Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Teaching Machines to Learn by Metaphors

Omer Levy and **Shaul Markovitch**
Computer Science Department
Technion – Israel Institute of Technology
32000 Haifa, Israel
{omerlevy,shaulm}@cs.technion.ac.il

Abstract

Humans have an uncanny ability to learn new concepts with very few examples. Cognitive theories have suggested that this is done by utilizing prior experience of related tasks. We propose to emulate this process in machines, by transforming new problems into old ones. These transformations are called metaphors. Obviously, the learner is not given a metaphor, but must acquire one through a learning process. We show that learning metaphors yield better results than existing transfer learning methods. Moreover, we argue that metaphors give a qualitative assessment of task relatedness.

Introduction

Despite its incredible success, machine learning still falls short of the human ability to recognize and induce new concepts from merely a few examples. Even state-of-the-art machine learning algorithms require significant amounts of data in order to learn a new non-trivial concept. Inevitably, we ask the age-old question of artificial intelligence: "How do humans do it?". Among many theories regarding the manner in which humans acquire new concepts, one has attracted particular attention within the machine learning community - *Transfer Learning* (Brown 1990). This theory claims that humans learn new concepts by relating them to old, familiar concepts, and utilizing known facts from those domains.

Many algorithms have been proposed for using existing (source) data while learning from new (target) examples. Existing state-of-the-art methods can be roughly categorized into three main approaches, depending on their assumptions (Pan and Yang 2010): *Common Inductive Bias*, inductive bias that performed well on the source should perform well on the target; *Common Instances*, certain instances of the source data can be used as examples in the target; *Common Features*, features that were discriminating in the source data should be discriminating in the target. These methods have been shown to improve the learning rate when their assumptions hold. Nevertheless, each method makes its own assumptions on the underlying relation between the source and the target, and these assumptions do not necessarily coincide. As a matter of fact, all of these assumptions are too strict to grasp a general notion of concept relatedness.

The question persists: how should we define whether two concepts are related? Though several studies have been conducted on *when* transfer learning should be used (Thrun and O'Sullivan 1996; Rosenstein et al. 2005), and some metrics for measuring relatedness between learning tasks have been proposed (Silver 1996; Eaton, desJardins, and Lane 2008; Ben-David and Borbely 2008), we still lack a *qualitative* definition of concept relatedness.

This study presents a computational framework for solving the problem of transfer learning, based on an understanding of how concepts are related to one another. The core notion of our framework is the *metaphor* - a transformation that converts one feature space into another.

One of the most striking examples of knowledge transfer in children, is their amazing ability to learn new animals from very few examples. Take a hypothetical three year-old child, for instance. Like most children his age, he can recognize horses, and classify every animal he sees as "horse" or not with superb accuracy. What if we were to take this child to the zoo, and show him a zebra for the first time in his life? It is only reasonable to assume that the child will make some sort of association between the never-before-seen zebra and his old acquaintance, the horse. A new rule for zebra classification could theoretically form in the child's mind: a zebra is a horse with stripes. In the future, the child will be able to classify zebras as accurately as he is able to classify horses.

A metaphor is a mapping of instances from a new problem (target) into instances of an old problem (source). Zebras, for example, would be mapped into horses by removing their stripes. Once an instance has been transformed from target to source, a source classifier (hypothesis) classifies it, and its result should indicate whether the original *target* instance belongs to the *target* concept. In our example, the metaphor would remove the white stripes of any given animal, and classify the result according to the horse classifier. A zebra would come out positive, while a tiger would not.

As mentioned earlier, the untackled issue of transfer learning is how to determine when two learning tasks are related. The notion of metaphors sheds new light on the very definition of concept relatedness. Instead of "measuring the distance" between two concepts, metaphors *describe the difference*; they can explain *how* concepts relate to one another. Learning a metaphor from one concept to another is, in effect, learning their difference.

Problem Definition

First, let us define the notions of domain and concept learning. A *domain* $\mathcal{D}$ is a trio $\langle X, P, f \rangle$ where X denotes a feature space, P a probability distribution over X, and $f : X \to \{0, 1\}$ a characteristic function of some subset of X. Intuitively, a real-world object is represented as a point in X, while P tells us how likely we are to encounter it. The labeling function f designates which instances belong to a special subset of X, known as the *concept*.

Let $\mathcal{D}$ be a domain. Given a loss function $\ell : \{0, 1\}^2 \to [0, 1]$, and a sample $S \subseteq X \times \{0, 1\}$ drawn from P and labeled by f, find a hypothesis $h : X \to \{0, 1\}$ for which the expected loss $E_{x \sim P} [\ell (f (x) , h (x))]$ is as small as possible. A *concept learning problem* is also called a *learning task* and is denoted by $\mathcal{T} = \langle \mathcal{D}, \ell, S \rangle$.

Let $\langle \mathcal{T}_s, \mathcal{T}_t \rangle$ be two learning tasks, *source* and *target*, respectively. The *transfer learning problem* is to solve $\mathcal{T}_t$, but instead of utilizing only target domain data, the learner can use source data as well. We will assume that the learner is better acquainted with the source task ($|S_s| \gg |S_t|$).

Metaphors: Theoretical Background

A *metaphor* is a mapping from the target learning task to the source, which preserves label and probability.

Perfect Metaphor Let $\langle X_s, P_s, f_s \rangle$, $\langle X_t, P_t, f_t \rangle$ be two domains, source and target. A function $\mu : X_t \to X_s$ is a perfect metaphor if:

1. $f_t (x_t) = f_s (\mu (x_t))$ for all $x_t \in X_t$.
2. $x_t \sim P_t \Rightarrow \mu (x_t) \sim P_s$ for all $x_t \in X_t$.

Perfect metaphors contain two very powerful assumptions. The first assumption, *label preservation*, demands that x is part of the target concept if and only if $\mu (x)$ is part of the source concept. This, however, is not enough; for a transformation to be a perfect metaphor, it must also *preserve the probability* of sampling instances. This means that a set of instances sampled from P_t must be translated into a set that is distributed by P_s. Without this criterion, a metaphor might convert target instances into unexplored regions of the source feature space, where h_s may perform poorly.

One may notice the similarity between metaphors and reductions (from complexity theory). In effect, they represent the same principle: solving problems by translating their instances into those of other, previously-solved, problems. So technically, if we were to obtain a perfect metaphor, we would be able to solve a given transfer learning problem at least as well as we can solve the *source* learning problem.

The Perfect Metaphor Theorem Let:

1. $\langle \mathcal{T}_s, \mathcal{T}_t \rangle$ be a transfer learning problem.
2. h_s be a hypothesis for $\mathcal{T}_s$ with less than ε_s loss.
3. $\mu : X_t \to X_s$ be a perfect metaphor.

Then $h_t (x) = h_s (\mu (x))$ is a hypothesis for $\mathcal{T}_t$ with less than ε_s loss.

This result is theoretically encouraging, but has little significance in practice; having a perfect metaphor at hand is very improbable. That said, obtaining an *approximated* perfect metaphor (a *metaphor*) seems more feasible. Our main theoretical result shows that even non-perfect metaphors can perform well in conjunction with a source hypothesis.

ε-Perfect Metaphor Let $\langle X_s, P_s, f_s \rangle$, $\langle X_t, P_t, f_t \rangle$ be two domains, source and target. A function $\mu : X_t \to X_s$ is an ε-perfect metaphor if:

1. $P_t (f_t (x_t) \neq f_s (\mu (x_t))) < \varepsilon_f$
2. $\|\mu (P_t) - P_s\| < \varepsilon_P$
3. $\varepsilon_f + \varepsilon_P < \varepsilon$

The Metaphor Theorem Let:

1. $\langle \mathcal{T}_s, \mathcal{T}_t \rangle$ be a transfer learning problem.
2. h_s be a hypothesis for $\mathcal{T}_s$ with less than ε_s loss.
3. $\mu : X_t \to X_s$ be an ε-perfect metaphor.

Then $h_t (x) = h_s (\mu (x))$ is a hypothesis for $\mathcal{T}_t$ with less than $\varepsilon_s + \varepsilon$ loss.

Thus, the problem of obtaining a metaphor becomes our core focus. Given a transfer learning problem $\langle \mathcal{T}_s, \mathcal{T}_t \rangle$, a *metaphor learning problem* is to find an ε-perfect metaphor such that ε is as small as possible. According to the Metaphor Theorem, we can solve a transfer learning problem by composing a learnt metaphor with a previously learnt hypothesis: $h_t = \mu \circ h_s$. We have now transformed the *transfer* learning problem into a *metaphor* learning problem.

So why should learning a metaphor be any easier than simply learning the concept? Because metaphors represent the *difference* between concepts. Each concept may be overwhelmingly intricate by itself, but fairly easy to explain using an already known concept. A fundamental assumption of this research is that if two concepts are closely enough related, the associated metaphor will be a relatively simple function, and therefore, considerably easier to learn than the entire target concept.

How to Learn Metaphors

The common approach in many machine learning scenarios is to select an appropriate hypothesis space and search it for the best hypothesis with respect to some utility function. We present a similar framework for learning metaphors.

Metaphor Spaces

Metaphor spaces define the family of possible transformations. This is a key ingredient when learning metaphors; they must be generic enough to capture the relation between the target and source concepts. On the other hand, metaphor spaces must have a limited amount of degrees of freedom to render them learnable from small target samples.

Metaphor spaces are also a means of inserting representation-specific bias. For example, if the transfer learning problem is that of image recognition, optical manipulations (such as rotation) may be used. Other representations, such as text, will have no use for optical manipulations, but may have their own specific metaphors. Below are a few examples of metaphor spaces.

Orthogonal Linear Transformations ($\mathcal{M}_{lin}$) Perform a linear transformation on each feature independently.

Orthogonal Polynomial Transformations ($\mathcal{M}_{pol(n)}$) Perform a polynomial transformation on each feature independently. Divided into sub-spaces by degree.

Feature Reordering ($\mathcal{M}_{ord}$) Re-order features, reassigning the values of each feature. For example, rearranging pixels in a bitmap image is a feature reordering. Another example is word-by-word translation from two different bag-of-words representations.

Linear Transformations ($\mathcal{M}_{mat}$) Generate new feature spaces by applying matrix multiplication.

Geometric Transformations ($\mathcal{M}_{geo}$) Perform geometric manipulations based on rotation, scaling, translation, and reflection. Using the family of geometric metaphors assumes that the data represents images.

Note that some of these metaphor spaces contain others. For example, $\forall p\colon \mathcal{M}_{pol(p)} \subset \mathcal{M}_{pol(p+1)}$. While larger metaphor spaces increase our descriptive power, they may also hinder our ability to generalize by over-fitting.

Metaphor Evaluation

The Metaphor Theorem dictates that a good metaphor adheres to two criteria: label and distribution preservation. To assure label preservation, we would like to minimize $P_t\left(f_t\left(x\right) \neq f_s\left(\mu\left(x\right)\right)\right)$. This value can be estimated by the empirical error over the target training set. For regression problems, mean square error (MSE) is a fine estimate.

Distribution preservation demands that we minimize the distance between $\mu\left(P_t\right)$ and P_s. The statistical distance between two samples has many empirical estimates. We will use the method of moments (Hansen 1982) to estimate distribution parameters and measure the statistical distance by comparing these parameters. This metric is easily computable, and has strong analytical properties.

Combining these two metrics by weighted sum is problematic, since label preservation and statistical distance may have entirely different scales, rendering their sum meaningless. Instead, we propose a different strategy for combining label and distribution preservation: calculate the statistical distance *per class*. In other words, positive target instances are compared only to positive source instances. In the case of a binary class:

$$SD\left(S_t, S_s\right) = SD\left(S_t^+, S_s^+\right) + SD\left(S_t^-, S_s^-\right)$$

where SD is the statistical distance metric and the sign notation indicates that only instances of that class are considered. This metric (*the metaphor heuristic*) can easily be generalized to accommodate multiple classes, and combined with binning techniques for regression problems.

Algorithms for Metaphor Learning

Given a metaphor space $\mathcal{M}$, we can use the metaphor heuristic to search for the most suitable metaphor $\mu \in \mathcal{M}$. This is a de-facto optimization problem, where search algorithms from the hill-climbing/gradient-descent family can be used. While these algorithms have proven themselves empirically across many domains, we can actually tailor efficient algorithms to certain metaphor spaces, by using the analytical properties of the heuristic. For example, finding the best *orthogonal linear transformation* can be reduced to n different optimization problems, one for each feature, which are solvable by partial derivatives. Another example is *feature reordering* metaphors, which can be described as an assignment problem with weighted edges (costs). Polynomial-time solutions such as the Hungarian algorithm (Kuhn 1955) may be used to find the best feature reordering that minimizes the metaphor heuristic.

Automatic Selection of Metaphor Spaces

Selecting a suitable metaphor space is critical to the learner's success. Alas, matching a metaphor space to a given problem is not a trivial task. For the metaphor framework to be truly robust, we require a method of selecting a metaphor space - that fits the problem at hand - from the arsenal of available spaces.

Given multiple metaphor spaces $\mathcal{M}_1, ..., \mathcal{M}_m$, we will require that they be sorted by preference. Informally, this preference relation will be called complexity, and its goal is to bias our choice of metaphor space towards simpler spaces, coinciding with Occam's Razor. While there are many definitions of complexity, our algorithm does not require that the ordering be dependent on one specific metric or another.

The algorithm learns the best metaphor from each space, with respect to the metaphor heuristic, and evaluates each metaphor's accuracy on the target sample. However, the selected metaphor is not chosen by maximal accuracy alone, as that may result in over-fitting. Instead, a pairwise comparison of metaphors using McNemar's test (1947) is conducted; metaphors that originated in complex spaces are preferred to simpler metaphors only if they are significantly better, beyond a predetermined significance parameter α. Hence, the selection algorithm will only select a complex metaphor if it is significantly better than its simpler counterparts.

Accuracy was chosen as a means of evaluation to avoid over-fitting. Had we used the metaphor heuristic instead, we would be giving an unfair advantage to complex metaphor spaces, which have more degrees of freedom.

Algorithm *Metaphor Space Selection Algorithm*

Input: source data S_s, target data S_t, source hypothesis h_s, the metaphor heuristic SD, list of metaphor spaces $\mathcal{M}_1, ..., \mathcal{M}_m$, significance α.

Output: The simplest metaphor that classifies significantly better.

1. **for** each $\mathcal{M}_i$: $\mu_i = \underset{\mu \in \mathcal{M}_i}{\operatorname{argmin}} SD\left(\mu\left(S_t\right), S_s\right)$
2. $\mu = \mu_1$
3. **for** $i = 2$ **to** n
4. **if** $\mu_i \circ h_s$ classifies S_t better than $\mu \circ h_s$ with α significance: $\mu = \mu_i$
5. **return** μ

Empirical Evaluation

Though our theoretical results are encouraging, the algorithms presented in the previous section are of a heuristic nature, and must be evaluated empirically.

Methodology

Protocol An excellent method of determining how well an algorithm performs with small sample sizes is by observ-

ing its learning curve. In our setting, the size parameter will affect only the amount of target instances available to the algorithm. The amount of source instances will remain constant at a large number (1000), since we are always under the core assumption of metaphor learning: the source concept is well-known and can be classified with small error.

Performing classic cross-validation is insufficient, because the minimal target training set is half of the original set. Since we are interested in understanding how metaphor learning performs under very small sample sizes, a variation was used. In essence, we partition the pool of target instances into chunks of size $2n$, and perform two-fold cross-validation on each chunk. This gives us an amount of results that is double the number of chunks, and can then be used for testing statistical significance. Apart from the protocol's versatility in terms of training set size, it also ensures that each instance is trained upon once and tested upon once (exactly). This protocol was used in every experiment, with a source dataset of 1000 instances and a pool of 600 instances from the target task (with the exception of Latin and Cyrillic, which has 100 target instances).

A variety of base learners was reviewed for each domain, including SVM, C4.5, Naive Bayes, and Nearest Neighbor. We used WEKA's implementation for these algorithms with their default parameters (Hall et al. 2009). Results based on the Nearest Neighbor learner will be presented. Similar behavior was observed across base classifiers.

Transfer Learning Tasks Transfer learning tasks are composed of two samples from the source and target domains. These domains must be *related*; there is no sense in trying to learn one concept based on another when there is no clear relation between the two. To truly evaluate algorithms in transfer learning conditions, the domains must also be significantly *different* from one another. Finally, we will also assume that all features (excluding the class feature) are numerical. While we have also looked into metaphors for nominal features, numerical features simplify the experimental process due to the continuous nature of many metaphor spaces. Summing up, we would like our transfer learning tasks to be *related*, *different*, and *numerical*.

The research literature lists only few transfer learning tasks. However, even these few do not satisfy the above criteria. In addition, we would like to examine transfer learning tasks in which the source and target feature spaces differ, but unfortunately, were unable to find such datasets. Therefore, we present several new transfer learning tasks.

Negative Image The source task is that of the optical digit recognition dataset from the UCI ML Repository (Frank and Asuncion 2010). While the source data consisted of black ink on a white background, the target data is inverted - white on black.

Higher Resolution This scenario simulates the case where we have much data from a low resolution camera, and little data of high resolution. The target data will be the original digits dataset. To create the lower resolution source data, we will merge each two-by-two quad of pixels into one pixel, where its intensity is their average. Note that the source feature space consists of only 16 dimensions, while the target feature space has 64.

Latin and Cyrillic The source data contains images of typeface uppercase Latin characters where only those depicting the letter 'R' are labeled positive. Similarly, the target consists of uppercase Cyrillic letters in which 'Я' is labeled positive and any other letter negative.

Wine The wine quality prediction task (Cortez et al. 2009) is a regression problem that consists of two distinct datasets: red wine (source) and white wine (target).

Reference Methods Three baseline methods were compared against metaphor learning algorithms: *Target Only* learns only from the target dataset; *Identity Metaphor* learns only from the source dataset; *Merge* learns from the union of source and target datasets. An additional three state-of-the-art algorithms were compared: *Frustratingly Easy Domain Adaptation (FEDA)* (Daumé III 2007); *Multi-Task Learning (MTL)* (Caruana 1997) using an implementation by (Tu, Fowler, and Silver 2010); *TrAdaBoost* (Dai et al. 2007). Besides *Target Only*, all methods are applicable only when both source and target feature spaces are identical.

Performance of Metaphor Learning

We will show that when we are given a suitable metaphor space, heuristic search finds a good metaphor that describes this relation. Moreover, we shall demonstrate that classifiers based on these metaphors are more accurate than classifiers generated by other transfer learning algorithms.

Metaphor Space Selection To meet the assumption that our metaphor space $\mathcal{M}$ contains an adequate metaphor, we must select one that is generic enough on one hand, yet specific enough to enable some form of bias. We will later relax the specificity demand when discussing automatic selection of metaphor spaces.

In the negative images task, each pixel in the target domain relates to a corresponding pixel in the source; therefore, orthogonal linear transformations are a good choice. The same metaphor space was used for wine, which also requires feature alignment. Intuitively, the Latin and Cyrillic tasks are related by mirroring, so the family of geometric transformations should be suitable. Changes in resolution require metaphors that can handle different source and target feature spaces: non-orthogonal linear transformations.

Where applicable ($\mathcal{M}_{pol}$ and $\mathcal{M}_{ord}$), we used analytical methods to find the best metaphor. Steepest-ascent hill-climbing was used to search the remaining metaphor spaces.

The previous section described $\mathcal{M}_{geo}$ in a manner that is too abstract to reconstruct, so we shall therefore elaborate. The base transformations are: 3 rotations ($90°, 180°, 270°$), 16 translations (8 horizontal, 8 vertical), and 2 reflections (horizontal axis, vertical axis). This space is closed under composition, allowing a combination of several base transformations to be a metaphor in $\mathcal{M}_{geo}$. The hill-climbing algorithm starts with no transformations (the identity metaphor) and fuses base transformations with the current state until a local minimum has been reached. This state (metaphor) is eventually returned.

Task	Instance	Sample Size			
		1	2	5	10
Negative Image					
Higher Resolution					

Table 1: Metaphor invocation across target sample sizes

Quantitative Results Figure 1 shows the performance of each method on each transfer learning task, as a function of the target sample size. Metaphors dominated all other methods with 95% significance in nearly all sample sizes. Nevertheless, with sufficient amounts of target data, metaphors will eventually fail to provide better classification than the target-only method, as can be seen in figure 1(d). This was also observed among the other datasets when the target sample size was substantially larger than 20. The phenomenon coincides with human cognition; when learning some concept for the first time, we will attempt to project it onto another concept, but eventually, we will become experts from studying the relevant data alone.

Another interesting observation is that state-of-the-art methods did not perform better than baseline methods. While it has been demonstrated in previous work that these methods perform well, this claim does not hold when the data does not meet the methods' underlying assumptions.

Qualitative Analysis A closer look at the actual metaphors that were found should provide a broader understanding of how metaphors work. An arbitrary target instance was selected from *Negative Image* and *Higher Resolution*, on which we invoked actual metaphors that were found for different sample sizes. Table 1 shows the input (target instance) and the outputs (translated instances).

It can be observed that even a few target instances are enough to learn a good metaphor. With only two examples, the outputs already resemble the source data, and five are sufficient for nearly perfect classification. The learning curve is most visible with the *Negative Image* task, where one target example creates an unintelligible image, two already form the general shape of 6, and five remove any shadow of a doubt regarding the digit's class.

Performance Across Base Classifiers We repeated the experiments across three other base classifiers: C4.5, Naive Bayes, and Linear SVM. Changing the source classifier did not have a significant effect on the metaphor learner's performance. Metaphors are indifferent to the base classifier's type because the metaphor heuristic is *independent* of the base classifier; the same metaphors will be selected by the algorithm, regardless of the base classifier. As long as each of the base classifier's inductive bias is general enough to capture the source concept with small error, classification by metaphors will display the same performance across base classifiers.

Sample Size	$\mathcal{M}_{geo}$	$\mathcal{M}_{ord}$	$\mathcal{M}_{lin}$	$\mathcal{M}_{pol(2)}$
1	100%	0%	0%	0%
2	100%	0%	0%	0%
3	100%	0%	0%	0%
4	71.3%	5.3%	20.7%	2.7%
5	25.8%	11.7%	55%	7.5%
6	13%	14%	71%	2%
7	2.4%	9.5%	85.7%	2.4%
8	1.4%	10.8%	86.4%	1.4%
9	0%	6.1%	87.8%	6.1%
10	0%	8.3%	88.4%	3.3%
11	0%	1.9%	98.1%	0%
12	0%	4%	96%	0%
13	0%	0%	100%	0%

Table 2: Metaphor space selection (*Negative Image*)

Automatic Selection of Metaphor Spaces

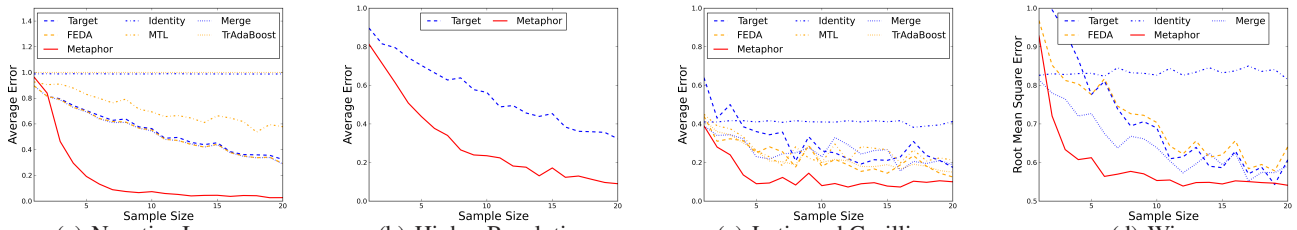
After evaluating the metaphor framework under the assumption of a single metaphor space, we shall proceed to generalize this setting by testing the Metaphor Space Selection Algorithm. We selected two optical recognition tasks (*Negative Image*; *Latin and Cyrillic*) and four metaphor spaces that were applicable ($\mathcal{M}_{geo}$; $\mathcal{M}_{ord}$; $\mathcal{M}_{lin}$; $\mathcal{M}_{pol(2)}$). The metaphor spaces were given in order of complexity, and a significance threshold of $\alpha = 90\%$ was used.

In *Latin and Cyrillic*, $\mathcal{M}_{geo}$ was selected at every fold. Not only did the space of geometric manipulations benefit from the algorithm's bias, it was also the best metaphor space for the task at hand. The *Negative Image* task was not as simple, since geometric manipulations do not describe the negative image relation. The space of orthogonal linear transformations ($\mathcal{M}_{lin}$), however, proves as a significantly better metaphor space as the number of target examples grows. Note that while three examples or less do not provide 90% significance in McNemar's test, five target examples are enough to convince the algorithm to select the $\mathcal{M}_{lin}$ more than half of the time. Table 2 shows the portion of folds in which each metaphor space was selected, by target sample size, in the *Negative Image* task.

These results show quick convergence into the best metaphor space, even when the algorithm is strongly biased towards simple spaces. The Metaphor Space Selection Algorithm's performance in classification also suggests that the algorithm converges into the right metaphor space within a small amount of target examples.

Related Work

The problem of learning from few examples has three main approaches, all of which add additional information beyond the original examples; however, they differ by the *type* of that information. Explanation-based learning (DeJong and Mooney 1986; Mitchell, Keller, and Kedar-Cabelli 1986) claims that by relying on known rules (axioms), one can logically deduce a hypothesis that *explains* an observation. Thus, a learner equipped with enough axioms can grasp entire concepts from a single observation. Semi-supervised learning (Board and Pitt 1989; Blum and Mitchell 1998) assumes that in addition to a few labeled examples, the learner



(a) Negative Image (b) Higher Resolution (c) Latin and Cyrillic (d) Wine

Figure 1: Comparison of transfer learning methods as function of target sample size

is given many *unlabeled* examples. Understanding the data from these unlabeled examples may assist the learner in forming a better hypothesis. The transfer learning setting incorporates prior knowledge as an additional dataset of a related concept. Metaphors fall into this category.

Many methods of transfer learning have been suggested. An interesting conclusion of Pratt's work (1991) was that the same inductive bias performed well on related tasks. This paved the way to additional methods (Thrun and Mitchell 1995; Ando and Zhang 2005; Eaton, desJardins, and Lane 2008; Ruckert and Kramer 2008) that used the parameters of a source classifier as an inductive bias for learning the target. An additional approach that assumes that certain instances of the source data can be used as examples in the target, has been demonstrated on SVMs (Wu and Dietterich 2004) and in a Bayesian framework (Daumé III and Marcu 2006). A series of boosting algorithms (Rettinger, Zinkevich, and Bowling 2006; Dai et al. 2007) have also proven to be effective when this assumption holds. A third approach harnesses discriminating features that are common to all tasks (Caruana 1997; Daumé III 2007; Raina et al. 2007; Pan, Kwok, and Yang 2008).

Metaphors do not assume that the source and target have anything in common, but rather that a transformation function from one to another exists. In this sense, metaphors differ dramatically from previous methods.

The notion of metaphors and analogies in human cognition had been studied by Gentner throughout the 80's (Gentner 1983). At the same time, Analogical Reasoning (also called Computational Metaphors) was developed by Carbonell and others (Carbonell 1981; Holyoak and Thagard 1989). Their prime focus was to use prior knowledge to assist reasoning tasks (such as logical deduction and planning) in new domains. In inductive learning, an instance-based method that incorporates analogical reasoning was recently introduced (Miclet, Bayoudh, and Delhay 2008).

While there are some similarities between metaphors and previous work in Case-Based Reasoning (CBR) (Riesbeck and Schank 1989), it is important to notice the fundamental differences. While CBR maps target problems to *previously observed* source problems, metaphors may translate target instances into never-before-seen instances in the source feature space. Metaphors are not necessarily similarity-based (as in CBR), and do not even require the source and target feature spaces to be identical. Another disparity is that CBR retrieves a different set of source problems for each given target problem, while metaphors translate the entire target feature space.

Discussion

We presented a novel transfer learning approach, inspired by human cognition: metaphors. The Metaphor Theorem shows that if two concepts are related by metaphor, the new (target) concept can be classified as accurately as the original (source) concept. Metaphor spaces and their automatic selection, alongside the metaphor heuristic and our efficient search methods, provide a robust toolbox for learning metaphors. These tools were tried and tested in a real transfer learning setting, and performed better than state-of-the-art transfer learning methods.

When the target and source concepts are not related, metaphors do not work; neither will they perform well in a scenario where the concepts are too distant for a simple metaphor to describe their relation. If translating from target to source requires a sophisticated metaphor, we might as well learn the target *without* using the source at all. Even humans are sometimes required to learn entirely foreign concepts, and in these particular situations, tabula rasa is the only way to go. We can therefore conclude that if a clear relation between target and source does not exist, the problem at hand does *not* fit the definition of a transfer learning task.

However, when such a relation exists, metaphors double our profit. Not only are we rewarded with better classification, we are also provided with an explanation as to *how* the new concept relates to the old. Metaphors provide a unique assessment of task relatedness - a qualitative difference rather than a numerical measure.

Since relatedness between concepts may take on many forms, the selection of a suitable metaphor space is critical. Selecting an appropriate metaphor space is not a trivial choice, and one may question the amount of engineering involved in this process. For this precise reason, we have designed and tested the automatic Metaphor Space Selection Algorithm. As demonstrated, the algorithm is able to select the best metaphor space from a variety of spaces after observing a very small amount of examples. Nevertheless, we must re-ask the question at a higher level: how does one go about selecting an adequate kernel function, hypothesis space, or even feature space? Much research has been devoted to answering these questions, and their answers should easily be applicable to metaphor spaces without prejudice.

References

Ando, R. K., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR* 6:1817–1853.

Ben-David, S., and Borbely, R. S. 2008. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning* 73(3):273–287.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT '98*, 92–100. New York: ACM Press.

Board, R. A., and Pitt, L. 1989. Semi-supervised learning. *Machine Learning* 4:41–65.

Brown, A. L. 1990. Domain-specific principles affect learning and transfer in children. *Cognitive Science* 14(1):107–133.

Carbonell, J. G. 1981. A computational model of analogical problem solving. In Hayes, P. J., ed., *IJCAI '81*, 147–152. William Kaufmann.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* 47(4):547–553.

Dai, W.; Yang, Q.; Xue, G.-R.; and Yu, Y. 2007. Boosting for transfer learning. In Ghahramani, Z., ed., *ICML 2007*, volume 227 of *ACM International Conference Proceeding Series*, 193–200. ACM.

Daumé III, H., and Marcu, D. 2006. Domain adaptation for statistical classifiers. *JAIR* 26:101–126.

Daumé III, H. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

DeJong, G., and Mooney, R. J. 1986. Explanation-based learning: An alternative view. *Machine Learning* 1(2):145–176.

Eaton, E.; desJardins, M.; and Lane, T. 2008. Modeling transfer relationships between learning tasks for improved inductive transfer. In Daelemans, W.; Goethals, B.; and Morik, K., eds., *ECML/PKDD 2008*, volume 5211 of *Lecture Notes in Computer Science*, 317–332. Springer.

Frank, A., and Asuncion, A. 2010. UCI machine learning repository.

Gentner, D. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science* 7(2):155–170.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: an update. *SIGKDD Explorations* 11(1):10–18.

Hansen, L. 1982. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society* 1029–1054.

Holyoak, K. J., and Thagard, P. R. 1989. *A Computational Model of Analogical Problem Solving*. New York, NY, USA: Cambridge University Press. 242–266.

Kuhn, H. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1-2):83–97.

McNemar, Q. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12(2):153–157.

Miclet, L.; Bayoudh, S.; and Delhay, A. 2008. Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. *JAIR* 32:793–824.

Mitchell, T. M.; Keller, R. M.; and Kedar-Cabelli, S. T. 1986. Explanation-based generalization: A unifying view. *Machine Learning* 1(1):47–80.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng* 22(10):1345–1359.

Pan, S. J.; Kwok, J. T.; and Yang, Q. 2008. Transfer learning via dimensionality reduction. In Fox, D., and Gomes, C. P., eds., *AAAI 2008*, 677–682. AAAI Press.

Pratt, L. Y.; Mostow, J.; and Kamm, C. A. 1991. Direct transfer of learned information among neural networks. In *AAAI '91*, 584–589.

Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. Y. 2007. Self-taught learning: transfer learning from unlabeled data. In Ghahramani, Z., ed., *ICML 2007*, volume 227 of *ACM International Conference Proceeding Series*, 759–766. ACM.

Rettinger, A.; Zinkevich, M.; and Bowling, M. 2006. Boosting Expert Ensembles for Rapid Concept Recall. In *AAAI 2006*, volume 21, 464.

Riesbeck, C. K., and Schank, R. C. 1989. *Inside Case-Based Reasoning*. Hillsdale, N.J.: Lawrence Erlbaum Assoc.

Rosenstein, M.; Marx, Z.; Kaelbling, L.; and Dietterich, T. 2005. To transfer or not to transfer. In *NIPS 2005 Workshop, Inductive Transfer: 10 Years Later*.

Ruckert, U., and Kramer, S. 2008. Kernel-based inductive transfer. In Daelemans, W.; Goethals, B.; and Morik, K., eds., *ECML/PKDD 2008*, volume 5212 of *Lecture Notes in Computer Science*, 220–233. Springer.

Silver, D. L. 1996. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. *Connection Science* 8(2):277–294.

Thrun, S., and Mitchell, T. M. 1995. Learning one more thing. In *IJCAI '95*, 1217–1225.

Thrun, S., and O'Sullivan, J. 1996. Discovering structure in multiple learning tasks: The TC algorithm. In *ICML*, 489–497.

Tu, L.; Fowler, B.; and Silver, D. L. 2010. CsMTL MLP for WEKA: Neural network learning with inductive transfer. In Guesgen, H. W., and Murray, R. C., eds., *FLAIRS-23*. AAAI Press.

Wu, P., and Dietterich, T. G. 2004. Improving SVM accuracy by training on auxiliary data sources. In Brodley, C. E., ed., *ICML 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM.