Explaining Effective Learning by Analogical Reasoning

Helmar Gust (hgust@uos.de)
Institute of Cognitive Science, University of Osnabrück
Albrechtstr. 28, 49076 Osnabrück, Germany

Kai-Uwe Kühnberger (kkuehnbe@uos.de)
Institute of Cognitive Science, University of Osnabrück
Albrechtstr. 28, 49076 Osnabrück, Germany

Abstract

Machine learning algorithms are usually considered as explicit learning strategies requiring large data samples. Contrary to these accounts, cognitive learning seems to be based on significantly less amounts of training data and occurs often in the form of implicit learning. In order to close this gap we propose to explain these discrepancies by a form of analogical learning, bridging these two traditions. Using heuristic-driven theory projection (HDTP) as the framework for analogy making we can productively model learning aspects with sparse training data.

Keywords: Analogical Reasoning, Learning, Creativity.

Introduction

In cognitive science, cognitive modeling, and artificial intelligence, a large number of different learning theories were proposed. Considering learning mechanisms from a computer science perspective, two major types of learning theories can be roughly distinguished:

- On the one hand, lazy learning algorithms have been proposed, i.e. learning theories which store each exemplar (or example) explicitly in a database without identifying abstract features.
- On the other hand, eager learning algorithms have been extensively discussed, i.e. learning theories which minimize the storage load by an abstraction process. The identification of important (common) features of the exemplars is a necessary prerequisite for this type of learning algorithm.

Lazy learning occurs in a variety of different versions: It is also called instance-based learning (Aha, Kibler & Albert, 1991), exemplar-based learning (Salzberg, 1991), case-based learning (Kolodner, 1993), or memory-based learning (Stanfill & Waltz, 1986). Eager learning is a collection of learning algorithms that abstracts from the sample data like in ID3 (Quinlan, 1986), version space (Mitchell, 1982), inductive learning (Muggleton & Feng, 1990), connectionist-style learning algorithms (Bishop, 1995), or explorative learning (Watkins & Dayan, 1992). The proposed learning paradigms were applied to a variety of different domains: Examples are perception, and motor control in artificial agents, induction of grammars in natural language processing, expert systems, or categorization. The different accounts have been proven to be successfully applicable to many domains of interest.

Both classes of learning strategies are based on a common idea: As input data a (more or less) large number of examples is needed to guarantee a successful learning procedure. Based on these examples the corresponding learning algorithms can establish generalized hypotheses. Although the mentioned techniques are reasonable accounts from an engineering perspective, cognitive adequacy cannot be achieved by these learning concepts:

- The available data samples for successful human learning are in general rather limited, as can be seen in language acquisition, learning of new concepts, or perception. The mentioned machine learning accounts usually fail in learning effectively from sparse data samples.
- Human learning abilities are embedded in and crucially connected to a context, for example, coordinated with a particular perception or a motor action. Learning occurs often implicitly in these context, i.e. as a side-effect of problem solving. In contrast to human learning, machine learning accounts are often based on explicit learning accounts and usually abstract from the situational context.
- The ability of humans to learn new concepts cannot be underestimated as one of the most important cognitive capacities. In contrast, machine learning algorithms are rather limited concerning their productivity in generating new concepts.

Considering this situation from a cognitive science perspective, other learning models need to be developed, in order to take the mentioned points into account. We will propose to use analogical reasoning as a possibility to partially bridge this gap between machine learning accounts and cognitive learning. The paper has the following structure: First, we will roughly discuss some aspects of learning and cognition. We will continue by presenting the crucial ideas of heuristic-driven theory projection (HDTP) as a model for analogy making. After sketching a simple example, we will discuss how a learning theory based on analogies can be developed.

Learning and Cognition

The algorithms mentioned in the introduction have a common feature: Large data samples are usually necessary for successful learning. Dependent on the particular learning algorithm the size of the necessary training data sample may vary. In the case of learning by neural networks, a larger data sample is usually needed than in applications where case-based reasoning techniques are applied. On the other hand, neural networks may perform better in some domains than case-based reasoning. Nevertheless the required data of both accounts exceeds by far corresponding learning data for cognitive agents like humans.

In many cases, cognitive learning seems to work differently in comparison to the proposed learning approaches: The Chomsky - Skinner debate can be seen as a prominent example focusing on this gap (Chomsky, 1959). In order to explain the human ability to learn natural language, Chomsky's conclusion was that a universal grammar must be presupposed. Researchers from the psychological field put a certain emphasize on imitation, projecting the behavior of others to oneself, like in observational learning (Bandura & Walters, 1963), or stressed the influence of situations, like in situated learning (Lave & Wenger, 1990). Unfortunately no convincing machine learning approach realizing these psychologically motivated theories in a computer is available yet.

We propose to bridge the gap between machine learning on the one hand and the rather effective learning strategies of cognitive agents on the other hand by analogical reasoning. Analogical reasoning was discussed in domains like proportional analogies in string domains (Hofstadter & The Fluid Analogies Research Group, 1995) and analogies between geometric figures (Dastani, 1998; Evans, 1968). Further discussions were based on the relation between analogies and metaphors (Indurkhya, 1992; Gentner et al., 2001) and on analogical problem solving (Anderson & Thompson, 1989). Methods used for modeling analogies range from algebraic accounts (Dastani, Scha & Indurkhya, 1997; Indurkhya, 1992) to graph-based approaches (Gentner, 1983; Falkenhainer, Forbus & Gentner, 1989) and similarity-based approaches (Gentner, 1989). Although the mentioned models for analogical reasoning differ quite significantly from each other in some aspects certain other aspects issues seem to be uncontroversial: Analogical relations between a well-known domain (source domain) and a formerly unknown domain (target domain) can be established without taking much input data into account. Rather it is the case that a conceptualization of the source domain is sufficient to productively generate knowledge about the target domain. This can be achieved by associations of attributes and relations of the source domain and the target domain. Moreover, a projection of attributes and relations from source to target can productively introduce new concepts on the target domain. The result is that cognitive agents can learn a new conceptualization of the target domain without perceiving a huge number of examples.

We mention some domains where we think analogical reasoning can be used, in order to explain learning aspects of human cognition:

- Learning how to use new software: If a human knows how to use a text-processing software like MS-Word, she can easily adapt to a new software like the text processing tool of Open Office, although the menu and the overall structure of the new software may be completely different. An explanation can be given by an analogical transfer.

- Learning new concepts: The productivity of metaphoric expressions to generate new meanings can be modeled by analogies (Gust, Kühnberger & Schmid, 2006b). A well-known domain is the introduction of technical concepts, for example, in the IT world: Concepts like *(computer) mouse*, *daemon*, *virus*, or *backbone* are metaphors that can be explained by establishing a metaphorical relation (Gust, Kühnberger & Schmid, 2006a).

- Learning of abstract concepts: Learning in high schools and universities as well as insights of scientists in research projects are strongly based on a concept interpretation "as if it were" a well-known concept. To a large extent this can be explained by an analogical transfer.

In the following section, we will sketch the ideas of a mathematically sound theory that can be used for establishing analogical transfers. We propose a model for analogical reasoning where no large numbers of examples are needed. Rather it is the case that analogical learning occurs as a side-effect by the process of generating generalizations, i.e. learning is an implicit feature of the system. Furthermore learning occurs in stepwise processes, where generating generalizations is only one part among others. An equally important part of analogical learning concerns the transfer of knowledge from the source to the target governed by a testing procedure (experiment) that either accepts or rejects certain transfers. Last but not least, even dynamic updates of the source domain may be possible in case the analogical relation cannot be established using the given input data.

Heuristic-Driven Theory Projection

The Idea of HDTP

Heuristic-Driven Theory Projection (HDTP) is a formally sound theory for computing analogical relations between a source domain and a target domain. HDTP computes analogical relations not only by associating concepts, relations, and objects, but also complex rules and facts between target and source domain. In Gust, Kühnberger & Schmid (2006b) the syntactic, semantic, and algorithmic properties of HDTP are specified. Unlike to well-known accounts for modeling analogies like the structure-mapping engine (Falkenhainer, Forbus & Gentner, 1989) or Copycat (Hofstadter & The Fluid Analogies Group, 1995), HDTP produces abstract descriptions of the underlying domains, is heuristic-driven, i.e. allows to include various types of background knowledge, and has a model theoretic semantics induced by an algorithm. HDTP was applied to a variety of domains, for example, naive physics (Schmid, Gust, Kühnberger & Burghardt, 2003; Gust, Kühnberger & Schmid, 2003a) and metaphors (Gust, Kühnberger & Schmid, 2006a). The algorithm HDTP-A is implemented in SWI-Prolog. The core program is available online (Gust, Kühnberger & Schmid, 2003b).

Syntactically, HDTP is defined on the basis of a many-sorted first-order language. First-order logic is used in order to guarantee the necessary expressive power of the account. An important assumption is that analogical reasoning crucially contains a generalization (or abstraction) process. In other words, the identification of common properties or relations is represented by a generalization of the input of source and target. Mathematically this can be modeled by an extension of the so-called theory of anti-unification (Plotkin, 1970), a mathematically sound account describing the possibility of generalizing terms of a given language using substitutions. More precisely, an anti-unification of two terms

Table 1: A simplified description of the algorithm HDTP-A omitting formal details. A precise specification of this algorithm can be found in Gust, Kühnberger & Schmid, 2006b.

Input: A theory Th_S of the source domain and a theory Th_T of the target domain represented in a many-sorted predicate logic language.
Output: A generalized theory Th_G such that the input theories Th_S and Th_T can be reestablished by substitutions.

Selection and generalization of fact and rules.
    Select an axiom from the target domain (according to a heuristics h).
    Select an axiom from the source domain and construct a generalization (together with corresponding substitutions).
Optimize the generalization w.r.t. a given heuristics h'.
    Update the generalized theory w.r.t. the result of this process.
Transfer (project) facts of the source domain to the target domain provided they are not generalized yet.
    Test (using an oracle) whether the transfer is consistent with the target domain.

t_1 and t_2 can be interpreted as finding a generalized term t (or structural description t) of t_1 and t_2 which may contain variables, together with two substitutions Θ_1 and Θ_2 of variables, such that $t\Theta_1 = t_1$ and $t\Theta_2 = t_2$. Because there are usually many possible generalizations, anti-unification tries to find the most specific one. An example should make this idea clear. Assume two terms $t_1 = f(X, b, c)$ and $t_2 = f(a, Y, c)$ are given. Generalizations are, for example, the terms $t = f(X, Y, c)$ and $t' = f(X, Y, Z)$ together with their corresponding substitutions.[1] But t is more specific than t', because the substitution Θ substituting Z by c can be applied to t'. This application results in: $t'\Theta = t$. Most specific generalizations of two terms are commonly called anti-instances.

In order to guarantee the necessary expressive strength, HDTP extends the theory of anti-unification in several ways: First, not only terms but also formulas can be generalized. Second, predicates and functions can be generalized (second-order case). Third, we allow to generalize whole theories, because the input for the source and the target domains are usually given as a (more or less complex) theory about this domain. Fourth, the account is heuristic-driven, i.e. background knowledge governs the generalization process. Fifth, data from the source domain can be projected to the target domain in order to make the introduction of new concepts on the target side possible.

Given two input theories Th_S and Th_T for source and target domain respectively, the algorithm HDTP-A computes anti-instances together with a generalized theory Th_G. Table 1 specifies the most important steps of this algorithm:

[1]We assume that symbols $a, b, c, \ldots$ denote constants whereas capital symbols $X, Y, Z, \ldots$ denote variables, similar to the usage in Prolog.

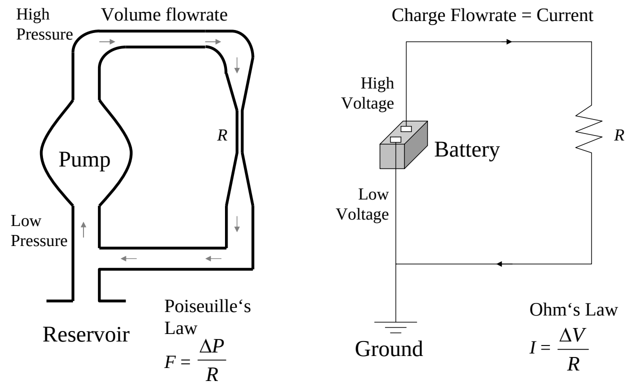

Figure 1: The analogy between a water pipe system and an electric circuit in a diagrammatic representation. The Figure contains more information than is necessary for an interpretation of the metaphorical description (1).

First, an axiom from the target domain is selected, guided by an appropriate heuristics h, for example, measuring the syntactic complexity of the axiom. Then an axiom of the source domain is searched in order to construct a generalization together with substitutions. The generalization is optimized using another heuristics h', for example, the length of the necessary substitutions. Finally axioms from the source domain are projected to the target domain. Then the transferred axioms are tested for empirical validation with the target domain using an oracle. If the test renders the axiom as invalid the transfer is blocked.[2] Furthermore the transferred must be consistent with the target theory, i.e. if a contradiction can be detected the transfer must rejected. Technically this can be implemented by a theorem prover.

Learning with HDTP

An Example

Many prototypical examples of analogies and analogical transfers can be found in the analogy related literature (like the Rutherford analogy establishing an analogical relation between the solar system and the atom model or the heat-flow analogy in which water-flow and heat-flow are associated by an analogy). In this subsection, we emphasize some important aspects of our modeling with HDTP. Consider the following analogy (represented as a metaphorical expression):

(M1) *Current is the water in the electric circuit.*

Figure 1 depicts the situation represented by this analogy.[3] The analogy associates water-flow in a water pipe system with the flow of current in an electric circuit. In a learn-

[2]In our view, an oracle represents a function mapping formulas containing only observables to truth values. Such formulas can be interpreted as specifying an experiment, i.e. they can loosely be compared with the role of a physicist who is performing experiments.

[3]The figure is based on a graphical representation of this analogy found on a physics webside of the Georgia State University (cf. http://hyperphysics.phy-astr.gsu.edu/hphys.html.)

Table 2: Examples of corresponding concepts in the source and the target domains of the analogy between water-flow and the flow of current in an electric circuit after a successful establishment of an analogical relation. The shortcut *ws*1 denotes an instance of a water pipe system and *es*1 an instance of an electric circuit.

	Source	Target	Generalization
(1)	*water_circuit(ws1,water,p1)*	*electric_circuit(es1,current,b1)*	*Circuit(A,C,S1)*
(2)	*closed(ws1)*	*closed(es1)*	*closed(A)*
(3)	*pump(p1)*	*battery(b1)*	*Source(S1)*
(4)	*pres(p1)* $> 0 \rightarrow$ *flow_in_circuit(water)*	*pres(b1)* $> 0 \rightarrow$ *flow_in_circuit(current)*	*pres(S1)* $> 0 \rightarrow$ *flow_in_circuit(C)*
(5)	*flow_in_circuit(water)*	*flow_in_circuit(current)*	*flow_in_circuit(C)*

ing situation of a high school student, clearly Ohm's law and Poiseuille's law are not available to the students. Therefore, Figure 1 depicts more than what can be learned by this analogy. Nevertheless an important new conceptualization about electricity can be learned by students using this analogy, namely that current is flowing in a circuit and that a battery has the function similar to a pump in the water pipe system.

We would like to achieve a modeling of metaphor (M1) using HDTP. Table 2 specifies the corresponding concepts in the target and the source domains that are associated with each other. The association established by HDTP is realized by a generalization process of the input theories. Hence the concept of a closed water system and a closed electric system generalize to an abstract concept $closed(A)$ where A is a variable. The terms *water* and *current* are associated explicitly in the metaphoric expression (M1). From the background knowledge a rule is available stating that if the pressure caused by the pump *p*1 in a water pipe system is different from 0, then water is flowing in the circuit (from high pressure to low pressure). This can be projected to the target side, inferring that due to the "pressure" of the battery *b*1 (realized by a positive voltage), current is flowing in the electric circuit. Hence, we end up with the conclusion (5 in Table 2) that current is flowing in the electric circuit (provided there is a "pressure" source). By the generalization process and the corresponding substitutions of the variables, we get a structural description of the two domains. The substitutions Θ_1 and Θ_2 can be summarized as follows:

$$
\begin{array}{lrcl}
\Theta_1/\Theta_2: & A & \longrightarrow & \textit{ws1 / es1} \\
 & C & \longrightarrow & \textit{water / current} \\
 & Source & \longrightarrow & \textit{pump / battery} \\
 & S1 & \longrightarrow & \textit{p1 / b1} \\
 & Circuit & \longrightarrow & \textit{water\_circuit / electric\_circuit}
\end{array}
$$

Clearly the proposed modeling is more complex than a pure and direct modeling of metaphor (M1). We tried to cover some important aspects of Figure 1 in the representation. In the case of metaphor (M1) the situation would be quite simple. Concepts like *battery*, *pump*, or *pressure* do not occur. Rather from the conceptualization that in a circuit of water, it is usually the case that water is flowing, we achieve the fact that current is flowing in an electric circuit directly by projecting the source to the target.

Establishing successfully an analogical relation between the involved domains by projecting facts and rules from source to target domains results in learning a new conceptualization of a formerly unknown concept, namely some property current depicts in an electric circuit. The acquired type of knowledge is clearly not a precise physical theory about electricity, rather a type of pre-conceptualization of a new domain.

It should be stressed that this type of learning does not require a large amount of input data. Only a sufficiently rich conceptualization of the source domain is necessary, usually given by background knowledge. In the following discussion we sketch how this can be used in order to develop a theory of learning by analogical reasoning.

Discussion

Contrary to the case of inductive learning, HDTP provides a generalization of quite parsimonious input data. Instead of extracting abstract features of a given sample of examples, or storing large data samples in a data base, the heuristics and a relatively rich source domain govern the generalization process. Because of the fact that various generalizations are possible, a testing procedure needs to be implemented inspired by the idea of performing experiments. Two criteria are implemented in the underlying algorithm HDTP-A and seem to be crucial for the design of such experiments.

- Is the resulting theory logically consistent?
- Does the theory predict the right outcome of measurements of the observables?

Clearly the real situation is more complicated. If we leave the very broad qualitative modeling of the example towards the quantitative modeling (the linear relation between pressure and throughput), the situation changes. We have to distinguish between charge and current, because current is defined by charge per time. In this situation, water must be related to charge and not current. On the other side, Poiseuille's law must be constrained to the case of laminar flow (smooth flow). Therefore, the correctness of an analogical relation can be restricted by the range of certain parameters.[4]

We think that this dependency on parameters is important for all types of analogical learning. Concrete analogies are based on instantiations of more or less abstract concepts building a conceptual basis of the reasoning process. The task is to figure out which properties of the source and target domains are dependent on the particularly chosen instantiation and which properties are dependent on the underlying concepts. Whereas the latter ones are not explicitly considered

[4]As long as these parameters are in a certain range, the above analogy is able to make quite good predictions about fairly complex situations, for example, circuits of parallel and serial components.

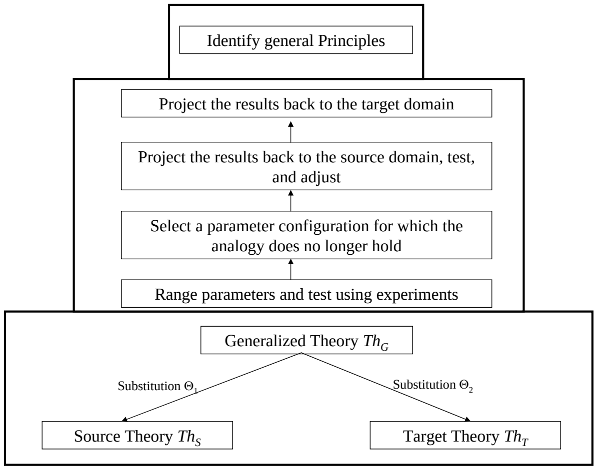

Figure 2: A graphical representation of learning stages.

to be under revision in HDTP, it is the first class of properties that need to be investigated and updated under appropriate circumstances.

The overall strategy of analogical learning in HDTP is summarized in a modular way in the following list:

- Given a source and a target input S and T, apply theory projection based on the algorithm HDTP-A in order to get a generalized theory for the source and the target together with corresponding substitutions.
- Find ranges of parameters such that theory projection can be established in a basic way using experiments as validation tool.
- Explore the boundaries of the ranges of parameter values for which the established analogy holds.[5]
- Select a parameter configuration on a boundary for which the analogy does not hold and project the conflicting results of the experiment back to the source domain.
- Use an inference machine on the source domain in order to adjust parameters to become consistent with the projected results of the experiment. This may require a refinement or extension of the source domain using a heuristics for relevant parameters.
- Project the refined or extended theory back to the target domain and test using experiments.

Notice that experiments in the field of qualitative physics are restricted to observables. In other words, what cannot be measured is not subject for considerations in a testing procedure. This idea governs also the inference machine used to adjust the parameters, because only things that are measurable functions in the particular modeling of the problem can be adjusted.

[5]The exploration of boundaries can loosely be identified with explorative learning (Watkins & Dayan, 1992).

Learning by Levels

Considering classical artificial systems containing a machine learning module, it is typically assumed that learning takes places in a special component of the system that is not inherently integrated into the system. Classical architectures result therefore in a learning device that is (more or less) independent from the other modules and learns explicitly form input data.[6] In our approach, learning occurs as a side-effect of the modeling and can be considered as implicit: The generalization process yields new conceptualizations of the target domain for nothing. Furthermore analogical learning in our modeling does not end with a successful establishment of an analogical relation. Learning continues in stages as was already sketched above. We will give an idea what that means by specifying three levels of learning (Figure 2):

- First level of learning: Finding the most specific generalization to establish the analogical relation.
- Second level of learning: Adjusting the parameters in order to find new (and finer) conceptualizations of the source and the target domains by projecting the new facts and rules in both directions.
- Third level of learning: Identifying general principles that can be applied in a variety of domains

The first step in our modeling is the task to find a generalization of the two input theories Th_S and Th_T. Because there are many possibilities for generalizations we introduced the idea of anti-instances that determine the most specific generalization. Finding such anti-instances is already a learning step: They are well-known in the ILP community as (relative) least general generalization (Plotkin, 1969; Muggleton & Feng, 1990). Clearly the differences between these ILP approaches and the presented account are significant, due to the aspect of HDTP to be theory-based. Notice that the space of possible generalizations is strongly restricted by the source and the target domain. Hence, the search for possible generalizations is governed by the overall conceptualization of the two domains and certain heuristics.

The second step in the learning process concerns the reliability of the generalization and the identification of the parameters telling us to which extent a certain analogy holds: Both aspects can be tested by performing experiments or – as in our case – by asking an oracle that functions as an abstract experiment generator. Clearly an experiment can fail, resulting in a rejection of an analogical relation. Then, a new search for a generalization has to be performed. In case the experiment supports the analogy, not only an analogy can be established, but also an explanation for the conceptualization of the target domain is found.

The third level of learning is the identification of general principles in physics. In our running example we would end up with general laws of thermodynamics, in the case of the Rutherford analogy (identifying revolving electrons with planets of a solar system) the principle would be the

[6]New AI is probably an approach trying to circumvent this idea of modularity. Nevertheless there does not seem to exist a holistic approach in New AI to higher cognitive capabilities that are able to model our domains.

equilibrium of forces (*actio* = *reactio*). Because of the generality of such principles the modeling can be extended to other domains as well. It is clear that this third level of learning presupposes further applications of our modeling.

Concluding Remarks

Analogical reasoning is a crucial ability of human cognition, because analogies can be used to explain many aspects of cognitive creativity, productivity, and adaptivity: In the field of natural language, the creative interpretation of metaphoric expressions are an important reason for semantic productivity and in the field of concept learning, the analogical transfer of knowledge to new domains can explain the power of human conceptualization. In short, learning by analogies is a crucial factor for the adaptivity of humans without large input data.

In our modeling of analogical reasoning using HDTP, learning occurs implicitly due to the generalization process (together with the substitutions). Learning aspects are not represented as additional modules that are somehow added to the analogical reasoning process, rather is learning a side-effect of analogy making. In other words, learning occurs implicitly as a part of a more complex reasoning process. This complex reasoning process can be divided into three main stages, starting with simple generalizations, continuing with an exploration of parameter settings, and ending in the establishment of general principles. In naive physics, for example, knowledge gathered from different domains is generalized to abstract principles like the law of the conservation of energy.

The proposed model presupposes crucially background knowledge: Without any conceptualization of the source domain learning is simply not possible. Hence, the question about the bootstrapping aspect remains open. Humans start to learn somehow, but the presented theory cannot give an explanation how. Nevertheless a large part of cognitive learning abilities seems to be covered by an approach of learning by analogies.

Acknowledgments

This research was partially supported by the grant KU 1949/2-1 of the German Research Foundations (DFG).

References

Aha, D., Kibler, D. & Albert, M. (1991). Instance-based learning algorithms, *Machine Learning*, 7:37-66.

Anderson, J. & Thompson, R. (1989). Use of analogy in a production system architecture, in Vosniadou, S. & Ortony, A. (eds.): Similarity and analogical reasoning, 267-297, Cambridge.

Bandura, A. & Walters, R. (1963). Social Learning and personality development. New York.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.

Chomsky, N. (1959). A Review of B. F. Skinner's Verbal Behavior. *Language*, 35(1):26-58.

Dastani, M. (1998). Languages of Perception, ILLC Disseration Series 1998-05, 1998, http://www.illc.uva.nl/Publications/Dissertations/DS-1998-05.text.ps.gz

Dastani, M., Scha, R. & Indurkhya, B. (1997). *An algebraic method for solving proportional analogy problems involving sequential patterns*, Mind II: Computational Models of Creative Cognition (Dublin, September 1997), pp. 1- 15.

Evans, T. (1968). A program for the solution of a class of geometric-analogy intelligence-test questions, in: M. Minsky (ed.), *Semantic information processing*, MIT press, pp. 271-353.

Falkenhainer, B., Forbus, K. & Gentner, D. (1989). The structure-mapping engine: Algorithm and example, *Artificial Intelligence*, 41(1):1-63.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy, *Cognitive Science*, 7:155-170.

Gentner, D. (1989). The mechanisms of analogical learning, in: S. Vosniadou & A. Ortony (editors): *Similarity and Analogical Reasoning*, New York, Cambridge University Press.

Gentner, D., Bowdle, B., Wolff, P. & Boronat, C. (2001). Metaphor is like analogy, in *The analogical mind: Perspectives from cognitive science*, editors: Gentner, Holyoak, Kokinov, Cambridge, MA 2001, pp. 199-253.

Gust, H., Kühnberger, K.-U. & Schmid, U. (2006a). Ontologies as a Cue for the Metaphorical Meaning of Technical Concepts, to appear in Andrea Schalley & Drew Khlentzos (Eds.): *Mental States: Evolution, Function, Nature*, John Benjamins Publishing Company, Amsterdam, Philadelphia.

Gust, H., Kühnberger, K.-U. & Schmid, U. (2006b). Metaphors and Heuristic-Driven Theory Projection (HDTP). *Theoretical Computer Science* 354(1):98-117.

Gust, H., Kühnberger, K.-U. & Schmid, U. (2003a). Solving Predictive Analogy Tasks with Anti-Unification. In: P. Slezak (ed.), *Proceedings of the Joint International Conference on Cognitive Science 2003* (ICCS/ASCS-2003), 145-150, Sydney, Australia.

Gust, H., Kühnberger, K.-U. & Schmid, U. (2003b). *Anti-unification of axiomatic systems*. Available on the www: http://www.cogsci.uni-osnabrueck.de/~helmar/analogy/.

Hofstadter, D. & The Fluid Analogies Research Group (1995). *Fluid concepts and creative analogies*, New York.

Indurkhya, B. (1992). *Metaphor and Cognition*, Dordrecht, The Netherlands, Kluver.

Kolodner, J. (1993). *Case-based reasoning*, San Mateo, CA, Morgan Kaufmann.

Lave, J. & Wenger, E. (1990). *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press.

Mitchell, T. (1982). Generalization as Search, *Artificial Intelligence*, 18(2):203-226.

Muggleton, S. & Feng, C. (1990). Efficient induction of logic programs, *First Conference on Algorithmic Learning Theory*, Omsha, pp. 368-381.

Plotkin, G. (1970). A note on inductive generalization. *Machine Intelligence* 5:153-163.

Quinlan, J. (1986). Induction of decision trees, *Machine Learning*, 1:81-106.

Rojas, R. (1996). *Neural Networks – A Systematic Introduction*. Springer, Berlin, New York.

Salzberg, S. (1991). A Nearest Hyperrectangle Learning Method, *Machine Learning*, 6(3):251-276.

Schmid, U., Gust, H., Kühnberger, K.-U. & Burghardt, J. (2003). An Algebraic Framework for Solving Proportional and Predictive Analogies, in Schmalhofer, F., Young, R., and Katz, G. (eds.): *Proceedings of EuroCogSci 03*. The European Cognitive Science Conference 2003, 295-300. Lawrence Erlbaum Associates.

Stanfill, C. & Waltz, D. (1986). Toward memory-based learning, *Communications of the ACM*, 29(12):1213-1228.

Watkins, C. & Dayan, P. (1992). Technical Note: Q-Learning, *Machine Learning*, 8(3):279-292.