Structure Mapping Theory

More old notes to be edited later... so many of these to be done...

In analogy, the relations between objects (e.g. friend(Bill,Ted)) are matched rather than the individual attributes (e.g. tall(Bill)) of those objects. This is distinct from literal similarity where both attributes and relations are matched, mere-appearance matches where primarily attributes are matched, and anomaly where very few attribute or relations are matched. Metaphor (e.g. 'Bill is a rock') is seen as a reduced form of analogy with usually just one (contextual?) attribute being matched. Abstraction is seen as a similar process to analogy but with mappings between objects that have few or no attributes, and in abstraction, all the relations are matched rather than just some. This view of analogy appears to be confirmed by empirical psychological studies (Falkenhainer89).

Analogy can be split into three distinct subprocesses (Falkenhainer87.IJCAI): (i) access, (ii) mapping and inference and (iii) evaluation and use. Before processing begins, it is assumed that there is a current situation of interest (the target). Access finds a body of knowledge (the base) that is analogous or similar to the current target. Mapping finds similarities or correspondences (mappings) between the base and target, and may also transfer information (inference) from the base to the target. Evaluation of the analogy gives a measure or estimate of the quality of match in the context of other knowledge about the target's domain, against three different types (structural, validity and relevance) of quality criteria. Structural criteria include the degree of structural similarity, the number of similarities and differences and the amount and type of knowledge added to the target description (the candidate inferences) in using this analogy. Validity checks that any new knowledge makes sense within the current knowledge domain. Relevance assesses whether the analogy and candidate inferences are useful to the current task or aim of the system. Analogies thus formed are used for analogical reasoning, similarity-based generalisation, or analogical learning.

Structure mapping theory (SMT, Gentner83) asserts that an analogy is the application of a relational structure that normally applies in one knowledge domain (the base domain) to another, different, knowledge domain (the target domain); unlike less-structural psychological theories, it also sees analogy and similarity as connected processes. It uses graph-matching to describe the constraints that people use in creating analogies and interpreting similarity. SMT does not capture the full richness of analogy, but is primarily interested in, and applied to, the mapping and a subset of the evaluation subprocesses described above (e.g. it does not include the access subprocess, and uses structural criteria only).

The core of structure mapping algorithms is finding the biggest common isomorphic pair of subgraphs in two semantic structures [Veale98]. Much of the work in this field has concentrated on subgraph heuristics to produce near-optimal matches in polynomial time. All of it is based on or related to SMT. In SMT, graph matches are assumed to be exact correspondences only (structurally consistent); this constraint may need updating to better reflect the inexact matches and representations used in human analogical reasoning. Note that there are two matching processes in play here: the mapping of objects (graph nodes), and the mapping of relations (link labels), that these two processes are often run at the same time, and that the relational matches may determine the matches between objects. SMT is also tidy: attributes are discarded unless they are actively involved in a match, and higher-order relations and interconnections are more likely to be preserved (systematicity, Gentner83).

SMT underlies much work on computational models of analogy, metaphor, case-based reasoning and example-based machine translation [Veale98]. SMT is used for analogical learning (Falkenhainer87.IJCAI, JonesLangley95) and information retrieval. Analogical learning uses analogy to generate new knowledge about a domain; examples include the extension of the ARCS/ACME structure-matching programs with algorithms from the Eureka learning system. Similarity-based retrieval models include MAC/FAC (many are called but few are chosen, Gentner95); this is a two-stage algorithm, where the MAC stage is a rough filter, and the FAC stage is performed by a structure-matching engine (SME).

Implementations of SMT include the Structure Mapping Engine (SME, Falkenhainer89), SAPPER, ACME and ARCS. SME represents each objects as a description group (dgroup), a list of items (entities and expressions) associated with that object, where functions, attributes and relations within these items are represented as Prolog-style clauses. Each dgroup is mapped onto a graph with a node for every item in it, and an arc from every item i to any item j which is used as an argument to i.
The SME algorithm is divided into four stages: Local match construction, Gmap construction, Candidate inference construction and Match evaluation. A local match is a pair of base and target items which could potentially match; in local match construction, a match hypothesis is created for each of these pairs. SME does not create a match hypothesis for every possible pair of base and target items, but uses a set of rules (for instance, only matching items that appear in the same expression) to filter them. Which rules are used here determine whether the output of SME will be analogy, literal similarity or mere appearance. Because they are based on dgroups, match hypotheses also form a directed acyclic graph, with exact matches fixed only between predicates. GMAP construction is the heart of SME processing. Here, the local matches are combined into gmaps, where a global mapping (gmap) is a structurally consistent mapping between the base and target object, consisting of correspondences between the items and entities in their description groups, candidate inferences and a structural evaluation score. Candidate inferences are the structurally grounded inferences in the target domain suggested by each gmap, where the substitutions used in structurally grounded inferences are consistent with the existing target information, and there is an intersection between their ancestor nodes and the network representing the target domain. In match evaluation, a score is given for each local match, using a set of rules (for instance, expressions with the same functors are assigned a score of 0.5). These scores are combined using Dempster-Shafer theory (in a Belief Maintenance System) to give a score for each match hypothesis. The scores for all the match hypotheses that contribute to an

individual gmap are then summed to give a structural evaluation score for that gmap.

Veale's SAPPER system [Veale98] represents each domain as a graph in which nodes represent concepts (e.g. SME's items) and edges between nodes represent relations between those concepts. The algorithm uses much simpler rules than SME. It first looks for common nodes between the two graphs being compared, then uses two simple rules to update the match: the triangulation rule and the squaring rule. This creates a partial match pmap for the domain, and is a spreading-activation algorithm that Veale claims is more efficient than the corresponding part of SME [Veale97]. The pmaps formed are graded by richness, then the algorithm attempts to combine all the pmaps, starting with the richest as a base.

Contenders to SMT for modelling analogy include high-level perception (Hofstadter's CopyCat) and theories of semantic/pragmatic constraints. High-Level Perception emphasises analogy as a mixture of representation and mapping, and treats data gaps in a similar manner to recent work on protein structure alignment. Holyoak and Thagard's work on the Analogical Constraint Mapping Engine (ACME) and ARCS systems assume that analogy is goal-driven; Holyoak [Holyoak89] argues that the SMT view of analogy only addresses the structural constraints on a mapping, and that semantic and pragmatic constraints should be included in the mapping too.

Where an analogy is drawn between two separate areas of knowledge, then the use of information fusion by them is simple: the joining of the information held in the two pieces of knowledge into a coherent whole. More complex forms of creativity combine together several pieces of knowledge; this is where information fusion techniques will be most useful and provide most insight into the process.