Transfer Learning of Link Specifications

Axel-Cyrille Ngonga Ngomo
Universität Leipzig
Institut für Informatik
AKSW Research Group,
Postfach 100920
D-04009 Leipzig, Germany,
http://aksw.org

Jens Lehmann
Universität Leipzig
Institut für Informatik
AKSW Research Group,
Postfach 100920
D-04009 Leipzig, Germany,
http://aksw.org

Mofeed Hassan
Universität Leipzig
Institut für Informatik
AKSW Research Group,
Postfach 100920
D-04009 Leipzig, Germany,
http://aksw.org

***Abstract*—Over the last years, link discovery frameworks have been employed successfully to create links between knowledge bases. Consequently, repositories of high-quality link specifications have been created and made available on the Web. The basic question underlying this work is the following: Can the specifications in these repositories be reused to ease the detection of link specifications between unlinked knowledge bases? In this paper, we address this question by presenting a formal transfer learning framework that allows detecting existing specifications that can be used as templates for specifying links between previously unlinked knowledge bases. We discuss both the advantages and the limitations of such an approach for determining link specifications. We evaluate our approach on a variety link specifications from several domains and show that the detection of accurate link specifications for use as templates can be achieved with high reliability.**

I. Introduction

Link Discovery has gained significant momentum over the last years due to the growth of the Linked Data Web and the use of Semantic Web technologies across manifold applications including question answering [25], federated querying [21], large-scale inferences [26] and data integration [3]. Over the last years, several tools and libraries have been developed with the main aim of efficiently supporting the whole of the link discovery process [6], [24], [17]. In general, this process can be modeled as consisting of two steps: Once provided with a source and target set of instances, the first step consists of discovering a link specification for retrieving high-quality links. This step is of crucial importance the precision and recall of the link discovery process depend heavily on the link specification used. Once a specification has been decided upon, it has to be carried out. Several frameworks such as LIMES [10] and SILK [8] have been developed to address the quadratic runtime of link discovery. With the ongoing growth of the Linked Data Web, these tools and libraries have been employed successfully to create links between the different knowledge bases on the Linked Data Web. Consequently, different repositories of high-quality link specifications have been created and made publicly available on the Web. For example, the LATC project[1] generated 170 specifications for linking knowledge bases across several domains including persons, organizations and geo-spatial entities.

[1]http://latc-project.eu

The basic observation underlying this paper is the following: While several approaches for learning link specifications have been developed over the past years (e.g., [13], [14]), the discovery of a specification to link two datasets has been regarded as an isolated process. Hence, to the best of our knowledge, none of the previous approaches to detecting link specifications has made use of the already available knowledge available in repositories for link specifications. The primary aim of this paper is consequently to explore how this knowledge can be reused within the framework of transfer learning. More specifically, our goal is present a formal framework for the transfer learning of link specifications with the aim of improving the computation of link specifications.

The rest of this paper is structured as follows: We first begin by presenting formally the problem of link specification and the idea that underlie transfer learning. Thereafter, we present a formal framework that combines both ideas and allows implementing transfer learning for link specifications. We then present the heuristics we implemented to apply this approach to real link specifications created by domain experts. Subsequently, we evaluate our framework on 113 specifications retrieved from the LATC repository and show that we can effectively detect templates for link discovery. Our evaluation yet also shows the limitation of transfer learning on specifications. We discuss our findings and present some of the work related to our approach. Finally, we conclude with an overview of possibilities that arise in this novel field of research. Note that throughout this paper, we use RDF prefixes as available at http://prefix.cc.

II. Preliminaries and Notation

A. Link Discovery

The link discovery problem, which is similar to the record linkage problem, is an ill-defined problem and is consequently difficult to model formally [2]. In this work, we expand on the formalization presented in [12]. The goal of link discovery can be described as follows: Given two sets of resources S (source) and T (target) as well as a relation ρ, compute the set M of pairs of instances $(s,t) \in S \times T$ such that $\forall (s,t) \in M : \rho(s,t)$. The sets S resp. T are usually (not necessarily disjoint) subsets of the resources contained in two (not necessarily disjoint) knowledge bases $\mathcal{K}_S$ resp. $\mathcal{K}_T$. In

most frameworks, the computation of whether $\rho(s,t)$ holds for two elements is carried out projecting the elements of S and T based on their properties in a similarity space $\mathfrak{S}$ and setting $\rho(s,t)$ iff some similarity condition $\sigma(s,t) \geq \tau$ is satisfied, where $\sigma : S \times T \rightarrow [0,1]$ is a similarity function and $\tau \in [0,1]$. The specification of the sets S and T and of this similarity condition is usually carried out within a *link specification*. In general, a link specification consists of three parts:

1) two sets of restrictions $\mathcal{R}_1^S$ ... $\mathcal{R}_m^S$ resp. $\mathcal{R}_1^T$ ... $\mathcal{R}_k^T$ that specify the sets S resp. T,
2) a specification of a complex similarity metric σ via the combination of several atomic similarity measures σ_1, ..., σ_n and
3) a set of thresholds τ_1, ..., τ_n such that τ_i is the threshold for σ_i.

A restriction $\mathcal{R}$ is generally a logical predicate. Typically, restrictions in link specifications state the `rdf:type` of the elements of the set they describe, i.e., $\mathcal{R}(x) \leftrightarrow x$ `rdf:type someClass` or the features the elements of the set must have, e.g., $\mathcal{R}(x) \leftrightarrow (x$ `someProperty someValue`$)$. Each $s \in S$ must abide by each of the restrictions $\mathcal{R}_1^S$ ... $\mathcal{R}_m^S$, while each $t \in T$ must abide by each of the restrictions $\mathcal{R}_1^T$ ... $\mathcal{R}_k^T$. Note that the atomic similarity functions $\sigma_1, ..., \sigma_n$ can be combined to σ by different means. Also note that this formal model allows regarding link specifications as binary classifiers that assign the class $+1$ to pairs such that $\rho(s,t)$ and -1 to pairs that should not. This view of link specifications and the existence of repositories for link specifications automatically raises the question whether transfer learning can be applied to alleviate the costs necessary to create them. Addressing this issue is the core idea behind transfer learning.

B. Transfer Learning

Our formalization of machine learning in general and transfer learning in particular is based on [16]. The goal of most machine learning approaches is to find a predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$ which can compute the right classification $f(x_i) = y_i$ when given the input data x_i. We call the set $\mathcal{X}$ the *domain* in which we are to learn the right way to classify, while we dub the pair (f, Y) the *task* t of the machine learning approach at hand and write $t = (f, \mathcal{Y})$. In the case of link discovery, $\mathcal{X} = S \times T$ while $\mathcal{Y} = \{+1, -1\}$ with $f(x_i) = +1$ if $\rho(s,t)$ and $f(x_i) = -1$ in all other cases. Finding the function f for link discovery tasks is generally very costly, as it requires either (mostly manually) labeled training data [13] or a significant amount of computation [14]. The idea behind *transfer learning* (also coined *knowledge transfer*) [16] can be broadly described as follows: Given other machine learning tasks t' with known or unknown classification functions f' that are somehow "related" to f, use the functions f' or the domain knowledge available for determining f' (i.e., transfer the knowledge from the tasks t') to improve the process of finding $(f, \mathcal{Y})$.

In general, three categories of transfer learning from a task $t' = (f', \mathcal{Y}')$ to a task $t = (f, \mathcal{Y})$ can be differentiated: inductive, transductive and unsupervised transfer learning. *Inductive learning* assumes that training data for learning f is available and aims either to reuse labeled data available for learning f' (in which case inductive learning reduces to multi-task learning) or to use the data on which f' is to be learned (self-taught learning). *Transductive learning* assumes that labeled data is only available for the task t' and aims at reusing that data for learning f. If $\mathcal{X} = \mathcal{X}'$, then transductive learning reduces to adapting $(f', \mathcal{Y}')$ to a new task. Else, if the domain and task are the same, we are confronted with a machine learning task which requires dealing with sample selection biases and covariance shift. The last category of transfer learning, *unsupervised transfer learning*, is concerned with learning when labeled data is available for learning neither $(f, \mathcal{Y})$ nor $(f', \mathcal{Y}')$. In this paper, we address the following transductive transfer learning task: Given a set of functions f_i' computed for domains X_i', improve the determination of the function f for solving task $(f, \mathcal{Y})$ on the domain $\mathcal{X}$ with $\forall i : \mathcal{X} \neq \mathcal{X}_i'$.

III. Formal Framework for Transfer Learning of Link Specifications

The idea behind our approach to the transfer learning of link specifications is sketched in Figure 1. Instead of regarding the learning of link specifications as an isolated task (see left side of the figure), we aim to reuse existing specifications (see right side of the figure). Based on a repository of existing link specifications, the aim of our approach is thus to extract knowledge out of the existing (and often validated) specifications and transfer it to a link specification, which can be used as template (i.e., an initial solution f) for the linking task at hand. Let us assume that the current domain $\mathcal{X} = S \times T$ is fully known, i.e., that the set of restrictions that describe both S and T has already been computed[2]. In link discovery, the elements of the domain $\mathcal{X}$ are usually a set of pairs of points $(s,t) \in S \times T$ from sets of instances S and T described in a similarity space $\mathfrak{S}$. This similarity space is defined over a subset of $P_L \times P_L'$ of source properties P_L and target properties P_L'. Consequently, a link specification can be described as a binary classifier that maps elements of $(P_L \times P_L')^n$ to the classes +1 and -1.

One of the challenges of transferring knowledge from a link specification to another is that link specifications can be very complex in general since it combines restrictions, arbitrarily nested complex similarity metrics and thresholds. Still, given the formalization of link specifications presented in Section II-A, we can tackle the transfer learning of link specifications by reducing it to a set of three simpler problems: The measurement of restriction similarity, property similarity as well as determining the accuracy of link specifications. The *restriction similarity* ensures that the domains $\mathcal{X}'$ of known link specifications and the domain $\mathcal{X}$ of the link specifications to devise are as related as possible. Here the basic idea is

[2]Computing such restrictions is the object of ontology matching, which is out of the scope of this paper. A good overview of approaches for achieving such computations can be found in [23].

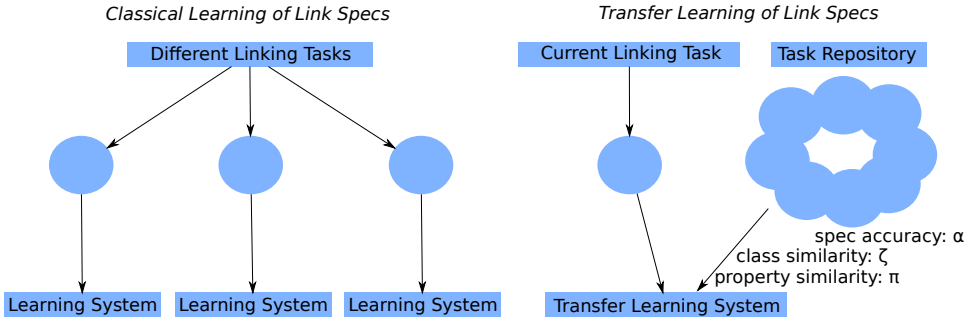


Figure 1. Overview of transfer learning of link specifications.

that the more related $\mathcal{X}$ and $\mathcal{X}'$ are, the higher the probability that the function f' is a good initial value for f. The *property similarity* ensures that the similarity space $\mathfrak{S}'$ used in $\mathcal{X}'$ can be mapped to a similarity space $\mathfrak{S}$ that describes the elements of $\mathcal{X}$. Therewith, we ensure that the transfer of $(f', \mathcal{Y}')$ to $(f, \mathcal{Y})$ is possible my mapping each of the dimensions of $\mathfrak{S}'$ to $\mathfrak{S}$. Finally, we assume that not all tasks were solved with the same accuracy. Thus we also measure the *accuracy* of each task $(f', \mathcal{Y}')$ so as to learn from the better ones. In the following, we describe formally how those three functions can be used for transfer learning, whereas in the next section, we describe implementations of those functions.

Transfer learning on link specifications assumes that 2 sets of restrictions are given and aims to find the right classifier based on known classifiers for other source and target classes mapped by known classifiers. As pointed out in Section II-A, these restrictions are mostly class restrictions of the form s `rdf:type someClass`. Thus, in the following, we will reduce the similarity of restrictions to the similarity of sets of classes. Note that this reduction does not diminish the applicability of our transfer learning framework as even triples of the form s `someProperty someValue` can be considered to be class expressions. In the following, we denote the set of all classes as C, the set of all properties as P and the set of all link specifications as Q. Lowercase variants of these letters are used to indicated elements of the respective sets.

The input of a transfer learning problem are two sets of classes $\mathcal{C} \subseteq C$ and $\mathcal{C}' \subseteq C$, which should be mapped. Furthermore, we assume a set $\{q_1, \ldots, q_m\}$ of existing link specifications from which we want to transfer knowledge. In addition, we assume the existence of two sets P_L and P'_L denoting the sets of properties relevant for the current linking task. Those sets can be computed automatically by determining the properties associated with instances of $\mathcal{C}$ and $\mathcal{C}'$ or provided manually.

Using these preliminaries, we can define the main functions for implementing transfer learning: $\zeta : 2^C \times 2^C \mapsto [0, 1]$ compares the similarity of two input class sets, possibly using background knowledge about those classes available in the underlying triple stores, and returns a value between 0 and 1, where 0 indicates no similarity and 1 indicates identity. Similarly, we assume the existence of a function $\pi : P \times P \mapsto [0, 1]$, which works analogously for properties. The set all such property similarity functions is denoted as Π. For each specification in the background knowledge, an assessment function $\alpha : Q \mapsto [0, 1]$ can be used to judge its quality. The aim of this function is to prioritize high quality link specifications. Specifying α is optional, i.e. it can be omitted by dropping it from Equation 1.

In order to implement transfer learning, the basic framework implements a set of methods: $r : Q \times P \times \Pi \mapsto T$ replaces each source property in a link specification with the most similar property in P according to the given similarity function π. Analogously, $r' : Q \times P \times \Pi \mapsto T$ performs the same replacement for target properties. Moreover, we assume the existence of two helper functions $\psi : Q \mapsto 2^C$ and $\psi' : Q \mapsto 2^C$, which return the source and the target classes in a link specification, respectively. Note that the formal specification of transfer learning can be used in all linking tools if the above four functions are implemented.

Combining all of the above notions, we can formally define the equation which allows computing a score ω for how well a link specification task t' (with link specification q') is suited to be for transfer learning for the task $t = (f, \mathcal{Y})$are hand:

$$\omega(t, t') = \alpha(q')\zeta(\psi(q'), \mathcal{C}) \cdot \zeta(\psi'(q'), \mathcal{C}') \cdot r'(r(q', P_L, \pi), P'_L, \pi) \quad (1)$$

Intuitively, the equation computes the product of the different similarity functions defined above and this depends on the quality of existing specifications and their similarity to the specification to devise. Furthermore, it maps properties and classes in existing specifications to those relevant for the current linking task. This weight function can be used in manifold ways. For example, it can be used to compute a weighted sum over all available link specifications. It can also be used to sort and rank the available link specifications and present them to the user in charge of devising the link specification for the task at hand. Manifold approaches can be used to implement the appropriate similarity functions σ, π and α for computing the similarity of properties and classes as well as the accuracy of link specifications. In the following sections, we present such functions.

IV. HEURISTICS

A. Accuracy of Specifications

The most common method to assess precision and recall of link specifications is to manually label created links as correct or incorrect. Several link discovery tools such as LIMES [13] as well as the LATC[3] and SAIM[4] platforms integrate user feedback directly and let users evaluate links. Furthermore, game-based approaches such as VeriLink[5] have been devised to evaluate links. In the LATC repository in particular (which we use as a base for our evaluation) 66.47% of the link specifications are associated with reference positive and negative links. To estimate the quality of a link specification, we can calculate the precision of the evaluated links as the number of correct links c divided by the total number n of evaluated links. The assumption that the precision of the verified links can be used as approximation of the precision of all links generated by the link specification would reflect current practice. However, such an approximation would not take into account that a higher number of evaluated links usually leads to a higher confidence estimate. For this reason, we define the specification accuracy as the center of the 95% confidence interval of the proportion of correct links, which is a statistical technique that allows avioding the aforementioned shortcomings of the precision measure. To compute this interval efficiently, we use the improved Wald method defined in [1]:

$$\max(0, p' - 1.96 \cdot \sqrt{\frac{p' \cdot (1-p')}{n+4}}) \text{ to } \min(1, p' + 1.96 \cdot \sqrt{\frac{p' \cdot (1-p')}{n+4}})$$

$$\text{with } p' = \frac{c+2}{n+4}$$

This formula can be computed very efficiently and has been shown to be accurate in [1]. To illustrate the effect in an example, 3 correct links out 3 evaluated links yields a specification accuracy of only 69.1% compared to 100% when directly using the proportion. For larger numbers of evaluated links, both approaches converge, i.e. 603 out of 610 correct links yield a specification accuracy of 98.5% whereas a division of both numbers results in a value of 98.8%.

One weakness of our approach to compute specification accuracy is that we cannot take any knowledge into account about how the references links were evaluated. The above approach assumes that they were randomly drawn from the set of generated links. However, in some cases algorithms may have just presented the user problematic links with unclear classification which is common in active learning approaches [12]. In that case, our specification accuracy estimate tends to be too pessimistic.

B. Similarity of Classes

In addition to measuring the accuracy of link specifications, it is central measure how similar the domain $\mathcal{X} = S \times T$ at hand is to domains $\mathcal{X}' = S' \times T'$ for which classifiers are known. As the domains are defined by a set of restrictions, measuring the similarity between two domains can be carried by measuring the similarity between the restrictions that define them. Most commonly, each source restriction $\mathcal{R}_i^S$ is a s `rdf:type` c_i. The call the set $\mathcal{R}(S) = \bigcup_i \{c_i\}$ the *restriction set* for a source S. We define $\mathcal{R}(T)$ analogously. Given this model, we implemented two different approaches to computing the similarity of classes (which we will denote $\zeta(S, S')$ for the sake of brevity): a *label-based* approach and a *data-centric* approach. The idea behind the label-based similarity is that two sets of instances are similar if the labels in the elements of their restriction sets are similar. We thus defined the first similarity $\zeta_l(S, S')$ as

$$\sum_{c_i \in \mathcal{R}(S)} \max_{c'_j \in \mathcal{R}(S')} sim(label(c_i), label(c'_j)). \quad (2)$$

Here, the function $label(c)$ returns the label of a resource while sim is a string similarity function.

Given that the label of resources is stored in endpoints that are not always available, we extended the similarity measure ζ_l by considering the local name $name$ of each resource as its label. We thus extended the similarity function ζ_l to ζ_u as follows:

$$\zeta_u(S, S') = \sum_{c_i \in \mathcal{R}(S)} \max_{c'_j \in \mathcal{R}(S')} sim(name(c_i), name(c'_j)). \quad (3)$$

A potential drawback of label-based similarity functions is that the labels of classes might not reflect the semantics of these classes. For example, specifications linking large geographical knowledge bases such as LinkedGeoData and Geonames contain restrictions such as `geonames:LK`, whose label and local names do not permit to determine the meaning of the restrictions. Thus, we devised a second category of similarity function that relies on the data contained in the endpoints. Here, we compute the similarity of two sets S and S' of instances by measuring the average string similarity of the datatype property values of the resources contained in these sets. Thus, we define the data-centric similarity of two resources $s \in S$ and $s' \in S'$ as

$$\zeta_d(s, s') = \frac{1}{|P(s)||P(s')|} \sum_{x \in P(s)} \sum_{y \in P(s')} sim(x, y), \quad (4)$$

where $P(s) = \{x : s \; p \; x \wedge p$ `rdf:type owl:DatatypeProperty`$\}$. The natural extension of ζ_d to measuring the similarity of restriction sets is then given by

$$\zeta_d(S, S') = \frac{1}{|S||S'|} \sum_{s \in S} \sum_{s' \in S'} \zeta_d(s, s'). \quad (5)$$

The similarity of properties $\pi(p_i, p'_j)$ can be defined analogously. We thus also implemented three property similarity functions π_l, π_u and π_d. Note that our approach can easily be extended to any type of restriction.

[3] http://latc-project.eu
[4] http://saim.sf.net
[5] http://verilink.aksw.org

V. Evaluation

A. Experimental Setup

The goal of our evaluation was two-fold. First, we wanted to evaluate how well how approach can suggest the right reference specification (i.e., the right function f') to use as template for building the prediction function for a previously unseen linking task. We retrieved 113 specifications from the LATC repository which contained a manual evaluation of the links generated by specifications. The distribution of the type of specifications across different domains is shown in Figure 2. We then ran a leave-one-out evaluation as follows: We removed each specification that was to be learned from the set of available specifications and ran our approach on the specification at hand. We then looked at the top-scored specification and compare the learned specification with that used originally. In case of a match (i.e., if the two specifications relied on exactly the same combination of atomic similarity functions), the specification was assigned a 1, else a 0. Note that this represents the lower bound of the Mean Reciprocal Rank that our approach can achieve. Note that we computed the similarity of the specifications based on the URI similarity ζ_u as both the source and target endpoint of only 12 of the 113 specifications (10.62%) were alive during the period of experiments.

The goal of the second series of experiments we carried out was to discover whether the functions f' for other domains could be used directly. Thus, we measured the precision and recall of the detected functions f' for the domain $\mathcal{X}$ for which a specification was to be devised. Note that this experiment was clearly biased due to the limitation to only 12 of the 113 specifications. Here, we were able to use all three similarity methods for classes and properties.

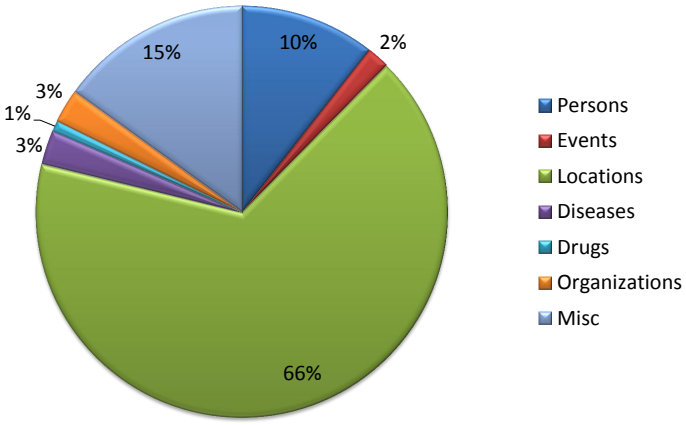


Figure 2. Distribution of specifications across different domains.

B. Results

The results of our first series of experiments are shown in Figure 3. Note that the whole experiment lasted less than 6 seconds on a laptop with a 2.5GHz processor running Windows 7 Enterprise, making the approach very time-efficient. On average, we were able to detect the right specification in 81% of the cases. This was mainly due to our approach achieving remarkably good results on specifications that aimed to link geo-spatial entities with each others. Here, we were able to detect the right specification in approximately 92% of the cases. The specifications in the category Misc. (short for Miscellaneous) were in many cases singletons, i.e., specifications designed for a particular domain that was unrelated to the other domains in our pool of specifications. For example, only one specification linked movies with each other, making the transfer learning for this specification difficult. While 12 specifications linked persons with each other, we were able to detect a fitting specification in only 58.3% of the cases. This was mainly due to the detection of persons being highly dependent on the schema of the ontology used as well as the domain in which the linking was to be carried out. For example, while our approach suggest the specification used for linking the persons from DBpedia with the authors from Open Library as template for linking the authors from the Gutenberg project with the persons in DBpedia, the mismatch between the schema of the Open Library and Project Gutenberg would have required significant modifications of the specification, leading us to score this mapping as a 0.

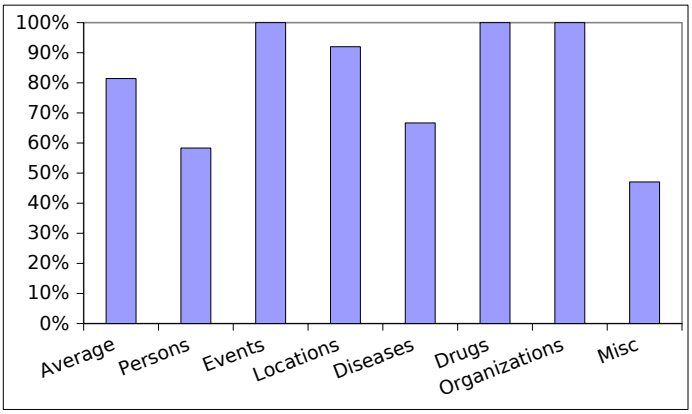


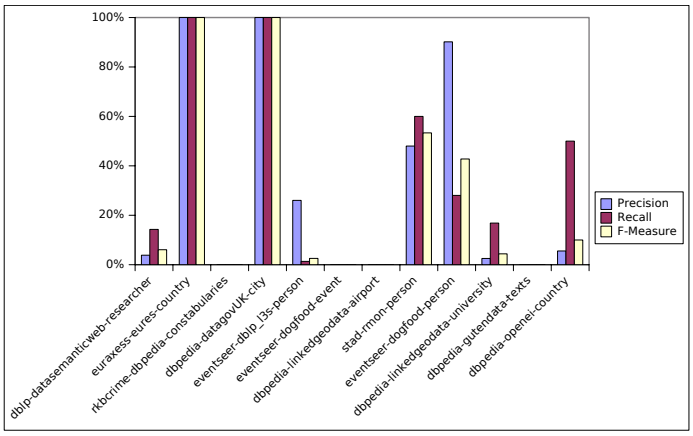
Figure 3. Average ranking results for specifications.

In the second step of the evaluation, we considered the specifications for which both the source and target endpoints where alive (see Figure 4. Note that the results for the label- and URI-based similarities were similar). Here, we were able to run the whole pipeline of transfer learning. We first learned a specification for each source and target pair out of the other 12 specifications. An excerpt of the specifications learned with ζ_u are shown in Table I. In many cases, we failed to determine the right template because only one correct template was available for learning. For example, we failed to determine the right template for linking DBpedia and LinkedGeoData as this was the only specification which reflected correctly how to link geo-spatial entities. Once removed from the template set, it could not be learned.

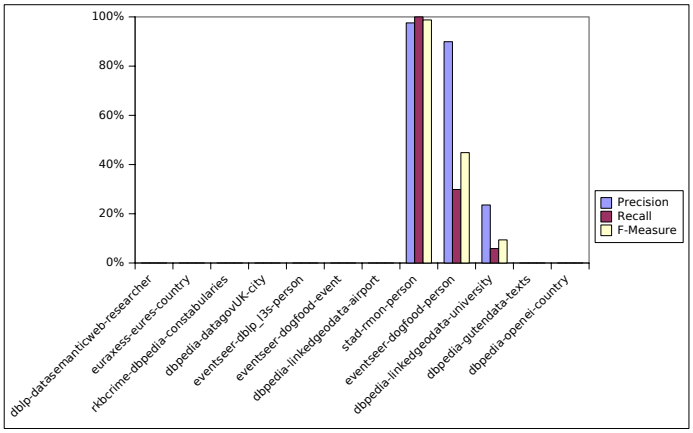
Running the specifications learned yet led to interesting even when not very surprising results: Even when the templates that we learned were correct, the thresholds were mostly erroneous,

Source	Target	Classes	Specification
DBpedia	OpenEI	Countries	trigrams(a.rdfs:label, b.rdfs:label) $\geq$ 0.7
DBLP	Semantic Web	Persons	levenshtein(a.foaf:name, b.foaf:name) $\leq$ 2
Eventseer	Dogfood	Persons	trigrams(a.foaf:name, b.rdfs:label) $\geq$ 0.7
Eventseer	DBLP	Persons	trigrams(a.foaf:name, b.foaf:name) $\geq$ 0.7
DBpedia	LinkedGeodata	Universities	trigrams(a.rdfs:label, b.rdfs:label) $\geq$ 0.7
DBpedia	DataGov.uk	Cities	trigrams(a.rdfs:label, b.rdfs:label) $>$ 0.5 AND euclidean(a.wsg84:lat|wgs84:long, b.wgs84:lat|wgs84:long) $>$ 0.9
Stad	Rmon	Persons	levenshtein(a.foaf:name, b.rdfs:label) $\leq$ 2
Euraxess	Eures	Countries	MAX(trigrams(a.rdfs:label, b.rdfs:label), trigrams(a.eures:countryCode, b.rdfs:label)) $\geq$ 0.7
Eventseer	Dogfood	Events	MAX(trigrams(a.rdfs:label, b.rdfs:label), trigrams(a.ns4:dtstart, b.rdfs:label)) $\geq$ 0.7
DBpedia	Sider	Drugs	MAX(levenshtein(a.foaf:page, b.rdfs:label), MAX(levenshtein(a.rdfs:label, b.rdfs:label), MAX(levenshtein(a.foaf:page, b.sider:sideEffect), levenshtein(a.rdfs:label, b.sider:sideEffect)))) $\leq$ 2

Table I
EXAMPLE OF SPECIFICATIONS LEARNED VIA TRANSFER LEARNING



(a) Results using URI similarity



(b) Results using the sampling-based similarity

Figure 4. Precision, recall and F-score achieved in our experiments. For the sampling we use a sample size of 100 and a similarity threshold of 0.25

leading to mostly very low F-scores. Note that our approach did not alter the thresholds nor the similarity measures of the specifications we detected. The low F-scores are clearly due to the right thresholds being significantly dependent on the actual data in the triple stores.

This is a not unexpected but still important limitation of transfer learning based solely for link specifications: By learning from the specifications alone, it is difficult (if not impossible) to detect the correct thresholds for running high-accuracy link specifications. Methods for detection the right thresholds can yet be easily derived from existing methods for learning link specifications such as those presented in [12], [13], [14]. Our results thus indicate that transfer learning is not only possible but can also be carried out with satisfying accuracy even when relying on only the top-ranked specification as template.

VI. RELATED WORK

Our work is mostly related to link discovery and transfer learning. Several frameworks have been developed with the aim of addressing the quadratic a-priori runtime of Link Discovery. [11] presents LIMES, a lossless and hybrid link discovery framework, which implements the HYPPO algorithm [10] for the time-efficient computation of links amongst others. [8] present a lossless approach called MultiBlock that allows to discard a large number of comparisons. Similar frameworks include those presented in [19], [20], [6], [24], [17]. The second core problem that needs to be addressed while computing links across knowledge bases is the detection of accurate link specifications. The problem has been addressed in manifold ways: The RAVEN approach [12] relies on active learning to learn linear and Boolean classifiers for discovering links. While [7] relies on batch learning and genetic programming to compute link specifications, the approach presented in [13] combines genetic programming and active learning to learn link specifications of high accuracy. In recent work, approaches for the unsupervised computation of link specifications have been devised. For example, [14] show how link specifications can be computed by optimizing a pseudo-F-measure within a genetic programming setup.

Transfer learning on the other hand has been used in manifold domains [16]. For example, authors such as [9] and [5] propose transfer learning models for the recognition of classes of objects in computer vision. [4] use transfer

learning to reduce the labeling cost for adapting sentiment analysis models across domains and show that it significantly reduces the annotation cost necessary necessary to perform well in new domains. [22] combine transfer learning, case-based reasoning and reinforcement learning to improve the performance of a gaming engine across different scenarios. [27] employ transfer learning for improving cross-domain text classification. [18] rely on a transfer learning approach to learn image classifications when confronted with sparse data. [15] address the problem of transferring localization models within the context of sensor-based location estimation. Transfer learning has also been shown to wrk well when employed for other tasks such as collaborative image clustering [28] or even planning [29]. Manifold other applications and references can be found in [16]. While transfer learning has been shown to achieve good results in several domains, it has not been applied to the detection of link specifications so far. This paper addresses this research gap by presenting how transfer learning could be used for this purpose.

VII. Conclusions and Future Work

In this paper, we presented a formal model for the transfer learning on link specifications. We showed that by using a simple instantiation of our model, we can detect the specification after which our specification should be modeled with a mean reciprocal rank larger or equal to 0.81. We also showed that transfer learning cannot replace the learning of link specifications in itself as the thresholds for the specifications have to be learned out of the data. (Semi-)Automatic ways for determining the right thresholds can be derived from previous approaches to learning link specifications can be combined with our approach. For example, specification templates can be used to seed genetic programming algorithms [14] such as to accelerate their convergence. In addition, knowing which template to use can help when choosing the right deterministic model (Boolean classifier, linear classifier) as well as its initialization for these models [12]. The main quest behind our future work will thus be twofold: First, we will aim to combine existing approaches to learning link specications to transfer learning and measure how much labeling effort can be saved when applying transfer learning to the detection of link specifications. Moreover, we will aim to combine our transfer learning approach with more sophisticated class and property similarity approaches to see how they affect our mean reciprocal rank score.

References

[1] A. Agresti and B. A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[2] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *SIGMOD Conference*, pages 783–794, 2010.

[3] D. Ben-David, T. Domany, and A. Tarem. Enterprise data classification using semantic web technologies. In *ISWC*, 2010.

[4] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*, pages 187–205, 2007.

[5] L. Fei-Fei. Knowledge transfer in learning to recognize visual object classes. *ICDL*, 2006.

[6] A. Hogan, A. Polleres, J. Umbrich, and A. Zimmermann. Some entities are more equal than others: statistical methods to consolidate linked data. In *NeFoRS*, 2010.

[7] R. Isele and C. Bizer. Learning Linkage Rules using Genetic Programming. In *Sixth International Ontology Matching Workshop*, 2011.

[8] R. Isele, A. Jentzsch, and C. Bizer. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In *WebDB*, 2011.

[9] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. *CVPR*, (1), 2000.

[10] A.-C. Ngonga Ngomo. A time-efficient hybrid approach to link discovery. In *Proceedings of OM@ISWC*, 2011.

[11] A.-C. Ngonga Ngomo and S. Auer. Limes - a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011.

[12] A.-C. Ngonga Ngomo, J. Lehmann, S. Auer, and K. Höffner. Raven: Towards zero-configuration link discovery. In *Proceedings of OM@ISWC*, 2011.

[13] A.-C. Ngonga Ngomo and K. Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *Proceedings of ESWC*, 2012.

[14] A. Nikolov, M. D'Aquin, and E. Motta. Unsupervised learning of data linking configuration. In *Proceedings of ESWC*, 2012.

[15] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok. Transferring localization models across space. In *Proceedings of AAAI*, pages 1383–1388, 2008.

[16] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[17] G. Papadakis, E. Ioannou, C. Niederee, T. Palpanasz, and W. Nejdl. Eliminating the redundancy in blocking-based entity resolution methods. In *JCDL*, 2011.

[18] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[19] Y. Raimond, C. Sutton, and M. Sandler. Automatic interlinking of music datasets on the semantic web. In *Proceedings of the 1st Workshop about Linked Data on the Web*, 2008.

[20] F. Scharffe, Y. Liu, and C. Zhou. Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In *Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US)*, 2009.

[21] A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. Fedx: optimization techniques for federated query processing on linked data. In *Proceedings of the 10th international conference on The semantic web - Volume Part I*, ISWC'11, pages 601–616, Berlin, Heidelberg, 2011. Springer-Verlag.

[22] M. Sharma, M. P. Holmes, J. C. Santamaría, A. Irani, C. L. I. Jr., and A. Ram. Transfer learning in real-time strategy games using hybrid cbr/rl. In *IJCAI*, pages 1041–1046, 2007.

[23] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, PP(99), 2011.

[24] J. Sleeman and T. Finin. Computing foaf co-reference relations with rules and machine learning. In *Proceedings of SDoW*, 2010.

[25] C. Unger, L. Bühmann, J. Lehmann, A.-C. N. Ngomo, D. Gerber, and P. Cimiano. Sparql template-based question answering. In *Proceedings of WWW*, 2012.

[26] J. Urbani, S. Kotoulas, J. Maassen, F. van Harmelen, and H. Bal. Owl reasoning with webpie: calculating the closure of 100 billion triples. In *Proceedings of the ESWC 2010*, 2010.

[27] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *SIGIR*, pages 627–634, 2008.

[28] Q. Yang, Y. Chen, G.-R. Xue, W. Dai, and Y. Yu. Heterogeneous transfer learning for image clustering via the social web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 1–9, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[29] H. Zhuo, Q. Yang, D. H. Hu, and L. Li. Transferring knowledge from another domain for learning action models. In *Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, PRICAI '08, pages 1110–1115, Berlin, Heidelberg, 2008. Springer-Verlag.