Hummel, J. E., Landy, D. H., & Devnich, D. (2008). Toward a process model of explanation with implications for the binding problem. In *Naturally Inspired AI: Papers from the AAAI Fall Symposium.*Technical Report FS-08-06, 79-86.

Toward a Process Model of Explanation with Implications for the Type-Token Problem

John E. Hummel **David H. Landy** **Derek Devnich**

Department of Psychology, University of Illinois
603 E. Daniel St., Champaign, IL, 61820, USA
Corresponding Author: jehummel@uiuc.edu

Abstract

The ability to generate explanations plays a central role in human cognition. Generating explanations requires a deep conceptual understanding of the domain in question and tremendous flexibility in the way concepts are accessed and used. Together, these requirements constitute challenging design requirements for a model of explanation. We describe our progress toward providing a such a model, based on the LISA model of analogical inference (Hummel & Holyoak, 1997, 2003). We augment LISA with a novel representation of causal relations, and with an ability to flexibly combine knowledge from multiple sources in LTM without falling victim to the type-token problem. We demonstrate how the resulting model can serve as a starting point for an explicit process model of explanation.

Keywords: Explanation; analogy; LISA.

Explanation and Understanding

People are constantly seeking, generating and evaluating explanations (Keil, 2006; Sloman, 2005; Thagard, 1989). As anyone who has ever given an essay exam knows, the ability to explain is a powerful index of understanding. Explanation also plays a critical role in problem solving. In order to solve an automotive problem, for example, it is first necessary to understand (i.e., explain) the nature of the problem.

Although there exists a relatively rich literature on how people evaluate explanations (see Keil, 2006; Lombrozo and Carey, 2006; Thagard, 2001), comparatively little is known about how people generate them in the first place (for progress in this direction, see Ahn et al., 1987; Patalano, Chin-Parker & Ross, 2006; VanLehn, Jones & Chi, 1992; Vosniadou & Brewer, 1987). This paper presents our early attempts to understand, at a detailed algorithmic level, the cognitive operations that underlie our ability to generate explanations. As the empirical literature on this question is comparatively thin, our starting point is one of first principles: What do we know about how people generate explanations, and how can those facts constrain our modeling?

We will assume that generating an explanation involves inferring a causal chain or tree leading from some hypothesized or believed initial state of affairs to the explanandum (Pearl, 2000; Sloman, 2005).

One thing we know about inferring explanations is that explanation depends on our ability to flexibly access and apply our existing knowledge (Ahn et al., 1987; Vosniadou & Brewer, 1987). The flexibility is central, as illustrated by an experiment by Patalano et al. (2006). In one condition, Patalano et al. gave subjects a novel explanandum of the form "In the population as a whole, people tend to prefer Pepsi to Coke as often as they prefer Coke to Pepsi. However, ministers tend to prefer Coke over Pepsi," and asked them to generate an explanation for this "fact". One of the explanations subjects typically generated took the general form: "Ministers tend to be conservative. Perhaps the Coke Corporation supports conservative causes." This explanation reflects a combination of knowledge about ministers, corporations and the kinds of factors that can lead a person to prefer one product to another, and reflects tremendous flexibility in the way that knowledge is assessed and combined.

Flexibility and Knowledge Representation

The way we generate explanations suggests three kinds of flexibility in the representations and processes underlying those explanations. The first is *relational* flexibility. For example, one way to account for the "conservative causes" explanation above is to assume that the subject has some sort of knowledge structure specifying that if some person agrees with the political leanings of some company then, all other things being equal, that person will tend to prefer the products produced by that company. Such a schema needs to be relationally flexible in the sense that it needs to be variablized (i.e., symbolic; see Hummel & Holyoak, 2003), so that, in the limit, it can be used to reason about any person, product and company.

Second, explanation requires *semantic* flexibility so that it can exploit partial but imperfect matches between the objects and relations composing an explanandum and the objects and relations encoded in potentially relevant schemas or examples in long-term memory (LTM). For example, imagine that our experimental subject did not have a "product preference schema" but did know of a prior case in which her friend preferred to use a particular cell phone company because of their liberal-leaning political activism.

The subject could use this prior example as a *source* analog (Gentner, 1983; Holyoak & Thagard, 1989) with which to reason about the situation with ministers and Coke; but they could only do so if their mental representations of the situations allowed them to tolerate the semantic differences between their friend, the cell phone company and the cell phone service on the one hand and ministers, the Coca Cola Corporation and Coke on the other (Hummel & Holyoak, 1997).

These same kinds of flexibility also characterize human reasoning using analogies, schemas and rules (Holyoak & Thagard, 1989, 1995; Hummel & Holyoak, 1997, 2003). Accordingly, as elaborated below, the point of departure for our attempt to simulate explanation is a model of analogy, relational reasoning and schema induction—namely, Hummel and Holyoak's (1997, 2003) LISA model.

Beyond the Flexibility of Analogy Explanation also requires a third kind of flexibility not exhibited by extant models of analogy (including LISA). Analogy is typically construed as a process of reasoning about a novel target problem or domain in terms of a familiar source (or base) domain (Gentner, 1983; Gick & Holyoak, 1983; Holyoak & Thagard, 1989). For example, in the analogy between the solar system and the Rutherford model of the atom, the solar system serves as the source, guiding inferences about the atom as the target (e.g., the inference that some force must cause the electrons to orbit the nucleus). Importantly, both in this example and in extant models of analogy, the mapping and inference are driven from a single source to a single target. This restriction that one source maps to one target greatly reduces the complexity of analogical reasoning by placing strong constraints on the critical step of analogical mapping—the process of discovering the correspondences between the elements of the source and those of the target.

Things are not so tidy in the case of explanation. Generating an explanation often requires integrating information from *multiple* sources in LTM. Returning to our ministers and Coke example, the reasoner likely has one schema (or set of schemas) describing the properties of ministers, another schema describing the conditions under which one's political leanings might lead to particular product preferences, and still a third schema describing what it means for one person (e.g., a minister) to agree with another person or entity (e.g., the Coke Corporation). In order to generate the "supports conservative causes" explanation for why ministers might prefer Coke, it is necessary to integrate these diverse sources of knowledge, somehow keeping track of what corresponds to what within and between the explanandum and the various schemas.

The Type-Token Problem Integrating multiple sources of information in the service of explanation thus requires solving a variant of the type-token problem in perception and cognition—specifically, the problem of knowing whether two or more representational elements (tokens) have the same referent (i.e., object or type in the world). Extant models of analogy (including SME, Falkenhainer, Forbus & Gentner, 1989; ACME, Holyoak & Thagard, 1989; LISA, Hummel & Holyoak, 1997, 2003; and CAB, Larkey & Love, 2002) solve this problem by (a) designating one token per element (object or relation) in each analog (source or target), (b) mapping exactly one source onto exactly one target, (c) restricting analogical mappings to tokens in separate analogs (e.g., tokens in the source can map to tokens in the target but not to other tokens in the source) and (d) imposing a (more or less strict) 1:1 mapping constraint such that each token in one analog may map to at most one unit in the other. These constraints avoid the type-token problem by making it unnecessary to worry about whether different tokens refer to the same entity: tokens across analogs either map or not as dictated by the structure of the analogy. However, the reason they work is precisely because one target is mapped to exactly one source at a time.

These constraints cease to be adequate when a single target (e.g., an explanandum) needs to map to multiple sources (e.g., schemas) in LTM. In the minister example, the token representing the minister in the explanandum must map to one token in, for example, the "product-preference" schema and to a different token in the "agreement" schema: Given this, how is the system to "know" that the token in the product preference schema that maps to the minister has the same referent as the token in the agreement schema that maps to the minister? The difficulty of this problem is both exacerbated and illustrated by the fact that, in some other explanation, these tokens might *not* have the same referent.

In short, the 1:1 mapping constraint, which is necessary to make analogical mappings in a psychologically realistic way, must be violable when integrating information from multiple information sources. As elaborated below, we present a solution to this problem that works by serializing the mapping of the explanandum onto the various schemas (and other knowledge structures) in LTM: Effectively, this approach "solves" the type-token problem by replacing the question *Do these tokens refer to the same entity?* with the question *Do these tokens map to one another in the current context?*

A Process Model of Explanation

Knowledge Representation As noted previously, the point of departure for our effort is Hummel and Holyoak's (1997, 2003) LISA model of analogical reasoning. LISA is an artificial neural network whose representations and processes are rendered symbolic (i.e., explicitly relational) by virtue of its solution to the problem of dynamically binding relational roles to their fillers. LISA represents propositions (such as *prefer* (ministers, Coke)) using a hierarchy of distributed and progressively more localist codes (Figure 1). At the bottom of the hierarchy objects and relational roles are represented as patterns of activation distributed over units coding for their semantic features (small circles in Figure 1).

At the next level, objects and roles are represented by localist *object* and *role* units (large circles and triangles in Figure 1), which share bi-directional excitatory connections with the semantic units describing them. For example, the object unit *minister* might share connections with semantics such as *human*, *adult*, *religious*, etc. Role-filler bindings are encoded by *sub-proposition* units (SPs; rectangles in Figure 1), which share bi-directional excitatory connections with the object and role units they bind together. At the top of the hierarchy, *proposition* (P) units (ovals in Figure 1) bind individual role bindings (SPs) together into complete propositions.

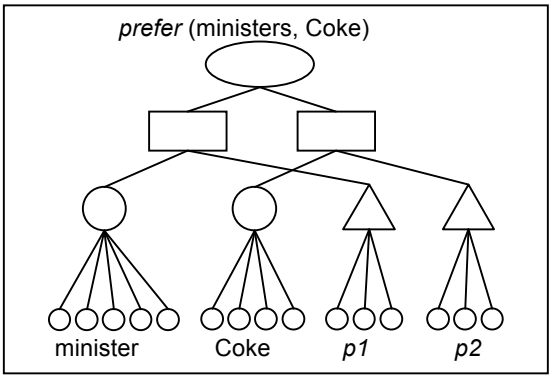

Figure 1. Representation of *prefer* (ministers, Coke) in LISA. Small circles are semantic units; triangles are role units; large circles are object units; rectangles are SPs and the oval is a P unit. Lines are excitatory connections. See text for details.

The hierarchy depicted in Figure 1 represents propositions both in LISA's LTM and, when a proposition becomes active, in its working memory (WM). In LTM, a proposition's role bindings are represented strictly by the conjunctive SPs. However, this kind of conjunctive code is inadequate as a general solution to the binding problem in WM (Hummel & Holyoak, 1997). When a proposition becomes active (i.e., enters WM) its role bindings are represented both conjunctively by the SPs and *dynamically*, by synchrony of firing: The separate SPs composing a proposition inhibit one another, and so fire out of synchrony with one another. As a result, relational roles fire in synchrony with the fillers to which they are bound, and separate role-filler bindings firing out of synchrony with one another. The result, on the semantic units, is a collection of mutually desynchronized distributed patterns of activation, one for each role-filler binding. These representations have the property that they represent relational roles and their arguments independently of one another (i.e., the same units will represent a given object or relational role, regardless of the role or object to which it happens to be bound at the time) and simultaneously specify how roles are bound to their fillers. They are therefore both distributed and explicitly relational, i.e., symbolic (see Hummel & Holyoak, 1997).

LISA's knowledge representations are compartmentalized into "analogs": Collections of propositions that together represent individual events, stories, concepts or schemas. Within an analog, a given object or role is represented by a single unit across all proposition in which it plays a role. However, separate analogs do not share object, role, SP or P units: A given object or role is represented by one unit in one analog and by a different unit in another analog. As such, object and role units do not represent objects or roles in the abstract; they represent specific instantiations or tokens of those objects or roles in specific analogs. (The same is true of SP and P units.) As such, we will refer to object, role, SP and P units collectively as *token units*. In contrast to the token units, all analogs connect to the same pool of semantic units. The semantic units thus represent the abstract *types* to which the tokens refer. (Albeit crucial for the various functions LISA performs, this division between type and token units is not sufficient, by itself, to solve the type-token problem described above; indeed, it causes that problem.)

For the purposes of LISA's operation, analogs are divided into three sets: a *driver* and one or more *recipients* are assumed to reside in *active memory* (a primed subset of LTM that is larger than WM; Cowan, 2001); the remainder are dormant in LTM. All of LISA's operations are controlled by the driver. One (or at most three) at a time, propositions in the driver become active and enter the *phase set*: The set of active but mutually de-synchronized role bindings. The phase set is LISA's WM, and like human WM (see Cowan, 2001), is limited to at most 4-6 role bindings at a time. The patterns of activation that propositions in the phase set generate on the semantic units excite other propositions in LISA's LTM (for memory retrieval) and in its active memory (for mapping, analogical inference and schema induction) and thereby bootstrap all the functions LISA performs.

Processing in LISA LISA performs memory retrieval as a form of guided pattern recognition (Hummel & Holyoak, 1997): Patterns of activation generated on the semantic units by one proposition tend to activate other, similar, propositions in LTM, retrieving them into active memory. For example, the patterns activated by the proposition *prefer* (ministers, Coke) might activate the proposition *prefer* (person, product) in the "product preference" schema.

Augmented with a simple algorithm for learning which structures in the recipient tend to become active in response to which in the driver, LISA's memory retrieval algorithm serves as a basis for analogical mapping: In this trivial analogy, *ministers* bound to *prefer-agent* activates *person* bound to *prefer-agent* in the schema and *Coke* bound to *preferred-object* activates *product* bound to *preferred-object*; LISA thus maps *ministers* to *person* and *Coke* to *product*. The same is true for corresponding roles of the *prefer* relation, and the SP and P units binding those roles to their fillers.

LISA represents these correspondences as learned *mapping connections* between corresponding structures

(e.g., between *ministers* and *person*, etc.). These connections serve not only to represent the learned mappings, but also to constrain future mappings: If LISA maps *ministers* to *person* in the context of *prefer*, then the resulting mapping connection will cause *ministers* to directly activate (and therefore map to) *person* in subsequent propositions. The learned mapping connections also play a central role in LISA's capacity for *self-supervised learning*—the core of its algorithm for analogical inference and schema induction (Hummel & Holyoak, 2003).

One of the main adaptive functions of analogical thinking is that it supports *analogical inferences* a.k.a. *relational generalization*: inferences and generalizations based on the relational roles that objects play, rather than just the literal similarity of the objects themselves. LISA's mapping algorithm is capable of exploiting the full power of relational thinking, mapping utterly dissimilar objects (and roles) to one another provided they are bound to similar roles (or, in the case of dissimilar roles, are bound to objects that are known to correspond based on an earlier mapping). In the case of the current example, once LISA maps *ministers* to *person* and *Coke* to *product* (along with their roles), it is then prepared to "copy with substitution and generation" (Holyoak & Thagard, 1989) the structure of the entire "product preference schema" over onto the "minister and Coke" situation, effectively filling in a (partial) explanation for why ministers prefer Coke. Through a novel process of repeated cycles of retrieval, mapping, and inference (elaborated below), the model is able to overcome the 1:1 mapping constraint to integrate multiple sources of knowledge through sequential analogical inference, and effectively side-step the type-token problem.

Finally, augmented with a simple algorithm for intersection discovery, LISA's algorithm for analogical inference also provides a very natural account of the induction of abstract schemas (such as the product preference schema) from concrete examples (such as the minister and Coke example and the cell phone example) (Hummel & Holyoak, 2003).

LISA's knowledge representations ("LISAese"), along with its algorithms for memory retrieval, mapping, inference and schema induction, provide a natural account of roughly 50 phenomena in the literature on analogical thinking, as well as 15 or more in cognitive development (see Doumas, et al., 2008; Hummel and Holyoak, 1997, 2003; Hummel & Ross, 2006; Morrison et al., 2004; Richland et al., 2006; Viskontas et al., 2004). These abilities derive from the fact that LISAese simultaneously enjoys the flexibility of distributed representations and the relational sophistication of symbolic representations. As such, they are an ideal platform on which to build a model of understanding and explanation.

Representing Causal Relations

Consider a set of propositions that together might form a "product preference" schema:

P1: *agree-with* (person, corporation)
P2: *produce* (corporation, product)
P3: *prefer* (person, product)

and another set of propositions that might form an "agreement" schema:

P1: *believe* (person, some-proposition)
P2: *believe* (entity, some-proposition)
P3: *agree-with* (person, entity)

(Recall that, because they reside in separate "analogs [in this case, schemas], *person* in the product preference schema is a different token than *person* in the agreement schema, even though they have the same name.) Assuming these propositions constitute reasonable caricatures of the preference and agreement schemas, then they are clearly causally related to one another. Specifically, P1 and P2 (*agree-with* and *produce*) in the preference schema jointly cause P3 (*prefer*), and P1 and P2 (*believe*) in the agreement schema jointly cause P3 (*agree-with*). How should these causal relations be represented for the purposes of generating explanations?

One straightforward approach would be to represent them as explicit propositions, for example:

P4: *and* (P1, P2)
P5: *cause* (P4, P3)

LISAese makes it possible for one proposition to take another as an argument, so this approach to representing the causal relations in a schema is perfectly plausible; and in some circumstances (e.g., when thinking explicitly enough about a causal relation to write about it), people can undoubtedly do so. However, LISAese assumes that explicit propositions are represented in WM and therefore consume finite WM capacity. As such, we suggest that this approach is likely to be too unwieldy to serve as a general solution to the problem of representing causal relations for the purposes of explanation: Note that P4 and P5 collectively introduce four additional role bindings into each schema; that's eight additional role bindings that would need to occupy slots (although not all at the same time) in our intrinsically capacity-limited WM. It seems intuitive that, although we are aware of the causal relations, and can name them when asked, we do not necessarily think so explicitly about them in the service of generating an explanation.

Alternatively, we could represent causal relations in an entirely implicit fashion, for example as associative links whose weights indicate causal strength. This approach would eliminate the WM problem caused by the explicit propositions, but it goes too far in the opposite direction, representing the causal relations only as implicit links rather than explicit structures that can be activated, analogically mapped and ultimately inferred (e.g., by analogical inference) into the emerging explanation.

We propose a third alternative: To represent groups of related propositions by connecting them to units that explicitly represent those groups (diamonds in Figure 2). For example, the fact that P1 and P2 in the agreement schema (the *believe* relations) jointly cause P3 (the *agree*

relation) can be represented by connecting P1 and P2 to a single *group* unit, and tagging that group as a *cause group* by connecting it to semantic units representing *cause*. Likewise, the fact that P3 is an effect can be represented by connecting it to a group unit, and connecting that unit to semantic units representing *effect*. Finally, the fact that the P1/P2 group is the cause of the P3 group can be represented by connecting the cause and effect group units to a higher-level *cause-effect* (CE) group unit. This latter unit represents the strength of the causal relation by connecting to semantic units coding for that strength.

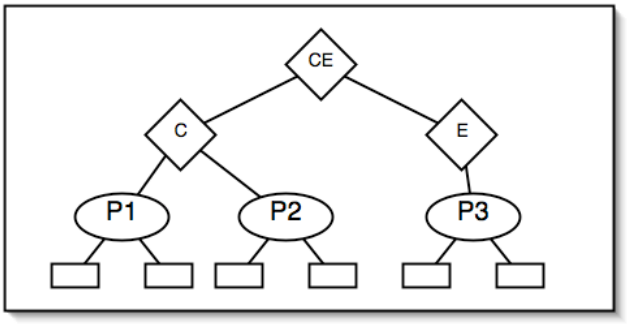

Figure 2. Illustration of group units (diamonds) as a representation of causal relations. Ovals are P units, rectangles are SPs. Objects, roles and semantic units are not shown. The upper-most diamond is a *cause-effect* group (CE); the groups below it to its left is a *cause* group (C); the one below it to its right is an *effect* group (E). P units are numbered as in the text above.

The resulting representation is more explicit than simply representing causal relations as associative links: causal relations are represented as collections of units that can be activated, mapped and inferred. But at the same time, it is less WM-demanding than explicit propositions: group units incur no additional WM burden over and above the propositions they link as causally related.[1]

Group units also serve to organize LISA's knowledge into meaningful packages (including but not limited to causes, effects, and cause-effect pairings). As a result, they play a central role in determining which propositions are likely to become active in close temporal proximity (i.e., what LISA is likely to "think about" in what order; see Hummel & Holyoak, 1997, 2003). Specifically, LISA's processing is constrained to activate propositions in group-based sets (in the driver), and it is constrained to retrieve propositions from LTM in group-based sets. As a result, if LISA is reminded of a familiar effect (e.g., some novel explanandum activates a proposition in LTM connected to an effect group), then it will tend to be reminded of the cause as well (via the shared CE group). In other words, group units not only play an important role in LISA's representation of causal relations; they also play a key role in metacognitive aspects of its operation, controlling what LISA "thinks about" together, and controlling what it is reminded of together.

Flow of Control

Armed with group-augmented LISAese, LISA's algorithm for explanation operates according to a retrieve-map-infer cycle that is applied iteratively to construct a causal chain representing an explanation of the explanandum. (The same retrieve-map-infer process also characterizes reasoning by analogy [see, e.g., Gentner, 1983]; but in analogical reasoning, it is performed only once, not iteratively.) This process is initiated by placing the proposition(s) representing the explanandum into the driver, connected to an isolated effect group (i.e., an effect group with no parent CE group and no sibling cause group). All of LISA's other knowledge resides dormant in LTM. In the case of our ministers and Coke example, the driver would contain the proposition *prefer* (ministers, Coke) connected to an effect group; LTM would contain all of LISA's other knowledge, including (most crucially for the current example) the preference and agreement schemas.

LISA initiates the explanation process by activating the proposition (and the effect group) in the attempt to retrieve a relevant schema or prior example from LTM. In the case of the current example, the isolated effect group activates the *effect* semantic, and the *prefer* proposition activates the semantics of *ministers+prefer-agent* and *Coke+preferred-object* (the semantics of *ministers* fire in synchrony with those of *prefer-agent* and out of synchrony with those of *Coke* and *preferred-object*, but all these units fire in synchrony with *effect*). The resulting patterns of activation on the semantic units represent the query *Why do ministers prefer Coke?*, and tend to activate effect groups (via the *effect* semantic) connected to semantically similar propositions in LTM (via the semantics connected to the proposition).

Groups in LTM are retrieved into active memory stochastically as a function of how active they become in response to the patterns generated by the driver. In the current example, P3 in the preference schema, *prefer* (person, product), is likely to be retrieved. Retrieval is group-based, with a bias in favor of retrieving higher-level groups over lower-level groups (e.g., CE groups rather than isolated cause or effect groups). As a result, the activation of P3 in the preference schema is likely to result in the retrieval of the whole product preference schema. (For convenience, we illustrate flow of control in LISA using the preference and agreement schemas, but the logic would be exactly the same if, instead of schemas, LISA had analogous specific examples, e.g., the cell phone example instead of the preference schema.)

If LISA fails to retrieve a CE group from LTM, then it halts, declaring the explanation complete. (If the resulting explanation is the empty set, then LISA's answer is

[1] Group units reside at different levels of the representational hierarchy than P, SP object and role units. As a result, they can be coactive with those units without having to occupy different slots in WM (see Doumas, et al., 2008; Hummel & Holyoak, 1997, 2003).

effectively "I don't know.") If it succeeds in retrieving a CE group, then it places a proxy of that group into a *workspace* (i.e., it copies the units comprising the group into a target analog; retrieval is thus a matter of activation and proxy creation rather than simply activating a structure in LTM) and maps the elements of the explanandum onto the proxy of the CE group, for example, mapping *ministers* onto *person*, *Coke* onto *product*, and *prefer* onto *prefer* (along with their SPs, P units and effect groups).

The model next makes the workspace the driver and the explanandum the recipient and, using analogical inference (i.e., self-supervised learning; Hummel & Holyoak, 2003), infers the missing elements in the explanantion. In this case, it would infer:

P2: *agree-with* (ministers, corporation)
P3: *produce* (corporation, Coke),

connecting both P2 and P3 to a cause group, and connecting both that cause group and the existing effect group (containing P1: *prefer* (ministers, Coke)) to a CE group. LISA's explanation now consists of the hypothesis "ministers prefer Coke because they agree with the corporation that makes Coke."

Finally, LISA attaches both P2 (*agree-with*) and P3 (*produce*) to their own effect groups, turns control back over to the explanandum (which is now an emerging explanation) and starts the whole cycle over again. Attaching P2 and P3 to effect groups is LISA's way of seeking new causes to explain these facts: *Why do ministers agree with the Coke corporation?* (P2) and (less sensibly) *Why does the Coke corporation produce Coke?* (P3). When the effect group connected to P2: *agree-with* (ministers, corporation) is used to drive retrieval, the result is very likely to be retrieval of the agreement schema (or an analogous specific example), in which case the same processes described above augment the explanation with the statements

P4: *believe* (ministers, some-proposition)
P5: *believe* (corporation, some-proposition),

connecting both P4 and P5 to a cause group linked via a CE group to the effect group connected to P3 (*agree-with*).

In the current instantiation of the model, this processes is repeated until the retrieval phase fails to retrieve a CE group. In the case of the current example, that will happen when LISA tries to retrieve a cause group describing why the corporation produces Coke (i.e., P3: *produce* (corporation, Coke)). It also happens with a small probability when other propositions fire (due to the stochastic algorithm for retrieving groups from LTM). This "explanation is done when retrieval fails" approach is a clear limitation of the model in its current state. Developing more intelligent halting criteria is the subject of ongoing research.

What is important to point out in the preceding description of the flow of control is the model's solution to the type-token problem: LISA maps *ministers* to *person* in the context of the preference schema, and then maps *ministers* to *person* (a completely different token) in the agreement schema. Seemingly (although not in fact) more impressively, it inferred *corporation* from the preference schema into the explanadum and then correctly mapped *corporation* onto *entity* in the agreement schema. How did it "know" that *corporation* in the preference schema had the same referent as *entity* in the agreement schema, or that *person* in the preference schema had the same referent as *person* in the agreement schema? The answer is that it did not know, and it did not have to. Rather than having to make the impossible decision of whether two tokens have the same referent, LISA's iterative retrieve-map-infer algorithm need only decide whether two units *correspond*, that is, map to one another, within the confines of the *current* retrieve-map-infer cycle. In so doing, it side-steps the question of whether the tokens "have the same referent". In short, LISA replaces the question "are they the same?" with the question "do they correspond?" and in so doing provides an effective solution to one particularly thorny variant of the type-token problem. Its ability to do so is a cornerstone of its ability to integrate multiple diverse sources of knowledge in LTM in the service of explaining a novel explanandum.

Simulations

The model described thus far, which consists of LISA augmented with units to represent causal groups and routines for using those groups to control LISA's meta-cognition, is still in an early stage of development. In order to test its potential, we ran several simulations based on an elaboration of the minister/Coke example given previously.

In a set of pilot simulations, the explanandum was the statement that ministers prefer Coke to Pepsi and we placed several relevant facts and schemas into LISA's LTM: (1) A (partial) *preference* schema stated that a person may prefer *x* to *y* either because they like *x* or because they dislike *y*. (2) A *Coke-and-cocaine* schema specified that Coke used to contain cocaine, rendering it and the Coke corporation "immoral". (3) A *minister* schema specified various properties of ministers, including their distaste for "immoral" things. (4) An *agreement* schema contained several statements specifying the conditions under which a person might like or dislike a product by virtue of their agreeing or disagreeing with the corporation that makes that product. And (5) a *support* schema specified how supporting common or inconsistent causes can cause a person or entity to agree or disagree with another person or entity. We also seeded its LTM with several irrelevant facts, so that we could evaluate the selectivity of the retrieval process.

Although we did not quantify the results of the pilot simulations, a typical result was an explanation such as:

P1: *prefer* (ministers, Coke) (the given explanandum)
P2: *believe* (ministers, some-proposition)
P3: *believe* (corporation, some-proposition)
P4: *agree-with* (ministers, corporation)
P5: *produce* (corporation, Coke)
cause (P2, P3) (P4)
cause (P4, P5) (P1)

where "*cause*" is shorthand for a collection of cause, effect and CE groups; the first pair of parentheses on each line enclose the P units connected to the cause group and the second pair enclose the propositions connected to the effect group. In other words, LISA inferred that ministers have some belief (e.g., some political belief) (P2), the corporation that makes Coke (P5) has that same belief (P3), these facts together cause the minister to agree with the corporation (*cause* (P2, P3) (P4)), and that this agreement, along with the fact that the company produces Coke, causes the ministers to prefer Coke (*cause* (P4, P5) (P1))).

This explanation represents the result of the majority of the pilot runs. Occasionally, other results obtained.

The model sometimes produced a truncated "explanation", in which the ministers are assumed to agree with the Coke corporation, but the model failed to recognize that they share views as a result. This explanation obtains when the explanandum, *prefer* (ministers, Coke), retrieves the preference schema on the first retrieve-map-infer cycle, but fails to retrieve anything on the next cycle.

A second result obtains when the explanandum retrieves nothing even on its first retrieval cycle. In this case, the model halts without generating any explanation at all (effectively saying, "I don't know").

Finally, the model occasionally retrieves the agreement schema (rather than the preference schema) on the first retrieval attempt. In this case, because analogical mappings are relationally flexible, *ministers* maps to *person*, and *Coke* maps successfully (but nonsensically) to *entity*. This particular mapping pattern is based on the slight semantic overlap between *ministers* and *person* (both are connected to the semantic *human*). In this case, the model generates the nonsensical "explanation":

P1: *prefer* (ministers, Coke) (the given explanandum)
P2: *believe* (ministers, some-proposition)
P3: *believe* (Coke, some-proposition)

In no cases did the model retrieve irrelevant information from LTM, illustrating that the algorithm is capable of selectively retrieving and mapping only potentially situation-relevant information.

In order to more precisely quantify the model's behavior, we also ran a suite of 50 simulations in which the exaplanandum was the statement "ministers prefer Pepsi". On 9 of these runs, the model produced no explanations, effectively saying "I don't know". On an additional 4 simulations, it produced incoherent explanations, for example, consisting of isolated facts not connected by causal relations. The remaining 37 runs resulted in coherent, causally-connected explanations. Of these, 15 contained three causal links (e.g., "ministers prefer Pepsi to Coke because they like Pepsi; they like Pepsi because they agree with the Pepsi Corporation; and they agree with PepsiCo because they both support some cause"). The remaining 22 explanations were of length two (e.g., "ministers prefer Pepsi to Coke because they dislike Coke; they dislike Coke because Coke is immoral").

Discussion

We described a process model of how people generate explanations of events and "facts", including novel ones (such as "ministers prefer Coke") that require the reasoner to integrate multiple sources of knowledge in LTM. The model is based on a psychological model of analogy (Hummel & Holyoak's, 1997, 2003, LISA model), reflecting our assumption that many of the core processes of explanation are also core processes of analogy making.

However, modeling explanation necessitates going beyond modeling analogy in at least two important respects: First, explanation, much more than analogy, depends on an understanding and an appropriate representation of causal relations. We model the representation of causal relations using units representing groups of propositions (and other groups). This representational format is more explicit than simple associative links between causes and effects (e.g., as in Bayesian models; Pearle, 2000; Tenenbaum, Griffiths, & Kemp, 2006), but less explicit than propositional statements about cause and effect relations. It permits the model to use cause, effect and cause-effect (CE) groups as units of both cognitive control and memory retrieval.

Second, explanation, unlike analogy, often requires the reasoner to integrate in processing multiple chunks of knowledge from diverse sources in LTM, which in turn requires a resolution of a difficult variant of the type-token problem, namely, the problem of knowing when different tokens refer to the same entity. We resolve this difficulty, not by solving it outright, but by replacing the question "are these the same" with the question "do these correspond?"

Preliminary simulation results suggest that the approach is promising as general way to understand the process of explanation, and indeed, the problem of understanding more broadly.

That said, the model is in an early stage of development, and many problems remain to be solved before we have a complete (much less correct) process model of explanation. First, we must address the problem of how a human reasoner knows when an explanation is complete. In the current version of the model, this decision is based strictly on the failure to retrieve additional causes from LTM. This is clearly incomplete, but what is right is harder to say. This question is a subject of ongoing research. Second, we must address the problem of explanation evaluation (for progress in this direction, see Thagard, 2001). One of the hard problems to be solved in this domain is contradiction detection: How does the cognitive architecture know when it has postulated something just plain stupid in the process of generating an explanation? Third, we must include a role for elaboration in explanation: in our example problem, for instance, the model is given the knowledge that ministers are politically conservative. Nevertheless, the model never suggests that the source of agreement between the Coke corporation and the ministers is one of conservative values. Such elaboration is not part of the causal chain approach

here, but seems to be a central component of explanation generation.

These issues remain serious hurdles in our attempt to understand how people perform this most mundane and everyday task of explanation. In the mean time, we believe our current work takes us at least one step closer to an answer to the difficult question of how we generate explanations.

Acknowledgments

This research was supported by AFOSR Grant # FA9550-07-1-0147. We are grateful to Brian Ross and Eric Taylor for many helpful discussions about this research.

References

Ahn, W-K., Mooney, R. J., Brewer, W. F., & DeJong, G. F. (1987). Schema acquisition from one example: Psychological evidence for explanation-based learning. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, 50-57.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity.*Behavioral and Brain Sciences*, *24*, 87-114.

Doumas, L. A. A., Bassok, M., Guthormson, A., & Hummel, J. E. (2006). A theory of reflexive relational generalization. In Proceedings of the Twenty Fourth Annual Conference of the Cognitive Science Society.

Doumas. L. A. A., Hummel, J. E., & Sandhofer, C. M. (2008). A theory of the discovery and predication of relational concepts. *Psychological Review, 115*, 1 - 43.

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

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

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, *13*, 295-355.

Holyoak, K. J., & Thagard, P. (1995). *Mental leaps: Analogy in creative thought*. Cambridge, MA: MIT Press.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427-466.

Hummel, J. E., & Holyoak, K. J. (2001). A process model of human transitive inference. In M. Gattis (Ed.). *Spatial schemas in abstract thought* (pp. 279-305). Cambridge, MA: MIT Press.

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, *110*, 220-264.

Hummel, J. E., & Ross, B. H. (2006). Relating category coherence and analogy: Simulating category use with a model of relational reasoning. In *Proceedings of the Twenty Fourth Annual Conference of the Cognitive Science Society.*

Keil, F.C. (2006). Explanation and understanding. *Annual Review of Psychology. 57*, 227-254.

Larkey, L. B., & Love, B. C. (2003). CAB: Connectionist analogy builder. *Cognitive Science*, *27*, 781-794.

Lombrozo, T., & Carey, S. (2006). Functional explanation and the function of explanation. *Cognition*, *99*, 167-204.

Morrison, R.G., Krawczyk, D.C., Holyoak, K.J., Hummel, J.E., Chow, T.W., Miller, B.L., & Knowlton. (2004). A neurocomputational model of analogical reasoning and its breakdown in Frontotemporal Lobar Degeneration. *Journal of Cognitive Neuroscience*, *16*, 260-271.

Patalano, A. L., Chin-Parker, S., & Ross, B. H. (2006). The importance of being coherent: Category coherence, cross-classification, and reasoning. *Journal of Memory and Language, 54*, 407-424.

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge Cambridge: Cambridge University Press.

Richland, L.E., Morrison, R.G., & Holyoak, K.J. (2006). Children's development of analogical reasoning: Insights from scene analogy problems. *Journal of Experimental Child Psychology*, *94*, 249–273.

Sloman, S. (2005). *Causal models: How people think about the world and its alternatives*. New York, NY: Oxford University Press.

Tenenbaum, J. B., Griffiths, T. L., and Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences, 10*(7), 309-318.

Thagard, P. (1989). Explanatory coherence. *Behavioral and Brain Sciences*, *12*, 435-467.

Thagard, P. (2001). How to make decisions: Coherence, emotion and practical inference. In E. Millgram (Ed.), *Varieties of practical inference* . Cambridge, MA: MIT Press. 355-371.

VanLehn, K., Jones, R. M., & Chi, M. T. H. (1992). A model of the self-explanation effect. *Journal of the Learning Sciences*, *2*, 1 – 59.

Viskontas, I., Morrison, R., Holyoak, K. J., Hummel, J. E., & Knowlton, B. J. (2004) Relational integration, inhibition, and analogical reasoning in older adults. *Psychology and Aging*, *19*, 581 – 591.

Vosniadou, S., & Brewer, W. F. (1987). Theories of knowledge restructuring in development. *Review of Educational Research*, *57,* 51-67.