Analogy in a General-Purpose Reasoning System

Pei Wang

Temple University, Philadelphia, USA

Abstract

This paper introduces the various forms of analogy in NARS, a general-purpose reasoning system. NARS is an AI system designed to be adaptive and to work with insufficient knowledge and resources. In the system, multiple types of inference, including analogy, deduction, induction, abduction, comparison, and revision, are unified both in syntax and in semantics. The system can also carry out relational and structural analogy, in ways comparable to (though different from) that in some other models of analogy, such as Copycat and SME. The paper addresses several theoretical issues in the study of analogy, including the specification and justification of analogy, the context sensitivity of analogy, as well as the role analogy plays in intelligence and cognition.

Key words: Non-Axiomatic Reasoning System (NARS), Artificial General Intelligence (AGI), term logic, experience-grounded semantics, extended syllogism

1. Introduction

Analogy has been extensively studied in Artificial Intelligence (AI) and Cognitive Science (CogSci) (Evans, 1969; Falkenhainer et al., 1989; Gentner et al., 2001; Hofstadter and FARG, 1995; Holyoak and Thagard, 1989; Indurkhya, 1992), and it is seen by some researchers as playing a key role in cognition (Hofstadter, 2001).

This paper introduces a novel treatment of analogy. What distinguishes it from the other works is that according to this treatment, analogy is not

Email addresses: `pei.wang@temple.edu` (Pei Wang)
URL: `http://www.cis.temple.edu/~pwang/` (Pei Wang)

a stand-alone cognitive process, but one aspect of a unified mechanism of intelligence and cognition. Furthermore, though analogy has various forms, they can be captured consistently in a formal model of intelligent reasoning.

This formal model has been implemented in an AI system called NARS. The system is a research project aimed at a unified theory of intelligence (Wang, 2006). Analogy plays an important role in the model, where it is tightly coupled with other cognitive processes and mechanisms.

In the following, the paper is going to start with a brief introduction to the relevant parts of NARS, then to describe how the system carries out "analogy", under various interpretations of the word, and finally, to discuss several theoretical issues on analogy.

2. NARS Basic

NARS stands for "Non-Axiomatic Reasoning System", and is an attempt to provide a unified theory and model of intelligence. Therefore, it can be referred to as an AGI (Artificial General Intelligence) project (Goertzel and Pennachin, 2007).

There have been many publications on various aspects of this research (most can be accessed via the author's homepage), with Wang (2006) as the most comprehensive description of the project. It is impossible to describe the whole NARS system in this paper, so, instead, this section summarizes the parts that are most relevant to analogy.

2.1. Theoretical foundation

NARS can process three types of *tasks*: (1) *knowledge* to be remembered (also called *judgment*), (2) *question* to be answered, and (3) *goal* to be achieved. The system processes each task according to its (available and relevant) *beliefs*, that is, the remembered knowledge (judgments) which is obtained from previous tasks. At the current stage, all input tasks of NARS are provided directly by human users in a formal language (to be introduced later), though in the future they may also come through a natural language interface or a sensorimotor interface.

The system is a *reasoning system*, and like other automated reasoning systems (Lenat, 1995; Nipkow et al., 2002; Robinson and Voronkov, 2001), it represents knowledge in a *formal language*, depends on a *semantic theory* to link the language to the outside environment, follows certain *formal inference* *rules* in each step, and has a *memory structure* and a *control mechanism* to organize inference steps into an inference process for each task.

NARS is "intelligent", in the sense that it is designed according to the opinion that "intelligence" is *the ability for a system to adapt to its environment while working with insufficient knowledge and resources*. It means the system only has a constant processing capability, but has to be open to novel tasks, and process them in real time. Under the restriction of available knowledge and resources, the system must learn from its experience, and make its best attempt to accomplish the tasks (Wang, 2007).

Designed under the assumption of adaptivity and knowledge-resources insufficiency, the components of NARS (language, semantics, rules, memory, and control) turn out to be fundamentally different from those in traditional reasoning systems. NARS is not a theorem prover or only does deduction in first-order predicate logic, and it does not use model-theoretic semantics, neither. Based on available knowledge, the inference rule may produce incorrect predictions. The system does not remember everything it is told or has derived, or access whatever it does remember with the same easiness. For a given task, the system's processing does not follow a predetermined algorithm, but is highly sensitive to its history and context. On the other hand, it does not mean that the processing is random or arbitrary.

In the following, the major components of NARS are briefly introduced one by one.

2.2. Representation language

The beliefs and tasks in NARS are expressed in a formal language, *Narsese*, using *terms* and *statements*. A "term" is an internal identifier, which can either be *atomic* (without inner structure) or *compound* (formed by an operator from component terms). "Statement" is a special type of compound term, which consists of an *inheritance relation* (or its variant) between two terms, indicating their substitutability in inference process.

In Narsese, the *inheritance relation* is symbolized as "$\rightarrow$", and defined by being reflexive and transitive in its idealized form. An *inheritance statement* "$S \rightarrow P$" consists of a *subject* term S and a *predicate* term P, linked together by the inheritance relation. Intuitively, it states that "S is a specialization of P, and P is a generalization of S". Therefore, "Bird is a kind of animal" can be represented in Narsese as statement "$bird \rightarrow animal$".

Being transitive, inheritance is a special case of substitutability. If both statements "$S \rightarrow M$" and "$M \rightarrow P$" are true, so is "$S \rightarrow P$". We can see the conclusion as obtained by substituting a term in a premise by another term in the other premise, as far as the directions of the involved inheritance relations are the same.

A variant of the inheritance relation, called *similarity* and symbolized as "$\leftrightarrow$", allows the two terms to substitute each other in any direction. Therefore "$S \leftrightarrow P$" (and "$P \leftrightarrow S$") means both "$S \rightarrow P$" and "$P \rightarrow S$".

Assuming insufficient knowledge and resources, an actual belief in NARS is never in the above idealized form, that is, being absolutely true. Rather, it is only *true to a degree*. Concretely, the system's belief on "Bird is a kind of animal" is represented as a *judgment* "$bird \rightarrow animal <f, c>$", that is, a statement with *truth-value*.

A truth-value in NARS has two factors, a *frequency* factor and a *confidence* factor. The frequency factor is a real number in [0, 1], defined as the proportion of positive evidence among available evidence, that is, $f = w^+/w$. The confidence factor is a real number in (0, 1), defined as the proportion of current available evidence among future available evidence, after the coming of a unit amount of new evidence, that is, $c = w/(w+1)$. For the current discussion, "evidence" can be understood intuitively, and its formal definition and measurement (such as the above w^+ and w) have been given in several publications, such as Wang (2005, 2006).

In this way, Narsese gets an *experience-grounded semantics* (Wang, 2005). According to this semantics, the truth-value of a statement is determined by how much the statement is supported (or refuted) by the evidence collected from the system's experience, rather than by how well the statement corresponds to a fact in the outside world; the meaning of a term is determined by the role the term plays in the system's experience (that is, how it is related to other terms by the inheritance relation and its variants), rather than by an entity in the outside world referred to by the term.

To express complicated content, a statement can have a compound term as its subject or predicate. For example, "The Sun is larger than the Earth" can be expressed in Narsese as statement "$(\times, Sun, Earth) \rightarrow larger$", where the subject term is a compound "$(\times, Sun, Earth)$", a *product* of term Sun and term $Earth$, representing the *relation* between the two terms. Therefore, the statement literally states that "The relation from *Sun* to *Earth* is a special case of the relation *larger* (*than*)".

In Narsese, a question is a statement with a unknown truth-value. For example, "$(\times, Sun, Earth) \rightarrow larger$" is a question "Is the Sun larger than the Earth?" A question can also include variable terms to be instantiated, like "$(\times, Sun, Earth) \rightarrow ?x$", a question asking for a term that best specifies the relation from Sun to $Earth$, and "$(\times, ?x, Earth) \rightarrow larger$", a question asking for something that is $larger$ than $Earth$. An answer to a question should be a matching belief, like "$(\times, Sun, Earth) \rightarrow larger <f, c>$" for the above three questions.

There are other types of compound terms in Narsese (such as *sets*, *intersections*, *differences*, *images*, *disjunctions*, *conjunctions*, and *negations*), and variants of inheritance (such as *implication* and *equivalence*, indicating the substitutability between statements). For a complete specification of the Narsese grammar, see Wang (2006).

2.3. Inference rules

As a reasoning system, NARS derives new tasks and beliefs from given ones, according to its inference rules, which is built into the system, rather than provided by the user (as the "rules" in production systems).

In NARS, a typical inference rule is given one task (which can be a judgment, a question, or a goal) and one belief (a judgment) as premise, and derives a task as conclusion. When the given task is a judgment, the rule carries out *forward inference*, and the derived task is a judgment, and it will also become a belief of the system; when the given task is a question (or goal), the rule carries out *backward inference*, and the derived task is a question (or goal), and its solution will contribute to the solution of the given task, as far as the system knows.

Since in NARS forward inference and backward inference are closely related, in the following we will focus on the former. Such an inference rule R has the form of

$$\{St_1 <t_1>,\ St_2 <t_2>\} \vdash St <t>$$

which takes two judgments as premises, and produces one judgment as conclusion, whose truth-value is a function of those of the premises, that is, $t = F_R(t_1, t_2)$. Since each truth-value consists of a frequency factor and a confidence factor, each "truth-value function" is actually a pair of equations for the two factors respectively.

All the inference rules of NARS are designed and justified according to the experience-grounded semantics, so F_R calculates the evidence to the conclusion provided by the premises *alone*.

One important rule is the *revision* rule

$$\{St <t_1>,\ St <t_2>\} \vdash St <t>$$

which is invoked when the two premises contains the same statement St, but with truth-values t_1 and t_2 that come from different evidences. The corresponding truth-value function F_{rev} calculates t from t_1 and t_2, according to the additivity of amount of evidence and the relationship between amount of evidence and truth-value:

$$F_{rev}: \ f = \frac{f_1 c_1(1-c_2)+f_2 c_2(1-c_1)}{c_1(1-c_2)+c_2(1-c_1)}, \ \ c = \frac{c_1(1-c_2)+c_2(1-c_1)}{c_1(1-c_2)+c_2(1-c_1)+(1-c_1)(1-c_2)}$$

Intuitively, the frequency factor of the conclusion is a compromise of the frequency factors of the premises (so it can be seen as conflict resolution), and the confidence factor of the conclusion is higher than the confidence factors of the premises (so it can be seen as evidence accumulation).

A major group of rules in NARS is the *syllogistic* rules, in each of which the two premises share exactly one common term, and the conclusion is about the other two terms. Considering all premise combinations involving inheritance and similarity relations, the basic syllogistic rules are listed in Table 1.

$St_1 <t_1>$ $St_2 <t_2>$	$M \rightarrow P$	$P \rightarrow M$	$M \leftrightarrow P$
$S \rightarrow M$	$S \rightarrow P \ <F_{ded}>$ $P \rightarrow S \ <F_{exe}>$	$S \rightarrow P \ <F_{abd}>$ $P \rightarrow S \ <F'_{abd}>$ $S \leftrightarrow P \ <F_{com}>$	$S \rightarrow P \ <F'_{ana}>$
$M \rightarrow S$	$S \rightarrow P \ <F_{ind}>$ $P \rightarrow S \ <F'_{ind}>$ $S \leftrightarrow P \ <F_{com}>$	$S \rightarrow P \ <F_{exe}>$ $P \rightarrow S \ <F_{ded}>$	$P \rightarrow S \ <F'_{ana}>$
$S \leftrightarrow M$	$S \rightarrow P \ <F_{ana}>$	$P \rightarrow S \ <F_{ana}>$	$S \leftrightarrow P \ <F_{res}>$

Table 1: *Basic Syllogistic Rules*

In the table, different rules associate to different truth functions, which are listed in Table 2, and function F'_R is obtained from F_R by switching t_1 and t_2.

To discuss each rule and its truth-value function is beyond the scope of this paper (and such discussions can be found in previous publications on

Inference Type	Function	frequency	confidence
deduction	F_{ded}	$f = f_1 f_2$	$c = f_1 c_1 f_2 c_2$
abduction	F_{abd}	$f = f_2$	$c = \frac{f_1 c_1 c_2}{f_1 c_1 c_2 + 1}$
induction	F_{ind}	$f = f_1$	$c = \frac{c_1 f_2 c_2}{c_1 f_2 c_2 + 1}$
exemplification	F_{exe}	$f = 1$	$c = \frac{f_1 c_1 f_2 c_2}{f_1 c_1 f_2 c_2 + 1}$
comparison	F_{com}	$f = \frac{f_1 f_2}{f_1 + f_2 - f_1 f_2}$	$c = \frac{c_1 c_2 (f_1 + f_2 - f_1 f_2)}{c_1 c_2 (f_1 + f_2 - f_1 f_2) + 1}$
analogy	F_{ana}	$f = f_1 f_2$	$c = c_1 f_2^2 c_2^2$
resemblance	F_{res}	$f = f_1 f_2$	$c = c_1 c_2 (f_1 + f_2 - f_1 f_2)$

Table 2: *Truth Value Functions of the Syllogistic Rules*

NARS), so only some of them (those directly related to various forms of analogy) will be introduced in the next section. At here, it is enough to notice that the seven truth-value functions associated with syllogistic rules can be divided into two groups, with respect to the confidence factor of its conclusion:

Strong Inference including *deduction*, *analogy*, and *resemblance*. The conclusions produced by these functions have 1.0 as the upper bound of their confidence factors. If both premises are absolutely true (i.e., frequency is 1, and confidence converges to 1), so is the conclusion. Therefore, these rules have counterparts in binary logic, coming from the transitivity of the inheritance and similarity relations in their idealized form.

Weak Inference including *abduction*, *induction*, *exemplification*, and *comparison*. The conclusions produced by these functions have 0.5 as the upper bound of their confidence factors. Even if both premises are absolutely true, the conclusion is not. Therefore, these rules have no counterpart in binary logic, and look like logical fallacy if the truth-values are omitted.

There are other rules in NARS that handle *structural* inference of compound terms, *higher-order* inference on implication and equivalence relations, as well as *matching* beliefs to questions and goals. For more details on them, see Wang (2006).

2.4. Memory and control

Given the requirement in the inference rules that the premises used together in an inference step must have at least one shared term, it is natural to cluster the beliefs and tasks in the system into "concepts", according to the terms in them. For example, the belief "$bird \rightarrow animal <f, c>$" can be accessed in concepts C_{bird} and C_{animal} (though not in concept C_{Earth}), and the question "$(\times, Sun, Earth) \rightarrow ?x$" can be accessed directly in concept $C_{(\times, Sun, Earth)}$, as well as indirectly in concepts C_{Sun} and C_{Earth}, since they correspond to the components of the compound.

In this way, a concept is labeled by a term, and contains tasks and beliefs associated with that term. A concept is a unit both for storage and for processing, since every inference step happens in some concept, and only uses premises stored locally. Roughly speaking, the system's memory contains a collection of concepts, and each concept contains a collection of tasks and a collection of beliefs. When the system gets an input task (from a human user or another computer system), it will be sent to the directly related concepts, which will process it, as well as send derived tasks to other concepts, which will do the same, and also send their processing results back (as derived tasks), which may contribute to the processing of the initial input tasks.

The major functions of the system are carried out by repeating the following working cycle:

1. Check the system-wide task buffer, and select new (input or derived) tasks to add into the corresponding concepts. If a task is a judgment, it is also added as a belief.
2. Select a concept from the memory, then select a task and a belief from the concept.
3. Use the task and the belief as premise to produce derived tasks, according to applicable rules.
4. Add the derived tasks into the task buffer, and send report to the environment (user or other system) if the task provides a best-so-far answer to an input question.
5. Return the belief, task, and concept back to memory.

All the "selections" in step (1) and (2) are *probabilistic*, in the sense that all the items (tasks, beliefs, or concepts) within the scope of the selection have priority values attached, and the probability for each of them to be selected at the current moment is proportional to its priority value. When an new item is produced, its priority value is determined according to its parent items, as well as the type of mechanism that produces it. At step (5), the priority values of all the involved items are adjusted, according to the immediate feedback collected during the current step.

The details of the priority functions are beyond the scope of this paper. Here we only need to say that they are designed to give higher priority to more *important* and *relevant* items, judged mainly according to the experience of the system.

This picture also shows why it is said previously that the meaning of a term is nothing but its relations with other terms, because all the information associated with a term is in its corresponding concept, which contains tasks and beliefs that relate the term to other terms. For this reason, we can similarly talk about the meaning of a concept, which is just that of the corresponding term.

With the inputting and deriving of new beliefs and tasks, as well as the removing of old ones ("forgetting"), the meaning of a term in the system changes from time to time, though not arbitrarily, but is accurately determined by the system's communication and inference history.

Since at any moment the system typically has many tasks competing for processor time, each of them usually does not have the chance to interact with all the beliefs residing in the same concept, but only those that happen to be selected during the processing of the task. Therefore, when a task is processed within a concept, the concept is used with a *partial meaning*, rather than with its *full meaning*. In some concepts, there are some stable and consistent beliefs with high priority, so they are almost always invoked whenever tasks are processed in the concept. We can say that these beliefs consist of the "essence" of the concept. On the contrary, some concepts lack such a stable core, and consequently their "current meaning" change dramatically from time to time. Of course, the distinction between these two types of concept is fuzzy and relative, but still, it will help us in the following discussion.

3. Analogy in NARS

Though Wang (2006) provides much more details on the components and aspects of NARS than the brief summary in the previous section, the book does not address analogy as a separate topic, and that is what this paper provides.

Like most words in a natural language, the word "analogy" has been used in many different, though related, senses. In this section, several major senses of the word are identified, and, for each of them, more details are provided to show how it shows up in NARS.

3.1. Simple analogy

First, in the previous section we have introduced the *analogy rule* in NARS: if concepts S and M are similar to each other, then S can substitute M in its inheritance relation with P. This rule indeed captures a special case of what we usually call "analogy", that is, "if two concepts are similar, one can be used as the other". In this paper, this form of analogy is called "simple analogy", since it is carried out as a single inference step.

This rule is applicable as far as the two premises contain exactly one shared term, which is related to the other two terms by an inheritance relation and a similarity relation, respectively. There is no additional requirement, such as whether the terms are in the same "domain" or not, whatever that means.

The truth-value function of the analogy rule is defined as

$$F_{ana} : f = f_1 f_2\,,\ c = c_1 f_2^2 c_2^2$$

where $<f_1, c_1>$ is the truth-value of the *inheritance* premise, and $<f_2, c_2>$ is that of the *similarity* premise.

Instead of deriving this function from certain general principles step by step (which can be found in Wang (2006)), in this paper we just list some special cases and features of this function:

- When the similarity is so strong as to make the two terms identical (i.e., $f_2 = 1$ and $c_2 \approx 1$), the truth-value of the conclusion is very close to that of the other premise — it is just like a substitution between two identical concepts, or synonyms.

- When there is no similarity (i.e., $f_2 = 0$), no conclusion can be derived — $c = 0$ means "I don't know", no matter what value f has.

- In general, the confidence of the conclusion is more sensitive to the truth-value of the *similarity* premise than to the other (*inheritance*) — only strong similarity can lead to confident analogy.

Obviously, though this analogy rule looks natural and general, it cannot cover all forms of "analogy" in the previous research, which typically correspond to problem-solving processes consisting of multiple steps. Even so, these processes usually can be carried out in NARS, though not merely by the analogy rule. Some of these cases are discussed in the following subsections.

3.2. Relational analogy

A common form of analogy has the form of a problem "A is to B as C is to what?". In Narsese, it is expressed as question "$(\times, A, B) \leftrightarrow (\times, C, ?x)$", which, as explained in the previous section, asks the system to find a constant term D to instantiate the variable term $?x$, so that the relation between A and B, $(\times, A, B)$, is similar to (or identical to, as an extreme case) the relation between C and D, $(\times, C, D)$. Since here the similarity are not between individual concepts, but between their *relations*, this type of analogy is called "relational analogy" in this paper.

A well-known model of this type of analogy is Copycat (Hofstadter and FARG, 1995), which is designed to work in the domain of letter strings, and can solve problems like "If *abc* is changed to *abd*, how would *ijk* be changed in the same way?", where *abc*, *abd*, and *ijk* correspond to the A, B, and C in the previous pattern of relational analogy, respectively.

To show how such a problem is solved in NARS, let us see the simplest case of relational analogy in the domain of letter string, by asking the system "$(\times, a, b) \leftrightarrow (\times, i, ?x)$", where letters a, b, and i correspond to the A, B, and C in the general pattern, respectively.

What will be the solution of NARS to this problem? Well, it depends. Since NARS is a general-purpose reasoning system, there is no built-in solution to domain-specific problems. Instead, the solution depends on what the system believes about the concepts involved. Therefore we can analyze this problem-solving process under different assumptions about the system's beliefs.

Assuming the system knows the order of all letters in the alphabet, its memory contains the following beliefs:

$$\begin{array}{ll} (1) & (\times, a, b) \rightarrow \mathit{successor} <1.00, 0.99> \\ (2) & (\times, i, j) \rightarrow \mathit{successor} <1.00, 0.99> \end{array}$$

Here frequency 1.00 indicates that all the evidence is positive, and confidence 0.99 indicates that the system is quite sure about these beliefs.

If the above two judgments happen to be selected as premises in an inference step within concept $C_{successor}$, from Table 1 we can see that one of the applicable rules is the *comparison* rule, and according to Table 2, the conclusion is

$$(3) \quad (\times, a, b) \leftrightarrow (\times, i, j) <1.00, 0.49>$$

which provides an answer to question "$(\times, a, b) \leftrightarrow (\times, i, ?x)$" when the variable term $?x$ is unified with the constant term j.

What if the system has belief (1) but not (or fail to recall) belief (2)? Then the answer may be different. Assuming the system takes the following beliefs into consideration:

$$\begin{array}{ll} (4) & (\times, i, h) \rightarrow \textit{predecessor} <1.00, 0.99> \\ (5) & \textit{predecessor} \rightarrow \textit{neighbor} <1.00, 0.99> \\ (6) & \textit{successor} \rightarrow \textit{neighbor} <1.00, 0.99> \end{array}$$

Then, using the comparison rule, from (5) and (6) the system gets

$$(7) \quad \textit{predecessor} \leftrightarrow \textit{successor} <1.00, 0.49>$$

that is, the *predecessor* relation and the *successor* relation are similar (to some extent) because both specify a *neighbor* in the alphabet.

From (4) and (7), using the analogy rule, the system gets

$$(8) \quad (\times, i, h) \rightarrow \textit{successor} <1.00, 0.24>$$

that is, the relation between i and h can be seen as successor, to certain extent (see the low confidence).

Finally, from (1) and (8), using the comparison rule again, the system gets a different answer to the question

$$(9) \quad (\times, a, b) \leftrightarrow (\times, i, h) <1.00, 0.19>$$

What if the system initially has all the beliefs (1,2,4,5,6)? In that case, both (3) and (9) may be produced as candidate answers to the question. Since (3) requests a single step while (9) needs three steps, we can expect the former to be found first, but given the probabilistic premise-selection mechanism, it is not guaranteed. Even so, when both of them are found, the system will choose (3) as a better answer, since its confidence factor is higher. It means if the system happens to find (9) first, it will be reported, but the system will change its mind, as well as its answer, if it finds (3) later. On the other hand, if (3) is found first, there won't be another answer reported, even if the system finds (9) afterward.

Of course, to fully solve the problems addressed by Copycat will be much more complicated than the previous examples, since the system needs to handle concepts like "leftmost", "opposite", "successor group", etc., as well as the relationships among them. Furthermore, letter string *abc* can be perceived either as compound term "$(\times, (\times, a, b), c)$" or "$(\times, a, (\times, b, c))$", so the number of candidate answers will be much larger. Even so, there is no *fundamental* difficulty for NARS to handle these problems, given the expressive power of Narsese and inferential power of the inference rules of the system.

Between Copycat and NARS, there is a recognizable "meta-level similarity" (or "family resemblance", since the author is a former member of Hofstadter's research group). When facing a relational analogy problem of the form "A is to B as C is to what?", both systems usually try to perceive, or categorize, the relation between A and B in such a general level that makes it possible to find something with the same, or a similar, relation with C. When there are multiple candidates, various factors will be taken into consideration to evaluate the "strength" of the analogy. Of course, given the differences between NARS and Copycat in technical details, they are not going to always provide the same results, though there will be a large overlap, given their common understanding of analogy as "using one concept as another". (Wang and Hofstadter, 2006)

3.3. Structural analogy

Another influential model of analogy in cognitive science is the Structure-Mapping Engine, SME. In this model, analogy is taken as "a mapping of knowledge from one domain (the base) into another (the target) which conveys that a system of relations known to hold in the base also holds in the target", and "Objects are placed in correspondence by virtue of corresponding roles in the common relational structure." (Falkenhainer et al., 1989)

For example, SME can establish an analogy between the situations of "water-flow" and "heat-flow", by establishing a one-to-one mapping between the objects in the two domains: "water" to "heat", "pipe" to "bar", "beaker" to "coffee", and "vial" to "ice cube". (Falkenhainer et al., 1989)

In general, the analogy problem is formalized in SME as: given the sets of items in the base and target domains B and T, respectively, as well as the within-domain relations among the items R, find a one-to-one mapping between items in their subsets b and t that keeps the relations. After such a *structure alignment*, other relations in the base domain can be mapped into the target domain. Given its stress on "structure", this type of analogy is called "structural analogy" in this paper.

To solve this type of problem in NARS, the first observation is that a "structure", in the current sense, can be represented in Narsese as a multi-component product, with the roles of the components implicitly indicated by the order of the components in the product. Using the example from Falkenhainer et al. (1989), the "simple-water-flow structure" can be represented in Narsese as $(\times, water, beaker, vial, pipe)$, and the "simple-heat-flow structure" as $(\times, heat, coffee, ice\text{-}cube, bar)$.

Structure analogy between domain B and domain T can be expressed in Narsese as finding similar products $(\times, b_1, ..., b_n) \leftrightarrow (\times, t_1, ..., t_n)$, with $b_i \in b$ and $t_i \in t$ $(i = 1, ..., n)$, that for a relation $r_j \in R$, $(\times, b_{j1}, ..., b_{jm}) \rightarrow r_j$ if and only if $(\times, t_{j1}, ..., t_{jm}) \rightarrow r_j$, where $j1, ..., jm$ is a subsequence of $1, ..., n$.

Again, let us use a simple example to show how NARS solves this type of problem. Assume the base domain contains terms *water*, *beaker*, *pipe*, and the target domain contains terms *bar*, *coffee*, *heat*. The system has the following initial beliefs:

$$\begin{array}{ll}
(1) & (\times, water, beaker) \rightarrow contained\text{-}in <1.00, 0.99> \\
(2) & (\times, water, pipe) \rightarrow flow\text{-}in <1.00, 0.99> \\
(3) & (\times, beaker, pipe) \rightarrow joined <1.00, 0.99> \\
(4) & (\times, heat, coffee) \rightarrow contained\text{-}in <1.00, 0.99> \\
(5) & (\times, heat, bar) \rightarrow flow\text{-}in <1.00, 0.99>
\end{array}$$

As in the previous example, products of the same relation will be judged as similar by the comparison rule, so from (1) and (4), the system can derive

$$(6) \quad (\times, water, beaker) \leftrightarrow (\times, heat, coffee) <1.00, 0.49>$$

and in the same way, from (2) and (5)

$$(7) \quad (\times, water, pipe) \leftrightarrow (\times, heat, bar) <1.00, 0.49>$$

NARS has the following theorem (Wang, 2006):

$$((S_1 \leftrightarrow P_1) \wedge (S_2 \leftrightarrow P_2)) \equiv ((\times, S_1, S_2) \leftrightarrow (\times, P_1, P_2))$$

That is, two products are similar if and only if the corresponding components are similar pairwisely.

As a special case, when the terms involved are products with common components, the system can "concatenate" them into "structures", that is, products with more than two components:

$$(((\times, S_1, S_2) \leftrightarrow (\times, P_1, P_2)) \wedge ((\times, S_1, S_3) \leftrightarrow (\times, P_1, P_3)))$$
$$\equiv ((\times, S_1, S_2, S_3) \leftrightarrow (\times, P_1, P_2, P_3))$$

This theorem, when applied on (6) and (7), gives us a structural analogy

$$(8) \quad (\times, water, beaker, pipe) \leftrightarrow (\times, heat, coffee, bar) <1.00, 0.24>$$

The details about how to use NARS theorems in inference and how to calculate the truth-values of compound statement (like conjunction) are explained in Wang (2006).

In a similar way, more terms, like *vial* and *ice-cube*, can be added into the structures.

The above theorem also means that similar products (structures) implies similar sub-products (sub-structures), that is,

$$((\times, S_1, S_2, S_3) \leftrightarrow (\times, P_1, P_2, P_3)) \supset ((\times, S_1, S_2) \leftrightarrow (\times, P_1, P_2))$$

so from this theorem and (8), the system get

$$(9) \quad (\times, beaker, pipe) \leftrightarrow (\times, coffee, bar) <1.00, 0.24>$$

Finally, from (3) and (9), the analogy rule produces

$$(10) \quad (\times, coffee, bar) \rightarrow joined <1.00, 0.06>$$

From this simple example, we see that NARS can carry out structure alignment by concatenating individual relations into longer ones, while keeping the correspondence between their components. After such a mapping is established, the additional relations in the base domain can be mapped into hypotheses (that is, beliefs with low confidence) in the target domain.

As described previously, when there are different ways for NARS to collect evidence for the same statement, the results will be combined using the revision rule, and when there are competing answers to a question, the choice is made mainly (though not completely) according to their confidence factors.

Restricted by insufficient resources, when facing a SME-style analogy problem, NARS will not try to exhaustively go through all possibilities. Instead, it will start from what it can find quickly, then build on it. Since NARS does not follow a predetermined algorithm when building structural analogy, its performance and efficiency cannot be directly compared to that of SME.

Once again, we have not seen fundamental problems for NARS to solve various types of structural analogy problems, even though many of them will surely be much more complicated than the previous example. Since NARS is very different from SME, for the same structural analogy problem, the two models do not process it by following the same procedure, and they may produce different results. Even so, some overlapping can still be expected.

3.4. General analogy

Finally, if we use the word "analogy" in its most general sense to mean "using one concept as another", we can even say that almost all inference in NARS is analogy.

As said above, NARS uses a term-oriented language and an experience-grounded semantics. Each belief in the system does not describe a "state of affairs" in the outside world, but rather describes the form and extent to which one term can be used as another (according to the system's experience) — this is how the inheritance relation and its variants (such as the similarity relation) are defined.

During the inference processes of NARS, the inference rules deal with this *can-be-used-as* relation: the revision rule pools evidence from different sources together, and the syllogistic rules extend the relations further to the previously unrelated terms, based on the transitivity in different relation types and directions. For example, in the similarity relation, the two directions have the same transitivity, while in the inheritance relation the transitivity is much "stronger" in one direction than in the other, though in the latter case it still exists.

In this general sense, "analogy" in NARS becomes the synonym of "reasoning", "inference", or even "thinking", so it is indeed "the core of cognition" (Hofstadter, 2001). We can see cognition, or intelligence, as the mechanism by which a system uses its past experience to deal with the current and to prepare for the future, by perceiving a novel situation as similar to a past situation. In this sense, analogy covers almost everything in AI and

CogSci, and NARS shows that such a process can be formally specified and computationally implemented.

4. Issues in Analogy

This section discusses several theoretical issues in analogy, as well as their treatments in NARS.

4.1. Working definition

As we have seen so far, the word "analogy" has different interpretations in AI and CogSci. Even in NARS, the word can be used to describe several different, though related, processes.

Since a word in a natural language often has multiple senses, to argue which sense is the "true meaning" of the word does not make too much sense. However, as far as we limit the discussion to a concrete situation, it is still better to use a word with a relatively clear and consistent meaning, chosen according to the requirement of that situation.

This is specially desired when designing a formal or computational model for certain notions, there we usually need to choose a "working definition" for each notion to be captured in the model. For this kind of usage, it is a bad idea to allow the same word to be used with different meanings. On the other hand, given the difference in research focus, it is normal for the same word to have different working definitions in different research projects.

In the technical descriptions of NARS, "analogy" has been used to mean "simple analogy", which refers to the inference rule defined previously. As shown in Table 1, in NARS different types of inference, including "revision", "deduction", "induction", "abduction", and "analogy", are defined in a purely *formal* way, that is, by the type of the relations and the position of the shared term in the premises. This treatment has the advantage of simplicity and clarity, while still keeping the intuitive sense of the cognitive functions that are usually associated with these notions: "revision" with "belief change", "deduction" with "demonstration", "induction" with "generalization", "abduction" with "explanation", and "analogy" with "substitution".

In NARS, the types of inference are not *defined* by their cognitive functions, because in the system such a function usually can be carried out in different ways, depending on the available knowledge and resources. For example, previously we have shown that a question "$(\times, A, B) \leftrightarrow (\times, C, ?x)$"

can be answered by following different inference paths and using different rules. To collectively call them "analogy" provides little help in understanding the process. Similarly, even though in the broad sense almost all inference in NARS can be considered as "analogy", to use the word in this way when talking about the details of the system is not a good idea — it is not really wrong, but not very informative.

For similar reasons, in NARS analogy is not defined as "structure mapping" (though it can be done), or being "cross domain" (though it often does) — there is no logical difference between "intra-domain" and "cross-domain" analogy in NARS.

The "simple analogy" process is obviously less complicated than the other forms of analogy in NARS, but it does not mean that it is necessarily less useful than the others, or that it does not qualified to be called "analogy". Since NARS can also cover the other forms, the decision of using "analogy" to name an inference rule does not restrict the system's capability in any manner, though it may initially cause some confusion among the readers who are used to a different usage of the word.

Please note that the claim "NARS can do analogy" is different from the claim "NARS can do analogy exactly like a human". Though it is possible for NARS to establish a similarity relation between concepts "water-flow" and "heat-flow" in the system, it does not mean that the system has built an analogy between these two English phrases as in the mind of a typical native speaker of the language. In a typical human mind and in a typical computer system, these two phrases have different associations, and therefore different meanings. However, these meanings have overlap, and the mechanism of similarity evaluation can be explained by the same principle. It is in this sense that an AI system like NARS can be said to be "making analogy".

4.2. Validity

NARS differs from the previous works on analogy in that it is a reasoning system following a logic.

It is well known that, though analogy plays an important role in providing novel ideas, the conclusions are not always reliable — actually analogy often produce wrong conclusions. Concepts C_1 and C_2 share *many* properties and relations does not mean that they share *all* properties and relations, otherwise they are not merely *similar*, but *identical*, and the inference from one to the other is not analogy, but deduction in the classical sense.

This situation is recognized by the researchers in the field, who therefore refer to the results of analogy as hypotheses, conjectures, or tentative conclusions, such as saying "Each candidate inference must be interpreted as a surmise, rather than a logically valid conclusion." (Falkenhainer et al., 1989)

If a conclusion derived by analogy could turn out to be wrong, in what sense it is still a *reasonable* conclusion? What makes us prefer analogy over arbitrary guess?

The common answer is "Analogy plays an important role in human cognition", which is of course justifiable. However, such an answer is not enough for a system like NARS, which is designed to be a *normative* model of intelligence in general, not a *descriptive* model of human intelligence. For this reason, analogy, as well as the other types of inference, must be justified in NARS as being "logically valid", rather than "psychologically real".

How is that possible, after we admit that analogy often produces wrong conclusions? The key here is to understand that there are different types of "logic" with different criteria for validity.

In its general and original sense, "logic" is the study of the principles of valid inference, where "valid" means justifiable according to certain criteria. In the last century, mathematical logic has achieved great success in formalizing valid inference in demonstrating conclusions in mathematics, to the extent that many people now implicitly take "logic" as a synonym of "mathematical logic". However, it is not hard to see that what is consider as "valid inference" in demonstrating mathematical conclusions is of a very special nature, and cannot be applied to inference in other domains.

NARS is designed to adapt to its environment while working with insufficient knowledge and resources, which means it is impossible to guarantee its conclusions to be confirmed by its future experience. However, it does not mean "validity" cannot be defined in this situation. On the contrary, it suggests that in an adaptive system validity can only be based on the system's *past* experience, not its *future* experience. Though in principle the future will always be different from the past, there is no way for a system to do better than to adapt according to the past, with the hope that the environment is at least relatively stable, so the future will not be completely different from the past.

In NARS, the justification for the validity of analogy, like that of the other types of inference, is provided by its experience-grounded semantics (Wang, 2005). As explained previously, in NARS truth-value is defined with respect to the evidence collected in the (past) experience of the system. Using it, the system can distinguish "full" analogies from "partial" analogies (by the frequency factor), as well as "strong" analogies from "weak" analogies (by the confidence factor). Though its analogical conclusions may turn out to be wrong, this function still helps the system in adaptation, therefore is justifiable in principle.

Since the human mind is evolved as an adaptive system, and it often has to work with insufficient knowledge and resources (Medin and Ross, 1992), it is not a coincidence if NARS seems similar to the human mind here or there. Especially, NARS can explain, justify, and reproduce certain well-known discrepancy between human behavior and traditional models of reasoning, like mathematical logic (Wang, 2001) and probability theory (Wang, 1996). Even so, the design of NARS is not directly guided, or justified, by psychological observations ("the *human* way"), but by more general principles ("the *right* way") — the two are not the same, though often similar to each other.

4.3. Selectivity

One aspect that makes analogy more tricky than the other types of inference is the selection of premises. When two concepts (including relations and structures) are compared, the resulting similarity largely depends on the features being compared. For some other inference, like induction, the usual preference is to consider as much evidence as possible, but many meaningful analogies can be obtained only by deliberately exclude some features as irrelevant. What makes the situation even more complicated is that the standard of relevance seems to be hard to specify, since what in one case is considered as relevant will become irrelevant in the next case, even in the same concept.

This is also what makes *metaphor*, a topic closely related to analogy, difficult to analyze. It has been revealed in the study that when a concept is used metaphorically, only part, but not all, of its properties and attributes are involved (Lakoff and Johnson, 1980).

As a general-purpose reasoning system working with insufficient knowledge and resources, NARS can neither limit itself to a predetermined microdomain, nor ask the user to only feed it with knowledge that is relevant to an analogy to be made. Instead, the system depends on a domain-independent memory and control mechanism to select the beliefs to be used.

As described previously, when the system is given a problem to solve (a new task), it does not first decide which type of inference to use. Instead, the task is added into the memory to interact with the available beliefs in the involved concepts. When the task and a belief are both selected (probabilistically), their formal features (the type of relation, the position of the shared term, etc.) decide the applicable rules. In this way, the system does not deliberately select relevant information for an analogy. Rather, it makes analogy that *happens to be recognized* at the moment.

Furthermore, not all the conclusions (analogical or not) are useful. Actually, most of them are not, and will be forgot soon by the system. An analogical conclusion must show some desired properties (such as suggesting a new answer to a question) to be kept. Therefore, NARS does not know how to only make *good* analogies. It makes all kinds of them, though only the good ones will be remembered for long.

This does not mean the system will blindly try all possibilities, of course. As described previously, there are priority distributions maintained among items competing resources of the system, and the more useful and relevant items will get more resources in the long run, even though there is no way for the system to always make the optimal decision, judged in hindsight.

One nature of this memory and control mechanism is that the meaning of a concept in the system is highly context-sensitive. Here "context" is the internal situation of the system at the moment, which changes constantly, due to the system's external interaction with the environment, as well as its internal inference activity. For a given concept, what the system believes about it, and which of the beliefs happen to be recalled, decide what kind of analogy the system is going to make on it. The analogical conclusions, when adding into the concepts, will more or less change the meaning of the concepts, and therefore they will influence what the system is going to do next.

4.4. Integrity

Another major feature that distinguishes NARS from other models of analogy is: though analogy plays an important role in NARS, it is not everything the system does (unless the notion is used in a very broad sense). As shown in Table 1, analogy is defined as one of the syllogistic rules, and it is used together with the others in solving problems. Except in trivial cases, a problem-solving process in NARS consists of multiple inference steps, with several types of inference interwoven. When the analogy rule is used in such a process, very often its promises are produced by other types of inference, and its conclusions will be used by other rules. The system rarely solves a problem using the analogy rule *alone*.

Furthermore, the selection of inference rule in the working cycle is "data driven", in the sense that it is the tasks and beliefs that happen to be recalled by the system at the moment that decide which rules are applicable at that step. When a user assigns a problem to NARS, she is like saying "Solve this in whatever way you can found", but not "Solve this using analogy", or "Solve this by induction, then analogy, finally deduction". Of course, the user can carefully prepare the input in such a way that the system can only find one possible way to solve it, but that will be exceptional, rather than the normal situation.

When talking about "relational analogy" and "structural analogy", we see that in NARS there may be other types of inference involved. Therefore, except at the level of a single inference step, "analogy" in NARS is not an inference process separable from the other inference processes.

Similarly, even if the system accomplishes something we call "analogy", it is often a by-product of many internal activities. The system may happen to notice the structural similarity between an atom and a planet system while working on another task, though not under the explicit instruction to look for such an analogy.

An interesting analogy in AI and CogSci is among the opposition between rule-based and case-based expert systems, between prototype and exemplar categorization models, as well as between neural networks making predictions based on the statistics of all data points and those only considering the nearest neighbors. At a general level, the question is the same: when we use past knowledge to solve new problems, should the knowledge be generalized first, or should it be directly transferred from old instances to new ones? It should not be a surprise if both sides can provide supporting evidence for their position, because we do, and should, use both methods. In a system like NARS, where multiple types of inference are consistently and uniformly implemented, this opposition does not exist anymore, since the system can simply use whatever information available with whatever methods applicable, given the current situation.

Some researchers argue that the analogy process (in the sense of "relational analogy" and "structure analogy") and the "high-level perception" process cannot be separated from each other, since the meaning of concept change *during* the establishing of a complicated analogy (Chalmers et al., 1992). NARS does not only treat the two processes as one, but also goes even further by treating other cognitive faculties as reasoning, such as learning (Wang, 2000) and categorizing (Wang and Hofstadter, 2006). In NARS, words like "reasoning" and "learning" (or "analogy" and "high-level perception") are usually just different descriptions of the same underlying process and mechanism — when we call it "reasoning", we focus on the relationship between the premises and the conclusions in the steps; when we call it "learning", we focus on the long-term effect of the process on the whole system. In the same way, when a process is considered as "analogy", the focus is on how the concepts correspond to each other; when the process is considered as "high-level perception", the focus is on how the boundary of certain concept is checked and adjusted. Wherever the focus is, the system cannot have one functionality without the others. Furthermore, the practice of NARS shows that the ideas like "fluid concepts" and "creative analogies" do not necessarily conflict with other desired features, such as the ones suggested in Forbus et al. (1998): memory, learning, and domain independence .

NARS can uniformly carry out multiple cognitive functions in a domain-independent manner, mainly because it is designed in the framework of reasoning system, where each inference step must follow a simple and rigid rule that is justified in isolation, while these steps can be linked together in run-time to form various inference processes, which can be very complex and flexible (Wang, 2006). This approach produces systems with higher adaptivity and generality than those following predetermined algorithms from input to output, though the latter are usually more efficient and reliable in solving problems in limited domains.

Even though general-purpose and domain-independent models are intuitively attractive, AI and CogSci are still dominated by special-purpose and domain-specific research, partly because of the belief that to study a complex phenomena like intelligence and cognition, we have to follow a reductionist approach. Literally speaking, this conclusion is correct, but it often obscures the fundamental difference between general-purpose systems and special-purpose systems, as well as between domain-independent systems and domain-dependent systems, because the former cannot be achieved by adding many the latter together, and the former can be solved using a reductionist method without changing its nature, as far as the reduction is done in proper places.

To study analogy together with other cognitive processes gives the research a different perspective. For this reason, NARS provides an interesting model for analogy, though it does not necessarily mean that it works better than models like Copycat and SME in their own domains.

5. Summary

NARS is designed to be a general-purpose and domain-independent intelligent system. Theoretically, it is special because the system is designed to be adaptive and to work with insufficient knowledge and resources. Technically, it is special because it uses a term-oriented formal language, an experience-grounded semantics, a set of rules unifying many types of inference, a dynamic memory structure allowing fluid concept, and a control mechanism supporting real-time resource allocation.

In NARS, there are several processes that are related to what people call "analogy":

Single step inference. There is an analogy rule that is designed and used consistently with rules for other types of inference.

Multi-step inference. The problems studied in existing "analogy" models, like Copycat and SME, can be solved in NARS as multi-step inference processes, typically with more than one type of rule involved.

Inference in general. The inference process in NARS can be considered as "analogy" in the sense that it is always about how to use one concept as another, though there are various concrete cases included.

As far as analogy is concerned, NARS has the following major differences from the other existing models:

- As a normative model of intelligence, in NARS all forms of analogy are carried out by a formal logic, and justified according to the principle of adaptation;
- As a domain-independent model of general intelligence, in NARS analogy is closely coupled with the other types of inference, as well as other cognitive processes.

Analogy indeed plays an important role in intelligence and cognition. Therefore, even though there have been many interesting works in this field, we still need to keep an open mind on how analogy should be perceived and modeled, given that we may have only explored a small part of the territory.

Acknowledgment

The author thank the anonymous reviewers for their helpful comments and suggestions.

References

Chalmers, D. J., French, R. M., Hofstadter, D. R., 1992. High-level perception, representation, and analogy: a critique of artificial intelligence methodology. Journal of Experimental & Theoretical Artificial Intelligence 4, 185–211.

Evans, T., 1969. A program for the solution of geometric-analogy intelligence test questions. In: Minsky, M. L. (Ed.), Semantic Information Processing. MIT Press, Cambridge, Massachusetts, pp. 271–353.

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

Forbus, K. D., Gentner, D., Markman, A. B., Ferguson, R. W., 1998. Analogy just looks like high level perception: why a domain-general approach to analogical mapping is right. Journal of Experimental & Theoretical Artificial Intelligence 10 (2), 231–257.

Gentner, D., Holyoak, K. J., Kokinov, B. N., 2001. The Analogical Mind: Perspectives from Cognitive Science. MIT Press, Cambridge, Massachusetts.

Goertzel, B., Pennachin, C. (Eds.), 2007. Artificial General Intelligence. Springer, New York.

Hofstadter, D. R., 2001. Analogy as the core of cognition. In: Gentner, D., Holyoak, K. J., Kokinov, B. N. (Eds.), The Analogical Mind: Perspectives from Cognitive Science. MIT Press, Cambridge, Massachusetts, pp. 499–538.

Hofstadter, D. R., FARG, 1995. Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. Basic Books, New York.

Holyoak, K. J., Thagard, P., 1989. Analogical mapping by constraint satisfaction. Cognitive Science 13, 295–355.

Indurkhya, B., 1992. Metaphor and Cognition: An Interactionist Approach. Kluwer Academic Publishers, Dordrecht.

Lakoff, G., Johnson, M., 1980. Metaphors We Live By. University of Chicago Press, Chicago.

Lenat, D. B., 1995. Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM 38 (11), 33–38.

Medin, D. L., Ross, B. H., 1992. Cognitive Psychology. Harcourt Brace Jovanovich, Fort Worth.

Nipkow, T., Paulson, L. C., Wenzel, M., 2002. Isabelle/HOL — A Proof Assistant for Higher-Order Logic. Vol. 2283 of LNCS. Springer.

Robinson, J. A., Voronkov, A. (Eds.), 2001. Handbook of Automated Reasoning. MIT Press, Cambridge, Massachusetts.

Wang, P., 1996. Heuristics and normative models of judgment under uncertainty. International Journal of Approximate Reasoning 14 (4), 221–235.

Wang, P., 2000. The logic of learning. In: Working Notes of the AAAI workshop on New Research Problems for Machine Learning. Austin, Texas, pp. 37–40.

Wang, P., 2001. Wason's cards: what is wrong? In: Proceedings of the Third International Conference on Cognitive Science. Beijing, pp. 371–375.

Wang, P., 2005. Experience-grounded semantics: a theory for intelligent systems. Cognitive Systems Research 6 (4), 282–302.

Wang, P., 2006. Rigid Flexibility: The Logic of Intelligence. Springer, Dordrecht.

Wang, P., 2007. The logic of intelligence. In: Goertzel, B., Pennachin, C. (Eds.), Artificial General Intelligence. Springer, Berlin, pp. 31–62.

Wang, P., Hofstadter, D., 2006. A logic of categorization. Journal of Experimental & Theoretical Artificial Intelligence 18 (2), 193–213.