

Measuring the Feasibility of Analogical Transfer using Complexity

Pierre-Alexandre Murena^{1,†}

¹*Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Espoo, Finland*

Abstract

Analogies are 4-ary relations of the form “A is to B as C is to D”. While focus has been mostly on how to solve an analogy, i.e. how to find correct values of D given A, B and C, less attention has been drawn on whether solving such an analogy was actually feasible. In this paper, we propose a quantification of the transferability of a source case (A and B) to solve a target problem C. This quantification is based on a complexity minimization principle which has been demonstrated to be efficient for solving analogies. We illustrate these notions on morphological analogies and show its connections with machine learning, and in particular with Unsupervised Domain Adaptation.

Keywords

Analogical reasoning, Analogical transfer, Minimum Message Length, Domain adaptation

1. Introduction

Analogies are 4-ary relations of the form “A is to B as C is to D”, denoted $A : B :: C : D$. Even though humans demonstrate strong capabilities of understanding and generating analogies, which has been intensively studied by cognitive sciences [1], these tasks are much more difficult for a machine. In particular, an important task consists in solving analogical equations: given A, B and C , find D such that $A : B :: C : D$ is a valid analogy. Solving such equations has been investigated in multiple domains: Boolean domains [2], formal concepts [3], structured character strings [4], semantic [5, 6] or morphological tasks [7].

In most cases, the aforementioned methods are designed to provide an answer to any equation $A : B :: C : x$, regardless of whether the equation makes sense. This is not the case for humans, who consider that some analogies make more sense than others. Consider for instance the domain of Hofstadter analogies [4]. It describes analogies between character strings, with strong domain constraints relative to the order of the alphabet. For instance, the analogy “ABC : ABD :: IJK : IJL”, which is a typical illustrative example of this domain, is based on the notions of *increment* and *last element*. Intuitively, not all analogical equations have a solution in this domain: for instance, it is difficult for a human to find a satisfying solution to the equation “ABC : HIC :: BFQ : x”. This is confirmed by the results of the user study conducted by Murena et al. [8].

IARML@IJCAI-ECAI'2022: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI-ECAI'2022, July, 2022, Vienna, Austria

✉ pierre-alexandre.murena@aalto.fi (P. Murena)

🌐 <http://pamurena.com/> (P. Murena)

🆔 0000-0003-4586-9511 (P. Murena)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Which analogical equations are indeed solvable, is a rather infrequently discussed question. However, it would have strong implications, be it in practical applications of analogies (e.g. in intelligent tutoring systems [9]) or in transfer learning. In this paper, we propose a first step toward this important direction, by considering how to measure the transferability of a source case (A, B) to a target case (C, D) . To do so, we propose a very general formalization of the problem, which is shown to apply to transfer in both symbolic tasks (like morphological analogies) and numerical machine learning. Based on this formalization, and getting inspiration from applications of Kolmogorov complexity in inference [10], we propose several potential definitions of transferability and discuss their main properties.

2. Preliminary Notions

2.1. Domains and Model Spaces

Although various definitions of analogy have been proposed in the literature, we consider in this paper the following definitions, inspired by the recent framework of Antic [11].

A domain \mathcal{D} is defined as the product of two spaces \mathcal{X} (the *problem space*) and \mathcal{Y} (the *solution space*). An element $(x, y) \in \mathcal{D}$ will be referred to as a *case*.

We introduce a set \mathcal{R} called *representation space*. A model space $\mathbb{M}_{\mathcal{R}, \mathcal{D}}$ of domain \mathcal{D} based on representation \mathcal{R} is defined as a subset $\mathbb{M}_{\mathcal{R}, \mathcal{D}} \subseteq \{f : \mathcal{X} \times \mathcal{Y} \times \mathcal{R} \rightarrow [0, 1]\}$ of functions mapping a problem $x \in \mathcal{X}$, a solution $y \in \mathcal{Y}$ and a representation $r \in \mathcal{R}$, to a real number. Any $M \in \mathbb{M}_{\mathcal{R}, \mathcal{D}}$ is called a model of \mathcal{D} . When the context is clear, we will use the notation \mathbb{M} instead of $\mathbb{M}_{\mathcal{R}, \mathcal{D}}$.

We illustrate these notions with two examples that will be further investigated in Sections 4 and 5.

Example 1 (Recursive model for morphology). *Given an alphabet \mathcal{A} , we define by \mathcal{A}^* the set of words of \mathcal{A} . The morphological domain consists of two forms of a word (e.g. declension of a word or conjugation of a verb in natural language): therefore, it is given by $\mathcal{D} = \mathcal{A}^* \times \mathcal{A}^*$.*

We define the representation space $\mathcal{R} = \bigcup_{n=1}^{\infty} (\mathcal{A}^)^n$. Let \mathcal{F} be the space of all recursive functions¹ from \mathcal{R} to \mathcal{D} . For all $\phi \in \mathcal{F}$, we define $M_{\phi} : \mathcal{A}^* \times \mathcal{A}^* \times \mathcal{R} \rightarrow \{0, 1\}$ as:*

$$M_{\phi}(x, y, r) = \begin{cases} 1 & \text{if } \phi(r) = (x, y) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We can then define the model space $\mathbb{M}_{\mathcal{R}, \mathcal{D}}$ as:

$$\mathbb{M}_{\mathcal{R}, \mathcal{D}} = \{M_{\phi} | \phi \in \mathcal{F}\} \quad (2)$$

This example has an easy interpretation. The morphological domain consists of two words w_1 and w_2 , which are typically two flections of a same word. For instance, the tuple “play : played” describes the flection of the English verb “play” from the present tense to the perfect tense. Similarly, the tuple “taloon : talossa” describes the flection of the Finnish noun “talo” from the illative case to the inessive case.

¹i.e. functions computable by a Turing machine. See Section 2.2.

For a better readability, we decompose the recursive function $\phi \in \mathcal{F}$ as $\phi(r) = (\phi_1(r), \phi_2(r))$. The function ϕ_1 (resp. ϕ_2) describes how the word w_1 (resp. w_2) is formed based on some representation r . For instance, in the “play : played” example, we can have $\phi_1(r) = r$, $\phi_2(r) = r + \text{“ed”}$ (where the $+$ operation is the string concatenation), and both functions are instantiated with $r = \text{“play”}$.

Example 2 (Probabilistic models on \mathbb{R}^d). We define the binary domain $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ with problem space $\mathcal{X} = (\mathbb{R}^d)^*$ and solution space $\mathcal{Y} = \mathcal{L}^*$, for some space \mathcal{L} of labels. When \mathcal{L} is discrete, the problem is called classification, otherwise regression. An observation on domain \mathcal{D} consists of one labelled dataset, where the problem is the unlabeled dataset (points in \mathbb{R}^d) and the labels (in \mathcal{L}).

Let \mathcal{P} be a set $\mathcal{P}_\Theta = \{p_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1] \mid \theta \in \Theta\}$ of probability density functions on \mathcal{D} parameterized by $\theta \in \Theta$. Fixing $\mathcal{R} = \emptyset$ and therefore identifying $M(x, y, r)$ to $f(x, y)$, we can define the model space $\mathbb{M}_{\mathcal{R}, \mathcal{D}} = \mathcal{P}_\Theta$.

In this paper, we assume that all considered quantities are computable. For those which are not computable (for instance domains of real numbers used in Example 1), we will introduce relevant computable approximations.

2.2. Kolmogorov Complexity

Our framework relies on the use of Kolmogorov complexity [10]. We propose a gentle introduction to this notion. In the following, we will use the notation \mathbb{B} to designate the binary set $\{0, 1\}$.

A function $\phi : \mathbb{B}^* \rightarrow \mathbb{B}^*$ is called partial recursive if its output $\phi(p)$ corresponds to the output of a given Turing machine after its execution with input p when it halts (otherwise, we use the convention $\phi(p) = \infty$). With this notation, the function ϕ can be improperly likened to a Turing machine. In this case, the input p is called a *program*. A partial recursive function ϕ is called *prefix* if, for all $p, q \in \mathbb{B}^*$, if $\phi(p) < \infty$ and $\phi(q) < \infty$, then p is not a proper prefix of q .

Complexity $K_\phi(x)$ of a string $x \in \mathbb{B}^*$, relative to a partial recursive prefix (p.r.p.) function ϕ , is defined as the length of the shortest string p such that $\phi(p) = x$:

$$K_\phi(x) = \min_{p \in \mathbb{B}^*} \{l(p) : \phi(p) = x\} \quad (3)$$

where $l(p)$ represents the length of the string p .

A key result of the theory of complexity is the existence of an additively optimal p.r.p. function ϕ_0 : for any p.r.p. function ϕ , there exists a constant c_ϕ such that for all $x \in \mathbb{B}^*$, $K_{\phi_0}(x) \leq K_\phi(x) + c_\phi$. These additively optimal p.r.p. functions are used to define Kolmogorov complexity. They present in particular invariance properties, which means that the difference between the complexities defined by two distinct universal p.r.p. functions is bounded. However, it can be shown that Kolmogorov complexity is not computable, and thus cannot be used in practice.

In practice, this limitation is overcome by fixing a reference p.r.p. function which is not optimal but leads to a computable complexity. By definition, this non-optimal complexity is an upper-bound of Kolmogorov complexity (up to an additive constant). This choice of a reference function is particularly restrictive and imposes some biases, which is inherent to any

inductive problem. In the following, we will refer to this upper-bound either as complexity or as description length.

2.3. Inference on a Single Domain

Consider a domain $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ and an observation $(x, y) \in \mathcal{D}$. Given a model space $\mathbb{M}_{\mathcal{D}}$, the standard *inference* task of supervised learning is to select a model $M \in \mathbb{M}_{\mathcal{D}}$ which optimally describes the observation.

Selecting a model which accurately describes the observation is not enough in general: In most situations there exists multiple such models, and sometimes even infinitely-many. It is then necessary to discriminate among all these models, in particular based on the intended use of the model. In statistical learning for instance, the discrimination is done by both the selection of a *simple* model family and a *penalization* of models [12].

The inference of the model describing an observed case can then be split into two aspects: selection of a model accurately describing the case and penalization over the model space. This idea has been formalized by Algorithmic Information Theory using Kolmogorov complexity. The *Minimum Message Length* (MML) [13] and the *crude Minimum Description Length* (MDL) [14] principles both formulate the general inference task over a domain as the following minimization problem:

$$\underset{M \in \mathbb{M}_{\mathcal{D}}}{\text{minimize}} \quad K(M) + K(x, y|M) \quad (4)$$

This two-part objective function corresponds to the trade-off between the accuracy of the model and its simplicity. For instance, in the morphological domain, it is possible to define a model accounting for all possible valid transformations. By construction, this model will have perfect accuracy, but it will require to encode every pair of problems and solutions in the language, and therefore will be particularly complex.

We remind the computation of complexity is relative to a choice of a reference Turing machine. Therefore, this choice imposes a strong bias over the intended outcome. A *refined* version of the MDL principle has been proposed, which overcomes this limitation. This version is beyond the scope of this paper, but we refer the interested reader to (Grünwald, 2007) [15].

3. A Definition of Transferability

Analogies involve two separate domains: a *source domain* \mathcal{D}^S and a *target domain* \mathcal{D}^T . Given a source problem-solution pair $(x^S, y^S) \in \mathcal{D}^S$ and a target problem $x^T \in \mathcal{X}^T$, the analogical transfer from (x^S, y^S) to x^T is informally defined as finding $y^T \in \mathcal{Y}^T$ such that the transformation $x^S \mapsto y^S$ is “similar” to the transformation $x^T \mapsto y^T$. This notion of similarity is problematic, especially when the source and target domains are distinct. Existing frameworks of analogy often assume that the source and target domains are the same [2], or at least share a common structure (e.g. are L -algebras of a same language L , such as proposed by (Antic, 2020) [11]).

In this section, we show how complexity can be used to properly define this similarity, even in the case of distinct domains. We will then show that this definition helps quantifying the

notion of *transferability*, i.e. how the source observation (x^S, y^S) is useful to find a solution to target problem x^T .

3.1. Inference of a Target Model

The task in the target domain consists in predicting the solution $y^T \in \mathcal{Y}^T$ associated to the problem $x^T \in \mathcal{X}^T$. Usually, this is done throughout a model M , in particular by finding $y^T \in \mathcal{Y}^T$ and $r \in \mathcal{R}$ maximizing the score $M(x^T, y^T, r)$:

$$y^{T*} \in \arg \max_{y \in \mathcal{Y}^T} \left\{ \max_{r \in \mathcal{R}} M(x^T, y, r) \right\} \quad (5)$$

In the context of Example 1, this corresponds to choosing a representation r that successfully describes x^T given the recursive function ϕ associated to M , i.e. finding r such that $\phi_1(r) = x^T$. The solution y^T is then estimated by taking $y^T = \phi_2(r)$. In the context of Example 2, Equation (5) corresponds to taking the most probable solution.

However, in practice, the model M is not known and needs to be inferred. The difficulty is that, in general, it is not possible to identify M given x^T only. Even worse, there is no guarantee that two models correctly describing x^T can yield the same values of y^T . For instance, in the morphological domain (Example 1), one can build degenerate functions ϕ such that $\phi(r) = x^T$ for all r . All such functions successfully describe x^T and yield all possible values for y^T .

The MML principle presented in Equation 5 is meant to be used in a single domain, in particular the source domain. We propose to use it as well to estimate the target model M^T . However, in the context of an analogy, some additional information is provided by the observation of the source domain, where a model M^S can be evaluated from observations $(x^S, y^S) \in \mathcal{D}^S$. The target model M^T can then be described with regards to the source model M^S .

The limitation of such a relative description of models is that the source model may not be relevant. In the following, we propose to quantify this relevance with two measures of model reusability,

3.2. Reusability of a Source Model

We measure the reusability of a source model to its ability to compress the information about the target domain. We identify two main possible definitions, that we call *weak* and *strong* reusability, and will show that a strongly reusable model is necessarily weakly reusable.

The main criterion for *weak reusability* is that knowing the source model M^S makes the target inference better, in the sense that it compresses more the description of the target case $(x^T, y^T) \in \mathcal{D}^T$.

Definition 1 (Weak reusability). *Let $\eta > 0$. A model $M^S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}$ is called weakly η -reusable for case (x^T, y^T) in target model space $\mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$ if:*

$$\begin{aligned} \min_{M \in \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}} \{K(M) + K(x^T, y^T | M)\} \\ \geq \eta + \min_{M \in \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}} \{K(M | M^S) + K(x^T, y^T | M)\} \end{aligned} \quad (6)$$

It is essential to keep in mind that this definition is relative to the choice of a model space $\mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$ for the target domain. A source model M^S may not be reusable for a case (x^T, y^T) depending on the chosen target model space. In the following, we will abusively omit to mention the target model space, for readability purposes.

Note that the models M defined in the left-hand side and in the right-hand side of inequality (6) are not the same. On the left-hand-side, the model corresponds to the optimal target model describing (x^T, y^T) while, on the right-hand-side, it is the optimal model describing (x^T, y^T) when M^S is known.

This notion of reusability means that providing the source model M_S helps finding a new description of the problem shorter than the optimal description by η bits. In particular, in the case where $\mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S} = \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$, the optimal model for (x^T, y^T) (i.e. the model that minimizes $K(M) + K(x^T, y^T|M)$) is trivially weakly reusable for (x^T, y^T) for all η . In other words, when the optimal model for the source domain is also optimal for the target domain, then it is obviously reusable for the target domain. The interesting case will be when the optimal source model is not optimal for the target domain.

We notice that the definition does not require $\mathcal{D}^S = \mathcal{D}^T$ and aims to quantify the reusability of a model even for a completely different task. This is possible since complexity only requires to have computable models: Indeed, the term $K(M|M^S)$ is defined as long as the models are computable. In practice, the issue of comparing objects of different nature is hidden within the choice of the reference p.r.p. function for complexity (see Section 2.2).

We propose an alternative definition of reusability, called *strong reusability*. It is based on the idea that M_S is reusable if it helps compressing the optimal model of target case (x^T, y^T) . Unlike previous definition, M_S is not directly involved in the description of (x^T, y^T) though.

Definition 2 (Strong reusability). *Let $\eta > 0$. A model $M^S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}$ is called strongly η -reusable for case (x^T, y^T) in target model space $\mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$ if:*

$$\begin{aligned} M \in \underset{M \in \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}}{\operatorname{arg\,min}} \{K(M) + K(x^T, y^T|M)\} \\ \implies K(M) \geq K(M|M^S) + \eta \end{aligned} \quad (7)$$

This definition also relies on the choice of a target model space. As for weak reusability, we will abusively omit the model space in the following notations.

Strong reusability is an extremely strong property of a source model. Indeed, it would be a natural property that, in case $\mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S} = \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$, any model minimizing $K(M) + K(x^T, y^T|M)$ is reusable to (x^T, y^T) . This is not the case for strong reusability: indeed, the definition implies that *any* model minimizing $K(M) + K(x^T|M)$ can be compressed given M^S , not only M^S itself. Note that this could be alleviated by weakening Definition 2 and requiring only the existence of a model M minimizing $K(M) + K(x^T, y^T|M)$ and such that $K(M) \geq K(M|M^S) + \eta$.

3.3. Properties of Reusability

We now present basic properties of these two notions of reusability. All the presented properties follow directly from the definitions.

The first property applies to both weakly and strongly reusable models: it states that the threshold η is not unique. The proof of this proposition is trivial and omitted.

Proposition 1. *Let $M^S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}$ and $(x^T, y^T) \in \mathcal{D}^T$. If M^S is η -reusable for (x^T, y^T) , then it is also η' -reusable for (x^T, y^T) for all $\eta' \leq \eta$. In the following, we will call degree of reusability of M^S to (x^T, y^T) the quantity:*

$$\rho(M^S, (x^T, y^T)) = \max \{ \eta ; M^S \text{ is } \eta\text{-reusable for } (x^T, y^T) \} \quad (8)$$

However, we insist on the fact that the degree of reusability is relative to a chosen target model space. It also depends on which of weak or strong reusability is considered. This dependency would not exist if these two notions of reusability were equivalent, which we will see is not the case. We will use the notation ρ_w and ρ_s to specify between the weak and strong cases.

We now show that strong reusability implies weak reusability:

Proposition 2. *Let $M^S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}$ a source problem, a target case $(x^T, y^T) \in \mathcal{D}^T$ and $\eta > 0$. If M^S is strongly η -reusable for (x^T, y^T) , then M^S is also weakly η -reusable for (x^T, y^T) .*

Proof. We call $M^* \in \mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T}$ an optimal model for (x^T, y^T) : $M^* \in \arg \min_M \{ K(M) + K(x^T, y^T | M) \}$. Under the assumptions of the proposition, it follows that:

$$\begin{aligned} & \min_M \{ K(M) + K(x^T, y^T | M) \} \\ &= K(M^*) + K(x^T, y^T | M^*) \\ &\geq K(M^* | M^S) + \eta + K(x^T, y^T | M^*) \\ &\geq \min_M \{ K(M | M^S) + K(x^T, y^T | M) + \eta \} \end{aligned}$$

which proves the proposition. \square

However, the converse is not true: weak reusability does not imply strong reusability. This is the consequence of the fact that the models implied in Equation (6) are not the same on the right hand side and on the left-hand side. We illustrate this with a simple example.

Example 3. *We consider a target model space made up of two distinct models: $\mathbb{M}_{\mathcal{R}^T, \mathcal{D}^T} = \{M_1, M_2\}$. We assume the following properties:*

- M_1 and M_2 describe equally well case (x^T, y^T) , in the sense that $K(x^T, y^T | M_1) = K(x^T, y^T | M_2)$
- Model M_1 is more complex than model M_2 : $K(M_1) > K(M_2)$
- Model M_1 is easily described by source model M^S : for simplicity, we can take $K(M_1 | M^S) = 0$
- Model M_2 is not well described by source model M^S : for simplicity, we can take $K(M_2 | M^S) = K(M_2)$

With these assumptions, it can be easily verified that M^S is weakly η -reusable for (x^T, y^T) , with $\eta \leq K(M_2)$. However, M^S is not strongly η -reusable for (x^T, y^T) , since M_2 minimizes $K(M) + K(x^T, y^T|M)$ but $K(M_2) < K(M_2) + \eta = K(M_2|M^S) + \eta$.

Putting together the results of Propositions 1 and 2, as well as the counter-example of Example 3, we can establish the following result:

Corollary 1. For $M_S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}$ and $(x^T, y^T) \in \mathcal{D}^T$:

$$\rho_s(M^S, (x^T, y^T)) \leq \rho_w(M^S, (x^T, y^T)) \quad (9)$$

3.4. Transferable Cases

The notion of reusability we proposed in Definitions 1 and 2 are not directly applicable to measure transferability from a source observation to a target problem. The property of transferability measures the ability to transfer knowledge from a source case $(x^S, y^S) \in \mathcal{D}^S$ to apply it to the target case $(x^T, y^T) \in \mathcal{D}^T$. It can be seen as an extension of reusability where the source model M^S is determined based on the source observation.

Definition 3 (Transferability). Let $(x^S, y^S) \in \mathcal{D}^S$ be a source case. The case (x^S, y^S) is said to be strongly (resp. weakly) η -transferable to the target case $(x^T, y^T) \in \mathcal{D}^T$ if the set of compatible models $\{M^S | K(M^S) + K(x^S, y^S|M^S) < K(x^S, y^S), M^S \in \mathbb{M}_{\mathcal{R}^S, \mathcal{D}^S}\}$ contains an element M^{S*} such that M^{S*} is strongly (resp. weakly) reusable for case (x^T, y^T) .

The proposed definition of transferability is very weak, since it only requires the existence of one source model that is reusable. However, it does not take into account the quality of this model in the source domain. For instance, it would acquire equal weight if the reusable source model is associated with complexity $K(M^S) + K(x^S, y^S|M^S)$ close to $K(x^S, y^S)$, as if the complexity is close to 0. Such situations are very likely to occur with probabilistic models, since many such models give positive probability to all observations.

In order to refine this notion, it would be important to define a coefficient of transferability, similar to the degree of reusability ρ defined in Proposition 1. We will denote such a coefficient $\tau((x^S, y^S), (x^T, y^T))$. We propose two possible definitions below. These definitions rely on the set of compatible models for (x^S, y^S) introduced in Definition 3, and denoted $\mathbb{M}(x^S, y^S)$. For simplicity, we will use the notation \mathcal{C}^S (resp. \mathcal{C}^T) to designate the case (x^S, y^S) (resp. (x^T, y^T)).

A first definition is a direct application of Definition 3: it associates the transferability of a problem to the maximum reusability of the corresponding model:

$$\tau_{\max}(\mathcal{C}^S, \mathcal{C}^T) = \max_{M^S \in \mathbb{M}(\mathcal{C}^S)} \rho(M^S, \mathcal{C}^T) \quad (10)$$

with the convention that $\max \emptyset = 0$. This definition has the same weakness as the introduced concept of transferability: it does not take into account that the most reusable model might also be very weak to describe (x^S, y^S) .

In order to alleviate this, our second definition proposes to average the score over the possible models. Therefore, we compute the posterior of the model for given (x^S, y^S) and compute the

score as the expected value over M_S of the degree of reusability.

$$\tau_{\text{avg}}(\mathcal{C}^S, \mathcal{C}^T) = \sum_{M^S \in \mathbb{M}(\mathcal{C}^S)} p(M^S | x^S, y^S) \rho(M^S, \mathcal{C}^T) \quad (11)$$

In order to compute the posterior $p(M^S | x^S, y^S)$, we use algorithmic probability [10] and assume a uniform prior:

$$p(M^S | x^S, y^S) = 2^{-K(x^S, y^S | M^S) - K(M^S)} \quad (12)$$

This second definition provides a better idea of how transferable a source observation can be, since it takes into account all possible models describing it. However, it has two major weaknesses. On a theoretical level, it does not reflect the variance of the reusability degree over the compatible models: based on ρ_{avg} only, it is impossible to know whether all models, only very few but very compatible models, or a large number of less compatibles are reusable. On a practical level, the τ_{avg} score may not be tractable, and would require some approximations.

4. Illustration: Transferability of Morphological Transformations

In this section, we propose to apply the notions introduced in Section 3 to the case of morphological analogies. We will mostly build upon the domain introduced in Example 1.

4.1. Introduction to Morphological Analogies

Morphological analogies are analogies on words involving transformations of a morphological nature, for instance declension or conjugation. Unlike semantic analogies (e.g. “king is to queen as man is to woman”), morphological analogies are mostly of a symbolic nature: they involve the detection of transformations of one form of a word into another form. Typical example of such morphological analogies could be “work : worked :: call : called” in English (transformation from present to preterit, in English), or “voihin : vuossa :: soihin : suossa” (transformation from illative plural to inessive singular in Finnish).

Various works have been proposed to solve such analogies, mostly based on the principles of proportional analogy [7]. Other approaches rely on an algebraic consequence of these principles [16], on deep learning approaches [17] or even on complexity minimization [18].

4.2. A Simplified Model for Morphology

The model proposed in Example 1 is relevant for a formal treatment of morphological analogies. However, following Murena et al. [18], we propose a simplification where the space of partial recursive functions is restricted to a simpler subset. This subset is defined by a simple descriptive language based on the concatenation of various strings. It allows to define functions of the form $\phi = (\phi_1, \phi_2)$ with, for $i \in \{1, 2\}$:

$$\phi_i(r_1, \dots, r_n) = w_0^i + \sum_{k=1}^{K_i} r_{\sigma_i(k)} + w_k^i \quad (13)$$

where the $+$ operation stands for string concatenation, $r_1, \dots, r_n \in \mathcal{A}^*$ and $w_0^i, \dots, w_K^i \in \mathcal{A}^*$ are words on the alphabet \mathcal{A} , and $\sigma_i : \{0, \dots, K_i\} \rightarrow \{0, \dots, n\}$. The vector $r = (r_1, \dots, r_n)$ corresponds to the representation, and the models are defined such as in Equation (1).

We notice that the proposed restriction accounts for various types of morphological transformations, such as prefixation, suffixation, change or prefix and/or suffix, but also duplication. However, the language is not Turing complete, and for instance does not cover any conditional statement.

In morphological analogies, the source and target domains are the same.

4.3. Computing Complexities

In order to compute the reusability and transferability, we must define three expressions of complexity: complexity of a model $K(M)$, complexity of a case given a model $K(x, y|M)$ and complexity of a target model given a source model $K(M^T|M^S)$.

A model is entirely defined by the function ϕ . A binary coding of such functions is proposed in [18]. The complexity of the model then corresponds to the length (in bits) of the binary code of the function.

Given a model M , the case (x, y) is coded by providing a correct representation (r_1, \dots, r_n) . In case no representation generates (x, y) with model M , the two words are hard-coded. In terms of binary representation, we propose the following coding. The string starts with a bit encoding whether the following bits code for the representation vector or for the two words. After this bit, the words are coded letter by letter, with specific delimiters to mark the end of each string. The complexity $K(x, y|M)$ is the length of this code.

For the model transfer, we propose an elementary description. More sophisticated versions have to be discussed in future works. We propose to reuse the source model either by reusing it directly (without modification), or by redefining completely. Such as previously, this choice is indicated by an initial bit. The model complexity is then given by:

$$K(M^T|M^S) = \begin{cases} 1 & \text{if } M^T = M^S \\ 1 + K(M^T) & \text{otherwise} \end{cases} \quad (14)$$

4.4. Examples of Reusability

Suffixation. We consider the source model M^S associated to transformation $\phi(r_1) = (r_1, r_1 + "s")$ which consists in suffixing an "s" at the end of a word. We can verify that M^S is η -reusable for the target case ("film", "films"). In that case, it can be verified that M^S minimizes quantity $K(M) + K(x^T, y^T|M)$. Consequently, M^S is weakly η -reusable for ("film", "films") with $\eta \leq K(M^S) - 1$. This shows that $\rho_w(M^S, ("film", "films")) = K(M^S) - 1$. It can also be verified that M^S is the *only* model minimizing $K(M) + K(x^T, y^T|M)$. Therefore, we also have that M^S is strongly η -reusable for ("film", "films") with $\eta \leq K(M^S) - 1$. In this case, we have $\rho_s(M^S, ("film", "films")) = \rho_w(M^S, ("film", "films"))$.

However, the model is not reusable for ("mouse", "mice") for instance, since the minimal transformation for this case is $\phi'(r) = (r + "ouse", r + "ice")$. We then have $\rho_s(M^S, ("mouse", "mice")) = \rho_w(M^S, ("mouse", "mice"))$.

Duplication. We consider the source model M^S associated to transformation $\phi(r_1) = (r_1 + \text{“-”} + r_1, r_1)$. This transformation is the reverse of a duplication (plural form in Indonesian). As for previous example, one can easily check that M^S is not reusable for the case (“orang”, “orang-orang”). This was expected, with the choice of the transfer representation $K(M^T|M^S)$ which forces toward reusing the source model or ignoring it completely. It would not have been the same with other choices though. In particular, if it allowed for model transformation of the form $(\phi_1, \phi_2) \mapsto (\phi_2, \phi_1)$.

5. Toward Transferability in Domain Adaptation

Domain adaptation is a machine learning task where a hypothesis on a source domain has to be transferred to a target domain where data are not equally distributed [19]. We conclude this paper with a quick investigation of how domain adaptation can fit to our proposed framework.

5.1. Related Works: Task-Relatedness

The question of transferability of one solved source problem to a target problem has played a predominant role in the theoretical understanding of domain adaptation. Ben-David et al. (2010) [20] propose a PAC bound for transfer between domains in a binary classification setting, which relies on the use of a measure of a specific *domain divergence*, called \mathcal{H} -divergence. It is noticeable that the introduced measure depends on the hypothesis class \mathcal{H} , i.e. on the model space. This proposed notion has then been refined, to account for a variety of loss functions [21] or to adapt to the PAC-Bayesian setting [22]. All these measures follow a similar idea of comparing the distribution over the input spaces, but ignore the labels. Closer to our proposal, Zhang et al. (2012) [23] propose to additionally take into account the label distribution $p(y)$ in the discrepancy measure. Independently from these studies, Mahmud (2009) [24] also proposed to quantify the task-relatedness using Kolmogorov complexity and Algorithmic Information Theory.

5.2. Open Question: Complexities for Probabilistic Models

We describe the domain adaptation task in the context of the probabilistic model space defined in Example 2, in which a model is associated to a probability distribution. Such as for the morphological domain (Section 4.3), we need to define the complexities $K(M)$, $K(x, y|M)$ and $K(M^T|M^S)$.

The term $K(x, y|M)$ is a standard quantity considered by Algorithmic Information Theory. It is commonly computed using the *Shannon-Fano coding*. Using this allows to define the complexity of an object (x, y) knowing a probability distribution p as $K(x, y|p) = -\log p(x, y)$. This corresponds to the natural choice when computing $K(x, y|M)$ in our probabilistic domain.

Defining $K(M)$ and $K(M^T|M^S)$ is more difficult thought and goes beyond the scope of this paper. Traditionally, the complexity of a probability distribution is assimilated to the probability of its density function, and specific computations are proposed to estimate these. We refer the interested reader to [24] for instance.

5.3. Discussion

Extending our framework to domain adaptation is both natural and complex. Indeed, the problem formulation in terms of models allow for a direct characterization of domain adaptation, where a domain is characterized by an unlabeled dataset (the problem) and a vector of predictions (the solution). However, the computation of the complexities may not be as simple as for the symbolic domain. Future works will have to bridge this gap.

6. Conclusion

In this paper, we introduced a novel understanding of analogical transfer, by focusing on whether the information contained in the source case were transferable to the target case. We proposed a formalism based on a notion of *models*, which we defined in a way that is consistent with both symbolic analogies and numerical machine learning. Even though this preliminary works cover only the question of measuring the transferability from a source case (x^S, y^S) to a target case (x^T, y^T) , it is a first step toward the key question of predicting whether knowledge of (x^S, y^S) could be reused to find a solution y to problem x^T . We think the proposed framework could provide a common basis to develop a general understanding of transfer, going beyond the current statistical theories [20] for instance.

Acknowledgments

The author wishes to thank the anonymous reviewers for their insightful comments and suggestions. Some of the presented ideas have been inspired by discussions with Antoine Cornuéjols, Jean-Louis Dessalles, Marie Al-Ghossein and Miguel Couceiro. This work was supported by the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence, FCAI.

References

- [1] D. Gentner, C. Hoyos, Analogy and Abstraction, *Top. Cogn. Sci.* 9 (2017) 672–693.
- [2] L. Miclet, H. Prade, Handling analogical proportions in classical logic and fuzzy logics settings, in: G. C. C. Sossai (Ed.), *Proc. 10th Europ. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009)*, Springer, Berlin/Heidelberg, 2009, pp. 638–650.
- [3] N. Barbot, L. Miclet, H. Prade, Analogy between concepts, *Artif. Intell.* 275 (2019) 487–539.
- [4] D. Hofstadter, M. Mitchell, *Fluid Concepts and Creative Analogies*, Basic Books, Inc., New York, NY, USA, 1995, pp. 205–267.
- [5] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies, 2013*, pp. 746–751.
- [6] S. Lim, H. Prade, G. Richard, Classifying and completing word analogies by machine learning, *International Journal of Approximate Reasoning* 132 (2021) 1–25.

- [7] Y. Lepage, Analogy and Formal Languages, *Electron. Notes Theor. Comput. Sci.* 53 (2001) 180–191.
- [8] P. A. Murena, J.-L. Dessalles, A. Cornuéjols, A complexity based approach for solving Hofstadter’s analogies, in: *CAW@ ICCBR-2017 Computational Analogy Workshop*, at International Conference on Case Based Reasoning, 2017.
- [9] P.-A. Murena, M. Al-Ghossein, Inferring Case-Based Reasoners’ Knowledge to Enhance Interactivity, in: *International Conference on Case-Based Reasoning*, Springer, 2021, pp. 171–185.
- [10] M. Li, P. Vitányi, et al., *An introduction to Kolmogorov complexity and its applications*, volume 3, Springer, 2008.
- [11] C. Antić, *Analogical Proportions*, 2020. URL: <https://arxiv.org/abs/2006.02854>.
- [12] S. Shalev-Shwartz, S. Ben-David, *Understanding machine learning: From theory to algorithms*, Cambridge university press, 2014.
- [13] C. S. Wallace, D. M. Boulton, An information measure for classification, *The Computer Journal* 11 (1968) 185–194.
- [14] J. Rissanen, Modeling by shortest data description, *Automatica* 14 (1978) 465–471.
- [15] P. D. Grünwald, *The minimum description length principle*, MIT press, 2007.
- [16] P. Langlais, F. Yvon, P. Zweigenbaum, Improvements in analogical learning: application to translating multi-terms of the medical domain, in: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 487–495.
- [17] S. Alsaidi, A. Decker, P. Lay, E. Marquer, P.-A. Murena, M. Couceiro, A Neural Approach for Detecting Morphological Analogies, in: *IEEE 8th DSAA*, 2021, pp. 1–10.
- [18] P.-A. Murena, M. Al-Ghossein, J.-L. Dessalles, A. Cornuéjols, Solving Analogies on Words based on Minimal Complexity Transformation., in: *IJCAI*, 2020, pp. 1848–1854.
- [19] A. Farahani, S. Voghoei, K. Rasheed, H. R. Arabnia, A brief review of domain adaptation, *Advances in Data Science and Information Engineering (2021)* 877–894.
- [20] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, *Machine learning* 79 (2010) 151–175.
- [21] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain adaptation: Learning bounds and algorithms, *arXiv preprint arXiv:0902.3430* (2009).
- [22] P. Germain, A. Habrard, F. Laviolette, E. Morvant, A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers, in: *International conference on machine learning*, PMLR, 2013, pp. 738–746.
- [23] C. Zhang, L. Zhang, J. Ye, Generalization bounds for domain adaptation, *Advances in neural information processing systems* 25 (2012).
- [24] M. H. Mahmud, On universal transfer learning, *Theoretical Computer Science* 410 (2009) 1826–1846.