# Using Analogies across Narratives to drive Dialogue

**David W. Knapp**
Independent Researcher
dwknapp@gmail.com

## Abstract

This paper describes a program called *Hob* that uses analogical mappings across narratives to drive an experimental conversational system. Analogical mappings are used to drive internal reasoning processes and supply response templates. The knowledge base is written in simple English, and consists of three parts: a dictionary, a collection of facts and heuristics, and a collection of stories that notionally correspond to an experience base. Internally, knowledge is stored and used in the form of near-canonical English parse trees. Thus the schema is not hardwired, but is derived from the knowledge base as interpreted under the rules and implications of English grammar. An experimental "sellbot" application is described, and example runs are presented.

## Introduction

This project is an effort to explore the possibilities of reasoning engines based on analogical mappings between input (current experience) and elements of a database of narratives that represent past experience. This mode of reasoning is attractive because it is cognitively plausible (Helman 1988), (Gentner, Holyoak, and Kokinov 2001), extensible, and intuitively accessible to writers of knowledge bases.

A "sellbot" application is the short-term driver for this project. A sellbot engages a customer in a conversation about the benefits, costs, and features of various products, and conceivably might also offer suggestions of alternatives, supplementary products, tips on the use of products, and so on. The sellbot has a number of convenient properties: it needs only a textual interface; it need not be very smart; and it is a plausible stepping-stone to more complex applications. Moreover, a sellbot requires only a minor modification to the basic engine in order to function, i.e. a procedural interface to a more traditional database containing products, prices, quantities on hand, etc. The rest of the sellbot can be implemented as a thin layer of "sales expertise" on top of a general-purpose knowledge base, and as such does not unduly distort the rest of the project.

### Representation

In order to function effectively a system based on analogies must contain a sufficient number of model systems onto which current problems can be mapped. It seems reasonable to expect any general-purpose knowledge base to be quite large; for example, (Lenat and Guha 1990) estimate the size of a general-purpose database at around $10^7$ basic facts. It therefore seems preferable to record the knowledge base in a form convenient to writers, i.e. in a natural language. Thus Hob's declarative knowledge files are almost entirely encoded in simple but standard English.

But it's difficult to transform natural-language parse trees into formal representations, as described in e.g. (MacCartney and Manning 2007). These difficulties can be summarized as stemming from ambiguity, grammatical complexity, and the sheer number of words needing to be mapped into predicates. In Hob, the issue was finessed by using normalized parse trees (NPTs), and structures trivially derived from them, as an internal representation. Thus sentences from the knowledge files have direct counterparts in the internal knowledge base.

Hob's internal knowledge base is divided into three main parts. These are a semantic net (Brachman and Levesque 2004), a collection of rule-like statements, and a collection of *narratives*. The semantic net primarily represents definitions and facts about objects; the statements represent heuristic knowledge about possibilities and probabilities; and narratives are the internal counterparts of stories, typically of from three to ten sentences. Again, these all contain essentially the same information as the text from which they were derived. On top of these, and intertwined with them, are various indices, tables, caches and other data structures that do not add any information but that make access speedy and convenient.

The core computational operation in Hob is, as stated above, analogical mapping. Following (Gentner 1983), (Gentner, Holyoak, and Kokinov 2001), an analogical mapping is a mapping from objects and relations of a model (the *reference*) to the objects and relations of a *target*. In the case of Hob, the target and reference are narratives. In a narrative, the objects are actors, props, scenery, and so on; relations can be either noun-based (e.g. "John's brother Richard"), preposition-based ("King of England") or verb-based ("John's barons disliked him").

**Example: narrative analogy** As described above, a *narrative* is the internal counterpart of a short story, consisting of one or more sentences written in simple English. Here is an example.

**(1)** I saw a wolf chase a rabbit. The wolf caught the rabbit and killed it. The wolf ate the rabbit.

In Hob's application of analogy to narratives, the objects are usually nouns, and the most important relations are derived from the ordering of events, the verbs used, and the grammatical roles and semantic properties of the nouns. Thus, for example, the wolf and rabbit of Story 1 are objects, having inherited attributes like *carnivorous* and *alive*; verb-mediated relations between them like *chaser/chasee* and *eater/eaten*; and temporal relations between actions, e.g. *chase preceded catch*.

Now consider a second story, about a cat named Rex:

**(2)** Rex saw a rat. He chased it. He caught it.

The internal representation of Stories 1 and 2 are shown schematically in Fig. 1.

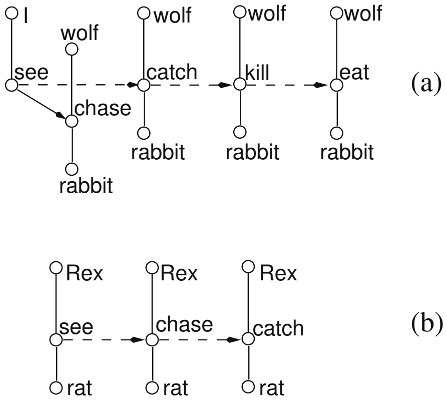

Figure 1: Graph representation of stories 1 and 2.

There is a simple and direct analogy between these two stories. In both stories the verbs *chase* and *catch* describe specific relationships between the actors; and furthermore in both cases we have *chase* preceding *catch*. Moreover, there is a consistent mapping of *Rex* to *wolf* and of *rat* to *rabbit*. So this is a pretty good analogy: it has many points of similarity, and much of the analogical mapping is between relations rather than objects, which means that the analogy is more capable of bearing semantic weight. And the analogy has some predictive power: Rex will probably kill and eat the rat, which is suggested by projecting the last two events of the first story (*killed, ate*) onto the end of the second. Now consider a third story:

**(3)** There was a viper and a weasel. The weasel caught the viper. The viper bit it. It died. The viper went away.

Here a mapping to Story 1 still exists, but it is weaker. The two events that anchor the mapping are those of *catch* and *kill/die*; but the actors in those events don't map consistently. That is, the catcher (*wolf*) causes the death (*kill* of the catchee (*rat*) in Story 1, but in Story 3 it is the catcher (*weasel*) that dies. Thus the analogy between Stories 1 and 3 is weak.

The mapping between the Stories 2 and 3 is also weak, in part because they only have only one event in common (*catch*), but also because the projection of Story 3 onto Story 2 (the rat bites Rex, who subsequently dies) is implausible, or at any rate less plausible than the projection of the Story 1 onto Story 3.

The key properties of a useful analogy are that one can project otherwise unmapped objects and relations of the reference into the target, and that such projection can carry semantics of the reference into the target. In other words, the projection allows us to borrow semantics from the reference in order to understand, flesh out, or predict outcomes in the context of the target.

Moreover, such semantic borrowing can be done without explicating the semantics being borrowed. That is, there is no primary requirement to "understand" either the target or the reference. One merely needs semantically significant objects and relations, ways to construct semantics-preserving mappings, and ways to rank mappings in terms of their quality and power. This dovetails nicely with the use of natural language as an internal representation, because natural language uses powerful and explicit semantically significant relationships (i.e. grammar) but is otherwise ambiguous, lexically populous, and difficult to understand.

For example, in Stories 1 and 2 above, there is some implicit causality associated with predators and prey. Why do cats chase rats? Beyond the bare statement that they do, causality becomes tricky. Should we devise an explicit system of FOL that works its way through a cognitive model of the cat and thence ultimately to its DNA? That doesn't seem particularly attractive, and it wouldn't end there: DNA has its causes also. But insofar as Story 1 has a good analogical correspondence to Story 2, its causality (whatever it is) can be borrowed without ever being explicitly modeled, and we can say with some confidence that the rat is likely to be eaten.

## Structure of this paper

The rest of this paper is organized as follows. The next section describes some related research and gives some useful pointers into the literature. Then the static structure of the Hob system is described, followed by a more operational description of how the various modules work, with the focus first on the construction of the knowledge base, then on the narrative analogy mapping process, and thence to the operation of a dialogue blackboard. Then two examples of dialogue interactions are given, following which future work is touched upon and conclusions drawn.

## Related work

In Hob, a simple English grammar based on an ESL textbook (Feigenbaum 1985) was used; a more complete descriptive treatment of English is given by (Huddleston and Pullum 2002). Hob's approach to parsing is similar to the chart-based parsers described in (Allen 1995), with additional clause-level semantic filtering after (Levin 1993) that helps reduce the number of alternative parsings. An adaptation of the centering method of (Brennan, Friedman, and Pollard 1987) was used for pronoun dereferencing. These methods are adequate for the simple standard English of Hob's knowledge base.

(Fellbaum 1998) and others have published machine-readable dictionaries, but these were not used in Hob because the author wanted to stay as close as possible to the style and content of an ordinary desk dictionary.

Recent work in natural logic (MacCartney and Manning 2007) is similar in spirit to the natural-language representation and inference methods used in Hob. However, the cited work is primarily aimed at detecting textual entailments, i.e. semantics-preserving reductions of texts, whereas Hob's core function is to find plausible and predictive analogical mappings between texts.

The basic strategy for analogical mapping follows the structure mapping theory of (Gentner 1983), reifications of which are described in (Forbus et al. 2007) and (Falkenhainer, Forbus, and Gentner 1989).

(Elson and McKeown 2007) and (Tomaszewski and Binstead 2007) describe high-level structural features of narratives, but Hob does not incorporate such structures in its reasoning process.

Hob's reasoning modes in some situations is similar to case-based reasoning(Leake 1996), where narratives could be viewed as self-constructing templates from which cases are to be derived on the fly. That is a closer description than those of planning (Allen, Hendler, and Tate 1990), (Russell and Norvig 2003), (Dean and Wellman 1991) or commonsense reasoning (Lenat and Guha 1990), (Mueller 2006).

## System Structure

Hob's structure is shown in Fig. 2. On the left is a knowledge base consisting of three sets of files: a dictionary, a collection of statements, and a collection of stories. They are read in and compiled by the English front end *engl2npt*. The compiled knowledge base KB is shown below the engl2npt module. To the right of engl2pt is a blackboard system (BB) containing a number of agents; user input to the blackboard is in English, via engl2npt. Output from the blackboard to the user is via an NPT printing routine, which is not shown.

To the right of the blackboard are two additional narratives, called the primary and internal narratives. The *primary* narrative is a verbatim record of the dialogue between the system and the user. The *internal* narrative is a monologue recording the blackboard's reasoning process in a linear form.

Below the blackboard is a notional database, communicating with the blackboard's agents via APIs. This database contains domain-specific sellbot information, i.e. products, prices, stock numbers, and so on.

### The Knowledge Base

As discussed above, Hob's knowledge is loaded from three sets of files: a small English dictionary, a collection of statements, and a collection of narratives.

**The Dictionary** Dictionary files look a lot like what's in an ordinary desk reference. For example, the verb *angle* is described as being *v. reg. tr. i.*, meaning that it is regular and both transitive and intransitive. For irregular verbs, participles and conjugations must be supplied.

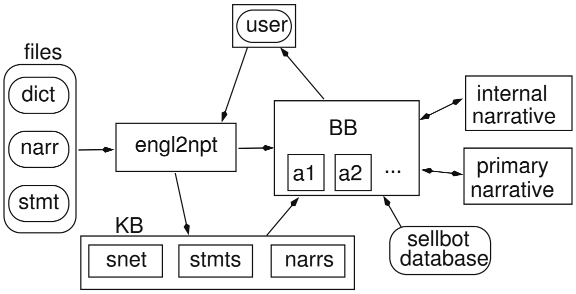

Figure 2: System structure.

In addition to the traditional transitive/intransitive tags, functionally similar but nonstandard verb tags are provided to express semantically valid syntactic alternatives at the clause level. For example, the tag *svio* indicates that the verb *give* can be formed into clauses containing a subject, verb, indirect object, and direct object; it would be semantically problematic to apply this tag to the intransitive verb *fall*. Hob has about 100 such tags, but of those only about 30 are independent; the others can be inferred.

The dictionary also contains definitions, which are short English fragments, sentences, or, in some cases, single words, again much like what is seen in a desk dictionary. For example, the verb *angle* is defined as "to fish with a hook and line." A word can have any number of such definitions.

At the time of writing there are about 3000 root words in the dictionary, of which about 2600 are nouns, verbs, adjectives, and adverbs. The rest are pronouns, prepositions, articles, determiners, quantifiers, conjunctions, and so on. This is interesting because most of Hob's word-specific procedural code is associated with the pronouns, prepositions, *et cetera*; these form a comparatively small and circumscribed set compared to the sets of nouns, verbs, and modifiers, which are neither small nor complete.

**Statements of general knowledge** Sentences from the statement files fall into two general classes. The first of these are plain statements of fact or supposition, of which some can be directly incorporated into semantic nets, while others are more temporally oriented and are stored in a single master narrative. The second class of statements represent conditional facts, which are stored as if/then pairs with a separate index. For example, the following sentences are in the current statement files.

A heron is a large ardeid.
Richard Lionheart died in 1199.
Dead animals don't eat.
Commoners usually eat porridge.
When you are unhappy about something, change it.
When a poisonous snake bites an animal, the animal will often die.

The first two of these are just facts: one is independent of time and could have been put into the dictionary, while the second describes an event and so is more properly part of the master narrative. The next four are less crisp, and are indexed as if/then pairs with varying structural details. In all cases the data structures remain NPT-equivalent.

At the time of writing there are about 280 sentences in the file of statements, about half of which could as properly have been put in the dictionary.

**Narrative knowledge** Story files contain stories that are compiled into narratives using the same apparatus as is used for dictionary and statement sentences. At the time of writing there are about 90 stories averaging about 7 sentences apiece in the narrative knowledge base.

**Sellbot database** In order to capture the sellbot application, a crude database of products, prices, and stocking data was added, and connected to one of Hob's blackboard agents via procedural APIs. These are of no interest except insofar as the blackboard agents can make use of them as described below. In addition, a small specialized dictionary of modern hardware-store terms (containing e.g. *switch*, *pliers*, *junction box*, and *circuit breaker*) was constructed.

## Compiling the knowledge base

As described above, the knowledge base is built by parsing and compressing English sentences from the various knowledge files.

The dictionary is processed in two passes. In the first pass, the words are read in and their various conjugations, plural forms, variant forms, etc. are entered into a vocabulary table. In the second pass definitions are parsed to NPTs and installed in a global semantic net.

NPTs differ from the parser's raw output in that they are more compact and more nearly canonical. For example, the sentence "Henry was blamed for the death of Becket" contains 17 nodes in the original parse tree, but only six in the normalized version. This is done without loss of information, e.g. by tagging the verb node *blame* with *passive past* and deleting the auxiliary verb *was*, and by collapsing intermediate layers.

Hob's semantic nets differ from most such nets in that there is no predefined set of edge predicates. Instead, edges have grammatically structured slots for verbs, adjectives, and so on. Thus edges are actually NPT-equivalent structures. They are constructed from dictionary definitions and other statements. Consider, for example, the sentences

John is the King of England.  
John is a bad King.  
John is a good lawyer.

Here the primary noun nodes *John*, *King*, and *lawyer* are linked by edges labeled *is*, with the extra adjectives *good*, *bad*, and the prepositional phrase *of England* remaining to be attached somewhere. But where? One cannot simply attach the adjectives *good* and *bad* to *John*; they would contradict one another. Similarly, additional links to *King* from e.g. *Henry* or *Arthur* would make it inconvenient to attach *bad* to the noun *King*. Thus such adjectives (and prepositional phrases, adjectival clauses, adverbial clauses, and so on) end up on the edges, which thereby become semantically rich and contain further references (e.g. *of England*) themselves.

The nodes of the net can represent nouns, adjectives, verbs, and adverbs, with the procedural constraint that verbs, adjectives, and adverbs when represented as nodes are being defined, as opposed to being used as parts of definitions. Noun nodes can represent either definitions or instances in the usual way, but instances of concrete nouns (*John Lackland*) are confined to the semantic nets associated with the master and other narratives.

The second part of the knowledge base is a collection of rule-like statements that represent general knowledge. These are used primarily to estimate the plausibility of internally constructed statements and narratives. Some of these are bald statements of possibility, e.g. "a cat can bite a rat." Others, which represent possible outcomes, are handled by using adverbs of frequency (*usually*, *seldom*, etc.) to describe conditional outcomes. For example, "When a poisonous snake bites an animal, the animal will often die" is present as an if/then pair whose predicate is the clause "poisonous snake bites animal" and whose outcome is "animal will die," modulated by the adverb *often*.

There are two sources for rule-like statements. The first of these is the file containing such statements described above. The second source is the narratives, from which possibility information can be derived directly, and from which (unreliable) frequency-tagged outcome information can be derived by looking for commonly occurring pairs of adjacent events.

The third part of the knowledge base is the collection of narratives. Each story describes some objects (actors, scenery, etc.) and a sequence of events involving the objects. A *narrative*, i.e. the internal counterpart of a story, consists of a local semantic net (its *namespace*) and a time-stamped list of events, which are just NPTs. Object information in a narrative's namespace is represented a local semantic net, with additional time-stamp tagging for attributes that change during the course of the story. The time stamps are those associated with the events that cause or describe changes. Thus for example if an event "Rex killed the rat" has time stamp $t = 5$, the rat will have an *alive* attribute with stamp $t < 5$ and a *dead* attribute stamped $t > 5$.

The local nets associated with narratives are not shared, but contain references to objects in a single common master narrative. This master narrative is constructed at run time by assembling events and instances from the statements file (*Richard died in 1199*) and instances from the story collection. Thus Rex, a cat, is present in two or three stories as an individual in his own right; these individuals are linked to an individual *Rex* in the namespace of the master narrative. This master Rex is then linked by an edge (containing e.g. the adjective *gray*) to the Platonic ideal *cat* in the global semantic net, which, in particular, contains neither instances nor time stamps. Thus the Rex of one story is young and playful, while the same Rex in another story is old and lame; and still a third story could refer to a completely different cat who is not gray but is also named Rex.

## Analogies across narratives

The construction of an analogical mapping of narratives will now be described in detail. We begin with two narratives, called the *target* and the *reference*. Usually the target is the narrative about which we want to draw inferences using the structure of the reference as a guide. Each is represented as an ordered list $(n_1, n_2, ...n_k)$ of NPTs (*events*) and a local namespace, as described above.

The first phase of the structure-mapping problem can be seen as a special case of weighted bipartite matching, where the two sets of nodes are the event lists of the target and reference respectively, and the weights on the edges correspond to a measure of semantic distance.

The first special property of this bipartite matching is an ordering constraint, which is a conservative measure intended to preserve causality. The matching must respect the orderings of both narratives. Formally, if a pair $(t_i, r_j)$ is present in the matching, there can be no other pair $(t_x, r_y)$ in the mapping such that either $(x < i)\&(y > j)$ or $(x > i)\&(y < j)$. Any such "crisscross" mapping represents a potential causality violation, and as such is disallowed.

The second special property is that the mapping of objects to one another must be consistent: if a rat is mapped to a rabbit in one sentence, then it must be mapped to (the same) rabbit everywhere the rabbit appears, and it cannot be mapped to a wolf in any sentence.

At the time of writing Hob has two algorithms for constructing the matching. The first is exhaustive and works by enumerating all matchings that respect the ordering constraint. This algorithm is exponential in the number of edges (which itself is quadratic in the length of the stories).

The second matching algorithm is greedy, beginning with a pair $(t_i, r_j)$ with a short semantic distance, and then working backward from $t_i$ and $r_j$ looking for pairs $(t_x, r_y)$ that are compatible with those edges already taken and that have minimum distance. This is cubic in the length of the stories.

In practice, though, there is no noticeable speed difference between these two algorithms. There are two reasons for this. First, the stories in the narrative database are quite short (typically three to ten events). The stories are short because they are designed that way, the intuition being that a short, simple story is more useful than a longer, more complex story. Second, these graphs are typically sparse, because the semantic distance between most random verb-verb pairs is enormous, and so most of the potential edges in the bipartite graph need never be constructed in the first place.

The distance calculation begins with a calculation of the distance between the two verbs that root the NPTs. In this calculation, for example, *want*, *desire*, *wish*, and *need* are computed as being quite close to one another, while *argue* and *go* are much more distant from one another. Verb-verb distances (as opposed to NPT distances) are computed by recursively referring to dictionary definitions. Thus, for example, "to flee" is defined as "to run away", and so the distance between *flee* and *run* is 1. If the evaluated verb-verb distance exceeds a maximum threshold, an edge is simply not constructed. Thus the sparsity property becomes a function of diversity of verb choice; and it just doesn't often happen that a short story contains very many repeated verbs, because such stories aren't very interesting.[1]

After the verb distance has been calculated, the general structures of the NPTs are compared. If, for example, one NPT has a subject and another has only an object, the distance is increased by a large amount, unless the one without a subject is passive and can be transformed to match the other (*Rex chased the rat : the rat was chased*).

Next, the nouns (subject, object, etc.) are compared. Again, this is done in the context of the dictionary: a rat is more like a rabbit than an insult is like a river. Nouns, in general, only contribute weakly to the overall distance measure; this is accomplished by weighting their contribution.

Prepositional phrases also contribute, either in terms of prepositionally connected nouns (e.g. *of England*) or phrases (e.g. *after betraying his brother*).

Finally, a contribution is derived from modifiers (adjectives and adverbs), which is in most cases weak, but very strong in the case of negative adverbials, as in e.g. "John's barons did not trust him."

**Plausibility** Once an analogical mapping between two narratives has been constructed, a further plausibility evaluation step can be applied to the mapping as a whole. This is done by constructing a composite narrative around the skeleton provided by the analogical mapping. This composite narrative consists of a copy of the namespace of the target, augmented with new objects as needed, and a sequence of events $(c_0, c_1, ...c_m)$ where each of the $c_i$ is either an untransformed event of the target, a mapped event of the reference, or a merged event of a matched target/reference pair. Such a composite of Stories 1 and 2 would be

Rex saw a rat. (from the target)
He chased the rat. (merged event)
He caught the rat. (merged event)
Rex killed it. (mapped from the reference)
Rex ate the rat. (mapped from the reference)

And for the Stories 1 and 3 a composite would be

Rex saw a rat. (from the target)
He chased the rat. (from the target)
He caught the rat. (merged event)
The rat bit Rex. (mapped from the reference)
Rex died. (mapped from the reference)
The rat went away. (mapped from the reference)

The plausibility of these two stories is evaluated by considering statements like

When a cat sees a rat, it usually chases it.
When a poisonous snake bites an animal, the animal will often die.

The first of these buttresses both composites, via the adverb "usually." The third statement supports only Story 3, because there is no snake in the composite of 1 and 3.

The support offered is modeled as a number, which is further modified by assigning quantitative values to adverbs like *often* and *seldom*. A number $dP$, representing a change

[1] Other than verbs like *is* and *has*. These end up in namespaces anyway and so don't directly figure in verb-verb matchings.

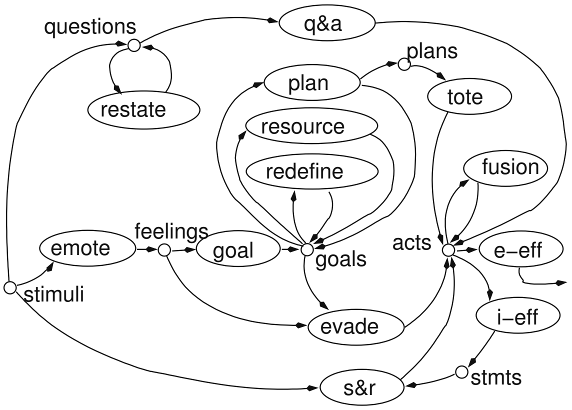

Figure 3: The blackboard.

in plausibility attributable to the composition, is then computed as $dP = 2Pc - (Pt + Pr)$, where $Pc$ is the intrinsic plausibility of the composite, and $Pt$ and $Pr$ are the plausibilities of the target and reference respectively. Thus in the Rex/wolf composite the plausibilities of all three stories are high, leading to a net $dP$ of about zero; whereas in the case of the Rex/weasel composite $Pr$ is increased, leading to a negative $dP$. Doing it this way allows the use of intrinsically implausible stories as target and reference.

## The Dialogue Blackboard

A blackboard (Engelmore and Morgan 1988) is used to organize various reasoning and bookkeeping tasks associated with user dialogue. The blackboard is conceptually organized as a set of data tables, a set of agents which read from and write to the tables, and a simple job scheduler.

The blackboard's tables contain sentences, i.e. NPTs. Each NPT is decorated with originator, confidence, and time stamp information. Agents are responsible for looking at these sentences, processing them, and generating new sentences, which are in turn put into the tables. Agents can also access the knowledge base and the primary and internal narratives. Input and output are handled by specialized agents.

Hob's blackboard as it exists at the time of writing is shown in Fig. 3. The small nodes are the tables and the larger ones are the agents.

*Stimuli* are sentences that originate with the user. These sentences are typed in by the user, parsed, normalized, and posted by a specialized input agent, which is not shown. Stimuli are processed by the *S&R* (stimulus-response) agent, the *Q&A* (query and answer) agent, and by the *emote* agent.

The S&R agent is responsible for direct stimulus/response coupling via narrative matching. For example, there is a story in which a bandit confronts the narrator.

Bandit: Give me your money.
Hob: I don't have any.
He took my pig.
I was angry and sad.

The S&R agent, upon finding a good match of some story to the tail of the interior monologue, will suggest an action or reply based on the story. The bandit story, in the absence of other applicable knowledge, means that the S&R agent will always respond "I don't have any" to an imperative demand for money.

*Feelings* are sentences of the general form "*actor* feels *adjective* about *clause*." They are inferred by the emote agent in response to stimuli as refracted through the knowledge base. An example is shown above: the narrator is angry and sad when his pig is taken. The emote agent is also responsible for projecting these "feelings" onto other participants whose situation is being considered. Thus reports of taking, theft, loss, etc. result in inferred statements of anger and sadness being felt by the victim.

*Goals* are statements of states, actions, etc. that the system will try to achieve. They are initially constructed by the *goal* agent in response to feelings; such initial goals are highly abstract, e.g. "change *clause*". They are then refined by the *redefine*, *resource*, and *plan* agents, each of which posts its output back to the goal table. So, for example, if the original goal is to change the fact of a sick relative, the redefine agent would suggest a goal of healing ("to cause a sick person to become healthy"). Thence the resource agent would suggest getting a doctor ("a person who heals sick people") but not a blacksmith ("a person who works iron") or hammer ("a tool for hammering").

The *plan* agent is responsible for looking up goals in the narrative base (as the redefinition agent looks them up in the semantic net) and trying to find fragments of narrative that plausibly map to the goals and seem to provide steps that would accomplish the goals. These steps are posted back to the goal table, and so in turn become goals themselves; they are also posted to the *plans* table.

The *evade* agent responds to a combination of feelings and goals. It represents a last resort, when nothing else can be done: at the time of writing its output is confined to more or less rude or sympathetic remarks.

The *questions* table contains questions originating with the user, which are posted from the stimulus table by a filtering agent (not shown), as well as restated questions generated by the *restate* agent, which looks up definitions and rewrites questions into alternative forms. Questions are answered by the Q&A agent, which does unification-based search on the knowledge base and the semantic nets associated with the primary and interior dialogues.

The *tote* (Miller, Galanter, and Pribram 1960) agent is responsible for sequencing plan steps. Each step of the most plausible plan is individually posted to the actions table. In the current system, which lacks realtime and physical i/o capabilities, that's all the tote agent reasonably can do.

The *acts* table contains imperative statements that have been addressed by Hob to itself. Action statements are consumed by effector agents. In the sellbot application, they are mostly of the form "Say *clause*" and "Look up *key*"; a robot application would presumably need a more elaborate set of effectors, recognizing e.g. *Look at* and *Grasp*. These verb phrases are used as indices into a table of procedurally coded functions, which at the time of writing consist only of database access APIs and the NPT printer, but which in a robot would presumably also include interfaces to motors, sensors, and so on. The internal effector agent *i_eff* posts its results to the statements table, while the external effector agent *e_eff* operates on the external world (at the time of writing this is just an NPT printer).

The *fusion* agent reads in statements from the actions table. Each is temporarily attached to the tail of the interior monologue, and then the interior monologue is mapped onto narratives in the knowledge base. If a plausible mapping is found, the fusion agent modifies the confidence stamp associated with the matched action. The confidence stamp is then used to select the most appropriate action for actual execution by either the internal effector or the external effector agent.

At the time of writing Hob consists of about 73 000 lines of ANSI C code. About 72% of that code parses, normalizes, represents, and indexes natural-language sentences, NPTs, etc. The unification-based query-answering subsystem comprises about 8% of the total, the blackboard with its agents is about 3%, and narrative mapping about 3%. The rest is miscellaneous utilities, debugging support, and so on. These figures illustrate the relative difficulty of acquiring and representing knowledge in English (72%) as compared to that of making basic use of it afterward (about 14%). This proportion is, however, expected to change as the knowledge-using algorithms are expanded.

## Examples

The first example makes no use of the sellbot database and story set. Instead, it makes use of a model of a medieval peasant, which is the base layer of knowledge.

In these examples the program output has been lightly edited to remove debugging information, remove some dead ends, and generally improve readability. Square brackets denote the originators of statements, and in some cases also contain a plausibility number or status flag. Table names and primary input/output are not indented; indented lines are the contents of blackboard tables. Some comments have been added as well.

```
Hob: Hello.
talk> my brother is ill. // user input
Stimuli:
  [1 input]: my brother is ill.
Feelings:
  [15 emote]: user is unhappy about
              his brother being ill
  [6 emote]: user is sad about his brother being ill
Goals:
  [15 goal]: change his brother is ill
  [15 goal]: evade his brother is ill
  [15 redef]: heal his brother
  [15 res]: get doctor
  [15 res]: get leech
  [15 res]: get Brother Andrew
Plans:
  [21 plan] // this is a four-step plan.
    [untried] run to Oxton
    [untried] find Brother Andrew
    [untried] tell him
    [untried] ask him
Actions:
  [0.89 s&r]: say Hob will fetch leech from Oxton
  [0.2 evade]: say I can not do anything
  [21 tote]: run to Oxton
[Hob] Hob will fetch leech from Oxton.
[Hob -act] run to Oxton
```

The salient points are that Hob infers that the user is unhappy; that Hob's goal therefore is to change the source of unhappiness; that a doctor is needed ("a doctor is a person who heals"); a reference to Brother Andrew as being a leech (i.e. a doctor); a match to a story that tells how Brother Andrew has been fetched in the past; and finally, a match to a story that contains an appropriate reply under the circumstances (*I will fetch a leech*). The first step of the plan (*run to Oxton*) is executed, as indicated by the bracketed *act* tag.

The next example triggers the sellbot layer via a story in which a query is handled by looking up a desired item. In this example time stamps have been retained inside the square bracket pairs of the statements and actions tables; these time stamps come after plausibility numbers and before agent names. This was done to explicate the sequence of events.

```
Hob: Hello.
talk> I need some junction boxes. // user input
Stimuli:
  [1 input]: I need some junction box.
Feelings:
  [23 emote]: user is sorry about needing junction box
  [7 emote]: user is unhappy about needing junction box
Statements:
  // note the timestamp (2). This is effector output.
  [1 2 i_act]: I see two kind of junction box
Actions:
  // note the timestamp (1). This triggers the effector.
  [.96 1 s&r1]: look up junction box.
  // this is the reply to the user at time 3.
  [.99 3 s&r2]: say we have two kinds of junction box
[Hob] we have two kinds of junction box.
talk> what do they cost? // user input
Stimuli:
  [1 input]: what do they cost?
Questions:
  [10 query]: what do they cost?
  [10 restate]: what is price of they?
Actions:
  [.99 1 s&r1]: say 1.09 is price // see below
  [58 3 q&a]: say 1.79 and 2.49 is price of they
[Hob] 1.79 and 2.49 is price of they. // answer
```

There are two things that need to be explained here. First, it is very important to notice the time stamps in the statements and actions tables. The sequence of events is that

1. The S&R agent posts an action, requesting a lookup.
2. The internal effector posts a statement in reply, stating that there are two kinds of junction box.
3. The S&R agent maps that reply into a speech action targeted at the user.

Second, there is an extra price (1.09) cited by the S&R agent in the actions table. This is not the price of junction boxes: it is a statement copied from a completely different sales dialogue, with low confidence (.99) in part because the price quoted is an unmapped constant, and hence unreliable.

This statement represents a fallback to be used in the event that no other response is forthcoming.

One should not read too much into these examples. They closely resemble dialogues in the knowledge base; as such, many holes are papered over by narrative matching.

For example, in the first example the goal of fetching a doctor is derived from the goal (to change the illness) via the definitions of *heal* (to cause to become healthy) and doctor (a person who heals). Then the KB can be used to infer that Brother Andrew would serve the purpose ("Brother Andrew is a leech").

But then there is a gap. How does the system know that Oxton is a good place to look for Brother Andrew? Or that in order to bring him, he will have to be told about the problem, and asked to come? This is inferred by matching to a story about needing a leech, which is also the source of Hob's statement that he will fetch the leech from Oxton. If that story is removed from the database, another story in which a priest is needed is matched (involving Brother John) but that story doesn't contain any speech acts, so the spoken output is quite different: Hob asks what is wrong with the user's brother. That happens because still a third story (one involving a lame horse) has now become the best match containing a speech act.

## Future Work

Hob is still at the laboratory-curiosity stage of develoment. Among the future challenges are:

- Improving the system's grasp of nonstandard English.
- Expanding the knowledge base.
- Performance improvements.
- Improved analogical mapping.
- Improved plausibility analysis.
- Adding facilities for efficiently reasoning about abstractions like quantities, time, space, substances, etc.

## Conclusion

This paper has described a program named Hob, which applies the theory of structure mapping to narratives in the context of a grammatically structured knowledge base. A dialogue application was described, and two simple examples were given.

## Acknowledgement

The author would like to thank the people who took the trouble to review this paper and provide feedback. Their comments, pointers, and suggestions were very helpful, both in the preparation of this paper and in pointing out issues and ideas of which the author was formerly completely unaware. Thank you.

## References

Allen, J.; Hendler, J.; and Tate, A. 1990. *Readings in Planning*. Morgan Kaufmann.

Allen, J. 1995. *Natural Language Understanding*. Benjamin/Cummings.

Brachman, R. J., and Levesque, H. J. 2004. *Knowledge Representation and Reasoning*. Morgan Kaufmann.

Brennan, S. E.; Friedman, M. W.; and Pollard, C. J. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual ACL*, 155–162.

Dean, T. L., and Wellman, M. P. 1991. *Planning and Control*. Morgan Kaufmann.

Elson, D. K., and McKeown, K. R. 2007. A platform for symbolically encoding human narratives. In *Intelligent Narrative Technologies (AAAI Fall Symposium 2007)*. AAAI Press. Technical Report FS-07-05.

Engelmore, R., and Morgan, T., eds. 1988. *Blackboard Systems*. Addison-Wesley.

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

Feigenbaum, I. 1985. *The Grammar Handbook*. Oxford University Press.

Fellbaum, C., ed. 1998. *Wordnet: An Electronic Lexical Database*. MIT Press.

Forbus, K. D.; Riesbeck, C.; Birnbaum, L.; Livingston, K.; Sharma, A.; and Ureel, L. 2007. Integrating natural language, knowledge representation, and reasoning, and analogical processing to learn by reading. In *Proceedings of AAAI-07*. AAAI.

Gentner, D.; Holyoak, K.; and Kokinov, B., eds. 2001. *The Analogical Mind*. MIT Press.

Gentner, D. 1983. Structure mapping: a theoretical framework for analogy. *Cognitive Science* 7(2).

Helman, D. H., ed. 1988. *Analogical Reasoning*. Kluwer.

Huddleston, R., and Pullum, G. K. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press.

Leake, D. B., ed. 1996. *Case-Based Reasoning*. MIT Press.

Lenat, D. B., and Guha, R. V. 1990. *Building Large Knowledge-Based Systems*. Addison-Wesley.

Levin, B. 1993. *English Verb Classes and Alternations*. University of Chicago press.

MacCartney, B., and Manning, C. D. 2007. Natural logic for textual inference. In *ACL Workshop on Textual Entailment and Paraphrasing*.

Miller, G.; Galanter, E.; and Pribram, K. 1960. *Plans and the Structure of Behavior*. Holt-Dryden.

Mueller, E. T. 2006. *Commonsense Reasoning*. Morgan Kaufmann.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice-Hall.

Tomaszewski, Z., and Binstead, K. 2007. The limitations of a Propp-based approach to interactive drama. In *Intelligent Narrative Technologies (AAAI Fall Symposium 2007)*. AAAI Press. Technical Report FS-07-05.