Augmenting Transfer Learning with Semantic Reasoning

Freddy Lécué1,2, Jiaoyan Chen3, Jeff Z. Pan4,5 and Huajun Chen6,7
1CortAIx Thales, Montreal, Canada
2Inria, Sophia Antipolis, France
3Department of Computer Science, University of Oxford, UK
4Department of Computer Science, The University of Aberdeen, UK
5Edinburgh Research Centre, Huawei, UK
6College of Computer Science, Zhejiang University, China
7ZJU-Alibaba Joint Lab on Knowledge Engine, China

Abstract

Transfer learning aims at building robust prediction models by transferring knowledge gained from one problem to another. In the semantic Web, learning tasks are enhanced with semantic representations. We exploit their semantics to augment transfer learning by dealing with when to transfer with semantic measurements and what to transfer with semantic embeddings. We further present a general framework that integrates the above measurements and embeddings with existing transfer learning algorithms for higher performance. It has demonstrated to be robust in two real-world applications: bus delay forecasting and air quality forecasting.

1 Introduction

Transfer learning [Pan and Yang, 2010] aims at solving the problem of lacking training data by utilizing data from other related learning domains, each of which is referred to as a pair of dataset and prediction task. Transfer Learning plays a critical role in real-world applications of ML as (labelled) data is usually not large enough to train accurate and robust models. Most approaches focus on similarity in raw data distribution with techniques such as dynamic weighting of instances [Dai et al., 2007] and model parameters sharing [Benavides-Prado et al., 2017] (cf. Related Work).

Despite of a large spectrum of techniques [Weiss et al., 2016] in transfer learning, it remains challenging to assess a priori which domain and data set to elaborate from [Dai et al., 2009]. To deal with such challenges, [Choi et al., 2016] integrated expert feedback as semantic representation on domain similarity for knowledge transfer while [Lee et al., 2017] evaluated the graph-based representations of source and target domains. Both studies encode semantics but are limited by the expressivity, which restricts domains interpretability and inhibits a good understanding of transferability. There are also efforts on Markov Logic Networks (MLN) based transfer learning, by using first order [Mihalkova et al., 2007; Mihalkova and Mooney, 2009] or second order [Davis and Domingos, 2009; Van Haaren et al., 2015] rules as declarative prediction models. However, these efforts still cannot answer questions like: What ensures a positive domain transfer? Would learning a model from road traffic congestion in London be the best for predicting congestion in Paris? Or would an air quality model transfer better?

In this paper, we propose to encode the semantics of learning tasks and domains with OWL ontologies and provide a robust foundation to study transferability between source and target learning domains. From knowledge materialization [Nickel et al., 2016], feature selection [Vicient et al., 2013], predictive reasoning [Lécué and Pan, 2015], stream learning [Chen et al., 2017] to transfer learning explanation [Chen et al., 2018], all are examples of inference tasks where the semantics of data representation are exploited for deriving a priori knowledge from pre-established statements in ML tasks.

We introduce a framework to augment transfer learning by semantics and its reasoning capability, as shown in Figure 1. It deals with (i) when to transfer by suitable transferability measurements (i.e., variability of semantic learning task and consistent transferability knowledge), (ii) what to transfer by embedding the semantics of learning domains and tasks with transferability vector, consistent vector and variability vector. In addition to expose semantics that drives transfer, a transfer boosting algorithm is developed to integrate the embeddings with existing transfer learning approaches. Our approach achieves high performance for multiple transfer learning tasks in air quality and bus delay forecasting.

![Figure 1: Ontology-based Transfer Learning Augmentation.](image-url)

2 Background

Our work uses OWL ontologies underpinned by Description Logic (DL) $\mathcal{EL}^{++}$ [Baader et al., 2005][Bechhofer et al., 2004] to model the semantics of learning domains and tasks.
2.1 Description Logics $\mathcal{EL}^{++}$ and Ontology

A signature $\Sigma$, noted $\langle \mathcal{N}_C; \mathcal{N}_R; \mathcal{N}_I \rangle$ consists of 3 disjoint sets of (i) atomic concepts $\mathcal{N}_C$, (ii) atomic roles $\mathcal{N}_R$, and (iii) individuals $\mathcal{N}_I$. Given a signature, the top concept $\top$, the bottom concept $\bot$, an atomic concept $A$, an individual $a$, an atomic role expression $r$, $\mathcal{EL}^{++}$ concept expressions $C$ and $D$ in $C$ can be composed with the following constructs:

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

A DL ontology is composed of a TBox $T$ and an ABox $A$. $T$ is a set of concept, role axioms. $\mathcal{EL}^{++}$ supports General Concept Inclusion axioms (GCIs e.g., $C \sqsubseteq D$), Role Inclusion axioms (RIs e.g., $r \sqsubseteq s$), $A$ is a set of class assertion axioms, e.g., $C(a)$, role assertion axioms, e.g., $r(a, b)$, individual in/equality axioms e.g., $a \neq b, a = b$. Given an input ontology $T \cup A$, we consider the closure of atomic ABox entailments (or simply entailment closure, denoted as $G(T \cup A)$) as $\{g|T \cup A \models g\}$, where $g$ represents an atomic concept assertion $A(b)$, or an atomic role assertion entailment $r(a, b)$, involving only named concepts, named roles and named individuals. Entailment reasoning in $\mathcal{EL}^{++}$ is PTime-Complete.

Example 1. (TBox and ABox Concept Assertion Axioms)

Figure 2 presents (i) a TBox $T$ where Road (1) denotes the concept of “ways which are in a continent”, and (ii) concept assertions (8-9) with individuals $r_0$ and $r_1$ being roads.

Figure 2: Sample of an Ontology’s TBox $T$ and ABox $A$.

![Image](image_url)

2.2 Learning Domain and Task

To model the learning domain with ontology, we use Learning Sample Ontology and Target Entailment, as in [Chen et al., 2018]. A learning domain consists of an LSO set (i.e., dataset) and a target entailment set (i.e., prediction task).

Definition 1. (Learning Sample Ontology (LSO))

A learning sample ontology $O = \langle \langle T, A \rangle, S \rangle$ is an ontology $\langle T, A \rangle$ annotated by property-value pairs $S$.

The annotation $S$ acts as key dimensions to uniquely identify an input sample of ML methods. When the context is clear, we also use LSO to refer to its ontology $\langle T, A \rangle$.

Example 2. (An LSO in Context of Ireland Traffic)

Assume an LSO is annotated by property-value pairs $S := \langle$ topic: Road, road: C.Way, country: UK $\rangle$. Its TBox $T$ includes static axioms like (1); its ABox $A$ includes facts e.g., hasAvgSpeed($r_0$, Low) that are observed in C.Way in UK.

Definition 2. (Learning Domain and Target Entailment)

A learning domain $D = \langle O, \mathcal{G}^T \rangle$ consists of a set of LSOs $O$ that share the same TBox $T$, and target entailments $\mathcal{G}^T$, each of whose truth in an LSO is to be predicted. Its entailment closure, denoted as $G(O)$ is defined as $\cup_{O \in S} G(O)$.

Definition 3 revisits supervised learning within a domain. In a training LSO, a target entailment is true if it is entailed by an LSO, and false otherwise. In a testing LSO, the truth of a target entailment is to be predicted instead of being inferred.

Definition 3. (Semantic Learning Task)

Given a learning domain $D = \langle O, \mathcal{G}^T \rangle$, whose LSOs $O$ are divided into two disjoint sets $O'$ and $O''$, a semantic learning task, denoted by $T$, within $D$, is defined as: $\langle D, O', O'', f(\cdot) \rangle$ i.e., the task of identifying a function $f(\cdot)$ with $O'$ and $\mathcal{G}^T$ to predict the truth of $\mathcal{G}^T$ in each $O$ in $O''$. Here, $\mathcal{G}^T$ is called a training LSO set, while $O''$ is called a testing LSO set.

Example 3. (Semantic Learning Task)

Given a domain composed of LSOs annotated by $\langle$ topic: Road, country: UK $\rangle$ and target entailments Closed($r_0$) and Disrupted($r_0$), the LSOs are divided into a training set $O'$ and a testing set $O''$ according to the type of roads involved, the objective is to identify a function from $O'$ to predict the condition of road $r_0$, namely the truth of Closed($r_0$) and Disrupted($r_0$) in each $O''$.

2.3 Transfer Learning Across Domains

Definition 4 revisits transfer learning where $D_S$ and $D_T$ are called source domain and target domain and their entailment closures are denoted as $G_S$ and $G_T$.

Definition 4. (Transfer Learning)

Given two learning domains $D_S = \langle O_S, G_S^T \rangle$ and $D_T = \langle O_T, G_T^T \rangle$, where the LSOs of $D_T$ are divided into two disjoint sets $O_T'$ and $O_T''$, transfer learning from $D_S$ to $D_T$ is a task of learning a prediction function $f_{T\langle S \rangle}(\cdot)$ from $O_S, G_S^T, O_T', O_T''$ and $G_T^T$ to predict the truth of $G_T^T$ in each LSO in $O_T''$.

Example 4. (Transfer Learning)

Assume $D_T$ is the domain in Example 3, $D_S$ is a domain with LSOs annotated by $\langle$ topic: Road, country: IE $\rangle$, an example of transfer learning is to identify a function using all the LSOs of Dublin traffic and the training LSOs of London traffic (O_T') for predicting the traffic condition of road $r_0$ in each testing LSO of London traffic (O_T'').

We demonstrate how ontology-based descriptions can drive transfer learning from one domain to another. To this end, similarities between domains are first characterized. We adopt the variability of ABox entailments [Lécué, 2015] in Definition 5, where (10) reflects variant knowledge between two domains while (11) denotes invariant knowledge.

Definition 5. (Entailment-based Domain Variability)

Given a source learning domain $D_S$ and a target learning domain $D_T$, let $G = G_S \cup G_T$, the variability from $D_S$ to $D_T$, denoted as $\mathcal{V}(O_T, O_S)$ are ABox entailments:

$$G_{var}^{[S,T]} = \{ g \in G | g \in G_T \land g \notin G_S \}$$

$$G_{var}^{[S,IE]} = \{ g \in G | g \in G_T \land g \notin G_S \}$$

Example 5. (Entailment-based Domain Variability)

Let Figure 3, which capture the contexts in IE and UK, be ontologies of $D_S$ and $D_T$ respectively. Table 1 illustrates some variabilities of $D_S$ and $D_T$ through ABox entailments. For instance $r_1$ as a disrupted road in $D_S$ is new (variant) w.r.t. knowledge in $D_T$ and axioms (1), (9) and (12-15).
3 Transferability

We present (i) variability of semantic learning tasks, (ii) semantic transferability, as a basis for qualifying, quantifying transfer learning (i.e., when to transfer), together with (iii) indicators (i.e., what to transfer) driving transferability. They are pivotal properties, as any change in domains, their transfer function and consistency drastically impact the quality of derived models [Long et al., 2015; Chen et al., 2018].

3.1 Variability of Semantic Learning Tasks

Definition 6 extends entailment-based ontology variability (Definition 5) to capture the learning task variability, where \( (\cdot)_{O_S, O_T} \) represents using target entailments in (10) (11).

**Definition 6. (Variability of Semantic Learning Tasks)**

Let \( T_S \) and \( T_T \) be semantic learning tasks of source learning domain \( D_S \) and target learning domain \( D_T \). The variability of semantic learning tasks \( \nabla(T_S, T_T) \) is defined by (22), where \( \cdot \) refers to the cardinality of a set.

\[
\left( \frac{|G_{TS}^{S_{var}}|_{\nabla} + |G_{TS}^{S_{inv}}|_{\nabla}}{|G_{TS}^{S_{var}}|_{\nabla} + |G_{TS}^{S_{inv}}|_{\nabla}} \right) \left( \frac{|G_{TS}^{Y_T}|_{\nabla} + |G_{TS}^{Y_T}|_{\nabla}}{|G_{TS}^{Y_T}|_{\nabla} + |G_{TS}^{Y_T}|_{\nabla}} \right)
\]

The variability of semantic learning tasks (22), also represented by \( \nabla(T_S, T_T) \), captures the variability of source and target domain LSOs as well as the variability of target entailments. The higher values the stronger variability. The calculation of (22) is in worst case polynomial time w.r.t size \( O_S, O_T, Y_S, Y_T \) in \( \mathcal{E} \mathcal{L} \mathcal{C} \). Its evaluation requires (i) ABox entailment, (ii) basic set theory operations from Definition 5, both in polynomial time [Baader et al., 2005].

**Example 6. (Variability of Semantic Learning Tasks)**

The variability of learning task between \( T_S \) and \( T_T \) in Example 4 is \((2/3, 0)\) as the number of variant and invariant ABox entailments are respectively 6 and 3, and \( Y_S = Y_T \), i.e., moderate variability of domains, none for target variables.

3.2 Semantic Transferability - When to Transfer?

We define semantic transferability from a source to a target semantic learning task as the existence of knowledge that are captured as ABox entailments [Pan and Thomas, 2007] in the source and have positive effects on predictive quality of the prediction function of the target semantic learning task.

**Definition 7. (Semantic \( \varepsilon \)-Transferability)**

Let \( T_S, T_T \) be source, target semantic learning tasks with entailment closures \( G_S, G_T \). Semantic \( \varepsilon \)-transferability \( T_S \xrightarrow{\varepsilon} T_T \) occurs from \( T_S \) to \( T_T \) if \( \exists S \subseteq O_S : m(f_{T_S}(\cdot)) - m(f_{T_T}(\cdot)) > \varepsilon \) \( G_S \neq G_T \) (24)

where \( f_{T_S}(\cdot) \) is the predictive function \( f_T(\cdot) \) w.r.t. \( O_T \cup S \). \( G_S \) is the ABox closures of \( S \).

\( S \) is knowledge from \( O_S \), to be used for over-performing the predictive quality of \( f_T(\cdot) \) with a \( \varepsilon \in (0, 1] \) factor (23) while being new with respect to ABox entailments in \( G_T \) (24).

**Example 7. (Semantic \( \varepsilon \)-Transferability)**

Let \( T_S, T_T \) be semantic learning tasks in \( D_S, D_T \) in Example 4, \( S \) be ABox entailment closure of (12-15) in \( O_S \), and \( m(f_{T_S}(\cdot)) > m(f_{T_T}(\cdot)) \). Semantic \( \varepsilon \)-transferability occurs from \( T_S \) to \( T_T \) as \( \varepsilon > 0 \), satisfying condition (23), exists, and (ii) (24) is true cf. Table 1 w.r.t. \( S \). Thus, knowledge \( S \) in IE traffic context \( (D_S) \) ensures transferability from \( D_S \) to \( D_T \) for traffic prediction in UK.

ABox entailments \( S \) satisfying Definition 7 are denoted as transferable knowledge while those contradicting (23) i.e., \( m(f_{T_S}(\cdot)) - m(f_{T_T}(\cdot)) \leq \varepsilon \) are non-transferable knowledge as they deteriorate predictive quality of target function \( f_T(\cdot) \).

**Example 8. (Transferable Knowledge)**

Consider entailments in \( S \): (i) \( \text{Disrupted}(r_4) \), derived from (13) (19-21), (ii) \( \text{Clear}(r_0) \), derived from (8) (12) (17-18). As part of knowledge \( S \) positively impacting the quality of the prediction task, they are also separate \( \varepsilon \)-transferable knowledge with max \( \varepsilon \in [0, 0.7] \) (computation details omitted).

3.3 Consistent Transferable Knowledge

Transferring knowledge across domains can derive to inconsistency. Definition 8 captures knowledge ensuring transferability while maintaining consistency in the target domain.

**Definition 8. (Consistent Transferable Knowledge)**

Let \( S \) be ABox entailments \( T_S \xrightarrow{T_T} T_T \). \( S \) is consistent transferable knowledge from \( T_S \) to \( T_T \) iff \( S \cup O_T \neq \bot \).

ABox entailments \( S \) satisfying \( S \cup O_T \models \top \) are called inconsistent knowledge. They are interesting ABox entailments as they expose knowledge contradicting the target domain while maintaining transferability. Evaluating if \( S \) is consistent transferable knowledge is in worst case polynomial time in \( \mathcal{E} \mathcal{L} \mathcal{C} \) w.r.t size of \( S \) and \( O_T \).

**Example 9. (In-)Consistent Transferable Knowledge**

\( \text{Disrupted}(r_4) \) in \( S \) of Example 8 is consistent transferable knowledge in \( T_T \) as \{ \text{Disrupted}(r_4) \} \cup O_T \models \top \). On contrary \( \text{Clear}(r_0) \) and \( \text{Disrupted}(r_0) \) in \( S \), derived from (16-18) are inconsistent (7). Thus, \( \text{Clear}(r_0) \) in \( O_S \) is inconsistent transferable knowledge in \( T_T \).
4 Semantic Transfer Learning

We tackle the problem of transfer learning by computing semantic embeddings (i.e., how to transfer) for knowledge transfer, and determining a strategy to exploit the semantics of the learning tasks (Section 3) in Algorithm 1.

4.1 Semantic Embeddings - How to transfer?

The semantics of learning tasks exposes three levels of knowledge which are crucial for transfer learning: variability, transferability, consistency. They are encoded as embeddings through Definition 9, 10 and 11.

Definition 9. (Transferability Vector)

Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be all distinct ABox entailments in $\mathcal{O}_S \cup \mathcal{O}_T$. A transferability vector from $\mathcal{T}_S$ to $\mathcal{T}_T$, denoted by $t(\mathcal{G})$, is a vector of dimension $m$ such that $t_j \in [0, 1]$: $t_j = \varepsilon_j$ if $g_j$ is $\varepsilon_j$-transferable knowledge, and $t_j = 0$ otherwise, with $\varepsilon_j | \varepsilon_j^* \geq \varepsilon_j$ and $g$ is $\varepsilon_j^*$-transferable knowledge.

A transferability vector (Definition 9) is adapting the concept of feature vector [Bishop, 2006] in Machine Learning to represent the qualitative transferability from source to target of all ABox entailments. Each dimension captures the best of transferability of a particular ABox entailment.

Example 10. (Transferability Vector)

Suppose $\mathcal{G} = \{\text{Disrupted}(r_1), \text{Clear}(r_0)\}$. Transferability vector $t(\mathcal{G})$ is $(1, 0)$, cf. $\varepsilon$-transferability in Example 8.

A consistency vector (Definition 10) is computed from all entailments by evaluating their (in-)consistency, either 1 or 0, when transferred in the target semantic learning task. Feature vectors are bounded to only raw data while transferability and consistency vectors, with larger dimensions, embed transferability and consistency of data and its inferred assertions. They ensure a larger, more contextual coverage.

Definition 10. (Consistency Vector)

Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be all distinct ABox entailments in $\mathcal{O}_S \cup \mathcal{O}_T$. A consistency vector from $\mathcal{T}_S$ to $\mathcal{T}_T$, denoted by $c(\mathcal{G})$, is a vector of dimension $m$ such that $c_j \in [0, 1]$: $c_j = 1$ if $\{g_j\} \not\subseteq \mathcal{O}_T$ and $c_j = 0$ otherwise.

The variability vector (Definition 11) is used as an indicator of variability weighting between the two learning tasks. It is a value in $[0, 1]$ with an emphasis on the domain ontologies and/or label space depending on its parameterization ($\alpha, \beta$). We characterize any variability weight above $\frac{1}{2}$ as inter-domain transfer learning tasks, below $\frac{1}{2}$ as intra-domain.

Definition 11. (Variability Vector)

Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be ABox entailments in $\mathcal{O}_S \cup \mathcal{O}_T$. A variability vector $v(\mathcal{G}, \alpha, \beta)$ from $\mathcal{T}_S$ to $\mathcal{T}_T$ is a vector of dimension $m$ with $\alpha, \beta \in [0, 1]$ such that $v_{j, j} \in [0, 1]$ is:

$$v(\mathcal{G}, \alpha, \beta) = \frac{\alpha(\mathcal{N}(\mathcal{T}_S, \mathcal{T}_T)|\alpha) + \beta(\mathcal{N}(\mathcal{T}_S, \mathcal{T}_T)|\beta)}{\alpha + \beta}$$

Example 11. (Variability Vector)

Applying (25) on the variability of semantic learning tasks between $\mathcal{T}_S$ and $\mathcal{T}_T$: $(2/3, 0)$ in Example 6 results in $v(\mathcal{G}, \alpha, \beta) = 1/3$, which represents moderate variability.

4.2 Boosting for Semantic Transfer Learning

Algorithm 1 presents an extension of the transfer learning method TrAdaBoost [Dai et al., 2007] by integrating semantic embeddings. It aims at learning a predictive function $f_{T|S}(\cdot)$ (line 20) using $\langle \mathcal{T}_S, \mathcal{O}_S \rangle, \mathcal{O}_T$ for $\mathcal{T}_T$. The semantic embeddings of all entailments in $\mathcal{G}_S \cup \mathcal{G}_T$ are computed (lines 7-8). They are defined through transferability, consistency, variability effects from source to target domain. Then, their importance / weight w are iteratively adjusted (line 9) depending on the evaluation of $f^i$ (lines 13-14) when comparing estimated prediction $f^i(e_j)$ and real values $Y_T(g_i)$.

The base model (lines 11-12), which can be derived from any weak learner e.g., Logistic Regression, is built on top of all entailments in source, target tasks. However, entailments from the source might be wrongly predicted due to tasks variability (Definition 6 - line 8) $\mathcal{T}_S, \mathcal{T}_T$. Thus, we follow the parameterization of $\gamma$ and $\gamma^*$ [Dai et al., 2007] by decreasing the weights of such entailments to reduce their effects (lines 17-19). In the next iteration, the misclassified source entailments, which are dissimilar to the target ones w.r.t. semantic embeddings, will affect the learning process less than the current iteration. Finally, StAdaB returns a binary hypothesis (line 20). Multi-class classification can be easily applied.

Algorithm 1: StAdaB($(\mathcal{D}_S, \mathcal{T}_S), (\mathcal{D}_T, \mathcal{T}_T), \mathcal{O}_T, \mathcal{G}, L, N, \alpha, \beta$)

1: Input: (i) Source/target learning domains and tasks $(\mathcal{D}_S, \mathcal{T}_S)$, $(\mathcal{D}_T, \mathcal{T}_T)$, (ii) a training LSO set of the target learning domain $\mathcal{O}_T$, (iii) distinct ABox entailments $\mathcal{G} = \{g_1, \ldots, g_m\}$ of $\mathcal{O}_S \cup \mathcal{O}_T$, (iv) a base learning algorithm $L$, (v) max. iterations $N$, (vi) $\alpha, \beta \in [0, 1]$.

2: Result: $f_{T|S}$: A predictive function by $\mathcal{D}_S, \mathcal{T}_S, \mathcal{O}_T, \mathcal{G}_S$ for $\mathcal{T}_T$.

3: begin

4: % Initialization of weights for transferability, consistency, and variability vectors of all m ABox entailments in $\mathcal{G}$.
5: Initialization of $w^1 = (w_1, \ldots, w_m)$. 
6: % Computation of semantic embeddings for all $g_i \in \mathcal{G}$.
7: $e_i \leftarrow (t(g_i), c(g_i), v(\mathcal{G}, \alpha, \beta))$, $\forall 1 \leq i \leq m$. 
8: foreach $t = 1, 2, \ldots, N$ do
9: \% Weight computation iteration
10: $p^t \leftarrow w^t / \sum_{i=1}^N w_i^t$; \% Probability distribution of $w^t$.
11: \% Predictive function $f^t$ over $\mathcal{O}_S \cup \mathcal{O}_T$.
12: $(f^t : e_i \rightarrow \mathcal{Y}_T(g_i)) \leftarrow L(e, p^t, \mathcal{Y}_T)$; 
13: \% Error computation of $f^t$ on $(\mathcal{T}_T, \mathcal{O}_T)$.
14: $\psi_i \leftarrow \sum_{j \in \mathcal{G}_S} \sum_{g \in \mathcal{G}_T} w_i^t [f^t(g_i) - \mathcal{Y}_T(g)]^2$; 
15: \% Weights for reducing errors on $\mathcal{T}_T$ over iteration.
16: $\gamma_t \leftarrow \psi_i / (1 - \gamma_t)$; $\gamma_t \leftarrow 1/(1 + e(t(\mathcal{O}_S) \cap \mathcal{O}_T))$; 
17: \% Weight update of source and target entailments in $\mathcal{G}$.
18: \% using $\gamma_t, \gamma_t$, and results from previous iteration: $w_i^t$.
19: $w_i^{t+1} \leftarrow \begin{cases} w_i^t - \gamma^t (f^t(g_i) - \mathcal{Y}_T(g_i)) & \text{if } g_i \in \mathcal{O}_T \\ w_i^t - \gamma^t (f^t(g_i) - \mathcal{Y}_T(g_i)) & \text{else} \end{cases}$

20: return Hypothesis ensemble:

$$f_{T|S}(e) = \begin{cases} 1, & \text{if } \prod_{t=1}^N \beta_t f^t(i) \geq \prod_{t=1}^N \beta_t^{-2} \\ 0, & \text{else} \end{cases}$$
A brute force approach would consist in generating an exponential number of models with any combination of entailments from source, target. StAdaB reduced its complexity by only evaluating atomic impact and (approximately) computing the optimal combination. As a side effect, StAdaB exposes entailments in the source which are driving transfer learning (cf. final weight assignment of embeddings).

5 Experimental Results

StAdaB is evaluated by two Intra-domain transfer learning cases: (i) air quality forecasting from Beijing to Hangzhou (IBH), (ii) traffic condition prediction from London to Dublin (ILD), one Inter-domain case: (iii) from traffic condition prediction in London to air quality forecasting in Beijing (ILB). Accuracy with cross validation is reported. All tasks are performed with a respective value of .3, .4, .7 for variability $v(G, \alpha, \beta)$. $\alpha$ and $\beta$ are set to .5.

In IBH, air quality knowledge in Beijing (source) is transferred to Hangzhou (target) for forecasting air quality index, ranging from Good (value 5), Moderate (4), Unhealthy (3), Very Unhealthy (2), Hazardous (1) to Emergent (0). The observations include air pollutants (e.g., PM$_{2.5}$), meteorology elements (e.g., wind speed) and weather condition from 12 stations. The semantics of observations is based on a DL $\mathcal{A} \mathcal{E} \mathcal{H}(D)$ ontology, including 48 concepts, 15 roles, 598 axioms, 1, 065, 600 RDF triples are generated on a daily basis. 18 (resp. 5) months of observations are used as training (resp. testing). Even though the ontologies are from the same domain, the proportion of similar concepts and roles are respectively .81 (i.e., 81% of concepts are similar) and .74. For instance, no hazardous air quality concept in Hangzhou.

In ILD, bus delay knowledge in London (source) is transferred to Dublin (target) for predicting traffic conditions classified as Free (value 4), Low (3), Moderate (2), Heavy (1), Stopped (0). Source and target domain data include bus location, delay, congestion status, weather conditions. We enrich the data using a DL $\mathcal{E} \mathcal{C} \mathcal{M}^+$ domain ontology (55 concepts, 19 roles, 25, 456 axioms). 178, 700, 000 RDF triples are generated on a daily basis. 24 (resp. 8) months of observations are used as training (resp. testing). The concept and role similarities among the two ontologies are respectively .73 and .77.

In ILB, bus delay knowledge in London (source) is transferred to a very different domain: Beijing (target) for forecasting air quality index. Data and ontologies from IBH and ILD are considered. Both domains share some common and conflicting knowledge. Inconsistency might then occur. For instance, both domains have the concepts of City, weather such as Wind but are conflicting on their importance and impact on the targeted variable i.e., bus delay in London and air quality in Beijing. The concept and role similarities among the two ontologies are respectively .23 and .17.

5.1 Semantic Impact

Table 2 reports the impact of considering semantics (cf. Sem. vs. Basic) and (in)consistency (cf. Consistency / Inconsistency) in semantic embeddings on Random Forest (RF), Stochastic Gradient Descent (SGD), AdaBoost (AB). “Basic” models are models with no semantics attached. “Plain” models are modelling and prediction in the target domain i.e., no transfer learning, while “TL” refers to transferring entailments from the source. As expected semantics positively boosts accuracy of transfer learning for intra-domain cases (IBH and ILD) with an average improvement of 13.07% across models. More surprisingly it even over-performs in the inter-domain case (ILB) with an improvement of 20.03%. Inconsistency has shown to drive below-baseline accuracy. On the opposite results are much better when considering consistency for intra-domain cases (63.55%), and inter-domain cases (187.89%).

<table>
<thead>
<tr>
<th>Case</th>
<th>Models</th>
<th>RF</th>
<th>TL</th>
<th>SGD</th>
<th>TL</th>
<th>AB</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBH</td>
<td>Basic</td>
<td>.61</td>
<td>.61</td>
<td>.59</td>
<td>.62</td>
<td>.59</td>
<td>.63</td>
</tr>
<tr>
<td></td>
<td>Consistency</td>
<td>.65</td>
<td>.74</td>
<td>.62</td>
<td>.69</td>
<td>.64</td>
<td>.73</td>
</tr>
<tr>
<td></td>
<td>Inconsistency</td>
<td>.56</td>
<td>.64</td>
<td>.52</td>
<td>.60</td>
<td>.49</td>
<td>.63</td>
</tr>
<tr>
<td></td>
<td>Cons. / Basic</td>
<td>+16.07%</td>
<td>+19.23%</td>
<td>+10.50%</td>
<td>+14.12%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semantic / Basic</td>
<td>+15.95%</td>
<td>+18.18%</td>
<td>+11.27%</td>
<td>+14.12%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ILD</td>
<td>Basic</td>
<td>.68</td>
<td>.71</td>
<td>.59</td>
<td>.62</td>
<td>.63</td>
<td>.69</td>
</tr>
<tr>
<td></td>
<td>Consistency</td>
<td>.75</td>
<td>.78</td>
<td>.66</td>
<td>.71</td>
<td>.76</td>
<td>.82</td>
</tr>
<tr>
<td></td>
<td>Inconsistency</td>
<td>.44</td>
<td>.52</td>
<td>.26</td>
<td>.49</td>
<td>.23</td>
<td>.46</td>
</tr>
<tr>
<td></td>
<td>Cons. / Basic</td>
<td>+90.22%</td>
<td>+102.58%</td>
<td>+102.58%</td>
<td>+102.58%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semantic / Basic</td>
<td>+10.07%</td>
<td>+14.70%</td>
<td>+19.42%</td>
<td>+19.42%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ILB</td>
<td>Basic</td>
<td>.62</td>
<td>.65</td>
<td>.60</td>
<td>.66</td>
<td>.61</td>
<td>.68</td>
</tr>
<tr>
<td></td>
<td>Consistency</td>
<td>.74</td>
<td>.79</td>
<td>.69</td>
<td>.78</td>
<td>.74</td>
<td>.85</td>
</tr>
<tr>
<td></td>
<td>Inconsistency</td>
<td>.23</td>
<td>.45</td>
<td>.29</td>
<td>.42</td>
<td>.18</td>
<td>.34</td>
</tr>
<tr>
<td></td>
<td>Cons. / Basic</td>
<td>+1.85%</td>
<td>+16.25%</td>
<td>+32.35%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semantic / Basic</td>
<td>+20.44%</td>
<td>+17.33%</td>
<td>+22.33%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Forecasting Accuracy / Improvement over State-of-the-art Models (noted as Basic) with Consistency / Inconsistency (Consistency ratio .8) based Knowledge Transfer.

Figure 4 reports the impact of consistency and inconsistency on transfer learning by analysing how the ratio of consistent transferable knowledge in [0, 1] is driving accuracy. Accuracy is reported for methods in Table 2 on intra- (average of IBH and ILD) and inter-domains (ILB). Max. (resp. min.) accuracy is ensured with ratio in [.9, .7] (resp. [.3, .1]). The more consistent transferable knowledge the more transfer for [.9, .1]. Interestingly having only consistent (resp. inconsistent) transferable knowledge does not ensure best (resp. worst) accuracy. This is partially due to under-/ (resp. over-) populating the target task with conflicting knowledge, ending up to limited transferability.

5.2 Comparison with Baselines and Discussion

We compare StAdaB (L = Logistic Regression, N = 800) with Transfer AdaBoost TrAB [Dai et al., 2007], Transfer Component Analysis (TCA) [Pan et al., 2011], TrSVM [Benavides-Prado et al., 2017] and SemiTr [Lv et al., 2012] (cf. details in Section 6). We considered intra-domains: IBH, ILD and inter-domains: ILB and ILB’ (i.e., ILB with same level of semantic expressivity covered by SemiTr). Results report that transfer learning has limitations in the Beijing - Hangzhou context cf. Figure 5(a). Although our approach over-performs other techniques (from 10.29% to 50%), accuracy does not exceed 74%. The latter is due to the context, which is limited by the (i) semantic expressivity and (ii) data availability in Hangzhou. The results show that TrSVM and...
TCA reach similar results (average difference of 9.1%) in all the cases. However our approach and TrAB tend to maximise the accuracy specially in inter-domains ILB and ILB* in Figures 5(c) and 5(d) as both favour heterogenous domains by design. Interestingly the semantic context of ILB* in Figure 5(d) (i) does not favour SemTr much (+7.46% vs. ILB), (ii) does not have impact for StAdaB compared to ILB, and more surprisingly (iii) does benefit TrAB (+9.15% vs. ILB). This shows that expressivity of semantics is crucial in our approach to benefit from (in-)consistency in transfer.

6 Related Work

We briefly divide the related work into instance transfer, model transfer and semantics transfer. Instance transfer selectively reuses source domain samples with weights [Dai et al., 2011] introduced a transfer component analysis for domain adaptation; [Benavides-Prado et al., 2017] selectively shares the hypothesis components learnt by Support Vector Machines. These methods however usually ignore data semantics.

Semantics transfer incorporates external knowledge to boost the above two groups, by using semantic nets [Lv et al., 2012] or knowledge graph-structure data [Lee et al., 2017] to derive similarity in data and features, with no reasoning applied. There are efforts on Markov Logic Networks (MLN) based transfer learning, by using first [Mihalkova et al., 2007; Mihalkova and Mooney, 2009] or second order [Davis and Domingos, 2009; Van Haaren et al., 2015] rules as declarative prediction models. However, these approaches do not address the problem of when is feasible to transfer. Our approach uses OWL reasoning to select transferable samples (addressing ‘when to transfer’), then enriching the samples with embedded transferability semantics. It can support different machine learning models (and not just rules).

7 Conclusion

We addressed the problem of transfer learning in expressive semantics settings, by exploiting semantic variability, transferability and consistency to deal with *when to transfer* and *what to transfer*, for existing instance-based transfer learning methods. It has been shown to be robust to both intra- and inter-domain transfer learning tasks from real-world applications in Dublin, London, Beijing and Hangzhou. As for future work, we will investigate limits and explanations of transferability with more expressive semantics (e.g, based on approximate reasoning) [Pan et al., 2016; Du et al., 2019].

Acknowledgments

This work is partially funded by NSFC91846204.
References


