Explaining Drug Discovery Hypotheses Using Knowledge Graph Patterns

Kara Schatz*, Cleber Melo-Filho†, Rada Chirkova*, and Alexander Tropsha†
*North Carolina State University, Raleigh, North Carolina, USA
Emails: kmschat2@ncsu.edu, rychirko@ncsu.edu
†University of North Carolina, Chapel Hill, North Carolina, USA
Emails: cleber@email.unc.edu, alex_tropsha@unc.edu

***Abstract*—Drug discovery is an important process in which biomedical experts identify new potential treatments for diseases. Currently, this requires much time and manual effort, so automating any part of the process would be a significant improvement. Thus, we propose to identify drug candidates via explainable automated fact-checking. That is, given a hypothesized drug-disease treatment relationship, we aim to generate explanations for the hypothesis as a means for determining whether or not the drug could be a potential treatment for the disease. Our goal in this paper is to develop such an approach that is well-suited for the biomedical domain.**

Direct application of existing fact-checking tools faces several challenges since most are not developed for use within the biomedical domain; both the explanation formats and the evaluation metrics are ill-suited for this domain application. We propose explanations in the form of knowledge graph patterns, which directly relate to existing structures used by biomedical experts, as well as evaluation metrics which rely solely on existing evidence present in knowledge graphs and make no domain-specific assumptions. We report experimental results, which suggest that, for the drug discovery task and potentially others, our metrics are accurate, and our explanations are understandable and reasonable to domain experts, as well as useful.

***Index Terms*—drug discovery, knowledge discovery, explainable fact-checking, link prediction, knowledge graph mining**

I. Introduction

Drug discovery is the process by which biomedical experts identify new potential treatments for diseases. This process has two broad steps: (1) identification of candidate drugs and (2) clinical trials of these candidates to test their viability. The first step can be solved by evaluating the truth of candidate facts, which is the common goal of fact-checking and link prediction systems. In this scenario, each candidate fact is a hypothesis that some drug is capable of treating the target disease, and evaluating this fact for a variety of drugs can help identify which are the strongest candidates, i.e., those with the highest truth score.

To this end, the problem of interest is to automate the drug discovery process by posing it as a fact-checking/link prediction problem. Our aim is to propose an automated fact-checking strategy which is well-suited for the biomedical domain. In particular, we aim to develop an explainable approach in which explanations are produced with each candidate fact to support or refute it, thereby allowing biomedical experts to further examine the candidates based on the evidence produced.

Currently, experts spend a considerable amount of time and effort on drug candidate identification because there are so many factors that they must consider, e.g., how well the drug manages the disease or some symptom of the disease, what the potential undesired effects of the drug are, whether or not the drug been used for other treatments, if the drug would be feasible for use on specific patients, etc. Automating this process would provide significant speedups and alleviate the manual efforts required. In addition, producing explanations can help domain experts to understand the machine output, as opposed to blindly accepting it, which would help make this process accepted and understood in the biomedical domain.

Most current fact-checking and link prediction tools were not developed with the biomedical domain in mind, so several challenges arise when trying to directly adopt these tools to address drug discovery. Challenges surrounding the explanations produced include that they are not in a form which is readily understood by biomedical experts, and they require unique explanations to be found for each candidate fact, which does not scale well for drug discovery where there is a large number of hypotheses to be considered [1]–[6]. Challenges surrounding the metrics used for evaluating explanations include that they make assumptions about the domain of interest that are not held in the biomedical domain, they evaluate how an explanation was derived, as opposed to evaluating the explanation itself, and they provide a local evaluation, i.e., specific to the fact, which does not consider the other data present [3], [4].

In this paper, we address these challenges by proposing a novel view of explanations along with three metrics for evaluating said explanations. Our proposed explanations take the form of knowledge graph patterns, which are subgraphs consisting only of type labels, i.e., no specific entities. These patterns are readily understood by biomedical experts as they are closely related to the concept of *clinical outcome pathways* (COPs). A COP is a sequence of biological events that explain the mechanism of action of a drug [7]. Fig. 1 shows an example of both a knowledge graph pattern and a COP which presents one possible way that a drug may act on

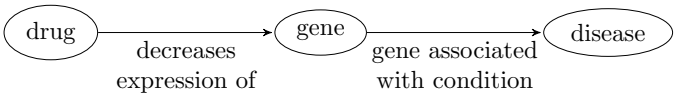


Fig. 1. An example of a knowledge graph pattern which consists only of edge and node type labels and serves as both an explanation for the relationship between the drug and disease, as well as a clinical outcome pathway revealing to biomedical experts how the drug acts on the disease.

a disease.[1] In addition to being understandable by domain experts, these patterns are general enough to apply to multiple candidate facts, so we can treat them as rules for inferring drug-disease relationships, thereby saving time on having to generate explanations specific to each hypothesis. Moreover, our proposed metrics do not require the domain to meet any assumptions and evaluate each explanation in and of itself, as opposed to how it was generated, by considering the existing data in knowledge graphs as evidence for or against it, which provides a global evaluation.

To the best of our knowledge, [2] is the only work to address fact-checking in the biomedical domain, like we do. Unfortunately, using their method still faces some of the key challenges presented. For example, their explanations are paragraphs, which are slower to understand than our patterns and must be unique for each candidate fact. In addition, their metrics for evaluating such explanations provide only a local evaluation without considering other domain evidence.

Our contributions are as follows:

- We pose drug candidate identification for drug discovery as a fact-checking problem and propose an explainable, automated approach as a solution;
- We present knowledge graph patterns as a novel, useful explanation format and implement an algorithm for deriving these;
- We treat these derived explanations as inference rules which can be applied to future queries beyond those for which they were derived, thereby saving time on future explanation tasks;
- We propose three data-supported scoring metrics for reliable, domain-specific evaluation of explanations;
- We present experimental results on the biomedical domain, specifically addressing drug discovery; our results suggest that, for the task of drug candidate identification and potentially others, our metrics are accurate and capable of identifying strong explanations, and our explanations are not only reasonable to domain experts but are also useful as inference rules for explaining future queries.

The rest of the paper is organized as follows. In Section II, we survey related work. We introduce necessary background in Section III and detail all steps of our approach in Section IV. In Section V, we present the results of our experiments. Finally, in Section VI, we offer conclusions of our work and suggestions for future work.

[1]There are many other COPs of varying length and specificity, see [7].

II. Related Work

A. Explanation Derivation

Producing explanations during automated fact-checking or link prediction has recently received significant attention, see [1]–[6], [8]–[16]. Arguably the most popular approach to explanation generation is evaluation of textual sources via deep learning [1], [2], [8]–[13]. Rule-based approaches, which aim to find paths or subgraphs in knowledge graphs that support facts, are also popular [3]–[6], as well as a variety of other approaches for finding supporting paths in knowledge graphs [14]–[16].

Backward chaining is a standard approach in artificial intelligence for synthesizing inference rules together to form proofs [17] and has been used in many applications over the years, e.g., theorem proving, inference engines, Prolog, etc. Our approach is a backward chaining style algorithm which uses inference rules over knowledge graphs to find explanations for input facts. Reference [4] also adopts a backward chaining approach; however, they consider an entirely different application domain, and their criteria for ideal explanations are not appropriate for our problem. In particular, their ideal explanations are (i) small, (ii) require few rules to generate, and (iii) contain more facts from knowledge graphs than text-based sources. We use knowledge graphs as our only source of reliable information, so (iii) does not apply. Criteria (i) and (ii) are inappropriate for our domain of interest because, as biomedical experts confirm, explanations which are longer and more detailed can often be stronger since they are able to more specifically describe the biological processes and relevant entity interactions; thus, the key criterion is simply that explanations are biomedically accurate and reasonable.

Apart from the variety of derivation methods, the resulting explanations can take many forms, e.g., text summaries [1], [2], article clips [8], highlighted words in clips [9]–[13], sets of inferences rules and facts [3]–[6], and paths/subgraphs in knowledge graphs [4], [14]–[16]. Our explanations take the form of knowledge graph patterns, see Section III-C for more detail. To the best of our knowledge, we are first to consider knowledge graph patterns, as opposed to specific subgraphs or paths. Therefore, our explanations are the only ones with the benefit of being applicable to multiple queries, instead of being query-specific.

As far as we know, [2] presents the only explanation derivation approach for the biomedical domain. While their goals are similar to ours, their explanations are paragraphs, which are slower to understand than our patterns and must be uniquely derived for each candidate fact.

B. Explanation Evaluation

Some of the explanation derivation methods mentioned also propose metrics for evaluating the explanations they produce [1]–[4], [14]–[16].

There are two existing metrics which are capable of scoring explanations in the form we use [3], [4], though they were originally designed for other explanation formats. Reference

[3] scores explanations by multiplying the weights of all the inference rules used to build the pattern. Unfortunately, this relies upon the assumption that these rules are independent of each other, which is not met in many cases, e.g., in the biomedical domain.

Reference [4] uses a heuristic explanation scoring metric which is based on the number of rules required to generate the explanation and how they were synthesized. This metric assumes that each rule can be used for perfectly reliable inference as using a single rule results in the maximum explanation confidence of 1. In many cases, inference rules do not hold all the time, including in the biomedical domain.

Additionally, the metrics of [3], [4] evaluate the process of generating an explanation, as opposed to the actual explanation itself, which has undesired side-effects, e.g., the same explanation can receive different scores if it can be derived in different ways. On the other hand, we present metrics which provide a data-supported evaluation of the explanation itself and do not make unreasonable, domain-dependent assumptions.

III. Background and Problem Statement

A. Knowledge Graphs

Knowledge graphs are a popular database format, which has arisen in a number of domains, including the biomedical domain on which we focus. We define a *knowledge graph* as follows.

Definition 1. A *knowledge graph* $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \mathcal{T})$ is composed of a the set of *entities* $\mathcal{E}$, a set of *predicates* $\mathcal{P}$, and a set of *triples* $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{P} \times \mathcal{E}$.

Each *triple* $(s, p, o) \in \mathcal{T}$ represents the fact that the *predicate*, i.e. relationship type, p holds between the *subject* s and the *object* o. For example, (*benazepril, treats, chronic kidney disease*) is a triple which states the real-world fact: "the drug benazepril treats chronic kidney disease."

B. Horn/Inference Rules

To generate explanations, we utilize sets of inference rules. In particular, we focus on Horn rules, which are useful for inference and are commonly used in the literature, see, for example, [3], [5], [18]–[24]. Horn rules are composed of *atoms*, which are (s, p, o) triples, where $p \in \mathcal{P}$ and s, o are either variables or entities from $\mathcal{E}$.

An example of a Horn rule r is

$$(x, \textit{treats}, z) \Leftarrow (x, \textit{treats}, y) \wedge (y, \textit{hasPhenotype}, z)$$

We refer to the left-hand side and the right-hand side of r as $head(r)$ and $body(r)$, respectively. The presence of $body(r)$ allows inference of $head(r)$. Thus, this rule represents the intuition that a drug x is likely to treat a disease z if x treats some disease y which has the same phenotype as z.

C. Explanations

As mentioned in Section II, there are many ways to view an explanation. In this work, we define an *explanation* as follows.

Definition 2. An *explanation* E is a Horn rule shown below.

$$(e_0, p, e_1) \Leftarrow a_1 \wedge a_2 \wedge \cdots \wedge a_n \tag{1}$$

In the above, $A = \{a_1, a_2, \ldots, a_n\}$ is a set of free atoms, i.e., atoms in which s, o are both variables, called the *explanation pattern*; p is the *target predicate*; e_0, e_1 are two entities (variables), from atoms in A, which are designated as the *endpoints*. Altogether (e_0, p, e_1) is the *target triple*. We say that the explanation pattern A explains the target triple (e_0, p, e_1).

For example, consider the explanation E below.

$$(w, \textit{treats}, z) \Leftarrow$$
$$(w, \textit{treats}, x) \wedge (y, \textit{biomarkerFor}, x) \wedge (y, \textit{biomarkerFor}, z)$$

Then, we say that the explanation pattern $A = \{(w, \textit{treats}, x), (y, \textit{biomarkerFor}, x), (y, \textit{biomarkerFor}, z)\}$ explains the triple $(w, \textit{treats}, z)$.

Since A explains (e_0, p, e_1), we consider A to allow inference of the triple (e_0, p, e_1). Therefore, in this work, we view an explanation E as a Horn rule, where the explanation pattern A is the body of the rule and the target triple (e_0, p, e_1), which E explains, is the head of the rule.

D. Problem Statement

We now introduce the formal problem statement for explanation derivation and evaluation of knowledge graph links.

Given (i) a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \mathcal{T})$, (ii) a set of Horn rules $\mathcal{R}$, and (iii) an existing or hypothesized query triple $q = (e_0, p, e_1)$, where $p \in \mathcal{P}$ and $e_0, e_1 \in \mathcal{E}$, our goal is to (i) derive a set $\mathcal{O}$ of possible explanations for q, and (ii) assign a reliable, understandable, data-supported score to each explanation $E \in \mathcal{O}$, which represents the extent to which the explanation justifies the existence of q.

IV. Methodology

We address each goal of the problem presented above separately. Deriving explanations is addressed in Section IV-A, and evaluating explanations is addressed in Section IV-B.

A. Explanation Derivation

Recall from our problem statement in Section III-D that explanation derivation requires three main inputs: a knowledge graph $\mathcal{G}$, a set of Horn rules $\mathcal{R}$, and a query triple $q = (e_0, p, e_1)$. Each serves a specific purpose in the explanation derivation process: the knowledge graph serves as a reliable source of information for which we can find groundings for the explanations built; the rule set allows us to rewrite current explanations by applying the rules appropriately; the query focuses our search and provides the starting point.

The proposed algorithm is based on backward chaining, which is a standard approach for reasoning in artificial intelligence in which one works backwards from the goal to find a solution. That is, the first explanation we consider is the query q itself. After all, if the queried triple is already in the KG, then it serves as its own explanation. The backward chaining

Algorithm 1: Deriving explanations

Input: knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{P}, \mathcal{T})$, set of Horn rules $\mathcal{R}$, query triple $q = (e_0, p, e_1)$, max search depth d
Output: set $\mathcal{O}$ of explanations for target triple q

1: $\mathcal{O} \leftarrow \emptyset$;
2: $q.depth \leftarrow 0$; $q.found \leftarrow$ **false**;
3: $\mathcal{Q} \leftarrow \{q \Leftarrow q\}$ // `build a queue`
4: **while** $\mathcal{Q} \neq \emptyset$ **do**
5: $\quad E \leftarrow$ some explanation from $\mathcal{Q}$;
6: $\quad$ **if** $\nexists$ atom $a \in E$ s.t. $a.found =$ **false then**
7: $\quad\quad \mathcal{O} \leftarrow \mathcal{O} \cup E$; // `E is a successful exp.`
8: $\quad$ **else**
9: $\quad\quad a \leftarrow$ some atom from E s.t. $a.found =$ **false**;
10: $\quad\quad$ // `first, ground E with a`
11: $\quad\quad \mathcal{Q} \leftarrow \mathcal{Q} \cup groundAtom(E, a, \mathcal{G})$;
12: $\quad\quad$ // `next, rewrite a in E`
13: $\quad\quad \mathcal{Q} \leftarrow \mathcal{Q} \cup rewriteAtom(E, a, \mathcal{R}, d)$;
14: **return** $\mathcal{O}$;

proceeds by identifying rules from $\mathcal{R}$ in succession that can be used to infer atoms of the explanation E under consideration. These rules are used to logically rewrite E in order to derive other potential explanations. Any explanation which exists entirely in the knowledge graph $\mathcal{G}$ is a successful explanation and is returned as a result of the backward chaining process.

We first provide a detailed explanation of the algorithm in Section IV-A1 and then provide a step-by-step walk through of an example in Section IV-A2.

1) Algorithm: We will now detail the main framework of our explanation derivation approach, shown in Algorithm 1.

Notice that the three inputs for explanation derivation, i.e., a knowledge graph $\mathcal{G}$, a set of Horn rules $\mathcal{R}$, and a query triple $q = (e_0, p, e_1)$, match the first three inputs of Algorithm 1. We add one additional input to ensure termination: a maximum search depth d. Depending on the set of Horn rules given as input, it is possible that the rewriting step, explained below, could always produce new explanations to consider. In this case, the algorithm terminates only because the maximum search depth d restricts the number of recursive rule rewritings allowed. Algorithm 1 produces a single result: a set of explanations $\mathcal{O}$ for the input query q, where each explanation $E \in \mathcal{O}$ is of the form described in Section III-C.

Algorithm 1 presents an iterative backward chaining process which searches the knowledge graph $\mathcal{G}$ for explanations for the query q. To suit the needs of our application, we have modify the standard iterative backward chaining algorithm in a few ways. One such modification is to return all possible explanations within the depth limit d, as opposed to stopping once a single explanation is found, so as to find the strongest explanations. To this end, we maintain an output set $\mathcal{O}$, initially an empty set (line 1), which stores successful explanations, i.e., explanations which exist in the knowledge graph $\mathcal{G}$ for the target triple q.

To track the progress of our search, we attach two attributes to each atom a of an explanation E: $a.found$ and $a.depth$. The value of $a.found$ is *true* if we have found a grounding for E with a in $\mathcal{G}$, i.e., the explanation pattern exists in $\mathcal{G}$; otherwise, $a.found$ is *false*. The value of $a.depth$ represents the number of recursive rewritings required for a to appear in E. For example, any atoms that appear in E after the first rewriting have depth 1; any atoms that appear in E by rewriting an atom of depth 1 have depth 2, and so on. Initially, $q.found$ and $q.depth$ must be set appropriately to *false* and 0, respectively (line 2).

In iterative backward chaining, a queue $\mathcal{Q}$ is maintained, which contains possible explanations to consider. Initially, it contains solely the input query q, which serves as the starting point for the search (line 3). There is a main loop (lines 4–13) in which, for each iteration, a single explanation E is removed from the queue and considered (line 5). If E has been ground entirely in $\mathcal{G}$, i.e., the entire explanation pattern exists in $\mathcal{G}$ and all atoms have been marked as found, then it is a successful explanation and is added to the output set $\mathcal{O}$ (lines 6–7).[2]

Otherwise, it must be processed further. Here, instead of considering the entire explanation at once, we consider only one atom, in particular one which has not yet been found (line 9). The atom a is first ground in $\mathcal{G}$ with the found atoms of E (line 11) to determine if the partial pattern exists in $\mathcal{G}$. If it does, then E is added back to the queue so the other atoms can be processed in later iterations of the main loop. Since an explanation pattern can exist in $\mathcal{G}$ only if its partial patterns also exist in $\mathcal{G}$, processing a single atom at a time helps to eliminate useless patterns early and, therefore, limits the number of large queries performed on $\mathcal{G}$.

Next, a is rewritten within E by rules from $\mathcal{R}$, and the rewritings are added to the queue to be considered in later iterations of the main loop (line 13). Here, we impose the depth limit d. Before rewriting, the subprocedure $rewriteAtom$ compares $a.depth$ with the maximum search depth d. If $a.depth$ is less than d, then rewriting proceeds as normal; otherwise, no rewritings are performed.

Algorithm 1 continues to process explanations in this way until the queue $\mathcal{Q}$ is empty and the main loop of lines 4–13 completes. At that point, all successful explanations of the query triple q within the depth limit d have been added to the output set $\mathcal{O}$, which is returned as the result of Algorithm 1 (line 14).

2) Example: The following example will outline the explanation derivation process described in Section IV-A1. For the sake of clarity, we will abbreviate *benazepril* with *BE*, *diabetic nephropathy* with *DN*, *kidney failure* with *KF*, and *chronic kidney disease* with *CKD*. Assume we have the following

[2]Before adding E to $\mathcal{O}$, the endpoints x, y of E are replaced with fresh variables as in the example of Section IV-A2.

inputs to Algorithm 1.

$$\begin{aligned}
\mathcal{G} = \{ & (\textit{BE}, \textit{treats}, \textit{DN}), (\textit{KF}, \textit{biomarkerFor}, \textit{DN}), \\
& (\textit{KF}, \textit{biomarkerFor}, \textit{CKD})\} \\
\mathcal{R} = \{ & (u, \textit{treats}, w) \Leftarrow (u, \textit{treats}, v) \wedge (v, \textit{hasPhenotype}, w), \\
& (x, \textit{hasPhenotype}, y) \Leftarrow \\
& (z, \textit{biomarkerFor}, x) \wedge (z, \textit{biomarkerFor}, y)\}, \\
q = \quad & (\textit{BE}, \textit{treats}, \textit{CKD}) \\
d = \quad & 2
\end{aligned}$$

As mentioned, the first explanation E that we consider is $q \Leftarrow q$. Our goal is to show that the right-hand side of E is true so that we can infer the left-hand side. Since $q.found =$ *false* to begin with, E is not yet a successful explanation. Therefore, we attempt to ground q in $\mathcal{G}$, which is not possible since $q \notin \mathcal{G}$. Next, we attempt to rewrite q with rules in $\mathcal{R}$ that allow inference of q. Notice that the first rule r_1 can be used to infer q under the substitution $\sigma = \{u/\textit{BE}, w/\textit{CKD}\}$. So, we rewrite q in E to get a new explanation E'.

$$q \Leftarrow (\textit{BE}, \textit{treats}, v) \wedge (v, \textit{hasPhenotype}, \textit{CKD})$$

The atoms on the right-hand side both have depth 1 since they were produced by a single rewriting.

Finding groundings in $\mathcal{G}$ for each atom in E' would complete the inference of $q = (\textit{BE}, \textit{treats}, \textit{CKD})$, and we would have a successful explanation. First, let's consider the atom $a = (v, \textit{hasPhenotype}, \textit{CKD})$. Note that a cannot be ground in $\mathcal{G}$, but it can be rewritten by the second rule. Therefore, we now have the following explanation E''.

$$\begin{aligned}
q \Leftarrow & (\textit{BE}, \textit{treats}, v) \wedge \\
& (z, \textit{biomarkerFor}, v) \wedge (z, \textit{biomarkerFor}, \textit{CKD})
\end{aligned}$$

Atoms $(z, \textit{biomarkerFor}, v)$ and $(z, \textit{biomarkerFor}, \textit{CKD})$ have depth 2 since they were produced by rewriting an atom of depth 1, i.e. a.

Now, we consider each atom in E''. First, consider $a_1 = (\textit{BE}, \textit{treats}, v)$. Notice that a_1 can be ground in $\mathcal{G}$ by the substitution $\sigma_1 = \{v/\textit{DN}\}$, so we set $a_1.found =$ *true*. For the sake of brevity, we will skip the rewriting step and instead consider the next atom in this explanation: $a_2 = (z, \textit{biomarkerFor}, v)$. Notice that a_2 can be ground in $\mathcal{G}$ along with a_1 by the substitution $\sigma_2 = \{v/\textit{DN}, z/\textit{KF}\}$, so $a_2.found =$ *true* as well. Since $a_2.depth = 2$, which is the depth limit d, no rewriting occurs. Finally, we consider $a_3 = (z, \textit{biomarkerFor}, \textit{CKD})$, which can be ground in $\mathcal{G}$ along with a_1 and a_2 by the substitution $\sigma_3 = \{v/\textit{DN}, z/\textit{KF}\}$, so $a_3.found =$ *true*. Again, $a_3.depth = 2$, so no rewriting occurs.

At this point, all atoms of E'' have been found, so E'' is a successful explanation for our query q. Before outputting E'', we perform the substitution $\sigma_{var} = \{\textit{BE}/e_1, \textit{CKD}/e_2\}$, which gives: $(e_1, \textit{treats}, e_2) \Leftarrow (e_1, \textit{treats}, v) \wedge (z, \textit{biomarkerFor}, v) \wedge (z, \textit{biomarkerFor}, e_2)$. Now the explanation is a knowledge graph pattern, matching the definition in Section III-C, which can be potentially reused for other specific $(e_1, \textit{treats}, e_2)$ queries.

For the sake of the exposition, we will stop our example here. However, it is important to recall that since we aim to find the strongest explanations, we do not stop after finding the first explanation. In particular, there are still atoms of both E' and E'' which can be rewritten to produce new explanations to consider. The backward chaining process continues until (1) the depth limit d has been reached for all atoms of all explanations, or (2) there are no rules left to facilitate rewriting.

B. Explanation Evaluation

Now, let us consider how we can evaluate the explanations generated by Algorithm 1.

In recent years, rule mining in knowledge graphs, a relative of the much older association rule mining, has become a relatively popular problem [18]–[25]. As a result, several metrics have been proposed for evaluating such rules. Since our explanations are inference rules, we propose to evaluate them by two of the strongest, most widely accepted of these pre-existing metrics, as well as a proposed combination of those two. To this end, we consider three metrics in total: confidence, head coverage, and the harmonic mean, or F1-score, of confidence and head coverage.

Before formally defining our metrics, we define a few background terms. Let E be the following explanation.

$$(e_0, p, e_1) \Leftarrow a_1 \wedge a_2 \wedge \cdots \wedge a_n$$

We define the *set of groundings of* E as follows.

$$groundings(E) = \{(e_0, e_1) : \exists\ \sigma : A\sigma \in \mathcal{G}\} \tag{2}$$

where $A = \{a_1, a_2, \ldots, a_n\}$ is the explanation pattern of E. That is, the set of groundings of E is the set of all endpoints (e_0, e_1) for which there is some substitution σ of entities in A which places A entirely in the knowledge graph $\mathcal{G}$. The corresponding *support* of E is defined as the number of groundings of E.

$$support(E) = |groundings(E)| \tag{3}$$

Note that this support metric was originally adapted from association rule mining for use in knowledge graph rule mining in [18].

Consider the example explanation E from Section III-C, duplicated below.

$$\begin{aligned}
& (w, \textit{treats}, z) \Leftarrow \\
& (w, \textit{treats}, x) \wedge (y, \textit{biomarkerFor}, x) \wedge (y, \textit{biomarkerFor}, z)
\end{aligned}$$

Then, in the ROBOKOP knowledge graph[3] used in our experiments, see Section V, $support(E) = 204$, which means that there are 204 pairs of endpoint entities (e_0, e_1) for the explanation E such that the entire explanation pattern exists in the knowledge graph.

[3] http://robokopkg.renci.org/browser/

1) Confidence: The first metric we propose for evaluating explanations is explanation *confidence*, which is a standard rule mining metric borrowed directly from association rule mining [26]. The confidence of an explanation E is defined as follows.

$$conf(E) = \frac{support(E')}{support(E)} \tag{4}$$

where E is $(e_0, p, e_1) \Leftarrow a_1 \wedge \cdots \wedge a_n$ and E' is $(e_0, p, e_1) \Leftarrow a_1 \wedge \cdots \wedge a_n \wedge (e_0, p, e_1)$.

This confidence value is the conditional probability that the fact (e_0, p, e_1) can be ground in $\mathcal{G}$, given that the explanation pattern $A = \{a_1, \ldots, a_n\}$ of E can be ground in $\mathcal{G}$. Therefore, it is a representation of how likely (e_0, p, e_1) is to be true based on the fact that A is known to be true. As a result of our considering explanations to be inference rules, confidence can also be viewed as a ratio of the inferences of the rule E that are in the knowledge graph $\mathcal{G}$. All in all, it is effectively a measure of how often the explanation pattern A is associated with (e_0, p, e_1) in $\mathcal{G}$ and, therefore, how much we can trust A to explain (e_0, p, e_1).

Consider, again, the explanation from Section III-C. Then, $support(E') = 144$ and $support(E) = 204$, which means that $conf(E) = 0.706$. In other words, of all the groundings of A in $\mathcal{G}$, 70.6% of them appeared with the triple $(w, \textit{treats}, z)$ as well. Therefore, we would expect that whenever we see the explanation pattern A, there is a 70.6% chance that $(w, \textit{treats}, z)$ is also true.

2) Head Coverage: The next metric we consider is that of head coverage, another standard rule mining metric proposed in [18]. The head coverage of an explanation E is defined as follows.

$$hc(E) = \frac{support(E')}{support(q)} \tag{5}$$

where E' is $(e_0, p, e_1) \Leftarrow a_1 \wedge \cdots \wedge a_n \wedge (e_0, p, e_1)$ and q is $(e_0, p, e_1) \Leftarrow (e_0, p, e_1)$.

This head coverage value both parallels and contrasts the value of confidence as it is also a conditional probability, but it is the conditional probability that the explanation E can be ground in $\mathcal{G}$, given that the fact (e_0, p, e_1) can be ground in $\mathcal{G}$. In other words, it is a representation of how likely $A = \{a_1, \ldots, a_n\}$ is to be true based on the fact that (e_0, p, e_1) is known to be true. Our consideration of explanations as inference rules allows for head coverage to be interpreted as the proportion of triples with predicate p that could be inferred by the rule $(e_0, p, e_1) \Leftarrow A$. Therefore, head coverage serves as a measure of how often (e_0, p, e_1) is associated with A in $\mathcal{G}$, and, therefore, how relevant the explanation pattern A is to the fact (e_0, p, e_1).

Again, let E be the example explanation E from Section III-C. Then, q is $(w, \textit{treats}, z) \Leftarrow (w, \textit{treats}, z)$, $support(E') = 144$ and $support(q) = 16135$, which means that $hc(E) = 0.009$. In other words, of all the groundings of q in $\mathcal{G}$, 0.9% of them appeared with the explanation pattern A as well. Therefore, we would expect that A only explains 0.9% of $(w, \textit{treats}, z)$ triples.

3) F1-score: Our final metric is a proposed combination of the two previous standard rule mining metrics. The confidence metric is a perfect analog to precision, i.e., the ratio of correct inferences to total inferences. Likewise, the head coverage metric is a perfect analog to recall, i.e., the ratio of correct inferences to correct facts. This lends itself very naturally to the definition of our third metric: the harmonic mean, i.e. F1-score, of confidence and head coverage.

$$F1\text{-}score(E) = 2 * \frac{conf(E) * hc(E)}{conf(E) + hc(E)} \tag{6}$$

The standard F1-score serves as a metric which helps balance the trade-off between precision and recall. In our case, we want to balance the trade-off between the confidence and the head coverage of our produced explanations to find explanations which are both highly associated and highly relevant to the input queries.

For our running example explanation E, $F1\text{-}score(E) = 0.018$, which is the harmonic mean of $conf(E) = 0.706$ and $hc(E) = 0.009$. This F1-score serves as a measure of the average quality of the explanation, where confidence and head coverage are two key aspects of explanation quality. It provides a single score which allows us to compare E to other explanations by balancing the importance of both confidence and head coverage.

V. Experiments

A. Implementation and Experimental Settings

We hosted the knowledge graph used in our experiments as a graph database in Neo4j version 4.2.1.[4,5] We have implemented the entirety of the explanation derivation algorithm along with all programs used for running experiments in Python 3.9; we have used the py2neo package to interact with the dataset used in our experiments. The experiments were conducted on a computer with an Intel i7-1165G7 CPU processor running at 2.8 GHz with 12GB of RAM and Windows version 10.

1) Knowledge Graph: In this work, we focus on the large-scale biomedical knowledge graph: Reasoning Over Biomedical Objects linked in Knowledge Oriented Pathways (ROBOKOP) [27].[6] The version that we used contains $8,891,404$ triples among $610,248$ biomedical entities. In particular, we focus in our experiments on deriving explanations for *treats* triples in ROBOKOP to address our goal of drug discovery.

We conduct all of our experiments on a sample of the ROBOKOP version mentioned above due to memory limitations when performing automatic rule extraction, see Section V-A2. The sample used contains $307,959$ entities of the following types: anatomy, biological process or activity, cell,

[4]https://neo4j.com

[5]Despite much effort to optimize the queries used in our programs, we found that Neo4j would timeout on queries for explanation patterns with more than roughly 4 atoms. We eliminated any such explanations from all experimental results and calculations.

[6]http://robokopkg.renci.org/browser/

cellular component, chemical substance, disease, gene, gene family, genetic variant, and phenotypic feature; it contains $4,016,135$ triples over 86 different predicate types. Construction of the sample is detailed in Sections V-A1a–V-A1b.

a) Subset Extraction: To extract the subset, we performed a random walk of the graph, as suggested by [28], to get a sample of 4 million triples from ROBOKOP. We randomly choose a starting node and then performed a random walk by repeatedly choosing a random neighbor of the current node. To facilitate extraction of an evenly spread out sample, we randomly jump back to the original start node with probability 0.15 at each step.[7] To ensure termination, if we have not accrued 4 million triples after $100 \times n$ steps, where $n = 610,248$, i.e. the number of nodes in the graph, we randomly select another starting node and repeat the process.

b) Negative Sampling: In our experiments, we attempt to find supporting explanations for $(x, \textit{treats}, y)$ triples, i.e., explanations with target predicate *treats*, as well as refuting explanations, i.e., explanations with target predicate *not_treats*. Thus, we require negative triples, e.g., (*cocaine, not_treats, chronic kidney disease*), which are not stored in knowledge graphs. To sample negative edges, we rely upon the popular assumption that triples not present in the graph can be considered false for the purposes of negative sampling, i.e., the closed world assumption [23], [29]–[32]. Of course, we recognize that this is not a perfect assumption as knowledge graphs inherently operate under the open world assumption. However, we reason that the vast majority of missing triples actually are false; thus, random selection of missing triples will result in a negligible amount of false negatives.

To perform negative sampling, we adopt the procedure used in [32]–[35]. Therefore, we consider each existing triple $(s, \textit{treats}, o)$ and randomly sample either a new subject s' or a new object o' of the same entity type m times, forming m new triples of the form $(s', \textit{not\_treats}, o)$ or $(s, \textit{not\_treats}, o')$, respectively. If any triple produced this way already exists in ROBOKOP as a true *treats* triple, then we repeat the process until this is not the case. Since there is no accepted value in the literature for the number of negatives to sample [23], [29]–[38], we randomly sample a single negative triple, i.e., $m = 1$, for each existing *treats* triple in ROBOKOP, in an effort to avoid class imbalance.

2) Ruleset: Unfortunately, manual rule design at the scale required for our experiments is infeasible. In particular, manual rule design would only be feasible by domain experts who have sufficient background knowledge regarding the biomedical concepts and relationships stored in ROBOKOP. Thus, to form the set of rules used in our experiments, we perform automatic rule extraction with the state of the art in knowledge graph rule mining: AMIE [18].

We ran AMIE using our sample of ROBOKOP and each of the 86 predicate types as the rule head type. We used no thresholds on AMIE's confidence or support and mined a maximum rule length of two, i.e. a maximum of two atoms in the body of each rule, since mining rules of length three or more would not terminate after more than a day. To avoid exponential blowup in the number of patterns considered during explanation derivation, we limited the resulting set of rules by imposing a threshold of 25 on AMIE's calculated support of each rule, i.e., the number of positive instances of the rule. Additionally, for each predicate, we use only the top ten rules with the highest standard confidence, as reported by AMIE. These restrictions simply eliminate weak rules which appear extremely infrequently and are therefore unlikely to produce explanations. The resulting rule set consists of 561 rules with 67 different predicates as rule heads.

[7]The value commonly used in the literature [28].

B. Metric Accuracy

1) Experimental Design: In this experiment, we test the accuracy and reliability of the scores assigned by our proposed metrics and, thus, their viability as valuable metrics for evaluating explanations of facts in knowledge graphs. In summary, our results suggest that our proposed metrics are significantly more accurate than existing baseline metrics. Therefore, they are more reliable metrics for evaluating explanations and, consequently, for determining the truth of a hypothesized query.

We follow the experimental design of [4], [39] as follows. We derive a set of positive explanations $\mathcal{O}^+$ as well as a set of negative explanations $\mathcal{O}^-$ for each fact (e_0, p, e_1) in our experimental data set of candidate facts; positive explanations are those for the fact (e_0, p, e_1), whereas negative explanations are those for the fact (e_0, not_p, e_1). We keep only the k top-scoring explanations from each set to avoid having an abundance of weak explanations that overshadow a few strong explanations. We define the quality of a set of explanations as the average score of the explanations in the set.

$$quality(\mathcal{O}) = \frac{1}{|\mathcal{O}|} \sum_{E \in \mathcal{O}} score(E) \tag{7}$$

where $score(E)$ takes the place of the explanation evaluation metric in use, e.g., $conf(E)$ or $hc(E)$. The quality metric allows us to incorporate several explanations into our evaluation of a single fact, which reduces the impact of potential outlier explanations, thus giving a more reliable evaluation.

We then assign a truth score to each fact (e_0, p, e_1) as follows.

$$truthScore((e_0, p, e_1)) = quality(\mathcal{O}^+) - quality(\mathcal{O}^-) \tag{8}$$

In effect, the truth score indicates whether we should consider the fact (e_0, p, e_1) to be true or false by determining which explanation set is stronger, i.e., has a higher quality score. A positive truth score indicates that the positive explanations $\mathcal{O}^+$ were stronger; a negative truth score indicates the opposite.

To form our experimental data set, we randomly selected n true treats triples from ROBOKOP. To avoid trivial explanations, i.e., $(e_0, p, e_1) \Leftarrow (e_0, p, e_1)$, we removed these n triples from the knowledge graph. Then, to match the experimental design of [4], [39], for each true fact, f_i, we generated m false alternative facts $f'_{i1}, f'_{i2}, \ldots, f'_{im}$ using the same negative

sampling process as before, see Section V-A1b. This produced n groups each containing 1 true fact and m false alternative facts. For example, let (*benazepril, treats, chronic kidney disease*) be the true fact. Then, possible false alternatives could be (*benazepril, treats, lung cancer*) and (*cocaine, treats, chronic kidney disease*).

We derive explanations via Algorithm 1 for each fact in each group and then assign a truth score to each fact. A reliable metric for evaluating explanations would assign higher scores to explanations of true facts than to explanations of false facts, thus causing the truth score of each true fact to be higher than the truth scores of the false alternative facts. Thus, by forming these groups we can determine which metrics are reliable and assign appropriate scores to explanations. To this end, we evaluate each the explanation metrics used on each group $G_i = \{f_i, f'_{i1}, f'_{i2}, \ldots, f'_{im}\}$ by computing its accuracy score as follows.

$$accuracy(G_i) = \frac{1}{m} \sum_{f'_{ij}} [truthScore(f_i) \geq truthScore(f'_{ij})] \tag{9}$$

where $[P] = 1$ if P is true; otherwise, $[P] = 0$. The accuracy score indicates the proportion of false alternative facts which scored lower than the corresponding true fact. Because we want each metric to score the true fact higher than the false alternative facts, this is a measure of how accurate the scoring metric is for group G_i.

2) Baselines: We compare our proposed metrics with two baseline explanation evaluation metrics, from references [3], [4]. These baseline metrics were designed to evaluate the explanation formats considered in their respective works, which are, in particular, different than ours. However, they can be easily modified to evaluate our novel explanation format: knowledge graph patterns.

Let E be the explanation $(e_0, p, e_1) \Leftarrow a_1 \wedge a_2 \wedge \cdots \wedge a_n$. The baseline metrics are defined as follows.

- Problog [3]:

$$Problog\text{-}score(E) = \prod_{r \in R} probability(r) \tag{10}$$

  where R is the set of rules used to produce the explanation and $probability(r)$ is the probability associated with rule r.[8]

- ExFaKT [4]:

$$ExFaKT\text{-}score(E) = \frac{1}{|A|} \sum_{a \in A} \frac{1}{a.depth} \tag{11}$$

  where $A = \{a_1, a_2, \ldots, a_n\}$ is the explanation pattern of E, and $a.depth$ is the same as the value $a.depth$ used in Algorithm 1, see IV-A1 for details.[9]

[8]We take the standard confidence score of rule r, as calculated by AMIE [18] during rule mining, as the probability of rule r since this matches the notion of rule probability presented in [3].

[9]The actual equation published in [4] incorporates a value called $trust(a.source)$. Since we use knowledge graphs as our only source, we simplify the equation, substituting the appropriate value for $trust(a.source)$, i.e., 1.

TABLE I

A COMPARISON OF THE ACCURACIES OF THE PROPOSED METRICS VERSUS THE BASELINE METRICS ON APPROPRIATELY SCORING THE TRUE FACT OF EACH FACT GROUP ABOVE THE CORRESPONDING FALSE ALTERNATIVE FACTS. THE ROW LABEL SPECIFIES THE PROPOSED METRIC, THE COLUMN LABEL SPECIFIES THE BASELINE METRIC AND THE VALUE INDICATES THE PERCENTAGE OF 143 FACT GROUPS WHERE THE ACCURACY OF THE PROPOSED METRIC WAS HIGHER THAN (STRICTLY HIGHER THAN) THE ACCURACY OF THE BASELINE METRIC.

	Problog-score	ExFaKT-score
Confidence	89.51% (18.18%)	91.61% (37.06%)
Head Coverage	84.62% (19.58%)	89.51% (32.17%)
F1-score	90.21% (18.88%)	92.31% (37.06%)

3) Experimental Details: For our experimental data set, we generated 250 fact groups with 5 false alternatives each, see Section V-B1 for details.[10] We derived our explanations using Algorithm 1 with the following inputs: (i) our extracted ROBOKOP sample, see Section V-A1, (ii) rules automatically extracted from AMIE, see Section V-A2, (iii) queried triples from our set of experimental data set, and (iv) maximum search depth 3.[11] We compute a truth score for each fact using both baseline metrics and our three proposed metrics, and we compare the scores assigned to each fact.

We eliminated any groups for which Algorithm 1 could not produce an explanation for the true fact and at least one of the false alternative facts as the accuracy of these groups could not be computed. In addition, we did not include false alternatives which had no explanations in our accuracy computations. After this post-processing, we had a total of 143 groups.

4) Results: We first report our results considering only the top 5 positive and top 5 negative explanations for each fact, as in [4]. To compare the performance of our three proposed metrics to the two baseline metrics, we computed the percentage of the 143 groups where our proposed metrics achieved accuracy scores higher than and *strictly* higher than the baseline metrics, see Table I. We observe that our proposed metrics achieve accuracy at least as high as the baseline metrics a vast majority of the time, indicating that the scores assigned by our metrics to the true facts are at least as accurate as the scores assigned by the baseline metrics. Moreover, there is a substantial number of groups where our proposed metrics strictly outperform the baseline metrics.

Next, we considered the accuracy of each metric while varying the top k value used when computing scores to determine how this might impact performance, i.e., we consider more than just the top 5 positive and top 5 negative explanations. Fig. 2 shows the average accuracy of each metric for $k \in \{1, 5, 10, 20, 30, 40, 50\}$. We first notice that our proposed metrics achieve significantly higher accuracy than the baselines metrics in most cases. In fact, head coverage is the only proposed metric which does not consistently achieve higher average accuracy than the baselines; Problog has higher

[10]We chose the number 5 because the datasets used in [4], [39], whose experimental design we follow, had an average of 3-5 alternatives per fact.

[11]Maximum search depth 3 allowed us to extract most, if not all, of the explanations that Neo4j could handle in light of footnote 5.

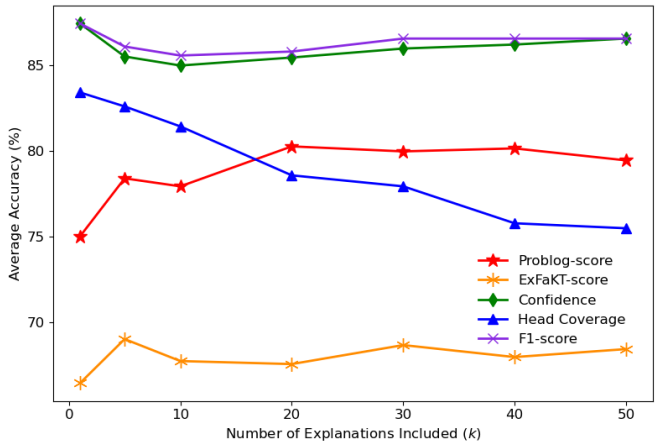


Fig. 2. The average accuracy of each metric on 143 fact groups when scoring each fact according to the top k positive explanations and the top k negative explanations derived by Algorithm 1. The X-axis indicates k, the number of explanations included when scoring each fact, and the Y-axis indicates the average accuracy.

average accuracy than head coverage for $k \geq 20$. Additionally, we notice that confidence and F1-score have comparable average accuracy for all k values despite the fact that the F1-score is intended to offer a good balance between confidence and head coverage. We conclude that our proposed metrics, especially confidence and F1-score, provide more reliable, accurate scoring no matter how many explanations are allowed for each fact.

C. Explanation Reasonableness

1) Experimental Design: In this experiment, we aim to show that the novel format of our explanations is productive for use by domain experts by showing that the explanations derived by our approach are both understandable and reasonable according to biomedical experts. In summary, the biomedical expert analysis indicates that our derived explanations provide reasonable justification for drug-disease treatment relationships. This suggests that our explanation derivation approach is capable of extracting meaningful knowledge graph patterns that summarize key relationships and interactions between biomedical entities.

To explore the quality of our explanations, we had a biomedical expert on our team analyze several explanations output by Algorithm 1. We sorted all the explanations generated in the experiment discussed in Section V-B according to each of our proposed metrics and presented the 5 top-scoring explanations for each metric to our domain expert. We instructed them to consider how reasonable each explanation pattern is as justification that the fact $(e_0, \textit{treats}, e_1)$ is true, i.e., how well it explains the target triple $(e_0, \textit{treats}, e_1)$. They were asked to rate each explanation pattern according to the following scale: (1) "reasonable," (2) "reasonable in some cases" (i.e., depending on other unknown factors), (3) "neutral," and (4) "unreasonable." In addition, we asked the domain expert to provide a short description of their reasoning as a qualitative analysis.

TABLE II
THE PRECISION OF EACH METRIC, I.E., THE PROPORTION OF THE 5 TOP-SCORING EXPLANATIONS WHICH WERE DEEMED ACCURATE BY OUR BIOMEDICAL EXPERT.

Metric	Precision
Confidence	80%
Head Coverage	100%
F1-score	80%

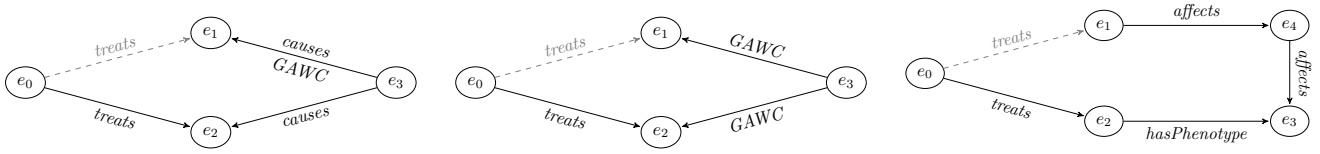
We deem any explanation which was rated as a (1) or (2) to be accurate and define the precision of each metric as the proportion of accurate explanations out of the 5 presented to the domain expert.

2) Results: Table V-C2 shows the precision of each of our proposed metrics, which indicates the proportion of explanations deemed accurate by our biomedical expert. This table indicates that each of our proposed metrics is capable of assigning high scores to explanations which are biomedically reasonable. In fact, for each metric, no more than one of the 5 top-scoring explanations was deemed inaccurate, i.e., a precision score of 80%, and no explanations were given a rating of (4) "unreasonable."

We next present a few explanations along with their ratings according to our biomedical expert and explore the qualitative feedback received. Fig. 3 shows three example explanations, discussed below. The target triple for each explanation is $(e_0, \textit{treats}, e_1)$.

First, the explanation shown in Fig. 3a received a rating of (1) "reasonable." Because diseases e_1 and e_2 share the common causative gene e_3, it is reasonable to deduce that the drug e_0, which treats disease e_2, would also treat disease e_1. This follows from the general understanding that similar diseases have similar treatments, especially diseases which have similar mechanistic causes, meaning that similar biological processes and interactions cause the disease. Thus, this explanation pattern shows a strong relationship between the drug e_0 and the disease e_1, based on biological interactions, and is therefore a reasonable explanation for the fact $(e_0, \textit{treats}, e_1)$.

Next, the explanation shown in Fig. 3b received a rating of (2) "reasonable in some cases." This explanation is very similar in structure to that of Fig. 3a with the only difference being the name of the relationship that gene e_3 has with diseases e_1 and e_2, i.e., *geneAssociatedWithCondition* instead of *causes*. Thus, it also follows from the intuition that similar diseases have similar treatments. However, this explanation received a lower rating since the *geneAssociatedWithCondition* relationship is not as specific as the *causes* relationship, i.e., it gives no indication of the type of association. A gene association is certainly still useful for determining whether diseases are similar or not, but a gene causation is stronger evidence that the diseases will have similar mechanistic behavior. In other words, because causation is a more specific, strong relationship, it is a better indicator that the diseases e_1



(a) An explanation which received rating (1) "reasonable."

(b) An explanation which received rating (2) "reasonable in some cases."

(c) An explanation which received rating (3) "neutral."

Fig. 3. Examples of explanations which received varying ratings according to our biomedical expert. Each edge with label p connecting nodes x and y represents the triple (x, p, y). For each explanation, the black triples represent the explanation pattern, and the gray, dashed edge represents the target predicate, which connects the endpoints e_0 and e_1. *GAWC* stands for *geneAssociatedWithCondition*. Based on the analysis of our biomedical expert, the explanations in (a) and (b) are reasonable since they indicate that diseases e_1 and e_2 are similar, and thus they are likely to have a common treatment, i.e., the drug e_0. However, the explanation in (a) is stronger since the *causes* relationship between gene e_3 and diseases e_1 and e_2 is stronger and more specific than the corresponding *geneAssociatedWithCondition* relationship in (b). The explanation in (c) is weaker due to the ambiguity of the *affects* relationship, i.e., there is no qualifier indicating the type of affect. Thus, there is not enough information to infer a strong relationship between drug e_0 and disease e_1.

and e_2 will behave similarly in their interactions with other biomedical entities, in particular, the drug e_0.

Finally, the explanation shown in Fig. 3c received a rating of (3) "neutral." This explanation is weaker than those previously mentioned largely due to fact that the explanation pattern reveals a much weaker connection between the drug e_0 and the disease e_1. In general, patterns with highly specific relationships are able to reveal more about the actual biological behavior of the entities involved. Thus, these patterns give a more clear indication of how the entities will interact. This is not the case for relationships like *affects*, which are useful, but ambiguous in that they give no clarification on the type of affect or how the affect happens. Thus, while this pattern does identify a connection between drug e_0 and disease e_1, it is not strong enough or specific enough to explain the fact $(e_0, \textit{treats}, e_1)$.

D. Pattern Usefulness

1) Experimental Design: In this experiment, we consider the potential of using the explanation derivation process as a means to generate new inference rules, i.e., explanations, which can provide explanations for facts other than those for which the explanations were originally derived. The explanations produced by other works cannot be generalized to new queries, as ours can, so we compare to no baselines in this experiment. In summary, our results illustrate that, by using knowledge graph patterns as explanations, we are able to explain many facts with the same explanation, which highlights that our explanations need not be derived anew for each query fact. In fact, our results suggest that a relatively small set of prevalent explanation patterns are actually capable of explaining most true facts.

To explore this potential use of explanations, we consider each explanation generated in the experiment discussed in Section V-B to be an inference rule. For each such explanation E, we calculate the prevalence of E as the proportion of true facts from the experimental data set of Section V-B whose set of explanations $\mathcal{O}$ output by Algorithm 1 included E.

$$prevalence(E) = \frac{|\{f_i \in T : E \in \mathcal{O}\}|}{|T|} \tag{12}$$

where T is the set of true facts from the experimental data set of Section V-B, and $\mathcal{O}$ is the set of explanations output by Algorithm 1 for input query f_i.

We form a new rule set comprised of the top k most prevalent explanations for target predicate *treats* and the top k most prevalent explanations for target predicate *not_treats*. We use a new experimental data set which consists of n true treats triples from ROBOKOP and is, in particular, distinct from that of Section V-B. We, again, remove these facts from ROBOKOP to avoid trivial explanations and then derive explanations for each fact via Algorithm 1 with our newly formed rule set as the input set of rules.

To evaluate the performance, we compute recall and accuracy, where recall is defined as the proportion of facts for which at least one explanation is found and accuracy is defined as the proportion of facts which were assigned an appropriate truth score, i.e. a positive score indicating "true," see Section V-B1 for details.

2) Experimental Details: In total, we extracted 614 unique explanations for target predicate *treats* and 579 unique explanations for target predicate *not_treats* in the experiment of Section V-B. For this experiment, we construct our rule set out of the top k most prevalent explanations for *treats* and the top k most prevalent for *not_treats*, for $k \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We use a maximum search depth of 1 since this ensures that each explanation is derived from a single rewriting of the input query by one of the input rules, i.e., each explanation is just one of the input rules. Thus, we can analyze the extent to which our previously derived explanations can directly serve as explanations for new facts. Our experimental data set consisted of 100 true treats triples from ROBOKOP, and we ran this experiment using each of our proposed explanation metrics: confidence, head coverage, and F1-score.

3) Results: Fig. 4 shows the recall and accuracy of this experiment when varying k from one to ten, where the top k most prevalent explanations for *treats* and for *not_treats* were used as the input rule set for Algorithm 1. The results of this experiment were identical for each of our proposed explanation metrics; for simplicity, we report these results only once in Fig. 4. We notice first that the accuracy is above

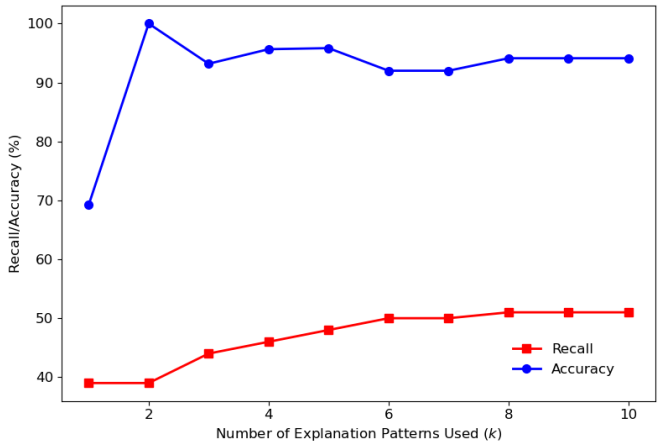


Fig. 4. The recall and accuracy achieved when explaining 100 new candidate facts by treating the top k most prevalent explanations, derived in the experiment of Section V-B, as the set of inference rules input to Algorithm 1. Recall indicates the proportion of candidate facts explained, and accuracy indicates the proportion assigned a positive truth score. The recall and accuracy scores shown were achieved by each of our proposed metrics: confidence, head coverage, and F1-score. The X-axis indicates k, the number of explanations used as rules; the Y-axis indicates the percent recall and percent accuracy.

90% for all k values except $k = 1$. This indicates that these rules are especially useful for reliably explaining facts and classifying them as true or false, even facts for which they were not originally derived. We are able to correctly identify most true facts based on their truth scores by using only these top k most prevalent explanations. Next, we notice that only 6 rules are required to achieve a recall of 50%. Therefore, the explanations we have derived occur frequently amongst *treats* triples, and only a small set of them may be required to explain most facts. Overall, this reveals that the explanations we derive are not too specific to the original query triples used to derive them; they are still valuable and applicable in new settings for new queries. Thus, we propose our explanation derivation approach as a potentially useful way to mine new, prevalent inference rules on knowledge graphs.

4) Limitations: In an effort to understand why our rules are not able to explain more facts, we analyze the density of ROBOKOP and observe that it is certainly not uniform; some areas are very dense while others are sparse. In particular, we note that sparse areas of the knowledge graph are much more difficult for tasks such as explanation derivation simply due to the lack of available data. We consider node degree to be a proxy for density and, for each fact in our experimental data set, we compute the minimum pair node degree, i.e., the minimum degree of its subject and object. We then classify each fact as (1) recovered: at least one explanation was found, or (2) unrecovered: no explanations were found. For this classification we consider the case $k = 10$, i.e., the 10 most prevalent explanations for *treats* and for *not_treats* comprised the rule set input to Algorithm 1. We present Fig. 5, which shows a boxplot of the distribution of the minimum

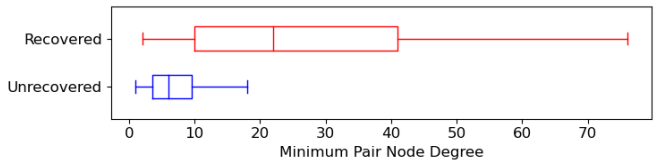


Fig. 5. A comparison of the distributions of the minimum pair node degree of recovered versus unrecovered facts. Recovered facts are those for which at least one explanation was found, and unrecovered facts are those for which no explanations were found, when using the top 10 most prevalent explanations for *treats* and for *not_treats* as the rule set input to Algorithm 1. The X-axis indicates the minimum pair node degree; the Y-axis indicates the group (recovered or unrecovered).

pair node degrees for each of the two sets.[12] We notice that both the spread and center of the two sets are very different, and, in particular, the minimum pair node degrees of covered facts tend to be much higher than those of the uncovered facts. This indicates that many of the uncovered facts come from sparse sections of the knowledge graph, and, thus, it is significantly more difficult to find explanations for these facts. To further confirm this analysis, we have conducted a Welch's t-test [40] between the two groups resulting in a p-value of 0.00001, which is significantly less than the standard cutoff of 0.05. This confirms that the node degrees are statistically significantly different between the two groups. Overall, this evidence suggests that graph sparsity is a key reason why many facts were unrecovered.

E. Discussion

We now summarize the key takeaways of our experimental results. We saw that not only are our proposed explanation evaluation metrics capable of achieving higher accuracy than existing metrics, see Section V-B, but they are also capable of identifying strong, biomedically reasonable explanations, see Section V-C. Therefore, we conclude that our metrics are more reliable overall than existing metrics and we propose them as potential universal metrics to fill the gap identified in [41]. Moreover, in Section V-C, we also saw that explanations in the form of knowledge graph patterns, and, in particular, those explanations derived by Algorithm 1 in our experiments, are both understandable and reasonable to biomedical experts. Finally, in Section V-D, we saw that our explanation derivation process is capable of producing valuable explanations which can serve as useful inference rules for future explanation tasks. Thus, treating explanations as knowledge graph patterns means that they need not be derived anew for each query fact; time can be saved by simply applying previous, reliable explanations to new queries.

VI. Conclusion and Future Work

In this work, we present drug discovery as an impactful application of the explainable fact-checking problem and design an approach suitable for the biomedical domain. To this

[12]We have omitted outliers from Fig. 5 to ensure that the boxplots are readable.

end, we present a novel view of explanations as knowledge graph patterns, which enables us to consider a new strategy for explanation evaluation that incorporates the existing data as evidence. Our proposed explanations and metrics are understandable and, thus, useful for biomedical experts. For our application to drug discovery and potentially other tasks, our experimental results indicate that our proposed metrics are accurate and can reliably identify strong explanations. Moreover, they indicate that our derived explanations are reasonable to domain experts and useful for explaining queries beyond those for which they were derived; thus, time deriving these explanations can be saved for future queries. Overall, the success of our experiments suggests that our proposed explanation derivation and evaluation is a viable approach for drug candidate identification for drug discovery, and potentially other tasks.

We identify the following extensions as possible future work. Performing a similar analysis with other relationship types would enable us to identify other useful connections between drugs, genes, diseases, etc., as well as provide a way to analyze the difference in explainability between different relationship types. Comparing directly with other explanation formats would provide more evidence that our knowledge graph patterns are a productive, useful view of explanations. Additionally, our next step is to directly incorporate entity types into our explanations as biomedical experts have indicated that this will improve the quality and reliability. Moreover, we plan to develop metrics which further refine those proposed in this work in an effort to incorporate the specific entities involved in each pattern, e.g., incorporation of triple weights and node promiscuity.

ACKNOWLEDGMENT

The authors would like to thank Jing Ao, Daniel Korn, Andrew Thieme, and Jon-Michael Beasley for many insightful discussions, as well as Chris Bizon, Patrick Wang, and the entire ARAGORN team at the Renaissance Computing Institute, who developed ROBOKOP.

REFERENCES

[1] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, "Generating Fact Checking Explanations," *arXiv:2004.05773 [cs]*, Apr. 2020.

[2] N. Kotonya and F. Toni, "Explainable Automated Fact-Checking for Public Health Claims," *arXiv:2010.09926 [cs]*, Oct. 2020.

[3] L. D. Raedt, A. Kimmig, and H. Toivonen, "ProbLog: A Probabilistic Prolog and its Application in Link Discovery," p. 6.

[4] M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum, "ExFaKT: A Framework for Explaining Facts over Knowledge Graphs and Text," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. Melbourne VIC Australia: ACM, Jan. 2019, pp. 87–95.

[5] N. Ahmadi, J. Lee, P. Papotti, and M. Saeed, "Explainable Fact Checking with Probabilistic Answer Set Programming," *arXiv:1906.09198 [cs]*, Jun. 2019.

[6] P. Lin, Q. Song, and Y. Wu, "Fact Checking in Knowledge Graphs with Ontological Subgraph Patterns," *Data Science and Engineering*, vol. 3, no. 4, pp. 341–358, Dec. 2018.

[7] S. J. Capuzzi, T. E. Thornton, K. Liu, N. Baker, W. I. Lam, C. P. O'Banion, E. N. Muratov, D. Pozefsky, and A. Tropsha, "Chemotext: A Publicly-Available Web Server for Mining Drug-Target-Disease Relationships in PubMed," *Journal of chemical information and modeling*, vol. 58, no. 2, pp. 212–218, Feb. 2018.

[8] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media," in *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*. Perth, Australia: ACM Press, 2017, pp. 1003–1012.

[9] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 22–32.

[10] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "dEFEND: Explainable Fake News Detection," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage AK USA: ACM, Jul. 2019, pp. 395–405.

[11] F. Yang, S. K. Pentyala, S. Mohseni, M. Du, H. Yuan, R. Linder, E. D. Ragan, S. Ji, and X. B. Hu, "XFake: Explainable Fake News Detector with Visualizations," in *The World Wide Web Conference on - WWW '19*. San Francisco, CA, USA: ACM Press, 2019, pp. 3600–3604.

[12] Y.-J. Lu and C.-T. Li, "GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 505–514.

[13] L. Wu, Y. Rao, Y. Zhao, H. Liang, and A. Nazir, "DTCA: Decision Tree-based Co-Attention Networks for Explainable Claim Verification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 1024–1035.

[14] P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia, "Finding Streams in Knowledge Graphs to Support Fact Checking," *arXiv:1708.07239 [cs]*, Aug. 2017.

[15] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational Fact Checking from Knowledge Networks," *PLOS ONE*, vol. 10, no. 6, p. e0128193, Jun. 2015.

[16] B. Shi and T. Weninger, "Discriminative predicate path mining for fact checking in knowledge graphs," *Knowledge-Based Systems*, vol. 104, pp. 123–133, Jul. 2016.

[17] S. Russell and P. Norvig, "Artificial intelligence: A modern approach, global edition 4th," *Foundations*, vol. 19, p. 23, 2021.

[18] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "AMIE: association rule mining under incomplete evidence in ontological knowledge bases," in *Proceedings of the 22nd international conference on World Wide Web - WWW '13*. Rio de Janeiro, Brazil: ACM Press, 2013, pp. 413–422.

[19] N. Ahmadi, V.-P. Huynh, V. Meduri, S. Ortona, and P. Papotti, "Mining Expressive Rules in Knowledge Graphs," *Journal of Data and Information Quality*, vol. 12, no. 2, pp. 1–27, May 2020.

[20] S. Ortona, V. V. Meduri, and P. Papotti, "Robust Discovery of Positive and Negative Rules in Knowledge Bases," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, Apr. 2018, pp. 1168–1179, iSSN: 2375-026X.

[21] A. Sadeghian, M. Armandpour, P. Ding, and D. Z. Wang, "DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs," *arXiv:1911.00055 [cs, stat]*, Oct. 2019.

[22] K. Kolthoff and A. Dutta, "Semantic Relation Composition in Large Scale Knowledge Bases," p. 14.

[23] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla, and H. Stuckenschmidt, "Fine-Grained Evaluation of Rule- and Embedding-Based Systems for Knowledge Graph Completion," in *The Semantic Web – ISWC 2018*. Cham: Springer International Publishing, 2018, vol. 11136, pp. 3–20, series Title: Lecture Notes in Computer Science.

[24] Z. Wang and J. Li, "RDF2Rules: Learning Rules from RDF Knowledge Bases by Mining Frequent Predicate Cycles," *arXiv:1512.07734 [cs]*, Dec. 2015.

[25] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek, "Fast rule mining in ontological knowledge bases with AMIE $$+$$ +," *The VLDB Journal*, vol. 24, no. 6, pp. 707–730, Dec. 2015.

[26] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, 1993, pp. 207–216.

[27] C. Bizon, S. Cox, J. Balhoff, Y. Kebede, P. Wang, K. Morton, K. Fecho, and A. Tropsha, "Robokop kg and kgb: integrated knowledge graphs

from federated sources," *Journal of chemical information and modeling*, vol. 59, no. 12, pp. 4968–4973, 2019.

[28] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. Philadelphia, PA, USA: ACM Press, 2006, p. 631.

[29] T. Trouillon, "Knowledge Graph Completion via Complex Tensor Factorization," p. 38.

[30] T. Trouillon, J. Welbl, and S. Riedel, "Complex Embeddings for Simple Link Prediction," p. 10.

[31] A. Garcia-Duran, A. Bordes, N. Usunier, and Y. Grandvalet, "Combining Two and Three-Way Embedding Models for Link Prediction in Knowledge Bases," *Journal of Artificial Intelligence Research*, vol. 55, pp. 715–742, Mar. 2016.

[32] A. Borrego Díaz, D. Ayala Hernández, I. C. Hernández Salmerón, C. R. Rivero, and D. Ruiz Cortés, "Cafe: Fact checking in knowledge graphs using neighborhood-aware features," *Semantic Web, 2020*, 2020.

[33] Q. Wang, J. Liu, Y. Luo, B. Wang, and C.-Y. Lin, "Knowledge Base Completion via Coupled Path Ranking," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1308–1318.

[34] J. Alspector, R. Allen, V. Hu, and S. Satyanarayana, "Stochastic Learning Networks and their Electronic Implementation," in *Neural Information Processing Systems*, D. Anderson, Ed. American Institute of Physics, 1988.

[35] S. Mazumder and B. Liu, "Context-aware Path Ranking for Knowledge Base Completion," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 1195–1201, Aug. 2017.

[36] S. M. Kazemi and D. Poole, "SimplE Embedding for Link Prediction in Knowledge Graphs," *arXiv:1802.04868 [cs, stat]*, Oct. 2018.

[37] A. Atiya, "Translating Embeddings for Modeling Multi-relational Data," in *Neural Information Processing Systems*, D. Anderson, Ed. American Institute of Physics, 1988.

[38] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," *arXiv:1412.6575 [cs]*, Aug. 2015.

[39] N. Nakashole and T. M. Mitchell, "Language-Aware Truth Assessment of Fact Candidates," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1009–1019.

[40] B. L. Welch, "The Generalization of 'Student's' Problem when Several Different Population Variances are Involved," *Biometrika*, vol. 34, no. 1/2, p. 28, Jan. 1947.

[41] N. Kotonya and F. Toni, "Explainable Automated Fact-Checking: A Survey," *arXiv:2011.03870 [cs]*, Nov. 2020.