Knowledge-based Transfer Learning Explanation

Jiaoyan Chen
Department of Computer Science
University of Oxford, UK

Freddy Lécué
INRIA, France
Accenture Labs, Ireland

Jeff Z. Pan
Department of Computer Science
University of Aberdeen, UK

Ian Horrocks
Department of Computer Science
University of Oxford, UK

Huajun Chen
College of Computer Science, Zhejiang University, China
Alibaba-Zhejiang University Frontier Technology Research Center

Abstract

Machine learning explanation can significantly boost machine learning's application in decision making, but the usability of current methods is limited in human-centric explanation, especially for transfer learning, an important machine learning branch that aims at utilizing knowledge from one learning domain (i.e., a pair of dataset and prediction task) to enhance prediction model training in another learning domain. In this paper, we propose an ontology-based approach for human-centric explanation of transfer learning. Three kinds of knowledge-based explanatory evidence, with different granularities, including general factors, particular narrators and core contexts are first proposed and then inferred with both local ontologies and external knowledge bases. The evaluation with US flight data and DBpedia has presented their confidence and availability in explaining the transferability of feature representation in flight departure delay forecasting.

Introduction

Prediction with machine learning (ML) has been increasingly applied in a variety of fields to assist humans in decision making. ML explanation work such as interpreting the prediction model or justifying the prediction result can significantly increase decision makers' confidence on the prediction and boost its application (Biran and Cotton 2017), especially in making critical decisions like cancer diagnosis when people need to understand how and why the prediction is made.

Most ML explanation studies such as designing inherently interpretable models (Wu *et al.* 2018) and approximating a "black box" model with multiple "white box" models (Ribeiro *et al.* 2016) aim at users with ML expertise. The explanations lack background and common sense knowledge, thus are too hard to be understood by non-ML-experts, those common users without ML expertise such as doctors. There are only a limited number of human-centric ML explanation studies. Most of them adopt some corpus (e.g., Wikipedia articles (Biran and McKeown 2017)) or Link Data (Tiddi *et al.* 2014) to generate text to describe model components (e.g., effective ML features) or justify the prediction results (e.g., data clusters). They adopt background knowledge but are limited by expressivity, which in turn restricts the reasoning and inhibits rich explanations.

On the other hand, transfer learning which utilizes samples, features (i.e., representations of original data) or models of one learning domain (i.e., a pair of dataset and prediction task) to enhance prediction model training in another learning domain (Pan and Yang 2010) has been widely applied, especially in dealing with critical challenges like lacking training data. Its explanation aims at justifying the good or bad performance of the prediction model trained by a specific transfer learning algorithm with a specific parameter setting. Current work on transfer learning explanation such as analyzing the impact of a feature's specificity and generality on its transferability (Yosinski *et al.* 2014) aim at ML experts and represent the insights in a machine understandable way. It's hard for common users to understand why transfer from one learning domain contributes to an accurate prediction model (i.e., positive transfer) while transfer from another learning domain contributes to an inaccurate prediction model (i.e., negative transfer).

In this paper, we propose an ontology-based knowledge representation and reasoning framework for human-centric transfer learning explanation. It first models a learning domain in transfer learning, including the dataset and the prediction task, with expressive OWL (Web Ontology Language (Bechhofer 2009)) ontologies, and then complements the learning domain with the prediction task-related common sense knowledge using an efficient individual matching and external knowledge importing algorithm. The framework further uses a correlative reasoning algorithm to infer three kinds of explanatory evidence (i.e., general factors, particular narrators and core contexts) to explain a positive feature or a negative transfer from one learning domain to another. Some technical challenges such as feature transferability measurement and core context (entailment subset) searching are overcome.

As far as we know this is the first work to study human-centric transfer learning explanation and ontology-based ML explanation. It achieves confident and rich human understandable evidence for explaining both positive and negative transfers in predicting US flight delay, where the feature learned by a Convolutional Neural Network (CNN) is transferred. For example, we find that transferring between flights

Accepted by International Conference on Principles of Knowledge Representation and Reasoning, 2018.

carried a big airline company is an evidence to explain positive transfers, while transferring between flights departing from the airport of SFO is an evidence to explain negative transfers (cf. Example 4 for more examples).

The remainder of the paper is organized as follows. The next section introduces the ML background with ontologies, and defines the problem of transfer learning explanation. Then we present the ontology-based framework and report the evaluation. In the final two sections, we review the related work and conclude the paper.

Background and Problem Definition

We use Description Logics (DL) based ontologies written in the W3C OWL 2 standard[1], in particular the $\mathcal{EL}^{++}$ (Baader *et al.* 2005) fragment of the OWL 2 EL profile. In this section, we first introduce $\mathcal{EL}^{++}$ based ontology, then revisit the notions of learning domain, supervised learning and transfer learning with ontologies, and eventually define the problem of transfer learning explanation.

The $\mathcal{EL}^{++}$ Description Logic

Given a signature $\Sigma = (\mathcal{N}_C, \mathcal{N}_R, \mathcal{N}_I)$, consisting of 3 disjoint sets of atomic concepts $\mathcal{N}_C$, atomic roles $\mathcal{N}_R$, and individuals $\mathcal{N}_I$, the top concept $\top$, the bottom concept $\bot$, an atomic concept A, an individual a, an atomic role r, $\mathcal{EL}^{++}$ concept expressions C and D can be composed with the following constructs:

$$\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$$

An $\mathcal{EL}^{++}$ ontology is composed of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. The TBox $\mathcal{T}$ is a set of concept and role axioms. $\mathcal{EL}^{++}$ supports General Concept Inclusion axioms (GCIs e.g., $C \sqsubseteq D$), Role Inclusion axioms (RIs e.g., $r_1 \sqsubseteq r_2$, $r_1 \circ r_2 \sqsubseteq s$), where C, D are concept expressions, r_1, r_2, s are atomic roles. The ABox $\mathcal{A}$ is a set of class assertion axioms e.g., $C(a)$, role assertion axioms e.g., $r(a, b)$, individual equality and inequality axioms e.g., $a = b$, $a \neq b$, where C is a concept expression, r is an atomic roles and a, b are individuals. Entailment reasoning in $\mathcal{EL}^{++}$ is PTime-Complete.

Learning with Ontology

In order to support ML, we need to specify the input and output. To this end, we introduce the notions of learning sample ontology (LSO) and target entailment. We use an LSO as an input for ML methods, and the truth of a target entailment as an output. A learning domain in ML equals to a combination of an LSO set (i.e., a dataset) and a target entailment (i.e., a prediction task).

Definition 1. ***(Learning Sample Ontology (LSO))***
A learning sample ontology $\mathcal{O} = \langle\langle \mathcal{T}, \mathcal{A}\rangle, S\rangle$ is an ontology $\langle \mathcal{T}, \mathcal{A}\rangle$ annotated by property-value pairs S. Its ABox entailment closure $\{g | \mathcal{T} \cup \mathcal{A} \models g\}$ is denoted as $\mathcal{G}(\mathcal{O})$, where g represents an entailment.

The annotation S in Definition 1 acts as key dimensions to uniquely identify an input sample of ML methods. When the context is clear, sometimes we also use LSO to refer to its ontology $\langle \mathcal{T}, \mathcal{A}\rangle$. By entailment reasoning with both TBox and ABox axioms, we get a complete set of ABox entailments i.e., $\mathcal{G}(\mathcal{O})$ for modeling the input sample.

Example 1. ***(An LSO on Departure Flights)***
Figure 1 displays some axiom examples of an LSO annotated by property-value pairs $S := \{dat : 01/01/2018, car : DL, ori : LAX, des : JFK\}$. The LSO corresponds to one ML input sample that is related to a flight departure from Los Angeles International Airport (LAX) to John F. Kennedy International Airport (JFK) on 01/01/2018, carried by Delta Air Lines (DL). The examples include some TBox axioms (1)-(6) and ABox axioms (7)-(24), with some atomic concepts (e.g., $Airport$), defined concepts (e.g., $DelayedDep$), individuals(e.g. LAX) and roles (e.g., $hasCarrier$).

$Dep \sqcap \exists hasDelMin.\{Pos\} \sqsubseteq DelayedDep$	(1)		
$Dep \sqcap \exists hasDelMin.\{Neg\} \sqsubseteq OnTimeDep$	(2)		
$hasCarrier \circ hasCarHub \sqsubseteq hasDepHub$	(3)		
$hasNebApt \circ hasRecDep \sqsubseteq hasRecNebDep$	(4)		
$Dep \sqcap \exists hasOri.\{CA\} \sqcap \exists hasDes.\{CA\} \sqsubseteq \exists withIn.\{CA\}$	(5)		
$\exists withIn.\top \sqsubseteq InStateDep$	(6)		
$Airport(LAX)$	(7)	$locatedIn(LAX, CA)$	(8)
$Carrier(DL)$	(9)	$Departure(d)$	(10)
$hasDelMin(d, Pos)$	(11)	$hasWea(d, wea)$	(12)
$hasOri(d, LAX)$	(13)	$hasCarrier(d, DL)$	(14)
$Airport(JFK)$	(15)	$hasDes(d, JFK)$	(16)
$LAX = ori$	(17)	$DL = car$	(18)
$hasRecDep(d, d_1)$	(19)	$hasCarrier(d_1, MU)$	(20)
$hasRecDep(d, d_2)$	(21)	$hasCarrier(d_2, AA)$	(22)
$DelayedDep(d)$	(23)	$HeavySnow(wea)$	(24)

Figure 1: Ontology Examples of An LSO on Departure Flights

Definition 2 revisits the concept of *learning domain* in ML and defines *target entailment*. A learning domain is also annotated by property-value pairs (cf. Definition 3).

Definition 2. ***(Learning Domain and Target Entailment)***
A learning domain $\mathcal{D} = \langle \mathbb{O}, g^t \rangle$ consists of a set of LSOs $\mathbb{O}$ that share the same TBox $\mathcal{T}$, and a target entailment g^t whose truth in an LSO is to be predicted.

Definition 3. ***(Learning Domain Annotation)***
The annotation property-value pairs of the learning domain $\mathcal{D}$ in Definition 2 are defined as $S(\mathcal{D}) = (\cap_{\langle\langle \mathcal{T}, \mathcal{A}\rangle, S\rangle \in \mathbb{O}} S) \cup \{t\_e : g^t\}$.

Example 2. ***(A Learning Domain on Departure Flights)***
Now we consider a learning domain $\mathcal{D}_0 = \langle \mathbb{O}_0, g_0^t \rangle$, where the target entailment g_0^t is $DelayedDep(d)$ and $\mathbb{O}_0$ contains the LSO in Example 1, as well as many similar LSOs with the same (carrier, origin airport and destination airport), but different dates. The domain annotation $S(\mathcal{D}_0)$ is $\{car : DL, ori : LAX, des : JFK, t\_e : DelayedDep(d)\}$.

With the above definitions, Definition 4 revisits the notion of *within domain supervised learning* task (Mohri *et al.* 2012). It reduces a prediction problem to a supervised learning problem with steps of learning and predicting.

Definition 4. ***(Within Domain Supervised Learning)***
Given a learning domain $\mathcal{D} = \langle \mathbb{O}, g^t \rangle$, whose LSOs $\mathbb{O}$ are divided into two disjoint sets $\mathbb{O}'$ and $\mathbb{O}''$, a supervised learning task within $\mathcal{D}$, denoted by $\mathcal{L} = \langle \mathcal{D}, \mathbb{O}', \mathbb{O}'', \mathcal{M} \rangle$, is a task of learning a model $\mathcal{M}$ with $\mathbb{O}'$ and g^t to predict the truth

[1] https://www.w3.org/TR/owl2-overview/

of g^t in each $\mathcal{O}$ in $\mathbb{O}''$. Here, $\mathbb{O}'$ is called a training LSO set, while $\mathbb{O}''$ is called a testing LSO set.

In the training LSO set, we assume the ABox axioms (observations) are complete. The target entailment is true if it is entailed by an LSO, and false otherwise. In the testing LSO set, we assume the ABox axioms are incomplete (some observations are missing or we predict before they are observed), and the truth of the target entailment is predicted by the model.

Example 3. ***(Within Domain Supervised Learning)*** *Given the learning domain $\mathcal{D}_0$ in Example 2, we train the model $\mathcal{M}$ with LSOs before or at date t_0 (i.e., $\mathbb{O}'$), and apply the model to predict the truth of $DelayedDep(d)$ in each LSO after t_0 (i.e., $\mathbb{O}''$).*

To feed an ML algorithm, each LSO in both training LSO set and testing LSO set is encoded into a real value vector, denoted as $\boldsymbol{x}$. We first transform it into (i) a value vector $\boldsymbol{v}$ with data properties by concatenating their numeric values and (ii) an entailment vector $\boldsymbol{e}$ by BOE embedding with a set of entailments entailed by domain LSOs (cf. Definition 5). Then we concatenate $\boldsymbol{e}$ and $\boldsymbol{v}$ as the real value vector: $\boldsymbol{x} = [\boldsymbol{e}, \boldsymbol{v}]$. The target entailment g^t in each training LSO is transformed into a binary existence variable, denoted as y. It's assigned 1 if $g^t \in \mathcal{G}(\mathcal{O})$, and 0 otherwise.

Definition 5. ***(Bag of Entailments)*** *Given an entailment set $\{g_i | i = 1, ..., n\}$, Bag of Entailment (BOE) is an ontology encoding method that represents an LSO (with ABox entailment closure $\mathcal{G}$) to a vector $\boldsymbol{e} = (e_1, e_2, ..., e_n)$ where $e_i = 1$ if $g_i \in \mathcal{G}$ and 0 otherwise.*

Explaining Transfer Learning

Definition 6 revisits the concepts of *transfer learning*, *positive transfer* and *negative transfer* (Pan and Yang 2010).

Definition 6. ***(Transfer Learning)*** *Given two learning domains $\mathcal{D}_\alpha = \langle \mathbb{O}_\alpha, g^t_\alpha \rangle$ and $\mathcal{D}_\beta = \langle \mathbb{O}_\beta, g^t_\beta \rangle$, where the LSOs of domain $\mathcal{D}_\beta$ are divided into two disjoint sets $\mathbb{O}'_\beta$ and $\mathbb{O}''_\beta$, transfer learning from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$, denoted by $\mathcal{F}_{\alpha\to\beta}$ is a task of learning a model $\mathcal{M}_{\alpha\to\beta}$ from $\mathbb{O}_\alpha$, g^t_α, $\mathbb{O}'_\beta$ and g^t_β to predict the truth of g^t_β in each LSO in $\mathbb{O}''_\beta$. $\mathcal{F}_{\alpha\to\beta}$ is defined as a positive transfer if $\mathcal{M}_{\alpha\to\beta}$ outperforms $\mathcal{M}_\beta$ which is learned within domain $\mathcal{D}_\beta$ according to Definition 4, and a negative transfer otherwise.*

In Definition 6, $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ are called *source domain* and *target domain*, respectively, while $\mathcal{O}'_\beta$ and $\mathcal{O}''_\beta$ are the training LSO set and testing LSO set in the target domain. In comparison with supervised learning within the target domain (Definition 4), transfer learning has the same settings except that it learns the model from not only the training LSO set but also the LSO set from a source domain.

The effect of $\mathcal{F}_{\alpha\to\beta}$, namely the *transferability*, can be measured by comparing the performance of $\mathcal{M}_{\alpha\to\beta}$ and $\mathcal{M}_\beta$. In Definition 6, positive transfer and negative are defined for qualitative description. We also define a metric called *Feature Transferability Index* (FTI) (to be specified in (25) on page 4) for quantitative measurement. The higher the FTI metric, the higher the transferability.

Explaining transfer learning aims at justifying the good or bad performance of a model trained by a transfer learning algorithm with a specific parameter setting. It describes human understandable factors that influence the transferability. Definition 7 defines this problem as *correlation-based transfer explanation*.

Definition 7. ***(Correlation-based Transfer Explanation)*** *Given a transfer $\mathcal{F}_{\alpha\to\beta}$ in Definition 6, correlation-based transfer explanation is a task of inferring a set of influential factors $\mathbb{X}$ such that each $\mathcal{X}$ in $\mathbb{X}$ is correlated with FTI and the absolute value of the correlation coefficient $\|\gamma(FTI, \mathcal{X})\| \geq \epsilon$, where ϵ is a parameter in $[0, 1]$. An influential factor $\mathcal{X}$ is called an explanatory evidence.*

In Definition 7, the confidence of an explanatory evidence for transferability explanation is proportional to its absolute coefficient value $\|\gamma(FTI, \mathcal{X})\|$. We abuse the notion and note it as $\|\gamma(\mathcal{X})\|$ in the remainder of the paper.

Three kinds of explanatory evidence (cf. Example 4) are proposed, including

- *General factors* which are those statistic indexes of ABox entailments that quantify the overall knowledge variance and invariance from the source learning domain to the target learning domain,
- *Particular narrators* which are those particular ABox entailments that have a high impact on the transferability,
- *Core contexts* which are those ABox entailment combinations that have a high impact on the transferability.

Example 4. ***(Explanatory Evidence)*** *In US flight departure delay prediction, we consider three learning domains whose target entailments are all $DelayedDep(d)$: $\mathcal{D}_{(DL,ORD,LAX)}$ for Delta Airlines from ORD to LAX, $\mathcal{D}_{(B6,LAX,JFK)}$ for JetBlue from LAX to JFK and $\mathcal{D}_{(AA,ORD,SFO)}$ for American Airlines from ORD to SFO, as well as a negative transfer from $\mathcal{D}_{(DL,ORD,LAX)}$ to $\mathcal{D}_{(B6,LAX,JFK)}$ and a positive transfer from domain $\mathcal{D}_{(DL,ORD,LAX)}$ to $\mathcal{D}_{(AA,ORD,SFO)}$. We explain the two transfers with (i) general factors e.g., the percentage of shared entailments between $\mathcal{D}_{(DL,ORD,LAX)}$ and $\mathcal{D}_{(AA,ORD,SFO)}$ ($\mathcal{D}_{(DL,ORD,LAX)}$ and $\mathcal{D}_{(B6,LAX,JFK)}$) is high (low), (ii) particular narrators e.g., entailment "$locatedIn(ori, East)$" plays a positive role in the transfer from $\mathcal{D}_{(DL,ORD,LAX)}$ to $\mathcal{D}_{(AA,ORD,SFO)}$, and (iii) core contexts, e.g., the entailment set composed of "$hasOri(dep, ORD)$" and "$locatedIn(des, CA)$" has a high impact on the positive transfer from $\mathcal{D}_{(DL,ORD,LAX)}$ to $\mathcal{D}_{(AA,ORD,SFO)}$.*

Method

Transferability Measurement

In transfer learning, the training of the model for the target learning domain (i.e., $\mathcal{M}_{\alpha\to\beta}$ in Definition 6) either directly integrates samples of the source learning domain or indirectly utilizes model parameters learned in the learning source domain (Pan and Yang 2010; Weiss *et al.* 2016). In this study, we adopt the latter. Features learned by Convolutional Neural Networks (CNNs) i.e., learned parameters of hidden network layers, are transferred from the source

learning domain to the target. As transferability measurement only depends on the performance of the model trained within the target learning domain (i.e, $\mathcal{M}_\beta$) and the model trained with transfer (i.e., $\mathcal{M}_{\alpha\to\beta}$), how transfer learning is implemented does not impact the generality of our explanation framework.

A CNN is stacked by convolutional (Conv) layers which learn the feature with data locality, and fully connected (FC) layers which learn the non-linear relationship between the input and output. As shown in Figure 2, we first train a CNN model within the source learning domain (i.e., $\mathcal{M}_\alpha$) using its LSO set (i.e., $\mathbb{O}_\alpha$) and target entailment (i.e., g^t_α), then transfer the model's feature (i.e., parameters of the Conv layers) to a CNN model in the target learning domain (i.e., $\mathcal{M}_{\alpha\to\beta}$) which has the same network architecture. We eventually fine-tune the parameters of the model in the target learning domain with its training LSO set (i.e., $\mathbb{O}'_\beta$) and target entailment (i.e., g^t_β). It's called *hard transfer* if we only fine-tune the FC layers and *soft transfer* if we fine-tunes both FC layers and Conv layers.

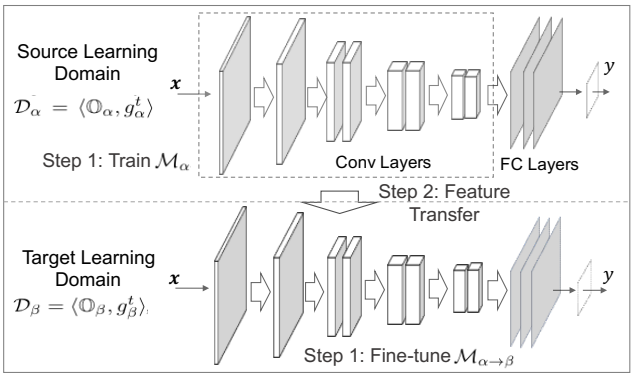

Figure 2: Transfer Learning with Convolutional Neural Networks.

A feature's transferability depends on its *specificity* to the learning domain where it is trained and its *generality* (Yosinski *et al.* 2014). By comparing the performance of the model trained within the target learning domain with the models trained with hard transfer and soft transfer, the specificity and generality can be measured (cf. Definition 8).

Definition 8. ***(Feature Specificity/Generality Index)***
Given transfer learning in Definition 6, let $\mathcal{M}^h_{\alpha\to\beta}$ and $\mathcal{M}^s_{\alpha\to\beta}$ be the models trained with hard transfer and soft transfer respectively, Feature Specificity Index (FSI) is defined as the performance drop of $\mathcal{M}^h_{\alpha\to\beta}$ over $\mathcal{M}_\beta$, while Feature Generality Index (FGI) is defined as the performance gain of $\mathcal{M}^s_{\alpha\to\beta}$ over $\mathcal{M}_\beta$, where the performance of all models are measured with the testing LSO set of the target learning domain.

We propose a comprehensive index called *Feature Transferability Index (FTI)* to measure the feature's transferability. It is proportional to its generality and inversely proportional to its specificity (cf. Property 1). The more the hard transfer or the soft transfer benefits the model in the target learning domain, the higher transferability the feature has. With FSI and FGI, we calculate FTI as follows:

$$FTI = \frac{\omega_1 \cdot FGI - \omega_2 \cdot FSI}{\omega_1 + \omega_2} \tag{25}$$

where ω_1 and ω_2 are weight parameters in $[0, 1]$ and are both set to 1 in the remainder of the paper if not specified. We denote the FTI value from learning domain $\mathcal{D}_\alpha$ to learning domain $\mathcal{D}_\beta$ as $f_t(\mathcal{D}_\alpha, \mathcal{D}_\beta)$.

Property 1. ***(Relation between FTI and FSI/FGI)***
FTI is proportional to FGI and inversely proportional to FSI.

External Knowledge

Each learning domain is extended with *external knowledge* from existent knowledge bases (KBs), such as DBPedia (Auer *et al.* 2007), for richer common sense knowledge about the prediction application. The extension includes two steps: (i) root individual selection, and (ii) external knowledge matching and importing.

Root Entailment and Root Individual. We define those entailments that play an import role in predicting the truth of the target entailment as *Root Entailments*, denoted as $\mathcal{G}^R$. Root entailments include *Frequent Entailments* (cf. Definition 9) and *Effective Entailments* (cf. Definition 10). Those individuals that are involved in at least one root entailment are defined as *Root Individuals*, denoted as $\mathcal{I}^R$.

Definition 9. ***(Frequent Entailment)***
Given a learning domain $\mathcal{D} = \langle \mathbb{O}, g^t \rangle$ and its local entailment closure $\mathcal{G}(\mathbb{O}) = \cup_{\mathcal{O}\in\mathbb{O}}\mathcal{G}(\mathcal{O})$, $g \in \mathcal{G}(\mathbb{O})$ is a frequent entailment if $|\{\mathcal{O} \in \mathbb{O} | g \in \mathcal{G}(\mathcal{O})\}|/|\mathbb{O}| \geq \sigma$, where $|\cdot|$ calculates the set cardinality, σ is a parameter in $[0, 1]$.

Definition 10. ***(Effective Entailment)***
In Definition 9, a κ-element entailment subset $\mathcal{G}_\kappa \subseteq \mathcal{G}(\mathbb{O})$ is a set of effective entailments if $r_e + r_i \geq \tau$, where $r_e = |\{\mathcal{O} \in \mathbb{O} | \mathcal{G}_\kappa \cup \{g^t\} \subseteq \mathcal{G}(\mathcal{O})\}|/|\mathbb{O}|$ and $r_i = |\{\mathcal{O} \in \mathbb{O} | (\mathcal{G}_\kappa \cup \{g^t\}) \cap \mathcal{G}(\mathcal{O}) = \emptyset\}|/|\mathbb{O}|$, $\kappa \geq 1$ and $\tau \in [0, 1]$ are parameters.

In Definition 9, we calculate the rate of LSOs that contain an entailment g. The entailments that appear in a large part of LSOs are frequent entailments. In Definition 10, r_e (r_i) represents the rate of LSOs where the entailment subset $\mathcal{G}_\kappa$ and the target entailment g^t co-exist (co-inexist). The higher $r_e + r_i$, the more effective $\mathcal{G}_\kappa$ in predicting the truth of g^t, according to the theory and practice of correlation-based ML feature selection (Hall 1999).

Example 5. ***(Root Individual Selection)***
We consider the learning domain $\mathcal{D}_{(B6,LAX,JFK)}$ in Example 4, "$hasOri(d, LAX)$" is a frequent entailment as it appears in all the LSOs of the domain, while the entailment subset composed of "$hasRecDep(d, d_2)$", "$DelayedDep(d_2)$" and "$hasCarrier(d_2, AA)$" are effective as they co-exist or co-inexist with the target entailment "$DelayedDep(d)$" in a large part of LSOs. The individuals d, d_2, LAX and AA that are involved in the above entailments are root individuals.

Knowledge Importing Workflow. For each learning domain, we match each of its root individuals with an entity of an external KB, and then import the concepts and roles of the entity. The workflow is shown in Algorithm 1.

In Line 5 and 6, we use root individuals to match external entities by name matching. Using root individuals signifi-

cantly saves computation and storage, as the non-root individuals take a large part but usually lead to external axioms that contribute little to the richness of explanatory evidence (cf. Evaluation for more details).

From Line 7 to Line 13, we (i) extract values of concepts and roles (i.e., $\mathcal{K}$) of each matched entity, (ii) transform them into ABox axioms with the terminologies defined in TBox, (iii) check their consistence with local LSOs and the constraint axioms (i.e., $\mathcal{C}$), and (iv) add them into the set of external axioms (i.e., $\mathcal{A}_e$) (cf. Example 6). The consistency checking is to avoid errors caused by name matching (cf. Example 7). Line 14 eventually computes the entailment closure of the learning domain, denoted as $\mathcal{G}(\mathcal{D})$, together with the local LSOs and external axioms.

Example 6. ***(External Axioms)***
The individual LAX in Example 5 is matched with the entity $Los\_Angeles\_International\_Airport$ in DBPedia. The triples related to the concept ("$rdf:type$") and roles, e.g., "$geo:lat$" and "$geo:long$", "$dbo:hubAirport$", of the entity are extracted and transformed into external ABox axioms e.g., "$hasLat(LAX, 38.94)$".

Example 7. ***(Consistency Checking)***
In Example 6, the individual LAX can be matched with the entity $L.A.InternationalAirport$ (a song by Leanne Scott) in DBPedia by name matching. The constraint axiom "$Location \sqcap Song \sqsubseteq \bot$", the local axioms "$Airport(LAX)$" and "$Airport \sqsubseteq Location$", and the external axiom "$Song(LAX)$" suggest that the entity matching is incorrect.

The constraint axioms, which may contain concept expressions that are more expressive than DL $\mathcal{EL}^{++}$ for the requirement of a specific prediction application, are an extension of the TBox of the learning domain, but are detached from the TBox to avoid increasing reasoning complexity in other steps.

Algorithm 1: `ExternalAxiomsImport`$\langle \mathcal{D}, \mathcal{I}^R, \mathcal{B}, \mathcal{C} \rangle$

1 **Input**: (i) A learning domain $\mathcal{D} = \langle g^t, \mathbb{O} \rangle$ with TBox $\mathcal{T}$, (ii) Root individuals $\mathcal{I}^R$, (iii) An external KB $\mathcal{B}$, (iv) Constraint axioms $\mathcal{T}_c$
2 **Result**: $\mathcal{G}(\mathcal{D})$: Entailment closure of the learning domain
3 **begin**
4 $\mathcal{A}_e := \emptyset$; % *Init. of the external axiom set*
5 **foreach** *root individual* $i \in \mathcal{I}^R$ **do**
6 $\mathcal{N}^i \leftarrow (\mathcal{B}, i)$ % *Match external entity by name.*
7 **foreach** *entity* $e \in \mathcal{N}^i$ **do**
8 $\mathcal{V} \leftarrow (\mathcal{B}, e)$ % *Extract concpepts and roles*
9 $\mathcal{K} \leftarrow (\mathcal{V}, \mathcal{T})$ %*Transform to external axioms*
10 % *Consistency checking*
11 **if** $\mathcal{O} \cup \mathcal{T}_c \cup \mathcal{K} \not\models \bot$ *for* $\forall \mathcal{O} \in \mathbb{O}$ **then**
12 $\mathcal{A}_e := \mathcal{A}_e \cup \mathcal{K}$
13 **break** % *Adopt the first matched entity*
14 $\mathcal{G}(\mathcal{D}) \leftarrow \cup_{\mathcal{O} \in \mathbb{O}} \mathcal{G}(\mathcal{O} \cup \mathcal{A}_e)$ % *Entailment reasoning*
15 **return** $\mathcal{G}(\mathcal{D})$;

Correlative Reasoning

We propose a method called correlative reasoning for calculating the explanatory evidence (i.e., general factors, particular narrators and core contexts) with entailment closures of the learning domains and the FTIs between learning domains. It is composed of two steps: evidence embedding and correlation analysis.

Evidence Embedding. It represents an explanatory evidence by a real value, without losing the evidence's semantics in analyzing the feature transferability. Given an evidence $\mathcal{X}$, we denote its embedding as $f_e(\mathcal{X})$.

General factors are statistic indexes that measure the overall difference and similarity between two learning domains. Definition 11 defines the embedding approach for three general factors: d^{new}, d^{obs} and d^{inv}, which are directed domain change rates from the source learning domain to the target, with *new*, *obsolete* and *invariant* entailments respectively.

Definition 11. ***(Entailment-based Domain Change Rates)***
Given source learning domain $\mathcal{D}_\alpha$ and target learning domain $\mathcal{D}_\beta$ in transfer learning (Definition 6), the entailment-based domain change rates from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$ are defined as:

$$\begin{cases} d^{new} = \frac{|\{g|g \in \mathcal{G}(\mathcal{D}_\beta), g \notin \mathcal{G}(\mathcal{D}_\alpha)\}|}{|\mathcal{G}(\mathcal{D}_\beta)|} \\ d^{obs} = \frac{|\{g|g \in \mathcal{G}(\mathcal{D}_\alpha), g \notin \mathcal{G}(\mathcal{D}_\beta)\}|}{|\mathcal{G}(\mathcal{D}_\alpha)|} \\ d^{inv} = \frac{|\{g|g \in \mathcal{G}(\mathcal{D}_\alpha), g \in \mathcal{G}(\mathcal{D}_\beta)\}|}{|\mathcal{G}(\mathcal{D}_\alpha) \cup \mathcal{G}(\mathcal{D}_\beta)|} \end{cases} \quad (26)$$

where the operation $|\cdot|$ calculates set cardinality.

Example 8. ***(Entailment-based Domain Change Rates)***
In the transfer from domain $\mathcal{D}_{(DL,ORD,LAX)}$ (entailment closure size: 25180*) to domain $\mathcal{D}_{(B6,LAX,JFK)}$ (entailment closure size:* 13412*) in Example 4, the sizes of new, obsolete and invariant entailments are* 11419*,* 23187 *and* 1193*. Thus the domain change rates d^{new}, d^{obs} and d^{inv} are calculated as $\frac{11419}{13412}$, $\frac{23189}{25180}$ and $\frac{1193}{13412+25180}$ respectively.*

A particular narrator is one single entailment that (i) is shared by the source and target learning domains, and (ii) has positive or negative impact on the feature's transferability. Core context is an extension of particular narrator from one single entailment to an entailment set (combination). To simplify the representation, we regard a particular narrator as a one-element entailment set, and use evidence note $\mathcal{X}$ to denote the entailment set involved in a particular narrator or core context.

Definition 12 defines the embedding approach for particular narrators and core contexts. It transforms a specific entailment or an entailment set into a binary variable called DEC with the entailments' co-existence in the source and target learning domains considered.

Definition 12. ***(Directed Entailment Co-existence)***
Given source learning domain $\mathcal{D}_\alpha$ and target learning domain $\mathcal{D}_\beta$ in transfer learning (Definition 6), Directed Entailment Co-existence (DEC) of an entailment set $\mathcal{G} \subseteq \mathcal{G}(\mathcal{D}_\alpha)$ is 1 *if $\mathcal{G} \subseteq \mathcal{G}(\mathcal{D}_\beta)$ and* 0 *otherwise. When $|\mathcal{G}| = 1$, it calculates the DEC of a single entailment.*

Example 9. ***(Directed Entailment Co-existence)***
In the transfer from learning domain $\mathcal{D}_{(DL,ORD,LAX)}$ to learning domain $\mathcal{D}_{(AA,ORD,SFO)}$ in Example 4, the particular narrator of "$hasOri(d, ORD)$" is embedded into 1*, while the core context composed of "$hasOri(d, ORD)$" and "$hasCarrier(d, DL)$" is embedded into* 0*.*

Different from BOE embedding in Definition 5, which transforms entailments of an LSO into one vector as ML input, evidence embedding transforms the entailment-based change from one learning domain to another into a real value for transferability analysis.

Correlation Analysis. We analyze the correlation between a given explanatory evidence $\mathcal{X}$ and its impact on the transferability of a feature with a set of learning domains $\mathbb{D}$, as shown in Algorithm 2. Line 6 traverses each pair of source and target learning domains, which correspond to one transfer. For particular narrators and core contexts, Line 7 and 8 skip the transfers whose source learning domains fail to entail all the entailments that are involved the evidence. Line 10 and 13 calculate the embedding of the evidence and the FTI value of the transfer, respectively. Line 14 calculates the Pearson Correlation Coefficient (Lee Rodgers and Nicewander 1988) and its p-value in a t-test with non-correlation hypothesis. The algorithm eventually returns the correlation coefficient (cf. Definition 7) and its p-value.

An evidence $\mathcal{X}$ is a valid explanatory evidence for the feature transferability if (i) its absolute value of correlation coefficient $\|\gamma(\mathcal{X})\| \geq \epsilon$ (cf. Definition 7) and (ii) the correlation analysis is significant (i.e., $\rho(\mathcal{X}) \leq 0.05$).

In correlative reasoning, $m(m-1)/2$ times FTI calculation are totally needed, where m is the size of the given domain set $\mathbb{D}$. Meanwhile, correlative reasoning costs $m(m-1)/2$ times evidence embedding calculation and one time correlation analysis calculation for each evidence. On the other hand, to compute a complete set of explanatory evidence, we need to traverse all the candidate evidence. The number of general factors is a constant, while the numbers of particular narrators and core contexts are n and $2^n - 1$ respectively, where n is the size of the entailment closure of the given learning domains (i.e., $\cup_{\mathcal{D}\in\mathbb{D}}\mathcal{G}(\mathcal{D})$). Directly searching all the core contexts is impractical; thus we need some optimized methods of searching for core context .

Algorithm 2: `CorrelativeReason`$\langle\mathbb{D}, \mathcal{X}\rangle$

1 **Input**: (i) A set of learning domains $\mathbb{D} = \{\mathcal{D}_k | k = 1, ..., m\}$ and (ii) an explanatory evidence $\mathcal{X}$
2 **Result**: Correlation coefficient $\gamma(\mathcal{X})$ and p-value $\rho(\mathcal{X})$
3 **begin**
4 $\boldsymbol{v}_e := \emptyset$ % *Init. of evidence value array*
5 $\boldsymbol{v}_f := \emptyset$ % *Init. of FTI value array*
6 **foreach** $\mathcal{D}_{k_1} \in \mathbb{D}$, $\mathcal{D}_{k_2} \in \mathbb{D}$ *such that* $\mathcal{D}_{k_1} \neq \mathcal{D}_{k_2}$ **do**
7 **if** *($\mathcal{X}$ is a particular narrator or a core context) and ($\mathcal{X} \not\subseteq \mathcal{G}(\mathcal{D}_{k_1})$)* **then**
8 **Continue**
9 % *Cal. evidence embedding*
10 $f_e(\mathcal{X}) \xleftarrow{Def.11, Def.12} (\mathcal{X}, \mathcal{G}(\mathcal{D}_{k_1}), \mathcal{G}(\mathcal{D}_{k_2}))$
11 $\boldsymbol{v}_e := [\boldsymbol{v}_e, f_e(\mathcal{X})]$
12 % *Cal. FTI for the transfer from $\mathcal{D}_{k_1}$ to $\mathcal{D}_{k_2}$*
13 $f_t(\mathcal{D}_{k_1}, \mathcal{D}_{k_2}) \xleftarrow{(25), Def.8} (\mathcal{D}_{k_1}, \mathcal{D}_{k_2})$
14 $\boldsymbol{v}_f := [\boldsymbol{v}_f, f_t(\mathcal{D}_{k_1}, \mathcal{D}_{k_2})]$
15 % *Pearson correlation analysis and t-test*
16 $\gamma(\mathcal{X}), \rho(\mathcal{X}) = corr(\boldsymbol{v}_e, \boldsymbol{v}_f)$
17 **return** $\gamma(\mathcal{X})$,$\rho(\mathcal{X})$;

Optimized Core Context Searching

A core context is composed of a subset of entailments of the given learning domain set (i.e., $\mathcal{X} \subseteq \cup_{\mathcal{D}\in\mathbb{D}}\mathcal{G}(\mathcal{D})$). Algorithm 3 presents our core context searching algorithm. It starts by traversing core contexts composed of two entailments (cf. Line 5 to 7), and then traverses core contexts with higher dimension by adding an entailment to the current core context (cf. Line 14 to 17). It adopts two approaches, `EarlyStop` and `FastExtend` to accelerate the search.

Algorithm 3: `CoreContextSearch`$\langle\mathbb{D}, \mathcal{X}\rangle$

1 **Input**: (i) A learning domain set $\mathbb{D}$, (ii) A candidate core context evidence $\mathcal{X}$
2 **Result**: Records of (evidence, coefficient, p-value)
3 **begin**
4 $\mathcal{G}_0 := \cup_{\mathcal{D}\in\mathbb{D}}\mathcal{G}(\mathcal{D})$ % *Cal. entailment closure*
5 **if** $\mathcal{X} = \emptyset$ **then**
6 % *Traverse core contexts with two entailments*
7 **foreach** $g_1 \in \mathcal{G}_0$, $g_2 \in \mathcal{G}_0$ *such that* $g_1 \neq g_2$ **do**
8 `CoreContextSearch`$(\mathbb{D}, \{g_1, g_2\})$
9 **else**
10 $\gamma(\mathcal{X}), \rho(\mathcal{X}) \leftarrow$ `CorrelativeReason`$(\mathbb{D}, \mathcal{X})$
11 `print` $\mathcal{X}, \gamma(\mathcal{X}), \rho(\mathcal{X})$
12 **if !** `EarlyStop` $(\mathbb{D}, \mathcal{X})$ **then**
13 % *Traverse core contexts with one more ent.*
14 **foreach** $g \in \mathcal{G}_0$ *such that* $g \notin \mathcal{X}$ **do**
15 **if** `FastExtend` $(\mathbb{D}, \mathcal{X}, g)$ **then**
16 `print` $\mathcal{X} \cup \{g\}, \gamma(\mathcal{X}), \rho(\mathcal{X})$
17 **else**
18 `CoreContextSearch` $(\mathbb{D}, \mathcal{X} \cup \{g\})$

Early Stop. The function `EarlyStop` returns true if adding more entailments to a core context evidence will not lead to any valid core contexts, and false otherwise. According to Algorithm 2 and the principle of t-test, enough *Evidence Domains* (cf. Definition 13) are needed for significant correlation analysis of an evidence, while Property 2 shows that when a core context evidence is extended by an entailment, the number of its evidence domains decreases. Thus when the correlation analysis of the current evidence is insignificant (i.e., $\rho(\mathcal{X}) > 0.05$), we stop extending this evidence with more entailments.

Definition 13. ***(Evidence Domains)***
Given a core context or particular narrator evidence $\mathcal{X}$ and a learning domain set $\mathbb{D}$, the evidence domains of $\mathcal{X}$, denoted as $\mathbb{D}(\mathcal{X})$, are defined as $\{\mathcal{D} \in \mathbb{D} | \mathcal{X} \subseteq \mathcal{G}(\mathcal{D})\}$.

Property 2. ***(Monotonicity of Evidence Domains)***
In Definition 13, for all the entailment g in $\cup_{\mathcal{D}\in\mathbb{D}}\mathcal{G}(\mathcal{D})$ and $g \notin \mathcal{X}$, we have $\mathbb{D}(\mathcal{X} \cup \{g\}) \subseteq \mathbb{D}(\mathcal{X})$.

Fast Extend. The function `FastExtend` returns true if an entailment is a *Synchronized Entailment* (cf. Definition 14) of another entailment in the current core context evidence, and false otherwise. According to Lemma 1, the new core context evidence with the synchronized entailment added

has the same impact on the feature transferability as the original one. `FastExtend` enables us to directly extend a core context evidence and avoid the calculation of evidence embedding and correlation analysis.

Definition 14. ***(Synchronized Entailments)*** *Given a learning domain set $\mathbb{D}$, two entailments g_1 and g_2 are synchronized, denoted by $g_1 \overset{\mathbb{D}}{=} g_2$, if for all the learning domain $\mathcal{D}$ in $\mathbb{D}$, $\{g_1, g_2\} \subseteq \mathcal{G}(\mathbb{D})$ or $\{g_1, g_2\} \cap \mathcal{G}(\mathbb{D}) = \emptyset$.*

Example 10. ***(Synchronized Entailments)*** *In our departure flights example, "$locatedIn(LAX, LA)$" and "$serveCity(LAX, LA)$" are synchronized entailments, w.r.t. the 92 learning domains used in our evaluation.*

Lemma 1. ***(Synchronized Evidence Extension)*** *In Definition 14, given a core context or particular narrator evidence $\mathcal{X}$, for all the entailment g in $\cup_{\mathcal{D}\in\mathbb{D}}\mathcal{G}(\mathcal{D})$ and $g \notin \mathcal{X}$, the new core context evidence $\mathcal{X}' := \mathcal{X} \cup \{g\}$ has the same correlation analysis result as $\mathcal{X}$, i.e., $\gamma(\mathcal{X}') = \gamma(\mathcal{X})$ and $\rho(\mathcal{X}') = \rho(\mathcal{X})$, if there is an entailment g_0 in $\mathcal{X}$ such that $g \overset{\mathbb{D}}{=} g_0$.*

Proof. $g \overset{\mathbb{D}}{=} g_0$ implies $\mathbb{D}(\{g\}) = \mathbb{D}(\{g_0\})$ (Definition 13 and 14), while $\{g_0\} \subseteq \mathcal{X}$ implies $\mathbb{D}(\mathcal{X}) \subseteq \mathbb{D}(\{g_0\})$ (Property 2), thus $\mathbb{D}(\mathcal{X}) \subseteq \mathbb{D}(\{g\})$; $\mathcal{X}' = \{g\} \cup \mathcal{X}$ implies $\mathbb{D}(\mathcal{X}') = \mathbb{D}(\mathcal{X}) \cap \mathbb{D}(\{g\})$ (Definition 13); thus $\mathbb{D}(\mathcal{X}') = \mathbb{D}(\mathcal{X})$; thus $\mathcal{X}$ and $\mathcal{X}'$ have the same FTI values ($\boldsymbol{v}_f$) in Algorithm 2. Meanwhile, $g_0 \in \mathcal{X}$ and $g \overset{\mathbb{D}}{=} g_0$ imply $f_e(\mathcal{X}) = f_e(\mathcal{X}')$ in Algorithm 2 (Definition 12 and 14); Thus $\mathbb{D}(\mathcal{X}') = \mathbb{D}(\mathcal{X})$ implies $\mathcal{X}$ and $\mathcal{X}'$ have the same evidence embedding ($\boldsymbol{v}_e$) in Algorithm 2. $\mathcal{X}$ and $\mathcal{X}'$ have the same $\boldsymbol{v}_e$ and $\boldsymbol{v}_f$ imply $\gamma(\mathcal{X}') = \gamma(\mathcal{X})$ and $\rho(\mathcal{X}') = \rho(\mathcal{X})$. □

Synchronized entailments are common especially when a large number of external axioms are imported. The time complexity of computing all the synchronized entailment pairs is $O(^{n(n-1)}/_{2})$, where n is the size of $\cup_{\mathcal{D}\in\mathbb{D}}\mathcal{G}(\mathcal{D})$. Meanwhile, with the transitivity property of synchronized entailment (cf. Property 3), we can merge two sets of synchronized entailments if an entailment of one set is synchronized with an entailment of another set, thus quickly calculating clusters of synchronized entailment.

Property 3. ***(Transitivity of Synchronized Entailment)*** *In Definition 14, $(g_1 \overset{\mathbb{D}}{=} g_2) \wedge (g_2 \overset{\mathbb{D}}{=} g_3) \rightarrow (g_1 \overset{\mathbb{D}}{=} g_3)$.*

Heuristics can also be developed to approximately search the core contexts. For example, in extending a core context, we can either ignore entailments that are not particular narrators, or only add entailments that are semantically close to the core context (e.g., about the same carrier). They are left in our future work.

Evaluation

Experiment Setting. In the experiment, we predict whether a flight's departure will be delayed or not, with observations of recent and surrounding flights, as well as meteorology[2]. The target entailment is set to $DelayedDep(d)$ for all the learning domains, and carrier, origin airport and destination airport are used to identify a learning domain. 92 learning domains composed of 10 airports and 11 carriers in US are adopted. One learning domain has 1,880 to 9,500 LSOs extracted from 01/01/2010 to 07/01/2017. 8372 transfers are evaluated, where FTI is measured with Area Under ROC Curve, a widely used performance metric for the prediction model. In deciding a valid evidence, the coefficient threshold ϵ in Definition 7 is set to 0.1.

We report results of (i) average number of root entailments, root individuals and external axioms per learning domain (Table 1), (ii) general factors (Figure 3), (iii) particular narrators (Figure 4) and (iv) core contexts (Figure 5), and at the same time analyze the impact of entailment reasoning and external knowledge importing on the explanation.

External Knowledge. Table 1 presents that the size of root entailments (including root concept assertion entailments and role assertion entailments), root individuals and external axioms all decreases when the parameters (σ, κ, τ) increase from P1 to P5. When (σ, κ, τ) are set to P5, only 9.3% of the individuals are selected as root individuals, reducing external axioms from around $21,000$ to 615.

On the other hand, importing less external axioms by selecting root individuals does not harm the richness of explanatory evidence. Firstly, Figure 3 (page 8, more explanatoins below) reports that setting (σ, κ, τ) to P4 (6271 external axioms imported) does not help infer more confident general factors than P5 (615 external axioms imported). In contrast, it reduces the confidence of general factors d^{new}, d^{obs} and d^{inv} by ($7.5\%, 51.3\%, 52.4\%$), when they are measured with external axioms alone. It means those additional external axioms in setting P4 bring more noise than effective information to general factors in explaining the transferability. Secondly, Figure 4 [Left] (page8) reports that the richness of particular narrators is kept from setting P4 to P5, as the total number decreases very little (e.g., positive entailments decrease from 833 to 828).

TBox Axi.: 541	Concept Ast. Ent.: 1824	Role Ast. Ent.: 4528	Individual: 1159	Ext. Axi.: $\sim$ 21 K
Parameters (σ, κ, τ)	Root Concept Ast. Ent.	Root Role Ast. Ent.	Root Individual	External Axioms
P1:(.90, 1, .40)	1105 (61%)	3805 (84%)	1103 (95%)	$\sim$ 20 K
P2:(.93, 1, .43)	990 (54%)	3459 (76%)	1080 (93%)	$\sim$ 19 K
P3:(.96, 1, .46)	540 (30%)	1816 (40%)	872 (75%)	$\sim$ 16 K
P4:(.99, 1, .49)	305 (17%)	980 (22%)	510 (44%)	6271
P5:(.99, 2, .49)	157 (8.6%)	402 (8.9%)	108 (9.3%)	615

Table 1: Average Number of Root Entailments, Root Individuals and External Axioms per Learning Domain.

General Factors. Figure 3 (Local ABox Ent. + External Axioms (P5)) presents that general factors d^{obs} and d^{new} have a significant negative impact on the feature's transferability ($\gamma(\mathcal{X}) < -0.2$). Thus we can explain a negative transfer like $\mathcal{F}_{(DL,ORD,LAX)\rightarrow(B6,LAX,JFK)}$ with explanations like *"There are a high percentage of new and obsolete entailments from domain $\mathcal{D}_{(DL,ORD,LAX)}$ to $\mathcal{D}_{(B6,LAX,JFK)}$"*. On the other hand, there is positive correlation between general factor d^{inv} and FTI ($\rho(\mathcal{X}) < 0.05$), which is opposite to d^{obs} and d^{new}, but the correlation is

[2] Codes and data: https://github.com/ChenJiaoyan/X-TL

weak ($\|\gamma(\mathcal{X})\| < 0.08$). Therefore, sharing a large percentage of entailments is not a confident evidence to explain a positive feature transfer.

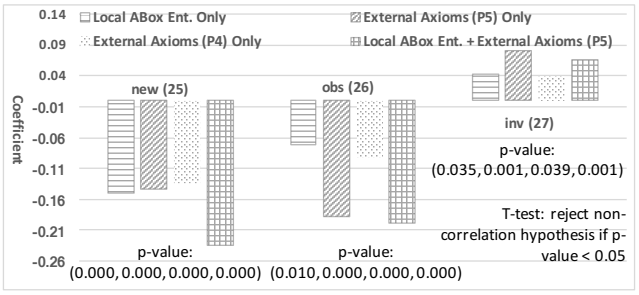

Figure 3: General Factors d^{new}, d^{obs} and d^{inv} Calculated with Different Knowledge Parts, and Parameter Settings of P4 and P5.

Entailment Narrator. Figure 4 [Left] shows that 11.3% (19.1%) of the entailments are positively (negatively) correlated with FTI in parameter setting P5. Those entailments are adopted as particular narrators for explaining a positive or negative feature transfer. According to the particular narrator examples, we can explain the positive transfer $\mathcal{F}_{(DL,ORD,LAX)\rightarrow(AA,ORD,SFO)}$ with descriptions like (i) *"the origin airport of both source and target learning domains is in the east part of US" (e2)* and (ii) *"the carriers of both source and target learning domains are public companies" (e4)*. We can explain the negative transfer $\mathcal{F}_{(DL,ORD,LAX)\rightarrow(B6,LAX,JFK)}$ with descriptions like *"the carriers of both source and target learning domains are small companies; it's hard to transfer a feature between two learning domains with small carriers" (e10)*.

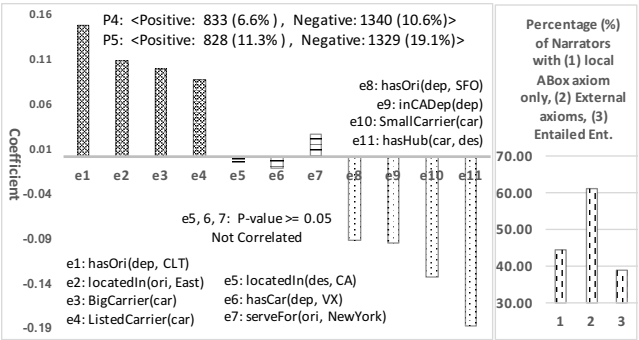

Figure 4: Examples and Statistics of Particular Narrators.

Entailment Reasoning and External Knowledge. Figure 3 (Local ABox Ent. Only vs. External Axioms (P5) vs. Local ABox Ent. + External Axioms (P5)) shows that combing local ABox entailments and external axioms for d^{obs} (d^{new}) achieves 178.9% (56.0%) and 5.9% (62.5%) higher absolute coefficient than using local ABox entailments alone and using external axioms alone respectively. This verifies external knowledge's positive impact on the confidence of general factors. Meanwhile, Figure 4 [Right] shows that 44.4% of particular narrators use local ABox axioms only, while 61.1% and 38.9% of them involve external axioms and entailed entailments respectively. This verifies the positive impact of entailment reasoning and external knowledge on the quality of particular narrators.

Core Context. Figure 5 [Left] and [Middle] present that the core contexts composed of 2 to 4 entailments have much higher absolute coefficient than general factors and particular narrators. For example, the average coefficient of the top $k\%$ most positively correlated core contexts ranges from (0.18, 0.28, 0.33) to (0.35, 0.59, 0.78) when the dimension C is $(2, 3, 4)$. They are more confident in explaining the transferability. For example, with the core context composed of $locatedIn(des, CA)$, $ListCar(car)$ and $BigCar(car)$, whose coefficient is 0.35, we can explain the positive transfer $\mathcal{F}_{(DL,ORD,LAX)\rightarrow(AA,ORD,SFO)}$ more confidently by *"The carrier of both source and target learning domain belongs to big and list airline companies, and their destination airports are both located in California"*.

Figure 5 [Right] reports that (19.9%, 11.6%, 4.8%) of all the (2, 3, 4)-dimension entailment subsets have significant correlation analysis with FTI (i.e., $\rho(\mathcal{X}) < 0.05$), while (13.6%, 1.8%, 0.2%) are valid core contexts (i.e., $\rho(\mathcal{X}) < 0.05$ and $\|\gamma(\mathcal{X})\| \geq 0.1$). On one hand, as the dimension increases, the percentage of valid core contexts significantly decreases. On the other hand, the fact that a very large part of the entailment subsets have insignificant correlation analysis verifies that `EearlyStop` in core context searching (Algorithm 3) is effective. For example, when the dimension of the current core context is 4, it avoids 95.2% of the traversing for core contexts with higher dimension.

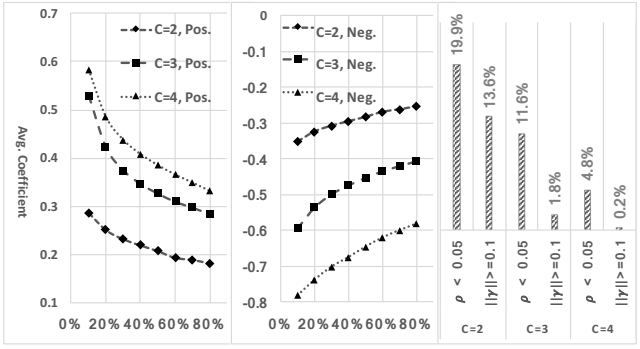

Figure 5: [Left] ([Middle]) Average Correlation Coefficient of Top $k\%$ Most Positively (Negatively) Correlated Core Contexts, [Right] Percentage of Valid Core Contexts (i.e., $\rho(\mathcal{X}) < 0.05$ and $\|\gamma(\mathcal{X})\| \geq 0.1$) , with Dimension $C = 2, 3, 4$.

Discussion and Lessons. The evaluation presents the explanatory evidence's confidence and percentage (i.e., the probability of being available as evidence). For confidence, we have core contexts $>$ general factors d^{new} and d^{obs} $>$ entailment narrators $>$ general factor d^{inv}, while for percentage (of being available), we have general factors (100%) $>$ entailment narrators (30.4% in average) $>$ core contexts (15.6% in average). General factors can successfully explain any negative transfers, but fail to provide confident evidence for positive transfers. Core contexts, especially those with high dimensions, have very high confidence but the percentage decreases quickly as the dimension grows. For both high confidence and availability, all the three kinds of evidence need to be used together.

The evaluation also analyzes the positive impact of our techniques, including (i) root individual selection, which

saves much computation but keeps high quality evidence, (ii) entailment reasoning and external axiom importing, which enrich the evidence and improve the percentage, and (iii) early stop strategy in core context searching, which significantly reduces unnecessary searching.

The explanations lead to insights of feature transfer for users without ML expertise, and in turn allow them to further improve a transfer learning approach with more optimized settings. For a specific target domain, the explanations can answer the question of what to transfer by comparing the evidence of different source learning domains. Meanwhile, we can infer explanatory evidence for different features such as different Conv layers of a CNN architecture. Thus for a specific pair of source and target learning domains, we can answer the question of when to transfer by selecting a feature that maximizes the positive evidence.

Related Work

ML explanation has been studied for years, mainly including model interpretation (i.e., understanding how decisions are made) and prediction justification (i.e., justifying why a particular decision is good) (Biran and Cotton 2017). In this section, we first overview the above two aspects, and then introduce the state-of-the-art in transfer learning explanation and human-centric explanation.

Model Interpretation. Some ML models are inherently interpretable. One type is sparse linear models such as Supersparse Linear Integer Models (Ustun and Rudin 2016). These models' variable coefficients can present how much each variable contributes to the decision. Another type is rule-based models such as sparse Decision Tree (Wu *et al.* 2018) and Bayesian Rule Lists (Letham *et al.* 2015). They can explain the decision inference procedure with internal probabilities and rules.

To interpret black-box models, visualization techniques have been applied. For example, (Zeiler and Fergus 2014) visualized the hidden layer output of a CNN to understand the feature representation of data. For another example, (Jakulin *et al.* 2005) proposed the algorithm `nomograms` to visualize Support Vector Machines. Recent advances in model interpretation include (i) attention mechanism for weighting the importance of different input parts (Qin *et al.* 2017), (ii) reasoning-based consistent sample selection for stream learning (Chen *et al.* 2017), (iii) data distribution summarization with prototypes and criticisms (Kim *et al.* 2016), etc.

Prediction Justification. A specific prediction can be explained by evaluating the effect of each meaningful input variable (Biran and McKeown 2017). It can be directly calculated in an interpretable model or estimated with input isolation strategies such as omitting a subset of input (Robnik-Šikonja and Kononenko 2008; Martens and Provost 2014). For complex and black-box models, (Baehrens *et al.* 2010; Ribeiro *et al.* 2016) proposed to approximate them by multiple linear models which have interpretable data representations and local fidelity.

Generating description text is another approach to justify predictions. Recent advances include (i) caption generation for visual decisions such as image classification (Hendricks *et al.* 2016), (ii) text description of effective ML features (Biran and McKeown 2017), etc.

Transfer Learning Explanation. Current studies on transfer learning explanation mainly lie in transferability analysis. Problems like when and what to transfer have been investigated in both theory and practice (Pan and Yang 2010; Weiss *et al.* 2016). Recent advances include (i) experimental quantification of the generality (transferable) and specificity (untransferable) of CNN feature (Yosinski *et al.* 2014), (ii) theoretic justification of the relation between feature structure similarity and transferability (Liu *et al.* 2017), etc.

These attempts of transferability analysis definitely benefit ML experts, but will fail to explain the learned model or justify the prediction to common people. The understanding to transferability is encoded in a machine understandable way (e.g., loss function) to enhance learning. The explanations are neither represented in a human understandable format nor enriched with common sense knowledge. To the best of our knowledge, there are currently no studies for human-centric transfer learning explanation.

Human-centric ML Explanation. Human-centric ML explanation aims at interpreting learned models or justifying predictions with background or common sense knowledge in a human understandable way (Biran and McKeown 2017). Most of the current studies are based on corpuses. (Hendricks *et al.* 2016) utilized external corpuses to generate captions to explain image classification decisions, while (Biran and McKeown 2017) used Wikipedia articles to describe effective features of a ML model. Few studies utilize semantic data in human-centric explanation. (Tiddi *et al.* 2014) proposed a framework to traverse Linked Data and use graph path commonalities to explain data clusters.

The current studies incorporate external knowledge, but ignore expressive knowledge e.g., ontology and their reasoning capability. It lacks a general knowledge representation and reasoning framework to utilize local ontologies and external knowledge bases for human-centric ML explanation. This work bridges the above gap and is among the first to study human-centric transfer learning explanation.

Conclusion and Outlook

In this study, we address the problem of human-centric transfer learning explanation. Our ontology-based framework exploits the reasoning capability and external knowledge bases like DBpedia to infer different kinds of human understandable explanatory evidence, including general factors, particular narrators and core contexts. It allows users without ML expertise to have a good insight of positive / negative transfers, and to answer the questions of what and when to transfer for more optimized transfer learning settings. The quality of explanatory evidence, including the confidence and availability, and the effect of our methods, are evaluated with US flight departure delay prediction, where features learned by CNNs are transferred. In the future, we will exploit more efficient core context search algorithms and the impact of semantic expressivity and compilation (Pan and Thomas 2007; Pan *et al.* 2016), with further experiments and evaluations.

Acknowledgments

The work was partially funded by the research center SIRIUS, the EPSRC project DBOnto, the EU Marie Curie K-Drive project (286348) and NSFC61673338.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.

Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 364–369, 2005.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Sean Bechhofer. OWL: Web ontology language. In *Encyclopedia of database systems*, pages 2008–2009. Springer, 2009.

Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*, page 8, 2017.

Or Biran and Kathleen McKeown. Human-centric justification of machine learning predictions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.

Jiaoyan Chen, Freddy Lécué, Jeff Z Pan, and Huajun Chen. Learning from ontology streams with semantic concept drift. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 957–963. AAAI Press, 2017.

Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

Aleks Jakulin, Martin Možina, Janez Demšar, Ivan Bratko, and Blaž Zupan. Nomograms for visualizing support vector machines. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 108–117. ACM, 2005.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288, 2016.

Joseph Lee Rodgers and W Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

Tongliang Liu, Qiang Yang, and Dacheng Tao. Understanding how feature structure transfers in transfer learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2365–2371, 2017.

David Martens and Foster Provost. Explaining data-driven document classifications. *MIS Quarterly*, 38(1):73–99, 2014.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

Jeff Z. Pan and Edward Thomas. Approximating OWL-DL Ontologies. In *the Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1434–1439, 2007.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Jeff Z. Pan, Yuan Ren, and Yuting Zhao. Tractable approximate deduction for OWL. *Arteficial Intelligence*, pages 95–155, 2016.

Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2627–2633, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.

Ilaria Tiddi, Mathieu dAquin, and Enrico Motta. Dedalo: Looking for clusters explanations in a labyrinth of linked data. In *European Semantic Web Conference*, pages 333–348. Springer, 2014.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

Mike Wu, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *AAAI*, 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.