10th Artificial Intelligence and Cognitive Science (AICS-99), UCC, pp 58-64, (1999).

Constraining Analogical Inference with Memory-based Verification

D. O'Donoghue, Dept. Computer Science, NUI Maynooth, Co. Kildare, Ireland.

Abstract

From, Analogies complete their primary task of generating new knowledge about a problem domain, based on a noted similarity between the problem and some other familiar domain. Analogical inferences are generated as a form of pattern completion occurring in the target domain, but not all comparisons between structurally similar domains generate valid inferences. We highlight the need for a domain independent verification mechanism. We describe an extended model of analogical reasoning that includes an explicit verification mechanism, employing existing memory to verify candidate inferences. We focus on the soundness of first-order predicates by comparing candidate inferences with background memory, with greater similarity leading to quicker and clearer verification.

Introduction

Analogical reasoning plays a central role in many cognitive processes affecting: problem solving (Gick and Holyoak, 1980), creativity (Boden, 1994), basic cognitive perceptions (Lakoff and Johnston, 1980) and especially learning (Duncker, 1945; Holyoak *et al*, 1994). However, analogy is primarily of interest as a workaday process, supporting inference in novel situations by comparison with past experience. Rather than reasoning from "first principles", analogical reasoning uses a noted similarity between some problem domain and a well-known one to infer useful facts about that problem domain.

Pattern completion is the ubiquitous model for analogical learning, and good analogies introduce a cluster of useful knowledge to the target domain. However, poor analogies do not support learning - or worse, they cause "negative learning" by adding incorrect or unsound information to the target domain. Markman (1997) and others point out that analogy is too profligate an inference mechanism, and constraints on inferences are necessary. Markman also notes that the one-to-one mapping constraint also acts to constrain the inference set, when n-to-m mappings might be generated. Because analogies use domains that are rarely fully isomorphic (Holyoak *et al*, 1994), computational models can easily over-generate inferences.

We propose a novel set of constraints on analogy that serve to constrain the inferences generated by an analogy. These constraints reject *unsound* inferences early during verification, making these constraints less expensive than previous pragmatic approaches to this problem. Furthermore, our memory based verification mechanism operates, like analogy, in a domain independent manner. First, we distinguish between the pragmatic utility and the soundness of an inference. Pragmatics have been shown to constrain the analogical mappings considered, under ambiguous structural conditions (Holyoak and Thagard, 1989; Hummel and Holyoak, 1997). However, inferences are still generated by a pattern completion process, extraneous source information can be carried over to the target. This generates unwanted inferences, but our simple validation mechanism rejects many of these inferences.

Specifically, inference rejection and acceptance requires a distinction between sound and unsound inferences. A sound inference is one that represents some reasonable belief about the external world, while an unsound inference attempts to represent information with no relevance to the real world. Consider the following analogy between a source <1 double 2> and <1 successor 2>, and the target domain <“a” successor “b”>. Inference by pattern completion generates the *grounded inference* (Gentner, 1983) <“a” double “b”>, and although systematicity theory mandates such an (unwelcome) inference, our verification process to rejects such inferences. Our verification process tests for successive degrees of similarity between the candidate inferences and the contents of long-term memory, in a simple domain-independent manner.

To highlight the difference between these approaches, pragmatic verification typically deals with systems of predicates, usually the candidate inference set and the pre-existing target domain. In stark contrast, our technique can determine the soundness of an individual (inferred) predicate, with reference only to background knowledge. Certifying the soundness of an inference halts the transfer of extraneous source material to the target. Finally, we point out the distinction between sound and true information, as a sound inference may actually be false. Consider an analogy between a last-minute cramming for an exam and preparing for a marathon. The obvious (and sound) inference creates a false understanding of the target domain. One must be careful as to which analogies are treated seriously, regardless of their plausibility.

Frameworks for Analogy

Since identifying the role of systematicity in analogy (Gentner, 1983) there has been much focused work on computational modelling of analogy, largely on identifying the inter-domain mapping. From this and other influences many larger frameworks for analogy research have arisen, and are typically multi-phase models operating primarily in a sequential manner. Many of these frameworks refer to an *evaluation* phase, but supply little detail on its operation. These frameworks are notable by their lack of an explicit verification activity, which operates on the candidate inferences mandated by an analogy. None propose a validation activity based on the soundness of candidate inferences.

For example, Kokinov (1994) identifies phases of *retrieval, mapping, transfer, evaluation* and *learning*; Holyoak and Thagard (1989) recognise *retrieval, mapping, transfer* and *subsequent learning;* Eskeridge (1994) recognises *retrieval, mapping* and *transfer and use*; Falkenhainer, Forbus and Gentner (1989) identify phases of *access, mapping,* and *evaluation and use.* Forbus, Gentner, Markman *et al* (1999) decompose analogy into *retrieval, mapping* (*alignment* and *projecting inference*) and *abstraction*. Hall (1989) compares models using phases of *recognition, elaboration, evaluation* and *consolidation*. Hummel and Holyoak's (1997) Lisa model encompasses phases of *access, mapping* and *induction*. However, throughout this paper we use Keane's (1994) framework as a reference point, and will later re-interpret its adaptation phase (Figure 1).

representation → retrieval → mapping → adaptation → induction

Figure 1 - Keane's Five Phase Model of Analogy

Of course pragmatic knowledge can server to reject inferences based on irrelevant aspects of the source, thereby focusing on the pragmatic utility of candidate inferences. The Phinneas model (Falkenhainer, 1988, 1990) takes this pragmatic approach, situating analogy within the context of physical modelling. By including a real world model Phinneas tests each inference against an "empirical envisionment" of that prediction. By use of a qualitative simulation, Phinneas compares analogical inferences to observed phenomena. This forms a reliable basis for verification but with considerable computation expense - a large section of Phinneas is devoted to this task for the physics domain alone (QPE and DATMI). Only sound inference need be considered by such an expensive operation. Holyoak, Novick and Melz (1994) also detail *evaluation/adaptation* along with *representation, retrieval, mapping, inference,* and *generalisation.* This evaluates candidate inferences, however it is also pragmatically based.

Analogical Inference and Pattern Completion

The ubiquitous scheme for analogical inference is as "pattern completion", and is realised by the CWSG algorithm - *Copy with Substitution and Generation* (Holyoak, Novick and Melz, 1994; Markman, 1997). When the inter-domain mapping identifies correspondences for all target entities, additional source structure can then be copied (with appropriate substitution) to the target domain. Semantically impoverished target domains leave source entities unmapped, and these must be generated in the target domain, to fulfil their role within that domain. In practice this can amount to positing the existence of skolem entities in the target domain (Falkenhainer *et al*, 1989), and this can lead to rampant inference generation for non-isomorphic domain pairs.

To reject unsound candidate inferences, we extended Keane's five phase model and introduce a domain independent verification sub-phase. We re-label this phase as validation because we see this as an activity makes valid all inferences that can be made valid, by a combination of verification and adaptation. This verification process relies heavily upon the contents of background knowledge, and specifically upon existing predicate structure. Many unsound candidate inferences may be rejected by comparison to predicates from a reasonable background memory store. This is the basis for our verification sub-phase (Figure 2). We shall not discuss the adaptation process, but merely wish to highlight it as a related phase.

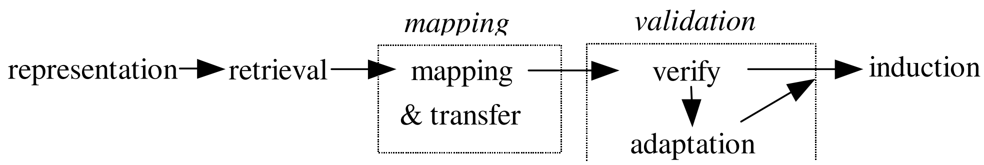

Figure 2 - An extended Model of the Validation Phase

Verification Sub-Phase

To develop the verification model, we adopt the quartic notation of Hofstadter (1995) and others and examine simple analogical inference; `A:B :: C:D` (read, A is-to B as C is-to D). As inference in Hofstadter's CopyCat is a form of pattern completion, inferences do not undergo the type of validation we propose. This notation identifies simple transfer-based analogies, and although they do not provide *grounded inferences* (Gentner, 1983) they can be extended to do so.

For example, comparing <man drive car> with the target <woman ? bus> (no connecting predicate) offers a verifiable inference, which might be grounded by adding the predicate <man inside car> to the source. The target <groceries inside car> is identified as dis-analogous because of the non-verifiable candidate inference <groceries drive car>. An adequate model of analogical reasoning should reflect peoples' ability to reject such non-analogical comparisons.

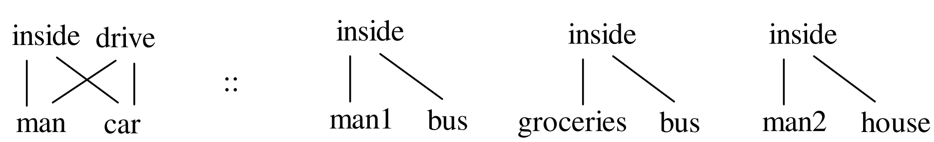

Figure 3 - **A source and three alternate targets**

Verification by Memory Matching

Verification may be achieved in a number of ways, depending upon the degree of overlap between the candidate inference and memory contents. Familiar inferences are more easily verified than less familiar ones. Here we focus on first-order predicates (between objects), where an identified source relationship is transferred to the target objects. The source domain identifies some relationship <A R1 B>, which we transfer to the target yielding <C R1 D>. For example, <2 double 4> applied to the target objects 3 and 6, yields <3 double 6>.

Our verification mechanism is sensitive to the commutativity of the inferred predicate, with additional verification mechanisms available to commutative predicates. Verification proceeds through a number of steps until one proves successful, beginning with the most specific and powerful, gradually uncovering extra information to aid verification.

1. Non-commutative Verification. Firstly, an existing instance of the inferred predicate <C R1 D> is sought from memory. If this predicate has previously been encountered, verification is deemed successful. Whether such a predicate is pragmatically useful in the target domain is left to processes beyond the scope of our validation phase. Thus the inference <x-ray go-down path> is only verified if previously encountered (our knowledge base being generally structure in an agent-patient ordering).

2. Commutative Verification. Commutative predicates may also be validated in a "piecemeal" fashion, so we search for either <C R1 D> or <D R1 C>, as commutative predicates can be validated with their arguments in any order. Thus, for <man next-to door> :: <house ? tree>; <tree next-to house> may be validated against itself and also against <house next-to tree>.

3. Partial-Predicate Verification. Many inferences form novel combinations of a predicate and its arguments. However, many novel inferences may be successfully verified in a piecemeal fashion. Validating the agent and patient roles separately, we may validate <C R1 ?> and <? R1 D>, with commutative predicates being validated in either agent-patient or patient-agent order.

Thus, <x-ray go-down path> may be validated if we have encountered the partial predicates; <x-ray go-down ?> and <? go-down path>. In other words, verification is achieved if we know an x-ray is something which fills the agent role of go-down and a path can fill its patient role. This would validate the "oesophagus" solution to the tumour problem, which Duncker (1945) regarded as "genuinely the solution of a problem", although its not the required one (we return to the issue in later section). This looser form of verification would allow <3 double 50> as a candidate inference, but this can be rejected by the later application of domain knowledge. However, verification would still reject the analogy <1 double 2> :: <"a" ? "b"> on the basis of the candidate inference <"a" double "b">.

4. Similarity Verification. Some support for verification may also be achieved by application of a similarity metric (Tversky, 1977) between the candidate inference and the "most similar" contents of memory. The inference <monkey drive motor-car> could be validated (against a reasonable knowledge base) only if "monkey" was sufficiently similar to some man, but "groceries" would never be sufficiently similar. Because verification interacts heavily with adaptation, the degree of similarity sought can depend upon how an inference is adapted. We do not describe adaptation herein.

This technique seems to account adequately for inferences that conform to previous usage of the relevant predicate. How then can it account for more novel usage of predicates, which introduce a new interpretation of a predicate or present a new aspect of some object? We argue that many of these inferences are the same ones that people have difficulty with, and typically require additional support. New meanings for polysemous predicates may be added and can be differentiated by their argument types, such as learning that the bounce predicate may take a cheque as well as a ball as an argument. Alternatively, introducing a metal that can flow along a glass tube might introduce highlighting the unique properties of mercury, or introducing a mammal that flies (bat), and birds that can't (emu).

Analogical Inference

We now return to Dunckers' (1945) analogy between a tumour and a fortress, which is centred on the inference that x-ray beams travel along "paths" before they converge to destroy the tumour. This convergence solution can only be generated if the inference <x-ray go-down oesophagus> is first rejected (Dunckers' protocol 1) - to make way for the required inference. Although rejected by the first two verification phases, partial predicate verification would accept such an inference. Thus, while the required inference is accepted, the "unwanted" inference is not first rejected by our verification method. As stated, Duncker regarded this as a (not prefereable) solution to the problem, so we contend that an analogy "use" phase is required to reject this inference. Clearly, not all unwanted inferences can be rejected by a simple verification scheme, some complex inferences require more complex reasoning, such as that of Phinneas (Falkenhainer, 1988).

Conclusion

Analogical verification is necessary to reject comparisons such as that between <1 double 2> and <"a" ? "b">, while accepting valid comparisons between <1 successor 2> and <"a" ? "b">. We present a model of domain independent memory-based verification, which ensures the structural soundness of candidate inferences. This enables a model to discriminate between a large number many analogical and non-analogical comparisons. Certifying the soundness of an inference stops inappropriate source material being transferred to the target - a necessary constraint because domain boundaries are rarely well-defined. These constraints greatly increase the inferential acuity of analogical reasoning models.

References

Boden, M. A. “The Creative Mind”, Abacus, 1994.

Duncker, K. “On problem solving”, Psychological Monographs, 58 (whole, no. 270), 1945.

Falkenhainer, B. “A unified approach to Explanation and Theory Formation”, In Computational Models of Scientific Discovery and Theory Formation, 1990.

Falkenhainer, B. Forbus, Gentner, D. “The Structure Mapping Engine: Algorithm and Examples”, Artificial Intelligence, 41, 1-63, 1989.

Falkenhainer, B. “Learning from Physical Analogies: A study in analogy and the explanation process”, PhD Thesis, Univ. Illinois at Urbana-Champaign, 1988.

Forbus, K. Gentner, D. Markman, A. Ferguson, R. “Analogy just looks like high level perception”, Journal of experimental and Theoretical Artificial Intelligence (to appear, 1999).

Gentner, D.“Structure-Mapping: A Theoretical Framework for Analogy”, Cognitive Science, 7, 155-170, 1983.

Gick, M. Holyoak, K. “Analogical Problem Solving”, Cognitive Psychology, 12, 306-355, 1980.

Gick, M. Holyoak, K. “Schema induction and Analogical Transfer”, Cognitive Psychology, 15, 1-38, 1983.

Hofstadter, D. “Fluid concepts and creative analogies”, Basic Book, NY, 1995.

Holyoak K. J. Novick L. Melz E. “Component processes in Analogical Transfer: Mapping, Pattern completion and Adaptation”, in Analogy, Metaphor and Reminding, Eds. Barnden and Holyoak, Ablex, Norwood, NJ: 1994.

Hummel, J. E. Holyoak, K. J. “Distributed Representation of Structure: A Theory of Analogical Access and Mapping”, Psychological Review, 104, 3, 427-466, 1997.

Keane, M. T. Ledgeway, T., Duff, S. “Constraints on Analogical Mapping: A comparison of Three Models”, Cognitive Science, 18, 387-438, 1994.

Kokinov, B. N. “A Hybrid Model of Reasoning by Analogy”, in Analogy, Metaphor and Reminding, Eds. Barnden and Holyoak, Ablex,. Norwood, NJ: 1994.

Lakoff, G. Johnson, M. “Metaphors We Live By”, Chicago, Illinois: University of Chicago Press, 1980.

Markman, A. “Constraints on Analogical Inference”, Cognitive Science, 21, 4, 373-418, 1997.

Tversky, A. “Features of Similarity”, Psychological Review, 84, 327-352, 1977.