings of Knowledge Graphs*", co-authored paper at 2019 European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)

2. "*Explaining Content-Based Recommendations with Topic-Models*", paper at IEEE 2019 Brazilian Conference on Intelligent Systems (BRACIS).

3. "*Faithfully Explaining Predictions of Knowledge Embeddings*", runner-up best paper undergraduate track at 2019 Encontro Nacional de Inteligência Artificial e Computacional (ENIAC).

4. "*Conversational Recommendation Systems within Explanations: Improving Coverage through Knowledge Embeddings*", paper at 2020 AAAI Workshop on Interactive and Conversational Recommendation Systems (WICRS).

5. "*Why should I not follow you? Reasons For and Against in Responsible Recommendation Systems*", paper at 2020 RecSys Workshop on Responsible Recommendation (FAccTRec)

REFERENCES

[1] RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. “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*. New York, NY, USA: ACM, 2016. (KDD ’16), p. 1135–1144. ISBN 978-1-4503-4232-2. Disponível em: <http://doi.acm.org/10.1145/2939672.2939778>.

[2] O’DONOVAN, J.; SMYTH, B. Trust in recommender systems. In: *Proceedings of the 10th International Conference on Intelligent User Interfaces*. New York, NY, USA: Association for Computing Machinery, 2005. (IUI ’05), p. 167–174. ISBN 1581138946. Disponível em: <https://doi.org/10.1145/1040830.1040870>.

[3] ESTOLA, E. When recommendation systems go bad. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA: Association for Computing Machinery, 2016. (RecSys ’16), p. 367. ISBN 9781450340359. Disponível em: <https://doi.org/10.1145/2959100.2959117>.

[4] MUSTO, C. et al. Linked open data-based explanations for transparent recommender systems. *International Journal of Human-Computer Studies*, v. 121, p. 93 – 107, 2019. ISSN 1071-5819. Advances in Computer-Human Interaction for Recommender Systems. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1071581918300946>.

[5] TINTAREV, N.; MASTHOFF, J. A survey of explanations in recommender systems. In: *2007 IEEE 23rd International Conference on Data Engineering Workshop*. [s.n.], 2007. p. 801–810. Disponível em: <https://doi.org/10.1109/ICDEW.2007.4401070>.

[6] HOEVE, M. ter et al. Faithfully explaining rankings in a news recommender system. *CoRR*, abs/1805.05447, 2018. Disponível em: <http://arxiv.org/abs/1805.05447>.

[7] SNEDEGAR, J. Reasons for and reasons against. *Philosophical Studies*, Springer, v. 175, n. 3, p. 725–743, 2018. Disponível em: <https://doi.org/10.1007/s11098-017-0889-2>.

[8] ALSHAMMARI, M.; NASRAOUI, O.; SANDERS, S. Mining semantic knowledge graphs to add explainability to black box recommender systems. *IEEE Access*, v. 7, p. 110563–110579, 2019.

[9] MILLER, T. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, v. 267, p. 1–38, 2019. ISSN 0004-3702. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>.

[10] HILTON, D. J. Conversational processes and causal explanation. *Psychological Bulletin*, p. 65–81, 1990.

[11] KOREN, Y.; BELL, R. M. Advances in collaborative filtering. In: RICCI, F. et al. (Ed.). *Recommender Systems Handbook*. Springer, 2011. p. 145–186. ISBN 978-0-387-85819-7. Disponível em: <http://dblp.uni-trier.de/db/reference/rsh/rsh2011.htmlKorenB11>.

[12] SMITH, B.; LINDEN, G. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 21, n. 3, p. 12–18, maio 2017. ISSN 1089-7801. Disponível em: <https://doi.org/10.1109/MIC.2017.72>.

[13] CHRISTAKOPOULOU, K.; RADLINSKI, F.; HOFMANN, K. Towards conversational recommender systems. In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2016. (KDD '16), p. 815–824. ISBN 978-1-4503-4232-2. Disponível em: <http://doi.acm.org/10.1145/2939672.2939746>.

[14] LIEBMAN, E.; STONE, P. Dj-mc: A reinforcement-learning agent for music playlist recommendation. *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AA-MAS 2014)*, abs/1401.1880, 2014. Disponível em: <http://arxiv.org/abs/1401.1880>.

[15] SCHEIN, A. I. et al. Methods and metrics for cold-start recommendations. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2002. (SIGIR '02), p. 253–260. ISBN 1-58113-561-0. Disponível em: <http://doi.acm.org/10.1145/564376.564421>.

[16] VOLKOVS, M.; YU, G. W.; POUTANEN, T. Content-based neighbor models for cold start in recommender systems. In: *Proceedings of the Recommender Systems Challenge 2017*. New York, NY, USA: ACM, 2017. (RecSys Challenge '17), p. 7:1–7:6. ISBN 978-1-4503-5391-5. Disponível em: <http://doi.acm.org/10.1145/3124791.3124792>.

[17] LU, Z. et al. Content-based collaborative filtering for news topic recommendation. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, 2015. (AAAI'15), p. 217–223. ISBN 0-262-51129-0. Disponível em: <http://dl.acm.org/citation.cfm?id=2887007.2887038>.

[18] ZHANG, F. et al. Collaborative knowledge base embedding for recommender systems. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 353–362. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939673>.

[19] HE, R.; KANG, W.-C.; MCAULEY, J. Translation-based recommendation. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2017. (RecSys '17), p. 161–169. ISBN 978-1-4503-4652-8. Disponível em: <http://doi.acm.org/10.1145/3109859.3109882>.

[20] W3. *RDF 1.1 Concepts and Abstract Syntax*. Disponível em: <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.

[21] BOLLACKER, K. et al. *Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge.* [S.l.], 2008. 1247–1249 p.

[22] NICKEL, M. et al. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, v. 104, n. 1, p. 11–33, Jan 2016. ISSN 0018-9219.

[23] MURPHY, B.; TALUKDAR, P.; MITCHELL, T. Learning effective and interpretable semantic models using non-negative sparse embedding. In: *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, 2012. p. 1933–1950. Disponível em: <https://www.aclweb.org/anthology/C12-1118>.

[24] CARLSON, A.; BETTERIDGE, J.; KISIEL, B. Toward an Architecture for Never-Ending Language Learning. *In Proceedings of the Conference on Artificial Intelligence (AAAI) (2010)*, p. 1306–1313, 2010. ISSN 1098-2345.

[25] LAO, N.; MITCHELL, T.; COHEN, W. W. Random Walk Inference and Learning in A Large Scale Knowledge Base. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Edinburgh, United Kingdom: Association for Computational Linguistics, 2011. p. 529–539. Disponível em: <https://www.cs.cmu.edu/ tom/pubs/lao-emnlp11.pdf>.

[26] GARDNER, M.; MITCHELL, T. M. Efficient and expressive knowledge base completion using subgraph feature extraction. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015.* EMNLP, 2015. p. 1488–1498. Disponível em: <https://www.aclweb.org/anthology/D15-1173/>.

[27] GUSMÃO, A. C. et al. Interpreting Embedding Models of Knowledge Bases : A Pedagogical Approach. In: *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018).* [S.l.: s.n.], 2018. p. 79–86.

[28] WANG, Y. et al. On evaluating embedding models for knowledge base completion. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019).* Florence, Italy: Association for Computational Linguistics, 2019. p. 104–112. Disponível em: <https://www.aclweb.org/anthology/W19-4313>.

[29] BORDES, A. et al. Translating Embeddings for Modeling Multi-Relational Data. *Advances in Neural Information Processing Systems*, p. 2787–2795, 2013. ISSN 10495258.

[30] WANG, Z. et al. Knowledge Graph Embedding by Translating on Hyperplanes. In: *Twenty-Eighth AAAI Conference on Artificial Intelligence.* [s.n.], 2014. p. 1112–1119. Disponível em: <https://pdfs.semanticscholar.org/2a3f/862199883ceff5e3c74126f0c80770653e05.pdf>.

[31] HENK, V. et al. Metaresearch recommendations using knowledge graph embeddings. 2018. Disponível em: <https://recnlp2019.github.io/papers/RecNLP2019$_{p}aper_{2}0.pdf¿$.

[32] HUANG, X. et al. Knowledge graph embedding based question answering. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* New York, NY, USA: ACM, 2019. (WSDM '19), p. 105–113. ISBN 978-1-4503-5940-5. Disponível em: <http://doi.acm.org/10.1145/3289600.3290956>.

[33] DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. *arXiv*, 2017. Disponível em: <https://arxiv.org/abs/1702.08608>.

[34] GRICE, H. P. Logic and conversation. In: COLE, P.; MORGAN, J. L. (Ed.). *Syntax and Semantics: Vol. 3: Speech Acts*. New York: Academic Press, 1975. p. 41–58. Disponível em: <http://www.ucl.ac.uk/ls/studypacks/Grice-Logic.pdf>.

[35] ANTAKI, C.; LEUDAR, I. Explaining in conversation: Towards an argument model. *European Journal of Social Psychology*, v. 22, n. 2, p. 181–194, 1992. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ejsp.2420220206>.

[36] WALTON, D. A dialogue system specification for explanation. *Synthese*, v. 182, n. 3, p. 349–374, 2011. Disponível em: <https://doi.org/10.1007/s11229-010-9745-z>.

[37] WALTON, D. A new dialectical theory of explanation. *Philosophical Explorations*, Taylor & Francis Group, v. 7, n. 1, p. 71–89, 2004.

[38] MADUMAL, P. et al. A grounded interaction protocol for explainable artificial intelligence. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2019. (AAMAS '19), p. 1033–1041. ISBN 978-1-4503-6309-9. Disponível em: <http://dl.acm.org/citation.cfm?id=3306127.3331801>.

[39] WEISSTEIN, F. L. et al. Examining impacts of negative reviews and purchase goals on consumer purchase decision. *Journal of Retailing and Consumer Services*, v. 39, p. 201–207, 2017. ISSN 0969-6989. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0969698917303120>.

[40] BAMBAUER-SACHSE, S.; MANGOLD, S. Brand equity dilution through negative online word-of-mouth communication. *Journal of Retailing and Consumer Services*, v. 18, n. 1, p. 38–45, 2011. ISSN 0969-6989. Disponível em: <https://www.sciencedirect.com/science/article/pii/S096969891000086X>.

[41] SEN, S.; LERMAN, D. Why are you telling me this? an examination into negative consumer reviews on the web. *Journal of Interactive Marketing*, v. 21, n. 4, p. 76–94, 2007. ISSN 1094-9968. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1094996807700397>.

[42] LEE, J.; PARK, D.-H.; HAN, I. The effect of negative online consumer reviews on product attitude: An information processing view. *Electronic Commerce Research and Applications*, v. 7, n. 3, p. 341–352, 2008. ISSN 1567-4223. Special Section: New Research from the 2006 International Conference on Electronic Commerce. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1567422307000415>.

[43] MILANO, S.; TADDEO, M.; FLORIDI, L. Recommender systems and their ethical challenges. *AI & SOCIETY*, v. 35, 2020. Disponível em: <https://doi.org/10.1007/s00146-020-00950-y>.

[44] FLOYD, K. et al. How online product reviews affect retail sales: A meta-analysis. *Journal of Retailing*, v. 90, n. 2, p. 217–232, 2014. ISSN 0022-4359. Empirical Generalizations in Retailing. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0022435914000293>.

[45] XIE, Q. et al. An interpretable knowledge transfer model for knowledge base completion. In: *ACL*. [S.l.: s.n.], 2017.

[46] KAZEMI, S. M.; POOLE, D. Simple embedding for link prediction in knowledge graphs. In: BENGIO, S. et al. (Ed.). *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018. p. 4284–4295. Disponível em: <http://papers.nips.cc/paper/7682-simple-embedding-for-link-prediction-in-knowledge-graphs.pdf>.

[47] ZHANG, W. et al. Interaction embeddings for prediction and explanation in knowledge graphs. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2019. (WSDM '19), p. 96–104. ISBN 978-1-4503-5940-5. Disponível em: <http://doi.acm.org/10.1145/3289600.3291014>.

[48] DREUNDEFINEDLER, K.; NGOMO, A.-C. N. Time-efficient execution of bounded jaro-winkler distances. In: *Proceedings of the 9th International Conference on Ontology Matching - Volume 1317*. Aachen, DEU: CEUR-WS.org, 2014. (OM'14), p. 37–48.

[49] MANNING, C. D.; SCHüTZE, H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999. ISBN 0-262-13360-1.

[50] LI, J.; FAN, Q.; ZHANG, K. Keyword extraction based on tf/idf for chinese news document. *Wuhan University Journal of Natural Sciences*, v. 12, n. 5, p. 917–921, Sep 2007. ISSN 1993-4998. Disponível em: <https://doi.org/10.1007/s11859-007-0038-4>.

[51] AUER, S. et al. Dbpedia: A nucleus for a web of open data. In: ABERER, K. et al. (Ed.). *The Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 722–735. ISBN 978-3-540-76298-0.

[52] CORREA, A. M. G. Jupiterweb chatbot. *Projeto de Formatura EPUSP*, 2018.

[53] ROMAN, N. T.; PIWEK, P.; CARVALHO, A. M. B. R. *A Web-Based Experiment on Dialogue Summarisation*. [S.l.], 2005.

[54] ROMAN, N.; PIWEK, P.; CARVALHO, A. A web-experiment on dialogue classification. 01 2006.

[55] LIKERT, R. A technique for the measurement of attitudes. *Archives of Psychology*, v. 140, p. 1–55, 1932.

[56] TINTAREV, N. Explanations of recommendations. In: *Proceedings of the 2007 ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2007. (RecSys '07), p. 203–206. ISBN 978-1-59593-730–8. Disponível em: <http://doi.acm.org/10.1145/1297231.1297275>.

[57] BONNER, S. et al. A review of biomedical datasets relating to drug discovery: A knowledge graph perspective. *arXiv preprint arXiv:2102.10062*, 2021.

[58] PALIWAL, S. et al. Preclinical validation of therapeutic targets predicted by tensor factorization on heterogeneous graphs. *Sci Rep*, v. 10, n. 1, p. 18250, 10 2020.

[59] TROUILLON, T. et al. Complex Embeddings for Simple Link Prediction. In: *33rd International Conference on Machine Learning*. [s.n.], 2016. Disponível em: <https://arxiv.org/pdf/1606.06357.pdf>.

Appendices

A APPENDICES

A.1 USPedia Construction

The entity linking procedure used to link keywords, from USP undergraduate courses syllabus, to DBpedia entities is the SparQL query depicted in Figure 22. Our procedure considers a match if the redirects, similar to synonyms, of a given DBpedia entity or its label itself contains the keyword string. We enforce one-to-one cardinality among keywords and DBpedia entities by limiting the number of entities a keyword can match to one. Conflicts where multiple entities contain a given keyword are solved arbitrarily through alphabetic order. We employed the virtuoso DBpedia SparQL API [1] to run the queries.

```
PREFIX  rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX   dbo:  <http://dbpedia.org/ontology/>
PREFIX   bif:  <bif:>

SELECT DISTINCT ?item ?label ?description
WHERE
  {
    {
      ?item  rdfs:label    ?label .
      ?label bif:contains  "<KEYWORD>" .
      ?item  dbo:abstract  ?description .
      FILTER (lang(?description) = 'pt')
      FILTER (lang(?label) = 'pt')
    }
    UNION
    {
      ?x     rdfs:label    ?label .
      ?label bif:contains  "<KEYWORD>" .
      ?x     dbo:wikiPageRedirects ?item .
      ?item  dbo:abstract  ?description .
      FILTER (lang(?description) = 'pt')
      FILTER (lang(?label) = 'pt')
    }
  }
ORDER BY ?label
LIMIT 1
```

Figure 22: Entity linking SparQL query for undergraduate courses keywords.

[1] https://dbpedia.org/sparql/

A.2 Application-Level Evaluation: November 2019

In this section we present the appendix for the application-level evaluation realized in November 2019. Figure 23 depicts the mean results from the user survey, in which two explanation methods PRED (embedding-based) and TRUE (graph-based) where compared among the five explanation aims (engagement, transparency, trust, persuasion and effectiveness). Even though the mean results for PRED method were consistently higher among all aims, we were not able to obtain statistical significance to tell whether PRED outperformed TRUE method in this experiment.

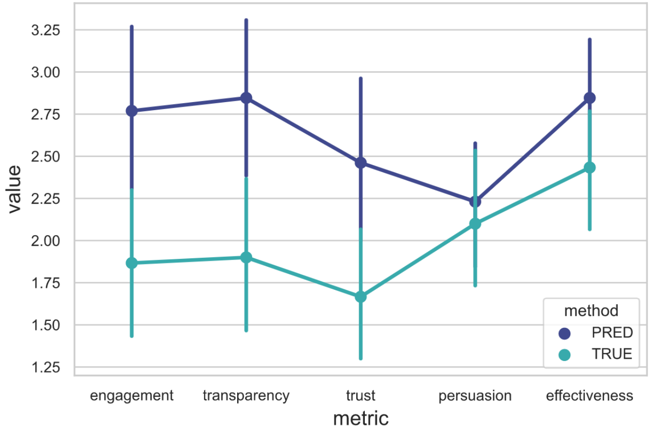
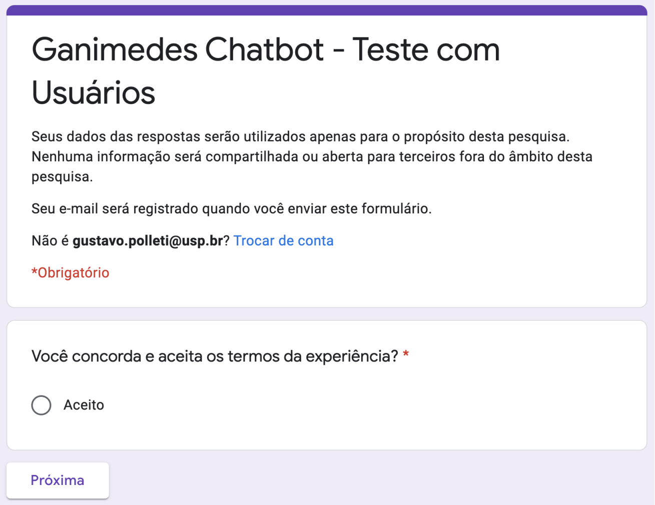


Figure 23: Visual representation for explanation metrics average scores.

Figures 24, 25, 26a, 26b and 26 depicts the survey screens presented to subjects during the experiment.



Ganimedes Chatbot - Teste com Usuários

Seus dados das respostas serão utilizados apenas para o propósito desta pesquisa. Nenhuma informação será compartilhada ou aberta para terceiros fora do âmbito desta pesquisa.

Seu e-mail será registrado quando você enviar este formulário.

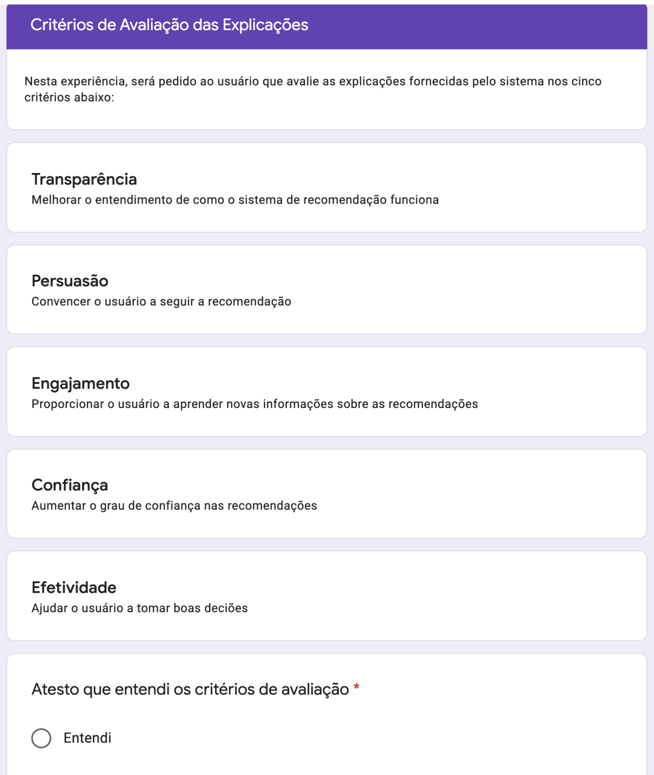
Não é **gustavo.polleti@usp.br**? Trocar de conta

*Obrigatório

Você concorda e aceita os termos da experiência? *

○ Aceito

Figure 24: Experiment flow first step. Subject is asked to accept disclosure agreement.



Critérios de Avaliação das Explicações

Nesta experiência, será pedido ao usuário que avalie as explicações fornecidas pelo sistema nos cinco critérios abaixo:

Transparência
Melhorar o entendimento de como o sistema de recomendação funciona

Persuasão
Convencer o usuário a seguir a recomendação

Engajamento
Proporcionar o usuário a aprender novas informações sobre as recomendações

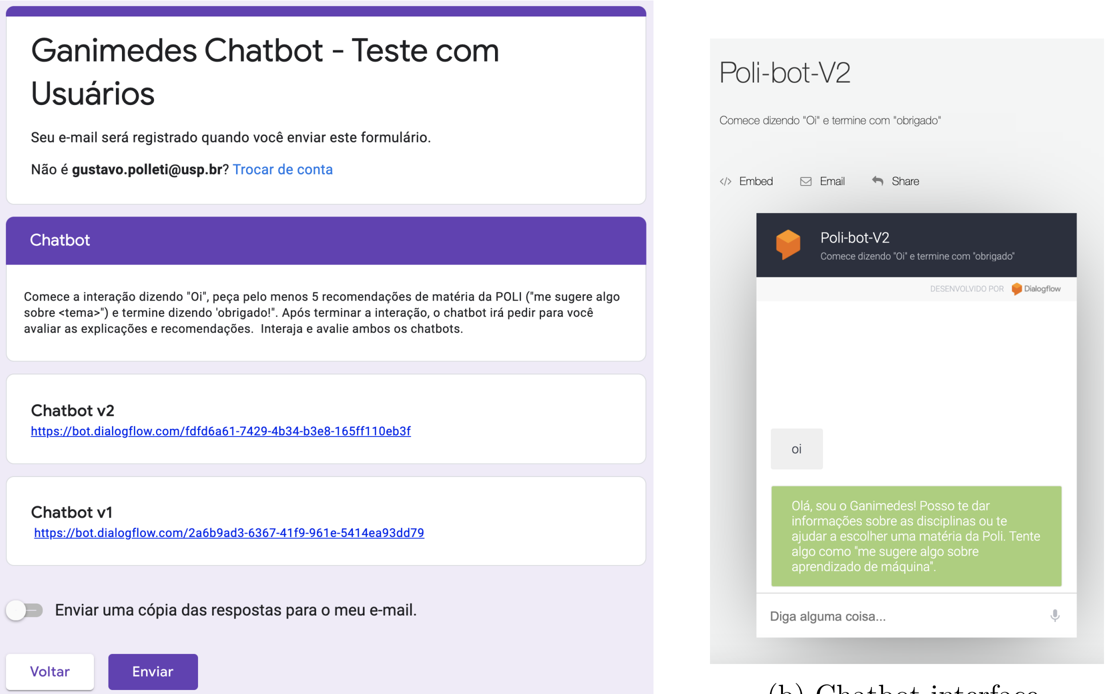
Confiança
Aumentar o grau de confiança nas recomendações

Efetividade
Ajudar o usuário a tomar boas deciões

Atesto que entendi os critérios de avaliação *

○ Entendi

Figure 25: Experiment flow second step. Subject is asked to acknowledge evaluation metrics.



Ganimedes Chatbot - Teste com Usuários

Seu e-mail será registrado quando você enviar este formulário.

Não é **gustavo.polleti@usp.br**? Trocar de conta

Chatbot

Comece a interação dizendo "Oi", peça pelo menos 5 recomendações de matéria da POLI ("me sugere algo sobre <tema>") e termine dizendo 'obrigado!". Após terminar a interação, o chatbot irá pedir para você avaliar as explicações e recomendações. Interaja e avalie ambos os chatbots.

Chatbot v2
https://bot.dialogflow.com/fdfd6a61-7429-4b34-b3e8-165ff110eb3f

Chatbot v1
https://bot.dialogflow.com/2a6b9ad3-6367-41f9-961e-5414ea93dd79

Enviar uma cópia das respostas para o meu e-mail.

Poli-bot-V2

Comece dizendo "Oi" e termine com "obrigado"

Poli-bot-V2
Comece dizendo "Oi" e termine com "obrigado"

oi

Olá, sou o Ganimedes! Posso te dar informações sobre as disciplinas ou te ajudar a escolher uma matéria da Poli. Tente algo como "me sugere algo sobre aprendizado de máquina".

Diga alguma coisa...

(a) Links to chatbots.

(b) Chatbot interface.

Figure 26: Experiment flow fifth step. Subject is asked to evaluate two chatbots with different explanation methods, respectively: embedding-based (PRED), graph-based (TRUE).

A.3 Human-Level Evaluation: July 2020

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

1. Do reasons for/against have value for users?
2. Do reasons against reduce 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 engineering undergraduate students from EPUSP, and asked them to evaluate two recommendation setups, 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. The described experimental procedure is depicted in Figure 27. First, the user is asked to accept an disclosure agreement granting access to her responses for the purpose of this research (step 1); this step is mandatory. Next, each subject first received a recommendation and one reason for, and was asked to select an item (step 2); then the subject received a recommendation with one reason for and one reason against, and was again asked to select an item (step 3). Note that we avoided presenting too many reasons at once. Figure 28 depicts the information presented. It is important to mention that this experiment purpose was focused solely on the evaluation of user perception of reasons against in explanations in the context of recommendation, so it does not address questions about application domain or implementation constraints.

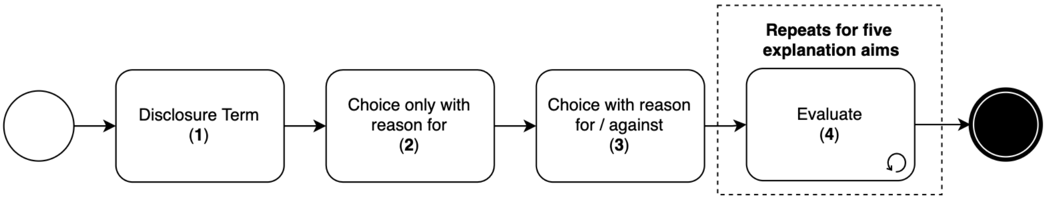

Repeats for five explanation aims
Disclosure Term (1)
Choice only with reason for (2)
Choice with reason for / against (3)
Evaluate (4)

Figure 27: Experiment flow diagram.

Each subject then evaluated the two recommendation systems individually in the five explanation aims (step 4 33), which are represented by the questionnaire in Table 9. Each subject ranked each system with respect to each explanation metric using a survey-based

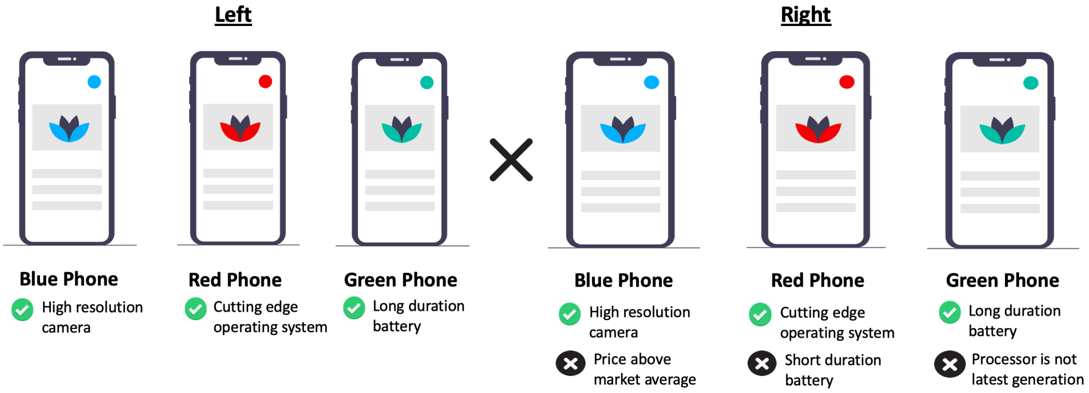


Figure 28: Experiment: just one reason for (left); one reason for and one reason against (right).

Likert scale from 1 to 5. Finally, each subject could write a short free text with thoughts about the experience; this final screen is shown in Figure 34. The screens presented to the subjects are presented at the end of this section.

Before we move forward, it is important to highlight the key assumptions and limitations of this experiment. First, since this is a *within subjects* experiment and the explanation approaches were presented in the same order for all subjects (only reasons for and reasons for/against respectively) these results disregard any learning effects that could potentially lead to a positive bias towards the second setup. In addition, since we do not collect demographic data from our subjects, we cannot tell whether these results generalize for other populations. Finally, our sample size was limited, so all conclusions are only preliminary indications and further research is required.

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

Figure 29 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 value with respect to trust, engagement and effectiveness. Figure 29 shows that there was a divergence amongst users about whether the proposed explanation paradigm increases transparency. Indeed, 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 4, this does not mean reasons for/against were adverse to transparency; it means that they were as good as just reasons for.

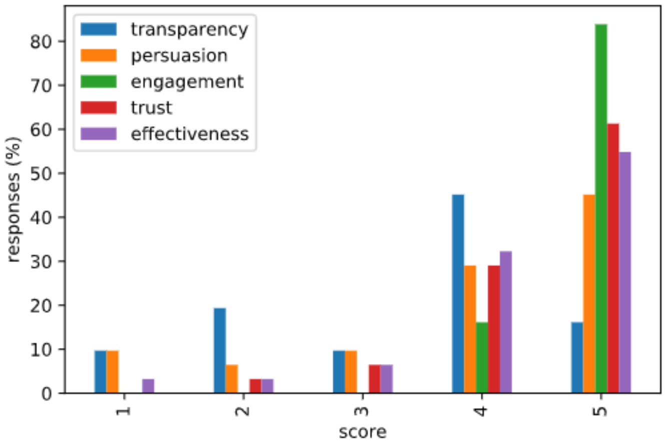


Figure 29: Visual representation for explanation metrics average scores.

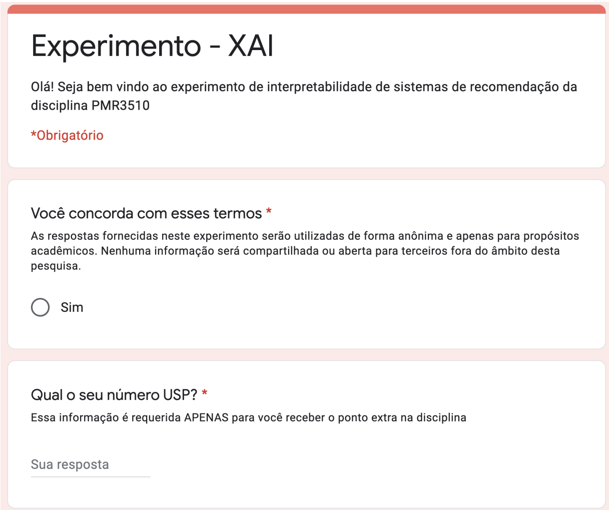
We expected a possible drawback of our proposal would be a reduction in persuasion (as reasons against might make the users less likely to follow recommendations). By doing a further analysis of textual comments, we found out that persuasion increases are produced by higher trust in the recommendation system. 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 indicated that many users expect the recommendation system 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 agree with our hypothesis while suggesting that reasons against have a significant positive impact on the user decision-making process. Furthermore, a full 45% of our test subjects changed their initial choices after we presented reasons against.



Experimento - XAI

Olá! Seja bem vindo ao experimento de interpretabilidade de sistemas de recomendação da disciplina PMR3510

*Obrigatório

Você concorda com esses termos *

As respostas fornecidas neste experimento serão utilizadas de forma anônima e apenas para propósitos acadêmicos. Nenhuma informação será compartilhada ou aberta para terceiros fora do âmbito desta pesquisa.

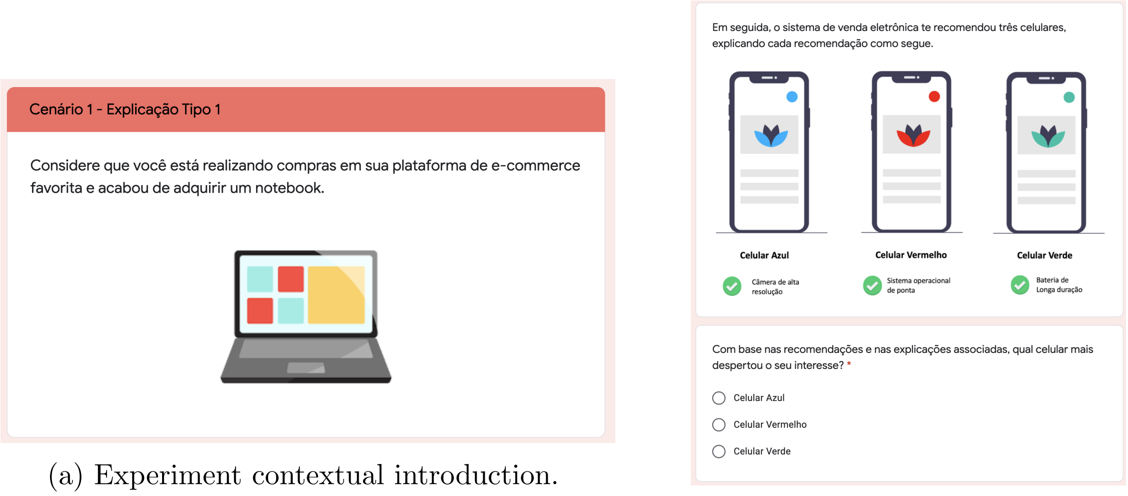
○ Sim

Qual o seu número USP? *

Essa informação é requerida APENAS para você receber o ponto extra na disciplina

Sua resposta

Figure 30: Experiment flow first step. Subject is asked to accept disclosure agreement.



Cenário 1 - Explicação Tipo 1

Considere que você está realizando compras em sua plataforma de e-commerce favorita e acabou de adquirir um notebook.

(a) Experiment contextual introduction.

Em seguida, o sistema de venda eletrônica te recomendou três celulares, explicando cada recomendação como segue.

Celular Azul
✓ Câmera de alta resolução

Celular Vermelho
✓ Sistema operacional de ponta

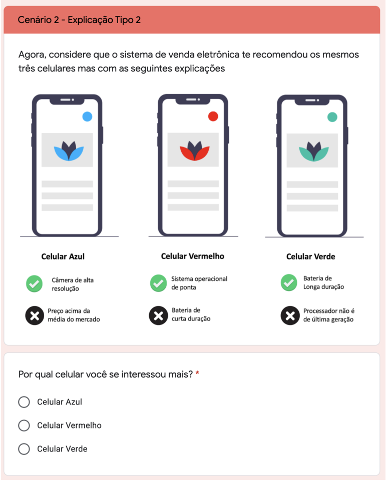
Celular Verde
✓ Bateria de Longa duração

Com base nas recomendações e nas explicações associadas, qual celular mais despertou o seu interesse? *

○ Celular Azul
○ Celular Vermelho
○ Celular Verde

(b) Three recommendation options with only reasons for.

Figure 31: Experiment flow second step. Subject is asked to choose among three recommendation when presented with only supporting reasons.



Cenário 2 - Explicação Tipo 2

Agora, considere que o sistema de venda eletrônica te recomendou os mesmos três celulares mas com as seguintes explicações

Celular Azul
✓ Câmera de alta resolução
✗ Preço acima da média do mercado

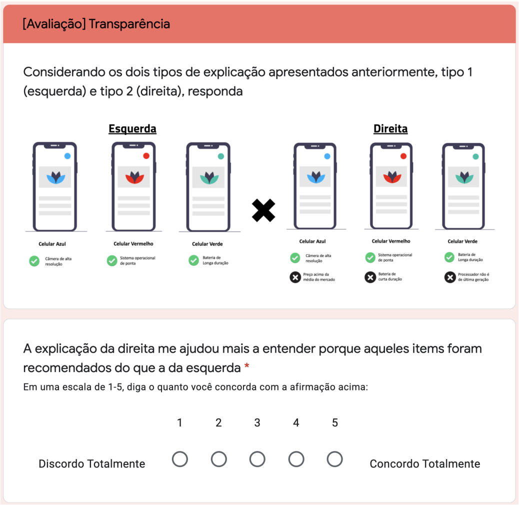
Celular Vermelho
✓ Sistema operacional de ponta
✗ Bateria de curta duração

Celular Verde
✓ Bateria de Longa duração
✗ Processador não é de última geração

Por qual celular você se interessou mais? *

○ Celular Azul
○ Celular Vermelho
○ Celular Verde

Figure 32: Experiment flow third step. Subject is asked to choose again after being presented with reasons against each option.



[Avaliação] Transparência

Considerando os dois tipos de explicação apresentados anteriormente, tipo 1 (esquerda) e tipo 2 (direita), responda

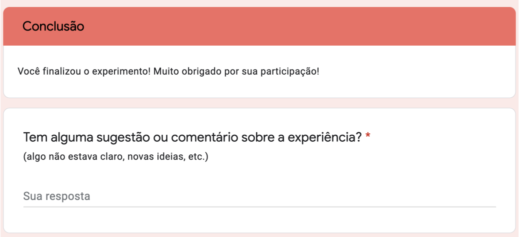
Esquerda
Direita
Celular Azul
✓ Câmera de alta resolução
Celular Vermelho
✓ Sistema operacional de ponta
Celular Verde
✓ Bateria de Longa duração
✖
Celular Azul
✓ Câmera de alta resolução
✗ Preço acima da média do mercado
Celular Vermelho
✓ Sistema operacional de ponta
✗ Bateria de curta duração
Celular Verde
✓ Bateria de Longa duração
✗ Processador não é de última geração

A explicação da direita me ajudou mais a entender porque aqueles items foram recomendados do que a da esquerda *

Em uma escala de 1-5, diga o quanto você concorda com a afirmação acima:

	1	2	3	4	5	
Discordo Totalmente	○	○	○	○	○	Concordo Totalmente

Figure 33: Experiment flow fourth step. Subject is asked to evaluate the two explanation scenarios, only reasons for (left) and reasons for/against (right), according to an explanation aim using a Likert-scale.



Conclusão

Você finalizou o experimento! Muito obrigado por sua participação!

Tem alguma sugestão ou comentário sobre a experiência? *

(algo não estava claro, novas ideias, etc.)

Sua resposta

Figure 34: Experiment flow final step. Subject is asked to write a general comment about the experiment.

A.4 Application-Level Evaluation: November 2020

In this section we present the appendix for the application-level evaluation realized in November 2020. Figure 35 depicts the mean results from the user survey, in which two reasons against schemes (S1 and S5) where compared against a reasons for only baseline among the five explanation aims (engagement, transparency, trust, persuasion and effectiveness). We were not able to obtain statistical significance to tell whether any methods outperformed the baseline in this experiment.

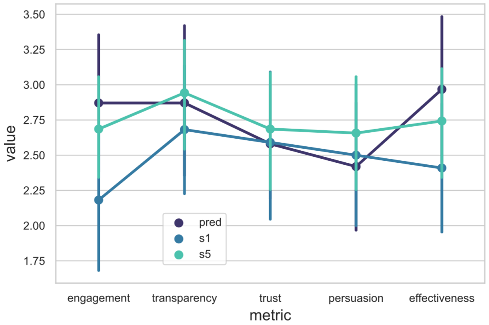
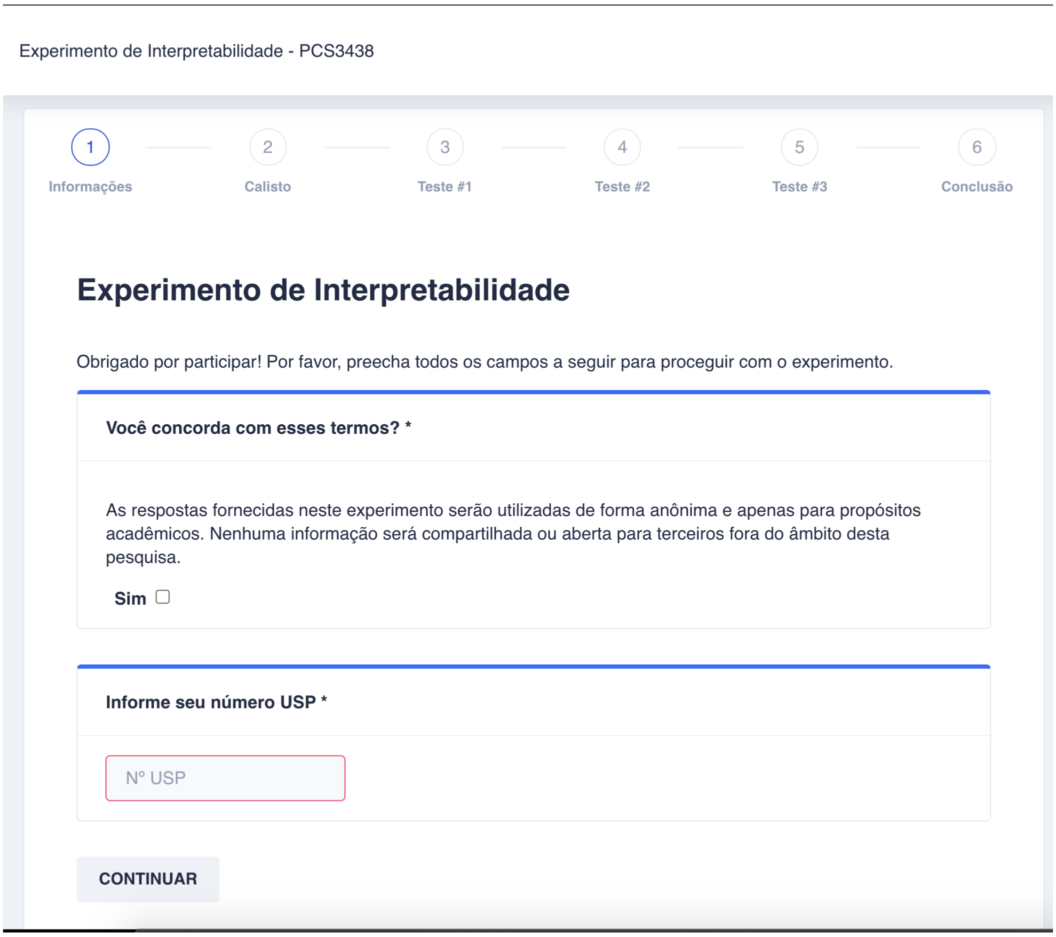


Figure 35: Visual representation for explanation metrics average scores.

Figures 36, 37, 38 depicts the survey screens presented to subjects during the experiment.



Experimento de Interpretabilidade - PCS3438

Experimento de Interpretabilidade

Obrigado por participar! Por favor, preecha todos os campos a seguir para proceguir com o experimento.

Você concorda com esses termos? *

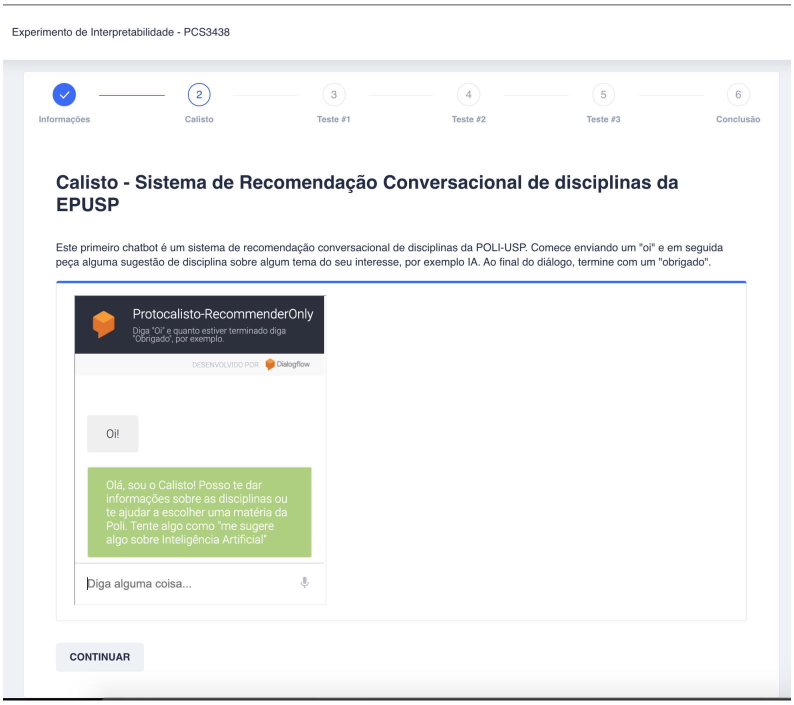
As respostas fornecidas neste experimento serão utilizadas de forma anônima e apenas para propósitos acadêmicos. Nenhuma informação será compartilhada ou aberta para terceiros fora do âmbito desta pesquisa.

Sim □

Informe seu número USP *

Nº USP

Figure 36: Experiment flow first step. Subject is asked to accept disclosure agreement.



Experimento de Interpretabilidade - PCS3438

Calisto - Sistema de Recomendação Conversacional de disciplinas da EPUSP

Este primeiro chatbot é um sistema de recomendação conversacional de disciplinas da POLI-USP. Comece enviando um "oi" e em seguida peça alguma sugestão de disciplina sobre algum tema do seu interesse, por exemplo IA. Ao final do diálogo, termine com um "obrigado".

Protocalisto-RecommenderOnly

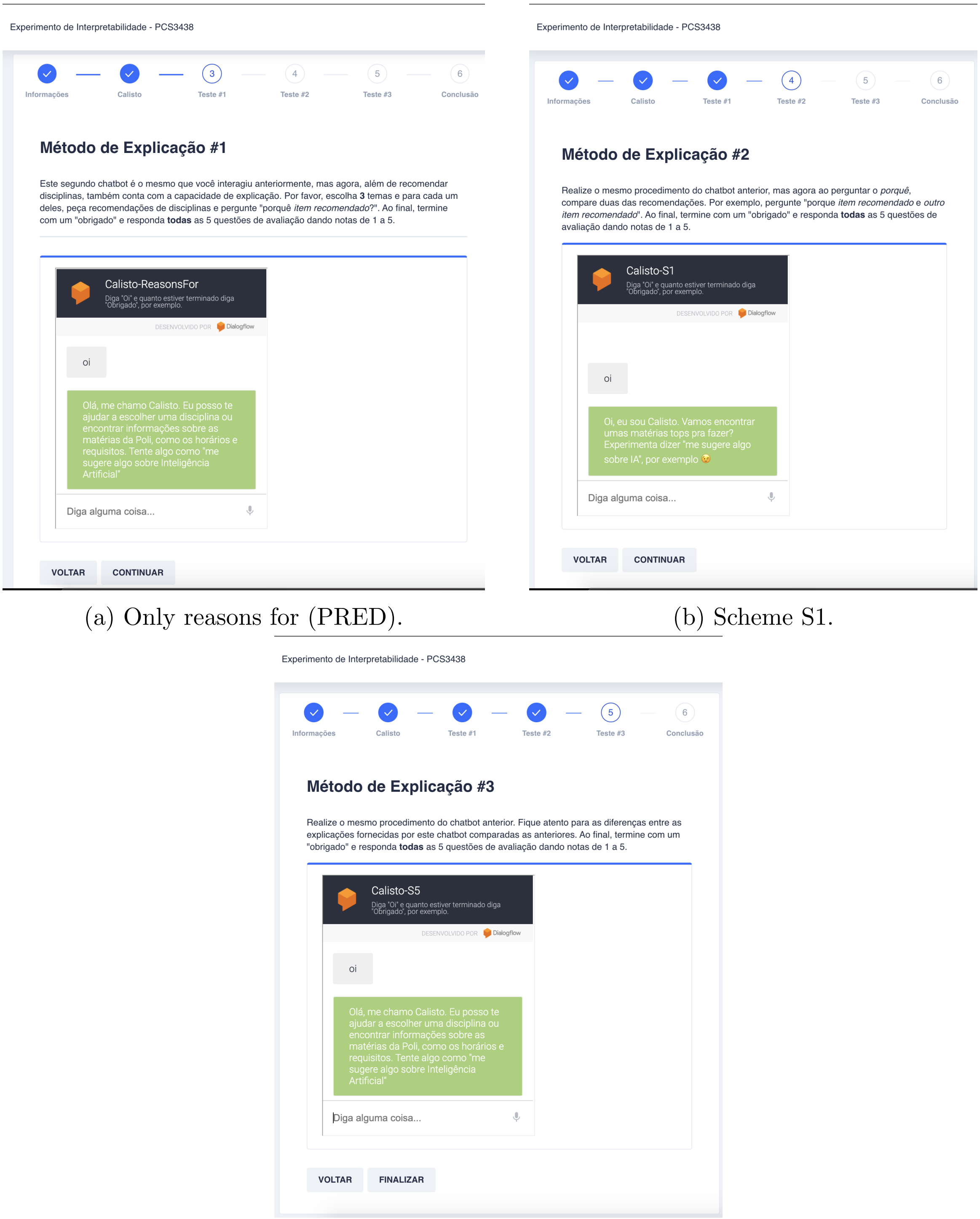
Diga "Oi" e quanto estiver terminado diga "Obrigado", por exemplo.

Oi!

Olá, sou o Calisto! Posso te dar informações sobre as disciplinas ou te ajudar a escolher uma matéria da Poli. Tente algo como "me sugere algo sobre Inteligência Artificial"

Diga alguma coisa...

Figure 37: Experiment flow second step. Subject is asked to freely interact with a conversational recommendation system as a warm up.



Experimento de Interpretabilidade - PCS3438

Método de Explicação #1

Este segundo chatbot é o mesmo que você interagiu anteriormente, mas agora, além de recomendar disciplinas, também conta com a capacidade de explicação. Por favor, escolha **3** temas e para cada um deles, peça recomendações de disciplinas e pergunte "porquê *item recomendado*?". Ao final, termine com um "obrigado" e responda **todas** as 5 questões de avaliação dando notas de 1 a 5.

Calisto-ReasonsFor

Diga "Oi" e quanto estiver terminado diga "Obrigado", por exemplo.

DESENVOLVIDO POR Dialogflow

oi

Olá, me chamo Calisto. Eu posso te ajudar a escolher uma disciplina ou encontrar informações sobre as matérias da Poli, como os horários e requisitos. Tente algo como "me sugere algo sobre Inteligência Artificial"

Diga alguma coisa...

(a) Only reasons for (PRED).

Experimento de Interpretabilidade - PCS3438

Método de Explicação #2

Realize o mesmo procedimento do chatbot anterior, mas agora ao perguntar o *porquê*, compare duas das recomendações. Por exemplo, pergunte "porque *item recomendado* e *outro item recomendado*". Ao final, termine com um "obrigado" e responda **todas** as 5 questões de avaliação dando notas de 1 a 5.

Calisto-S1

Diga "Oi" e quanto estiver terminado diga "Obrigado", por exemplo.

DESENVOLVIDO POR Dialogflow

oi

Oi, eu sou Calisto. Vamos encontrar umas matérias tops pra fazer? Experimenta dizer "me sugere algo sobre IA", por exemplo 😉

Diga alguma coisa...

(b) Scheme S1.

Experimento de Interpretabilidade - PCS3438

Método de Explicação #3

Realize o mesmo procedimento do chatbot anterior. Fique atento para as diferenças entre as explicações fornecidas por este chatbot comparadas as anteriores. Ao final, termine com um "obrigado" e responda **todas** as 5 questões de avaliação dando notas de 1 a 5.

Calisto-S5

Diga "Oi" e quanto estiver terminado diga "Obrigado", por exemplo.

DESENVOLVIDO POR Dialogflow

oi

Olá, me chamo Calisto. Eu posso te ajudar a escolher uma disciplina ou encontrar informações sobre as matérias da Poli, como os horários e requisitos. Tente algo como "me sugere algo sobre Inteligência Artificial"

Diga alguma coisa...

(c) Scheme S5.

Figure 38: Experiment flow third step. In order evaluation of three explanation schemes, respectively: only reasons for (PRED), S1 and S5.

A.5 Human-Level Evaluation: April 2021

In this section we present the appendix for the human-level evaluation realized in Abril 2021. Figure 39 presents the experiment subjects demographics distribution over self-declared: age, gender, place of birth and tech affinity. Our sample population was heavily concentrated on male young (age between 20 and 35) southeasters with high affinity to technology.

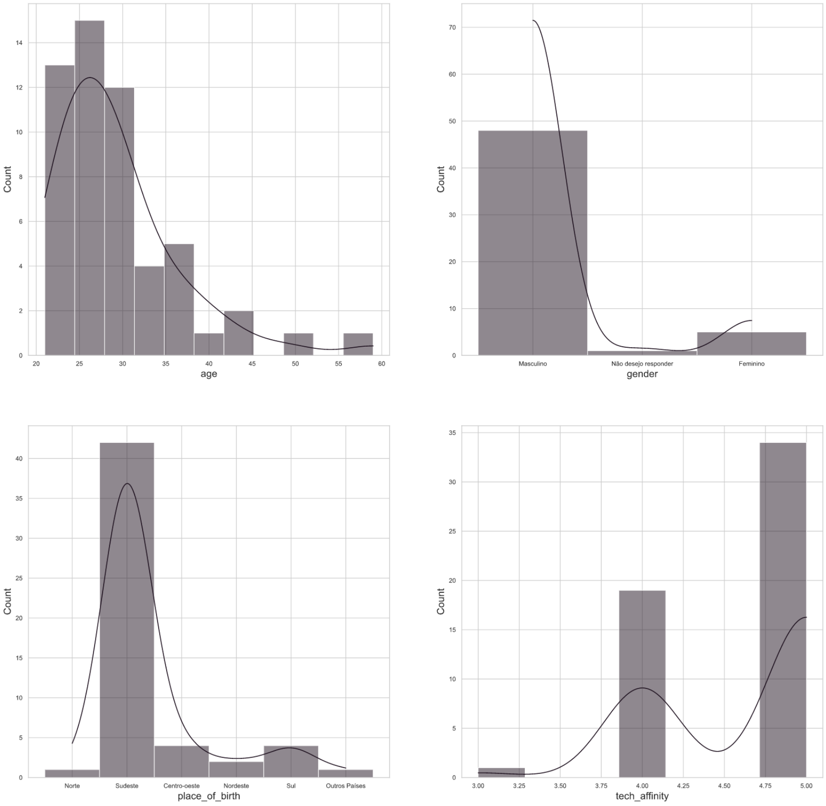
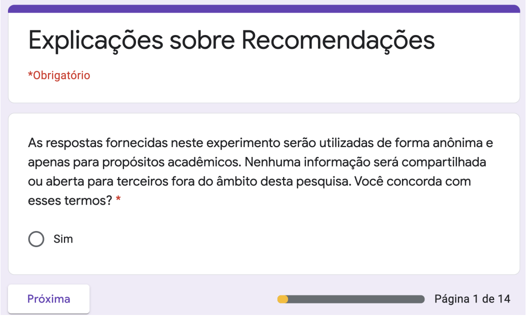


Figure 39: User study subjects demographic distribution according to age, gender, place of birth (Brazil macro regions) and tech affinity.

Figures 40, 41, 42, 43 and 44 depicts the survey screens presented to subjects during the experiment.



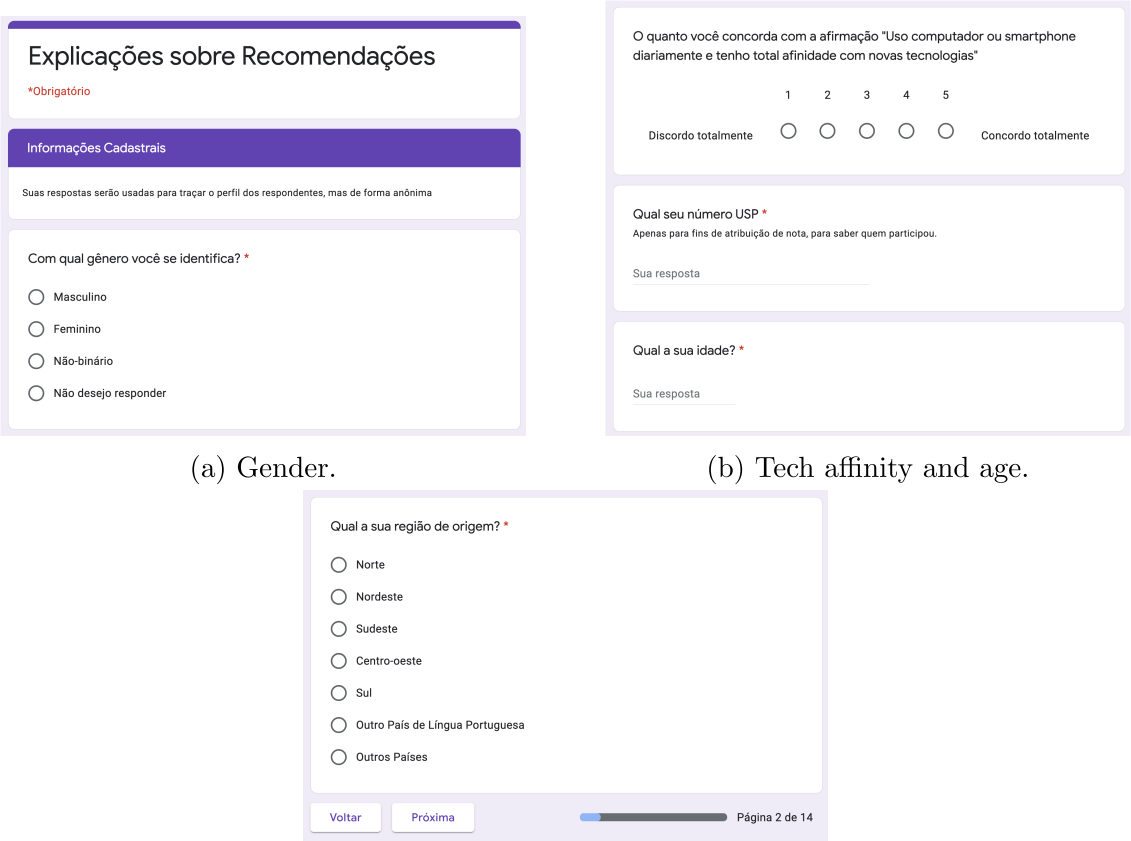
Explicações sobre Recomendações

*Obrigatório

As respostas fornecidas neste experimento serão utilizadas de forma anônima e apenas para propósitos acadêmicos. Nenhuma informação será compartilhada ou aberta para terceiros fora do âmbito desta pesquisa. Você concorda com esses termos? *

○ Sim

Figure 40: Experiment flow first step. Subject is asked to accept disclosure agreement.



Explicações sobre Recomendações

*Obrigatório

Informações Cadastrais

Suas respostas serão usadas para traçar o perfil dos respondentes, mas de forma anônima

Com qual gênero você se identifica? *

○ Masculino

○ Feminino

○ Não-binário

○ Não desejo responder

O quanto você concorda com a afirmação "Uso computador ou smartphone diariamente e tenho total afinidade com novas tecnologias"

	1	2	3	4	5	
Discordo totalmente	○	○	○	○	○	Concordo totalmente

Qual seu número USP *

Apenas para fins de atribuição de nota, para saber quem participou.

Sua resposta

Qual a sua idade? *

Sua resposta

(a) Gender.

(b) Tech affinity and age.

Qual a sua região de origem? *

○ Norte

○ Nordeste

○ Sudeste

○ Centro-oeste

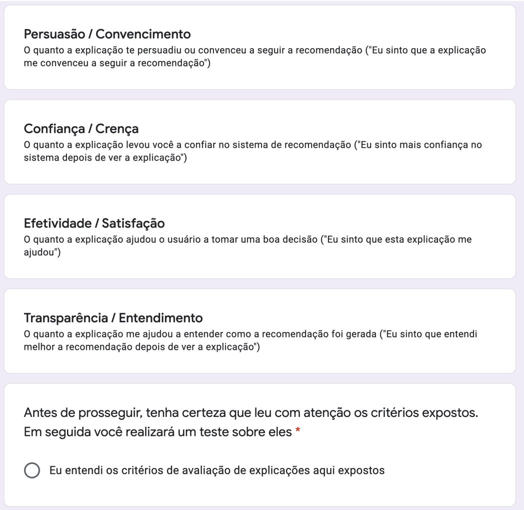
○ Sul

○ Outro País de Língua Portuguesa

○ Outros Países

(c) Place of birth. Each option corresponds to a Brazil macro region, non-Portuguese speaker country or other.

Figure 41: Experiment flow second step. Subject is asked to fill forms with her demographic information.



Persuasão / Convencimento
O quanto a explicação te persuadiu ou convenceu a seguir a recomendação ("Eu sinto que a explicação me convenceu a seguir a recomendação")

Confiança / Crença
O quanto a explicação levou você a confiar no sistema de recomendação ("Eu sinto mais confiança no sistema depois de ver a explicação")

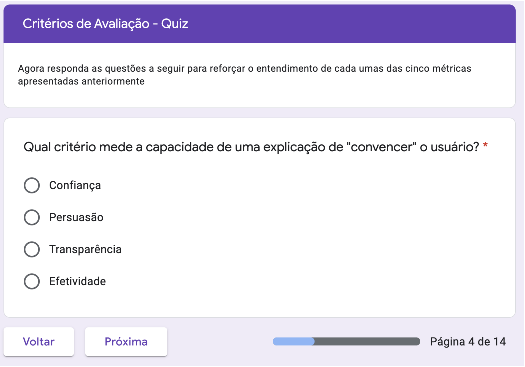
Efetividade / Satisfação
O quanto a explicação ajudou o usuário a tomar uma boa decisão ("Eu sinto que esta explicação me ajudou")

Transparência / Entendimento
O quanto a explicação me ajudou a entender como a recomendação foi gerada ("Eu sinto que entendi melhor a recomendação depois de ver a explicação")

Antes de prosseguir, tenha certeza que leu com atenção os critérios expostos. Em seguida você realizará um teste sobre eles *

○ Eu entendi os critérios de avaliação de explicações aqui expostos

Figure 42: Experiment flow third step. Subject is asked to acknowledge evaluation metrics.



Critérios de Avaliação - Quiz

Agora responda as questões a seguir para reforçar o entendimento de cada umas das cinco métricas apresentadas anteriormente

Qual critério mede a capacidade de uma explicação de "convencer" o usuário? *

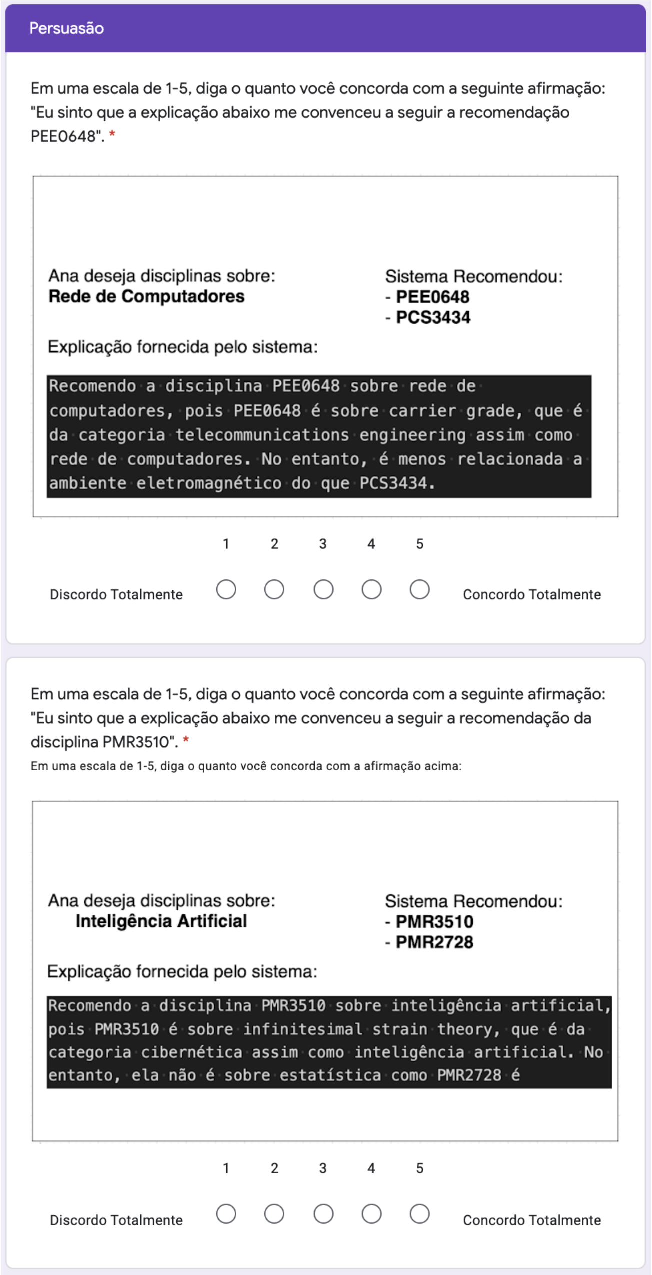
○ Confiança

○ Persuasão

○ Transparência

○ Efetividade

Figure 43: Experiment flow fourth step. Subject is asked to answer a quiz about evaluation metrics. All answers correct are required to proceed.



Persuasão

Em uma escala de 1-5, diga o quanto você concorda com a seguinte afirmação: "Eu sinto que a explicação abaixo me convenceu a seguir a recomendação PEE0648". *

Ana deseja disciplinas sobre:
Rede de Computadores

Sistema Recomendou:
- **PEE0648**
- **PCS3434**

Explicação fornecida pelo sistema:

```
Recomendo a disciplina PEE0648 sobre rede de
computadores, pois PEE0648 é sobre carrier grade, que é
da categoria telecommunications engineering assim como
rede de computadores. No entanto, é menos relacionada a
ambiente eletromagnético do que PCS3434.
```

	1	2	3	4	5	
Discordo Totalmente	○	○	○	○	○	Concordo Totalmente

Em uma escala de 1-5, diga o quanto você concorda com a seguinte afirmação: "Eu sinto que a explicação abaixo me convenceu a seguir a recomendação da disciplina PMR3510". *

Em uma escala de 1-5, diga o quanto você concorda com a afirmação acima:

Ana deseja disciplinas sobre:
Inteligência Artificial

Sistema Recomendou:
- **PMR3510**
- **PMR2728**

Explicação fornecida pelo sistema:

```
Recomendo a disciplina PMR3510 sobre inteligência artificial,
pois PMR3510 é sobre infinitesimal strain theory, que é da
categoria cibernética assim como inteligência artificial. No
entanto, ela não é sobre estatística como PMR2728 é
```

	1	2	3	4	5	
Discordo Totalmente	○	○	○	○	○	Concordo Totalmente

Figure 44: Experiment flow fifth step. Subject is asked to evaluate two explanation schemes according to an explanation aim using a Likert-scale.

A.6 Research Disclosure

The main peer reviewed papers reporting results of this work are:

1. "*Explaining Completions Produced by Embeddings of Knowledge Graphs*", co-authored paper at 2019 European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)

2. "*Explaining Content-Based Recommendations with Topic-Models*", paper at IEEE 2019 Brazilian Conference on Intelligent Systems (BRACIS).

3. "*Faithfully Explaining Predictions of Knowledge Embeddings*", runner-up best paper undergraduate track at 2019 Encontro Nacional de Inteligência Artificial e Computacional (ENIAC).

4. "*Conversational Recommendation Systems within Explanations: Improving Coverage through Knowledge Embeddings*", paper at 2020 AAAI Workshop on Interactive and Conversational Recommendation Systems (WICRS).

5. "*Why should I not follow you? Reasons For and Against in Responsible Recommendation Systems*", paper at 2020 RecSys Workshop on Responsible Recommendation (FAccTRec)