Combining Inductive and Deductive Reasoning for Query Answering over Incomplete Knowledge Graphs

Medina Andresel∗
AIT Austrian Institute of Technology
Vienna, Austria
medina.andresel@ait.ac.at

Trung-Kien Tran
Bosch Center for Artificial Intelligence
Renningen, Germany
trungkien.tran@de.bosch.com

Csaba Domokos
Bosch Center for Artificial Intelligence
Renningen, Germany
csaba.domokos@de.bosch.com

Pasquale Minervini
University of Edinburgh
Edinburgh, United Kingdom
p.minervini@ed.ac.uk

Daria Stepanova
Bosch Center for Artificial Intelligence
Renningen, Germany
daria.stepanova@de.bosch.com

ABSTRACT
Current methods for embedding-based query answering over incomplete Knowledge Graphs (KGs) only focus on inductive reasoning, i.e., predicting answers by learning patterns from the data, and lack the complementary ability to do deductive reasoning, which requires the application of domain knowledge to infer further information. To address this shortcoming, we investigate the problem of incorporating ontologies into embedding-based query answering models by defining the task of embedding-based ontology-mediated query answering. We propose various integration strategies into prominent representatives of embedding models that involve (1) different ontology-driven data augmentation techniques and (2) adaptation of the loss function to enforce the ontology axioms. We design novel benchmarks for the considered task based on the LUBM and the NELL KGs and evaluate our methods on them. The achieved improvements in the setting that requires both inductive and deductive reasoning are from 20% to 55% in HITs@3.

CCS CONCEPTS
• Computing methodologies → Description logics; Machine learning approaches; • Computer systems organization → Neural networks.

KEYWORDS
Knowledge Graphs, Ontologies, Embeddings, Query Answering, Neuro-Symbolic AI

∗The work has been done during the PhD sabbatical of Medina Andresel at Bosch Center for Artificial Intelligence

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM ’23, October 21–25, 2023, Birmingham, United Kingdom
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9007-0/23/10 $15.00
https://doi.org/10.1145/3583780.3614816

1 INTRODUCTION

Knowledge Graphs (KGs) have recently received much attention due to their relevance in various applications, such as natural question answering or web search. Prominent KGs include NELL [8], YAGO [31], and Wikidata [12]. A KG describes facts about entities by interconnecting them via relations, e.g., hasAlumnus(mit, bob) in Figure 1 states that Bob is an MIT alumnus.

A crucial task in leveraging information from knowledge graphs is that of answering logical queries such as Who works for Amazon and has a degree from MIT?, which can be formally written as $q(X) \leftarrow \text{degreeFrom}(X, \text{mit}) \land \text{worksFor}(X, \text{amazon})$. Answering such queries is very challenging when KGs are incomplete, which is often the case due to their (semi-)automatic construction, and obtaining complete answers typically requires further domain knowledge, i.e., the application of deductive reasoning. For instance, mary is a missing but desired answer of $q$ that cannot be obtained by combining the fact managerAt(mary, amazon), predicted using machine learning models, and the axiom stating that managerAt implies worksFor in the ontology $O$ of Figure 1. Therefore, such a task requires both inductive and deductive reasoning.

Recently, Knowledge Graph Embedding (KGE) techniques [33] that can be used to predict missing facts have been proposed to answer logical queries over incomplete KGs [20, 30, 36, 37, 39]. At the same time, in the Knowledge Representation and Reasoning area answering queries over incomplete data has also received a lot of attention and one of the most successful approaches for this task is to exploit ontologies when querying KGs, referred to as Ontology-Mediated Query Answering [OMQA, 38].

While promising, existing embedding-based methods do not take ontologies, which formalize domain knowledge, into account. Since large portions of expert knowledge can be conveniently encoded using ontologies, the benefits of coupling ontology reasoning and
embedding methods for KG completion are evident and have been acknowledged in several works, e.g. [18, 28]. However, to the best of our knowledge, coupling inductive and deductive reasoning to answer queries over incomplete KGs has not been considered yet.

Answering queries over the KG augmented with triples resulting from the naive process of interchangeably using embedding methods and ontology reasoning, comes with a big scalability challenge [27] and commonly known error accumulation issues. In practice, we need to restrict ourselves to computing merely small subsets of likely fact predictions required for answering a given query; thus more sophisticated proposals are needed. Hence, we investigate three open questions: (1) How to adapt existing OMQA methods to the setting of KGEs? (2) How do different data augmentation strategies impact the accuracy of existing embedding models for the OMQA task? (3) Does enforcing ontology axioms in the embedding space via loss function help to improve inductive and deductive reasoning performance?

We answer (1)-(3) by the following contributions:

- We formally define the novel task of Embedding-Based Ontology-Mediated Query Answering (E-OMQA), analyze and systematically compare various extensions of embedding-based query answering models to incorporate ontologies.
- We propose ontology-driven strategies for sampling queries to train embedding models for query answering, as well as a loss function modification to enforce the ontology axioms within the embedding space, and demonstrate the effectiveness of these proposals on widely-used representatives of query-based and atom-based models.
- As no previous benchmarks exist for E-OMQA, we design new ones using LUBM and NELL, i.e., well-known benchmarks for OMQA and embedding models, respectively.
- Extensive evaluation shows that enforcing the ontology via the loss function, in general, improves the deductive power regardless of how the training data is sampled, while ontology-driven sampling strategy has a further significant positive impact on performance. We obtain overall improvements, ranging from 20% to 55% in HIT@3, in the settings that require both inductive and deductive reasoning.

\[16\]

Table 1: DL syntax and semantics defined using FO interpretations \((\Delta^I, I^P)\) with a non-empty domain \(\Delta^I\) and an interpretation function \(I^p\). \(C\) and \(D\) denote concepts in DL-Lite\(_R\).

| DL Syntax | Semantics |
|-----------|-----------|
| \(e\) | \(e^I \in \Delta^I\) |
| \(A(p)\) | \(A^I \subseteq \Delta^I\) |
| \(\exists p\ r\|\) | \(\exists y \gamma \in (\Delta, \Delta^I)\) |
| \(\exists p\ r\|\) | \(= (d, d')\ in \Delta^I\) |
| \(\exists p\ r\|\) | \(\in \Delta^I\) |

2 PRELIMINARIES

Knowledge Graphs and Ontologies. We assume a signature consisting of countable pairwise disjoint sets \(E, C,\) and \(R\) of entities (constants), concepts (types), and roles (binary relations), respectively. A knowledge graph \(G\) (a.k.a. ABox) is a set of triples, such as \((\text{mit}, \text{type}, \text{University})\) and \((\text{bob}, \text{worksFor}, \text{mit})\), where \(\text{mit}, \text{bob}\in E, \text{worksFor}, \text{type}\in R, \) and \(\text{University}\in C\). These triples can also be represented as \(\text{University}(\text{mit})^1\) and \(\text{worksFor}(\text{bob}, \text{mit})\). An ontology \(O\) (a.k.a. TBox) is a set of axioms in Description Logics \([4]\) over the signature \(\Sigma = (E, C, R)\). We focus on ontologies in DL-Lite\(_R\) DL fragment \([7]\) that have the following syntax:

\[A \subseteq A' \subseteq \exists p \in A \exists p \in A \exists p \in A \exists p \subseteq p \subseteq p \subseteq s\ |

\[A(c) \in A', a \subseteq c, \text{ and } a \subseteq c' \subseteq p \subseteq p \subseteq s\]

where \(A, A' \subseteq C\) are concepts and \(p, s \subseteq R\) are roles and \(p^\perp\) denotes the inverse relation of \(p\). The KG and its ontology from Figure 1 are in DL-Lite\(_R\). The DL-Lite\(_R\) syntax and semantics are summarized in Table 1. Given a KG \(G\) and an ontology \(O\), an interpretation \(I\) is a model of \(G\) wrt \(O\) if \(I\) satisfies each fact in \(G\) and each axiom in \(O\). For DL-Lite\(_R\), a canonical model exists that can be homomorphically mapped into any other model, obtained from the deductive closure \(\Omega^\infty(\bar{G})\), which extends \(G\) with triples derived from existing facts in \(G\) by exhaustively applying axioms in \(O\) \([7]\).

Ontology-Mediated Query Answering. A query atom is an expression of the form \(p(T_1, T_2)\), where \(p \in R\), and each \(T_i \in V \cup \Sigma\) is called a term, with \(V\) disjoint with \(E, C,\) and \(R\) being a set of variables. A monadic conjunctive query (CQ) \(q(G)\) is a First-Order (FO) formula of the form \(q(G) = \exists \bar{Y} \left( p_1(T_1) \land \cdots \land p_n(T_n) \right)\) where each \(p_i(T_i)\) is a query atom, and \(\forall s(q) = \forall \bar{X} \cup \bar{Y}\) denotes the set of variables appearing in \(q\), with \(\bar{X} \neq \bar{Y}\) being the answer variable. In this work, we focus on monadic Existential Positive FO (EPFO) queries, i.e., unions of monadic CQs \([10]\). For a query \(q(G)\) and a KG \(G\), a constant \(a\) is an answer of \(q\) if a mapping \(\pi: \forall \bar{s}(q) \mapsto E\) exists, s.t. \(\pi(q) = A\) and \(\pi(a) = a\). \(q(G)\) are the answers of \(q\) on \(G\).

Ontology-Mediated Query Answering (OMQA) concerns answering queries by accounting for both the KG and the accompanying ontology. Since the model constructed from \(\Omega^\infty(\bar{G})\) can be homomorphically mapped to every other model, the deductive closure can be used to evaluate queries \([7]\).

Definition 2.1. Given a KG \(\bar{G}\) and an ontology \(O\), an entity \(a\) from \(\bar{G}\) is a certain answer of \(q(G)\) over \((\bar{G}, O)\) if \(a\) is an answer to \(q(G)\) over \(\Omega^\infty(\bar{G})\). We use \(q(\bar{G}, O)\) to denote the set of certain answers of \(q\) over \((\bar{G}, O)\).
Let \( q \) and \( q' \) be two monadic queries over \((G, O)\), then \( q \) is contained in \( q' \) w.r.t. \( O \) if \( q[G, O] \subseteq q'[G, O] \); we call \( q \) a specialization of \( q' \) (written as \( q \preceq q' \)), and \( q' \) a generalization of \( q \) (written as \( q \succeq q' \)). Query generalizations and specializations can be obtained by exploiting ontology axioms; such process (and result) is referred to as query rewriting.

Example 2.2. Consider the KG \( G \) in Figure 1 and the query \( q(X) \leftarrow \text{type(X, Professor)} \land \text{degreeFrom(X, mit)}. \) Since \( X \in q[\cdot] \), it is a certain answer. Moreover, according to \( O, \) if Professor is a sub-type of Professor and degreeFrom is inverse of hasAlumnus, then bob is also a certain answer. For \( q'(X) \leftarrow \text{type(X, AP:Professor)} \land \text{degreeFrom(X, mit)} \) it holds that \( q \preceq q' \) as \( q'[G, O] \neq q[G, O] \).

Embedding-Based Approximate Query Answering. Since, in reality, KGs might be missing facts, existing query answering techniques designed for complete data might not compute all answers. In such settings, one assumes that the given KG \( G \) is a subset of a complete but unobservable KG \( G^t \), and one aims at estimating the likely answers to \( q \) over \( G^t \). E.g., \( G^t \) for the graph \( G \) given in Figure 1 includes the links denoted by dashed edges. In practice, to evaluate the accuracy of a considered method, \( G^t \) is typically fixed at the beginning, and \( G \) is created by removing facts from \( G^t \).

The set of all answers to a given query \( q \) comprises those that can be obtained by directly querying the given KG \( G \) only and those that require predicting missing KG facts. Thus, one typically distinguishes easy and hard answers as follows:

Definition 2.3. Given a KG \( G \), a subgraph of a complete but unobservable KG \( G^t \) and a query \( q(X) \), \( a \) is an easy answer to \( q \) if \( a \in q[G] \), and it is a hard answer to \( q \) if \( a \in q[G^t] \setminus q[G] \).

Recently, embedding-based methods have been proposed for approximate answering of existential positive FO queries over incomplete KGs [9, 25, 30, 36, 37]. Broadly, such methods can be divided into two categories: query-based [9, 25, 30, 36, 37] and atom-based [2]. Generally, any neural QA model relies on an embedding function which maps entities and relations into a \( d \)-dimensional embedding space. It then computes a score of each entity \( c \) for being an answer to a given query \( q \) over \( G^t \) via a scoring function \( \phi_q(e) : \mathbb{R}^d \rightarrow [0, 1] \), where \( e \) denotes the embedding vector of \( c \). Using these scoring functions, the final embedding QA function \( \mathbf{E}_G \) takes as input a query and returns its approximate answers over the knowledge graph \( G^t \), i.e., answers that have the scoring above some predefined threshold. We say that \( \mathbf{E}_G \) is reliable w.r.t. \( G^t \) whenever for each query \( q, c \) is an approximate answer to \( q \) iff \( c \) is an answer to \( q \) over \( G^t \). Clearly, the challenge of identifying hard answers is still valid also for embedding QA models.

3 EMBEDDING-BASED OMQA

Existing methods for embedding-based query answering compute approximate answers to queries over an unobservable KG \( G^t \) by performing inductive reasoning. However, they are not capable of simultaneously applying deductive reasoning, and thus cannot account for ontologies with which KGs are often accompanied.

To address this shortcoming, we propose ways to combine inductive and deductive reasoning for approximate query answering over incomplete KGs. For that, we first formalize the task of Embedding-based Ontology-Mediated Query Answering (E-OMQA) in which both types of reasoning are exploited. The goal of this task is to approximate certain answers to OMQs over \( G^t \).

Definition 3.1 (E-OMQA). Let \( G \) be a KG, which is a subgraph of a complete but not observable KG \( G^t \), let \( O \) be an ontology and \( q \) a query. Embedding-based ontology-mediated query answering is concerned with constructing an embedding function \( \mathbf{E}_G, O \) that is reliable w.r.t. \( O^\text{om} \).

Note that, \( q[G^t, O] \) subsumes both \( q[G] \), the answers requiring inductive reasoning, and \( q[G, O] \), the answers computed via deductive reasoning only. Analogously as for embedding-based query answering, for E-OMQA, we distinguish between easy certain answers and hard certain answers as follows.

Definition 3.2. Given a KG \( G \), a subgraph of a complete but unobservable KG \( G^t \), an ontology \( O \) and a query \( q(X) \), \( a \) is an easy certain answer to \( q \) if \( a \in q[G] \), and it is a hard certain answer to \( q \) if \( a \in q[G^t] \setminus q[G] \).

Next, we discuss several embedding-based methods for ontology-mediated query answering under incompleteness.

Query Rewriting over Pre-trained Models. In the traditional OMQA setting, each query \( q \) can be evaluated by first rewriting \( q \) into a set of FO-queries \( Q_O \) and then evaluating each query \( q \) over \( G \) alone. For E-OMQA, this amounts to constructing an embedding QA function \( \mathbf{E}_G \) for \( G \) alone and using it to compute the answers to all queries in \( Q_O \). For DL-LiteR, such FO-rewriting is obtained by extensively applying ontology axioms in a specializing fashion, which results in the so-called perfect reformulation [7].

Ontology-Aware Models. An alternative to query rewriting is to develop an embedding query answering function that accounts for axioms in \( O \). To the best of our knowledge, there are no KGE models that directly address the problem of E-OMQA. Thus, we suggest the following: (1) Train existing embedding models for QA on the data derived from \( O^\text{om} \) instead of \( G \); (2) Develop an ontology-aware embedding model that will be trained on \( G \) but will have special terms in the training objective structurally enforcing \( O \). (3) Combine (1) and (2), i.e., train ontology-aware embedding models on the data derived from \( O^\text{om} \).

While the proposed approaches can be realized on top of any embedding model for logical query answering, in this work, we verify their effectiveness on a query-based model Query2Box [36] and an atom-based model CQD [2]. In Section 3.1, we present several effective ontology-driven training methods for realizing (1). As for (2), building on Query2Box, in Section 3.2 we develop its ontology-aware version. Moreover, we build an ontology-aware extension of CQD, on top of the neural link predictor using adversarial sets regularization (ASR) [32] to enforce the ontology axioms. We chose this approach, since it is general and allows us to incorporate rules into any off-the-shelf neural link predictor. In our experiments, we use ComplEx-N3 [29] as it requires minimal modification to CQD and outperforms other neural link predictors (see [29]). Finally, we evaluate the effectiveness of our ontology-driven strategies from Section 3.1 on the extended models described in Section 3.2 and
verify the feasibility of the classical query-rewriting approach in the knowledge graph embedding setting.

3.1 Ontology-Driven Data Sampling

Let \( Q_D \) be the set of all possible queries that can be formed using the signature \( \Sigma \). During the training process, existing embedding models are trained on a set of sampled queries of certain shapes and their answers over the KG \( G \).

3.1.1 Random Query Sampling. The existing sampling procedure from the literature [19, 36] arbitrarily chooses entities and relations in the graph to construct queries of various shapes. Query2Box is trained on complex queries involving multiple atoms, while CQD is trained on simple queries involving multiple atoms. For verifying how well the model generalizes, the test set includes queries whose shapes have not been encountered during training.

It is not guaranteed to capture ontological knowledge that comes with the knowledge graph, and using it over \( O^\infty(G) \) could generate a bias towards concepts and roles that are very general. Moreover, using all possible queries from \( Q_D \) with their certain answers might be infeasible in practice. In the following, we discuss various options for guiding the sampling of queries to train ontology-aware knowledge graph embedding models for query answering.

3.1.2 Incorporating Query Rewritings and Certain Answers. The first natural attempt to incorporate ontologies is to consider certain answers, which for DL-Lite\( \mathcal{Q} \) can be done efficiently. An example of this training case is to randomly sample query \( q(Y) \leftarrow \exists X.\text{hasAlumnus}(\text{mit},X) \land \text{worksFor}(X,Y) \) and, given \( (G, O) \) in Fig. 1, use it along with all its certain answers: mit, yale during training. To incorporate the ontology, we can randomly sample queries over the KG, using the standard procedure, and then add their generalizations and specializations obtained using the rules in Tab. 2. To rewrite a query we select an atom and apply an ontology axiom. For example, the first rule \( (R1) \) applies a concept inclusion axiom, while \( (R6) \) applies a role inclusion.

The specializations of a query \( q \) (i.e. \( \text{Spec}(q) \)), incorporate more specific information regarding the answers of \( q \), while the generalizations of \( q \) (i.e. \( \text{Geni}(q) \)) incorporate additional related entities.

Example 3.3. Take the queries \( q_1(X) \leftarrow \exists Y.\text{type}(X, \text{University}) \) and \( q_2(X) \leftarrow \exists Z.\text{teachesAt}(Z, X) \). Using \( R2 \) in Table 2 and (5) in Figure 1 we get \( q_1 \sim q_2 \), i.e. \( q_2 \) is a specialization of \( q_1 \).

In general, there are exponentially many rewritings, thus we fix a rewriting depth, up to which the respective training queries are generated, via a dedicated parameter.

3.1.3 Strategic Ontology-Based Sampling. While adding generalizations and specializations of randomly selected queries should partially reflect the background knowledge, many relevant axioms can be overlooked if they are not explicitly captured in the data. To overcome this, we consider the set of target query shapes as directed acyclic graphs (DAGs) of the form \((N, E)\), where \(N\) is a set of nodes and \(E \subseteq N \times N\) is a set of directed edges. The set of training queries is then obtained by applying a labeling function that assigns symbols in \( \Sigma \) to nodes and edges based on the ontology.

### Definition 3.4 (Query Shape).

A query shape \( S \) is a tuple \((N, E, n)\) such that \((N, E)\) is a DAG and \( n \in N \) is the distinguished node of \( S \) (i.e., the node for the answer variable). For a given set of relations and constants in \( \Sigma \), a labeling function \( f : N \cup E \rightarrow \Sigma \cup \mathbb{V} \) maps each node to either a variable or an entity and each edge to a relation symbol in the KG signature \( \Sigma \).

We rely on the ontology when labeling query shapes to create semantically meaningful queries. Let \( \Sigma' \) be the reflexive and transitive closure of \( \Sigma \). Then, for a given relation \( p \) we have:

- \( \text{inv}(p) = \{ p' \mid p \subseteq p' \subseteq \bigcup O \} \), \( \text{dom}(p) = \{ A \mid \exists p' \subseteq A \in O \) s.t. \( p \subseteq p' \subseteq A \subseteq A' \}
- \( \text{range}(p) = \{ A \mid \exists p' \subseteq A \in O \) s.t. \( p' \subseteq p \subseteq A' \subseteq A \}
- \( \text{follows}(p) = \{ p' \mid \text{range}(p) \cap \text{range}(p') \neq \emptyset \} \)
- \( \text{inter}_{\text{r}}(p) = \{ p' \mid \text{dom}(p) \cap \text{dom}(p) \neq \emptyset \} \)
- \( \text{inter}_{\text{d}}(p) = \{ p' \mid \text{dom}(p) \cap \text{dom}(p) \neq \emptyset \} \)

Intuitively, for a given relation \( p \), the set \( \text{inv}(p) \) contains all inverse relations of \( p \), \( \text{dom}(p) \) contains all domain types for \( p \), \( \text{range}(p) \) all range types for \( p \), \( \text{follows}(p) \) stores all relations \( p' \) which can follow \( p \), and \( \text{inter}_{\text{r}}(p), \text{inter}_{\text{d}}(p) \) contain resp. all relations \( p' \) which can intersect with \( p \) on range and domain positions. Then, for each shape we label nodes and edges to create queries...
1. If \( r_2 \in \text{follows}(r_1) \): \( r_1 \rightarrow \Delta \rightarrow r_2 \)
2. If \( A \in \text{dom}(r) \) or \( A \in \text{range}(r) \): \( r \rightarrow \text{type} \rightarrow A \)
3. If \( A \in \text{range}(r) \) or \( A \in \text{dom}(r) \): \( r \rightarrow \text{type} \rightarrow A \)
4. If \( r_2 \in \text{inter}(r_1) \): \( r_1 \rightarrow r_2 \)
5. If \( r_2 \in \text{inter}(r_1) \): \( r_1 \rightarrow r_2 \)
6. If \( A_1 \subseteq A \) and \( A_2 \subseteq A \): \( A_1 \rightarrow \text{type} \rightarrow A_2 \)

Figure 2: Ontology-driven rules to label query shapes; \( r^- \) denotes any of \( \text{inv}(r) \).

the query and its answers is minimized, while the distance to this box from other randomly sampled non-answers is maximized.

In what follows, we present our proposal for integrating ontological axioms into the Query2Box model. Similarly to [36], we define the distance between \( q \in \mathbb{R}^d \times \mathbb{R}^d \) and \( v \in \mathbb{R}^d \) as \( d(q, v) = ||v - c_v||_1 \), namely the L1 distance from the entity \( v \) to the center of the box. Using the sigmoid function we transform the distance into the (0, 1) interval, that is, \( p(v \mid q) = \sigma(-d(q, v) - \gamma) \), where \( \gamma > 0 \) is a margin, which denotes the probability of \( v \in q[G', O] \).

Note that for every ontological axiom its both left- and right-hand side can be turned into queries. When embedding those queries as boxes, axioms can be naturally enforced if in the vector space the inclusion of the boxes corresponding to the respective queries is ensured. For a query \( q \), let \( Gen(q) = \{ q_1 \ldots q_n \} \) be the set of all generalizations of \( q \) based on \( O \). Given a train query \( q \) and \( v \in q[G, O] \), we aim at maximizing \( \prod_{i=1}^{n} p(v \mid q_i)^{\beta_i} \), where \( \beta_i \geq 0 \) is a weighting parameter for all \( i = 1, \ldots, n \). This is achieved by minimizing the negative log-likelihood: \( -\log \left( \prod_{i=1}^{n} p(v \mid q_i)^{\beta_i} \right) = -\sum_{i=1}^{n} \beta_i \log (p(v \mid q_i)) \). By exploiting that \( \sigma(x) = 1 - \sigma(-x) \), for any \( v' \notin q[G', O] \), we have \( p(v' \mid q) = 1 - p(v \mid q) = \sigma(d(q, v) - \gamma) \).

Our goal is to enforce that if \( q' \in Gen(q) \) then the box of \( q' \) contains the box of \( q \). In order for that to hold, we need to ensure that, if \( a \) is an answer to \( q \) then the distance not only between \( a \) and \( q \) should be minimized, but also between \( a \) and all generalizations of \( q \). The following training objective reflects our goal:

\[
L = -\sum_{i=1}^{n} \beta_i \log \sigma(\gamma - d(q, q_i)) - \sum_{j=1}^{k} \frac{1}{k} \log \sigma(d(q', q) - \gamma),
\]

where \( q' \notin q[G', O] \) is a random entity for all \( j = 1, \ldots, k \) obtained via negative sampling. In our experiments, we use \( \beta_i = |Gen(q)|^{-1} = 1/n \).

**Example 3.5.** In Figure 3, the entities and relations are embedded into the vector space as points and projection operators, resp. The embedding of \( q(Y) \leftarrow \exists X \text{hasAlumnus}(mit, X) \land \text{worksFor}(X, Y) \) is represented by the larger gray box, obtained by applying the projection hasAlumnus to the embedding of entity mit followed
by the projection on worksFor. To enforce teachesAt ⊑ worksFor we ensure that the box of \( q'(Y) \leftarrow 3X.\text{hasAlumnus}(\text{mit}, X) \land \text{teachesAt}(X, Y) \), is contained in the box corresponding to \( q \).

Conceptually, our training data sampling techniques and the loss function modifications are flexible in terms of the DL, in which the ontology is encoded. The only restriction is the existence of efficient query rewriting algorithms.

3.3 Ontology-aware CQD

A prominent atom-based query-answering method is CQD [2], which relies on neural link predictors for answering atomic sub-queries, and then aggregates the resulting scores via t-norms.

We now describe how we inject the ontology axioms into the neural link predictor employed by CQD [2]. For that we rely on the FO translation of the DL axioms. Following [32], for each rule the goal is to identify the entity embeddings which maximize an inconsistency loss, i.e., the entities for which the scoring of the head is much lower compared to the scoring of the body. For example, given the rule \( \Gamma: \forall X, Y. \text{teachesAt}(X, Y) \rightarrow \text{type}(Y, \text{University}) \), the goal is to map the variables to \( d \)-dimensional embeddings, i.e., \( \phi : \text{var}(\Gamma) \mapsto \mathbb{R}^d \), s.t. \( \text{score}_{\text{teachesAt}}(\phi(X), \phi(Y)) - \text{score}_{\text{type}}(\phi(Y), \text{University}) \) is maximal, where \( \{x\}^+ = \max\{x, 0\} \) with \( x \) being the integral part of \( x \), and \( \text{score}_{\text{r}} \) being the scoring function for the relation \( r \) determining whether there is an \( r \)-edge between any two given entities. Mapping \( \phi \) determines a so-called adversarial input set, which is used as an adaptive regularizer for the neural link predictor. The inconsistency loss is then incorporated into the final loss function of the ontology-aware model which tries to minimize the maximal inconsistency loss while learning to predict the target graph over the given sets of correct triples. In experiments we rely on the existing implementation of the adversarial sets regularisation method in ComplEx-N3, which is the default neural link predictor for CQD.

4 EVALUATION

We evaluate our training strategies on popular models: Query2Box (Q2B) and CQD, as well as our ontology-aware adaptations O2B and CQD\textsuperscript{ASR}. Specifically, we test their ability to perform inductive reasoning, deductive reasoning, and their combination.

4.1 Benchmarks for E-OMQA

Since the task of embedding-based ontology mediated query answering has not been considered in the literature before, no benchmarks for it existed prior to our work. Thus, we have created two novel benchmarks based on LUBM [17] and NELL [8] KGs, available online\(^3\). LUBM has a rich ontology including domain and range axioms as well as concept and role inclusions, while the NELL KG is accompanied with a more simple ontology containing only (inverse) role inclusions. Following common practice, each input KG is completed w.r.t. inverse edges. In Table 3 we present the number of ontology axioms of various types as well as the number of (materialized) triples, entities and relations in these datasets.

4.1.1 Query and Answers Sampling. We use the same type of queries (corresponding to directed acyclic graphs with entities as the source nodes, also known as anchors) as in prior work [36] (see Figure 4).

We assume that each input KG is complete (i.e. \( G^\text{all} \)) and then partition it into \( G_{\text{valid}} \) for validation and \( G_{\text{train}} \) for training by discarding 10% of edges at each step; this yields \( G_{\text{train}} \subseteq G_{\text{valid}} \subseteq G \). We then create several training sets of queries according to our ontology-aware data sampling strategies from Section 3.1:

- **plain**: the training queries are randomly sampled from \( G_{\text{train}} \), and we take their plain answers, i.e. over \( G_{\text{train}} \).
- **gen**: queries in plain augmented with their ontology-based generalizations\(^4\); answers are certain, i.e., over \( O^\text{all}(G_{\text{train}}) \).
- **spec**: queries from gen augmented with specializations;
- **onto**: queries from Section 3.1, with randomly chosen percentage of valid entities as anchors; all answers are certain.

Specializations and generalizations non-compliant with the shapes from Figure 4 are discarded. Note that for NELL, the plain data is exactly the one from [37]. We observe that the number of 1p queries obtained for gen and spec settings are identical. This is probably because the set of 1p queries in plain covers all edges in the training KG. For the LUBM dataset, we have created the training and testing sets from scratch, and the 1p queries in plain do not contain the entire training KG. The onto set of queries leverages the proposed ontology-driven technique, given that the ontology covers all relations and concepts in the KG and describes how they interact, i.e., the ontology axioms support all the constructed queries, and we chose 50% of valid entities as anchors. As there are too many queries to chose from, due to the large number of relations, we had to select a smaller number of valid entities as anchors, namely 20-30%. This explains the smaller number of 1p queries. Moreover, the NELL ontology does not contain interesting axioms that can be leveraged by ontology-driven query sampling technique, thus to obtain onto we had to rely on the patterns from the data alone.

We generate three different test sets for verifying the ability of the query answering model to perform inductive reasoning, deductive reasoning and their combination. More formally,

- **Inductive case (I)**. Is the model able to predict missing answers to queries over the complete, but not observable KG \( G^\text{all} \)? (accounts for the standard test case)
- **Deductive case (D)**. Is the model able to predict answers that can be inferred from the known triples in \( G_{\text{train}} \) using ontology?
- **Inductive + Deductive case (I-D)**. Is the model able to predict missing answers inferred from the complete but not observable KG \( G^\text{all} \) using axioms in \( O \)?

For test case I, respectively I-D, test queries are randomly sampled over \( G_{\text{train}} \), respectively \( O^\text{all}(G_{\text{train}}) \), while for D they are randomly sampled over \( O^\text{all}(G_{\text{train}}) \) s.t. they cannot be trivially answered over \( G_{\text{train}} \), and unseen during training. In each test case the validation queries are generated similarly but over \( G_{\text{valid}} \).

The size of each training/testing set, and the number of queries per shape for each of the considered cases are presented in Table 4.

For each test and validation query, we measure the accuracy based on hard (certain) answers, i.e., those that cannot be trivially answered over \( G_{\text{train}} \) (or \( G_{\text{valid}} \) for test queries) and require prediction of missing edges and/or application of ontology axioms (see Definition 2.3 and 3.2).

\(^3\)This is similar to random sampling over \( O^\text{all}(G_{\text{train}}) \) but unlike the deductive closure, our procedure is guaranteed to terminate. We used the rewriting depth of up to 10.

\(^4\)https://github.com/medinaandresel/eamqa
Table 3: The total number of axioms $|O|$ and of each type, the size of the input KG $|\mathcal{G}|$, the number of entities $|E|$, the number of relations $|R|$, and the number of materialized triples $|\mathcal{G}^{\text{tr}}|$.

| Dataset | $|O|$ | $A \subseteq A'$ | $p \subseteq s$ | $p^+ \subseteq s$ | $\exists p \subseteq A$ | $\exists p^+ \subseteq A$ | $|\mathcal{G}|$ | $|E|$ | $|R|$ | $|\mathcal{G}^{\text{tr}}|$ |
|---------|------|----------------|----------------|----------------|----------------|----------------|-------|------|------|--------|
| LUBM    | 68   | 13             | 5              | 28             | 11             | 11             | 284K  | 55684| 28   | 565K   |
| NELL    | 307  | –              | 92             | 215            | –              | –              | 284K  | 63361| 400  | 497K   |

Table 4: Number of test and train queries of each shape in each of the settings.

| Dataset | Train/Test | Query Shape |
|---------|------------|-------------|
|         | 1p         | 2p          | 3p          | 2i          | 3i          | ip        | pi        | 2u       | up       |
| LUBM    | Plain      | 110000      | 110000      | 110000      | 110000      | –         | –         | –         | –         |
|         | Gen        | 117124      | 136731      | 150653      | 181234      | 208710    | –         | –         | –         |
|         | Spe        | 117780      | 154851      | 173678      | 271532      | 230085    | –         | –         | –         |
|         | Onto       | 116893      | 166159      | 333406      | 212718      | 491707    | –         | –         | –         |
|         | I          | 8000        | 8000        | 8000        | 8000        | 8000      | 8000      | 8000      | 8000      |
|         | D          | 1241        | 4701        | 6472        | 3829        | 4746      | 7393      | 7557      | 4986      |
|         | I+D        | 8000        | 8000        | 8000        | 8000        | 8000      | 8000      | 8000      | 7986      |

We consider $Q2B$, $O2B$, $CQD$ and $CQD^{\text{ASR}}$ trained in each described setting: i.e., $M_x$, where $M \in \{Q2B, O2B, CQD, CQD^{\text{ASR}}\}$ and $x \in \{\text{plain}, \text{gen}, \text{spec}, \text{onto}\}$; $Q2B^{\text{plain}}$ and $CQD^{\text{plain}}$ are taken as baselines. $Q2B$ and $O2B$ are trained on five query shapes that require projection and intersection, while $CQD$ and $CQD^{\text{ASR}}$ are trained on atomic queries. We have configured both $Q2B$ and $O2B$ systems as follows: The size of the embedding dimension was set to 1000, and the models were trained for $15 \times 10^5$ steps using Adam optimizer with an initial learning rate of $10^{-4}$ and the batch size of 512. The rest of the parameters were set in the same way as in [36].

For $CQD$, we used the code from [2] with ComplEx-N3 [29] employed as the base model. The embedding size was set to 1000, and the regularisation weight was selected based on the validation set by searching in $\{10^{-3}, 5 \times 10^{-3}, \ldots, 10^{-1}\}$. For LUBM, the regularisation weight was set to 0.1 in the gen, spe, and onto settings, and to 0.01 in the plain setting. For NELL, the regularization weight was set to 0.005 in the plain setting, to 0.001 in the gen and spe settings, and to 0.05 in the onto setting. For $CQD^{\text{ASR}}$, we have additionally used a regularisation weight of the following values: $10^{-2}$, $10^{-3}$ and $10^{-4}$. The batch size of the adversarial examples was set to 32.

We evaluated the models periodically and report the test results of the models with the best performance on the validation dataset. The performance of each trained model is measured using standard metric HITS@K for $K=3$ (HITS@3), indicating the frequency that the correct answer is ranked among the top-3 results.

4.3 Evaluation Results

4.3.1 Inductive-Deductive case. First, we present the detailed results for the most challenging setting (I+D) for LUBM and NELL in Table 5 and Figure 5.

Based on the average accuracy of the models across all query shapes reported in Table 5, the improvements of the proposed ontology-aware adaptations of $Q2B$ and $CQD$ are evident. For LUBM $O2B$ trained using the onto strategy improves the $Q2B$ baseline by almost 50%, while in case of $CQD$, 54% enhancement is achieved. For NELL similar behaviour is observed with the improvement of almost 20% for $Q2B$, and 25% for $CQD$. Next, we discuss the impact of each of the proposed techniques for the E-OMQA task.

The first observation is that incorporating the ontology in the training data is crucial as both $Q2B$ and $CQD$ trained in settings gen and onto yield significant improvements over the baselines. Additional incorporation of specializations (setting spec) does not seem
to have a major impact though (see Figure 5). On LUBM, for all models, the advantage of the ontology-driven query sampling (i.e., onto setting) is significant compared to gen setting. Remarkably, for LUBM CQDonto, resp. CQDonto, trained on less data than CQDgen resp. CQDgen results in higher accuracy. This shows that random query sampling is not adequate for E-OMQA. The ontology for NELL is not expressive enough, thus, when generating training queries in the onto setting (see Table 3 for statistics) we proceeded in a bottom-up fashion as follows: We randomly labeled query shapes which produce answers, and constructed their generalizations as before; thus the settings gen and onto are similar, but onto has significantly less atomic queries, which explains why CQDgen outperforms CQDonto on NELL.

On average ontology-aware models (i.e., O2B and CQDASR) significantly outperform their baselines (i.e., O2B and CQD, resp.) for the majority of training data sampling strategies. This trend is more prominent for atom-based models on the complex LUBM ontology, and for query-based ones on NELL, which is less expressive. In Table 6 we present the detailed results for the query rewriting over pre-trained embeddings. In order to evaluate this procedure, for each hard answer we take the best (i.e., minimum) ranking among all rankings generated by all queries in the rewriting of each test query. In other words, we take the minimal distance between the embedding of a and all rewritings of q. Note that, for measuring the performance we use the pre-trained models Q2Bplain, and CQDplain obtained after 450K training steps. Due to the reliance on particular query shapes of the respective models, the complete rewriting for each query is not guaranteed. In Table 6, we present the results for this method compared to the plain setting. Minor improvements of only at most 10% are observed.

4.3.2 Inductive case. Next, we present the average HITS@3 metric for the inductive I test case (see Figure 6). For I ontology-injection methods do not yield any improvement, which is expected, since ontologies cannot handle missing edges and facts in a KG that are not inferred from the data using ontological reasoning.
The problem of ontology-mediated query answering (OMQA) has recently received a lot of attention (see [35] for overview), but available methods, e.g., [11, 15], only focus on logic-based deductive reasoning, but do not aim at predicting missing links in knowledge graphs using machine learning approaches.

### 5 RELATED WORK

The task of answering queries that involve multiple atoms using embedding techniques has recently received a lot of attention (see [35] for overview). The existing proposals can be divided into query-based (e.g., [9, 25, 30, 36, 37, 40, 44, 45]) and atom-based (e.g., [2, 3]) methods.

The works [14] and [6] study the relation between the problem of conjunctive QA in the embedding space and over probabilistic databases. Our work is different from the above proposals in that along with the data we also rely on ontologies.

Table 6 shows the results for the test cases I and D.

| Models | Test Case D | Test Case I+D |
|--------|-------------|---------------|
|        | LUBM | NELL | LUBM | NELL |
| Q2B <sub>plain</sub> | 0.189 | 0.617 | 0.193 | 0.539 |
| Q2B <sub>rew</sub> <sub>plain</sub> | 0.248 | 0.683 | 0.261 | 0.639 |
| Gain | +0.059 | +0.066 | +0.065 | +0.1 |
| CQD <sub>plain</sub> | 0.225 | 0.656 | 0.231 | 0.621 |
| CQD <sub>rew</sub> <sub>plain</sub> | 0.228 | 0.743 | 0.249 | 0.708 |
| Gain | +0.003 | +0.087 | +0.018 | +0.087 |

#### 4.3.3 Deductive case. For the test case D, when the ontology is simple (e.g., NELL), CQD and Q2B are able to more or less learn to apply the ontology rules when they are explicitly injected in the training set. Moreover, the results on NELL in the plain setting show that rule enforcement is also competitive for deductive reasoning. For query-based models the best performance is achieved by combining ontology-driven sampling with rule enforcement, while for atom-based models, the inclusion of generalizations seems to be already sufficient.

For expressive ontologies, such as LUBM, the ontology-driven query sampling is crucial for optimal performance. For query-based models the best results are achieved when the ontology-driven query sampling is combined with rule enforcement, while for atom-based models rule enforcement does not seem to be necessary.

### 6 CONCLUSION

We have presented methods for embedding-based ontology mediated query answering (E-OMQA) that operate in the embedding space to enable simultaneous inductive and deductive reasoning over the incomplete data. Experiments show that embedding-based methods for query answering applied naively or combined with query rewriting techniques are not effective. At the same time, our ontology-aware extensions of the popular models for embedding-based QA and the proposed ontology-driven training strategies yield promising results on the novel benchmarks that we introduce for the considered task.

For future work we plan to study the effectiveness of our methods for embedding-based ontology mediated query answering for other more complex query forms [37, 40, 42], e.g., queries with negation, as well as evaluate the proposed approach for the cases when the ontology is more expressive.

### ACKNOWLEDGMENTS

Pasquale was partially funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 875160, ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence) EPSRC (grant no. EP/W002876/1), an industry grant from Cisco, and a donation from Accenture LLP; and is grateful to NVIDIA GPU donations. This work was partially funded by the European project SMARTEDGE (grant number 101092908).
